MAC-REALM: A VIDEO CONTENT FEATURE EXTRACTION AND MODELLING FRAMEWORK

by

Minaz Parmar

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy

School of Engineering and Design

Brunel University

2013

ABSTRACT

A consequence of the ‘data deluge’ is the exponential increase in digital video footage, while the ability to find relevant video clips diminishes. Traditional text based search engines are no longer optimal for searching, as they cannot provide a granular search of the content inside video footage. To be able to search the video in a content based manner, the content features of the video need to be extracted and modelled into a content model, which can then act as a searchable proxy for the video content. This thesis focuses on the extraction of syntactic and semantic content features and content modelling, using machine driven processes, with either little or no user interaction. Our abstract framework design extracts syntactic and semantic content features and compiles them into an integrated content model. The framework integrates a four plane strategy that consists of a pre-processing plane that removes redundant data and filters the media to improve the feature extraction properties of the media; a syntactic feature extraction plane that extracts low level syntactic feature and mid-level syntactic features that have semantic attributes; a semantic relationship analysis and linkage plane, where the spatial and temporal relationships of all the content features are defined, and finally a content modelling stage where the syntactic and semantic content features are integrated into a content model. Each of the four planes can be split into three layers namely, the content layer, where the content to be processed is stored; the application layer, where the content is converted into content descriptions, and the MPEG-7 layer, where content descriptions are serialised. Using MPEG-7 standards to produce the content model will provide wide-ranging interoperability, while facilitating granular multi-content type searches. The framework is aiming to ‘bridge’ the semantic gap, by integrating the syntactic and semantic content features from extraction through to modelling. The design of the framework has been implemented into a prototype called MAC-REALM, which has been tested and evaluated for its effectiveness to extract and model content features. Conclusions are drawn about the research output as a whole and whether they have met the objectives. Finally, future work is presented on how concept detection and crowd sourcing can be used with MAC-REALM.

PREFACE

The work of this thesis was carried out under the supervision of Prof. Marios Angelides in The School of Engineering and Design, at the University of Brunel, United Kingdom.

ACKNOWLEDGEMENTS

I would like to thank my colleagues, family and friends for all of their support, encouragement and ears! I would also like to give special thanks to Rahim Lakhoo, whose insight was a constant source of guidance and help during the writing of this thesis. Also special thanks to Emmy-Lou for being my rock through the final months of this PhD.

This thesis is dedicated to my mum, whose love, support and strength has seen me not just through this task, but everything I have ever done and will do.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS

Refereed Journal Articles

Parmar, M. J., Angelides, M. C., (2010). Automatic Feature Extraction to an MPEG-7 Content Model. *Advances in Semantic Media Adaptation and Personalization*, *2*, 399.

Angelides, M., Sofokleous, A., & Parmar, M. (2006). Neural Networks, Semantic Web Technologies and Multimedia Analysis (Special Session)-Classified Ranking of Semantic Content Filtered Output Using Self-organizing Neural Networks. *Lecture Notes in Computer Science*, *4132*, 55-64.

Refereed Edited Book Articles

Parmar, M. J. (2008). Multimedia Information Filtering. In M. Syed (Ed.), *Multimedia Technologies: Concepts, Methodologies, Tools, and Applications* (pp. 233-241). Hershey, PA: Information Science Reference. doi:10.4018/978-1-59904-953-3.ch019

Parmar, M. (2005). Review: Distributed Multimedia Database Technologies Supported By Mpeg-7 And Mpeg-21. *Computer Journal*, *48*(5).

Refereed Conference Proceedings Articles

Parmar, M. J. (2007, December). Automatic feature extraction to COSMOS-7 content models. In *Semantic Media Adaptation and Personalization, Second International Workshop on* (pp. 245-248). IEEE.

Parmar, M. J., & Angelides, M. C. (2007). XML-based Genetic Rules for Scene Boundary Detection in a parallel processing environment. **http://bura.brunel.ac.uk/handle/2438/601. Access Date: 23/09/13.**

Angelides, M., Sofokleous, A., & Parmar, M. (2006). Classified ranking of semantic content filtered output using self-organizing neural networks. In *Artificial Neural Networks–ICANN 2006* (pp. 55-64). Springer Berlin Heidelberg.

Parmar, M. J., & Angelides, M. C. (2005, March). Human readable genetic rules for scene boundary detection. In *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on* (Vol. 2, pp. 541-546). IEEE.

CHAPTER 1: CONTENT FEATURE EXTRACTION AND MODELLING

1.1 Introduction

This chapter aims to establish the thesis by introducing the overarching themes and by placing the inspiration for the research undertaken into context. Subsequently, the motivation and goals defined for the investigation of the thesis are discussed, followed by a summary of the thesis project. Finally, an overview of the dissertation is given on a chapter-by-chapter basis.

The explosion of multimedia content on the Internet and in digital archives over the last decade has led to a striking increase in data volume being transferred and stored (Vijayakumar & Nedunchezhian, 2012). The increase in data has lead to the need for better methods for processing and storage of content (Apache, 2013; Dropbox, 2013; Microsoft, 2013; SugarSync, 2013). Data can also be stored in a semantically rich way that allows for better links to be made between information stored in the content (Chiarcos, Nordhoff, & Hellmann, 2012; Mika & Greaves, 2012). The film industry amongst others (i.e. gaming industry) has made extensive use of multimedia content for their businesses (Fromme & Unger, 2012; Tryon, 2012) , including Internet and mobile streaming services (Lawrence et al., 2012; Sarmiento & Lopez, 2012). Multimedia content not only contains text, audio, video and metadata such as length and time, but can also convey a wealth of information in the content itself. The information that is conveyed by multimedia can also include descriptions and events. For example, in the film industry multimedia content may also convey shots, scenes, people and objects. In addition, the multimedia content includes low-level information such as structural and signal level descriptions. For example, 3-Dimensional (3D) content contains extra information to generate content for both eyes (Dal Mutto, Dominio, Zanuttigh, & Mattoccia, 2012). Even though multimedia contains and conveys a wealth of information, the information contained is not typically used for searching.

Searches for multimedia content such as images are by and large a manual process. Typically, the manual search process performs a coarse search, based on simple identifiers of the multimedia content, which are often misleading and result in a deluge of results. The majority, if not all, of the results are incorrect, or the user must painstakingly identify all multimedia content manually that could be relevant. Traditional text-based search and filtering cannot directly query the multimedia content causing these inaccuracies. The main disadvantage to this is that the abundance of semantic information available within the data itself is largely ignored. Other metadata apart from the semantics within the content are largely ignored, for example how the content was created and what formats it is available in. Typically, methods of searching require a granular description of the content in order to fully utilise semantic meanings within the media. There have been research endeavours to improve the representation and querying of multimedia content (Moens, Poulisse, & VRT, 2012; Weiming, Nianhua, Li, Xianglin, & Maybank, 2011). Google's image search[1] is one such endeavour that can now search for images, using an image as search criteria. However, research pertaining to video searching using similar methods are still not as readily available and is an on-going area of interest (Mezaris, Papadopoulos, Briassouli, Kompatsiaris, & Strintzis, 2009).

To allow for the searching of multimedia content, the content requires to be represented in a suitable fashion, to better describe the content. The multimedia description can be represented and organised into a content model (Marios C. Angelides, 2003). The aim of a content model is the presentation of such information to allow content producers/consumers to effectively query and retrieve content (Weiming et al., 2011). A content model can also facilitate as a container for the automatic extraction of content semantics and the intricacies pertaining to multimedia interpretation (Garg and Ramsay, 2011). With the increase of the amount of content being generated the content representation also needs to be automatic. Automatic content representation is an implicit requirement from the combination of the increase in the amount of content being generated, and the wealth of information stored in multimedia content itself (Lavee, Rivlin, & Rudzsky, 2009; Moens et al., 2012).

In Figure 1.1 we have an example of a video content extraction and modelling system environment. The video content feature extractor processes the raw media stream. Here the syntactic and semantic content features within the video stream are extracted. These are then modelled into a content model that can be accessed by a video search application. Consumers can query the content model via the video search application. The results are then sent back to the consumers' devices.

[1] https://www.google.co.uk/imghp?hl=en&tab=wi

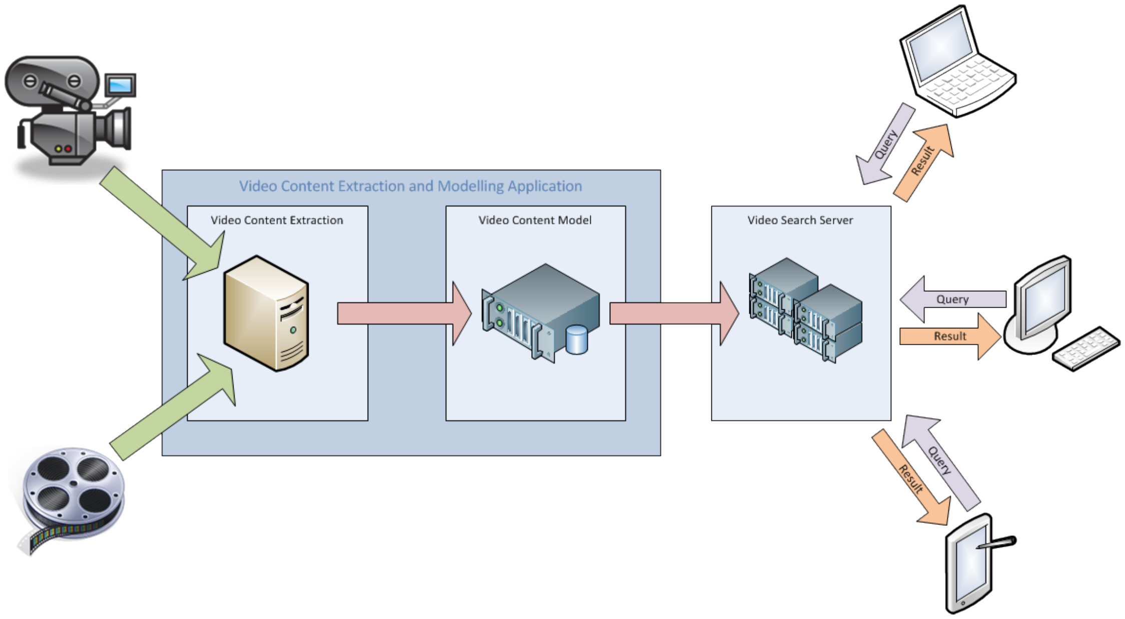


Figure 1.1: VIDEO CONTENT EXTRACTION AND MODELLING APPLICATION

The focus of this thesis is merging content models with automatic feature extraction into the MAC-REALM cross-functional framework. MAC-REALM uses automatic feature extraction techniques on video content and models them into a hierarchically linked scheme. The automatically extracted features are analysed and semantic relationships derived, to allow the user to query relationships between entities in the multimedia content effectively. The extracted features are also structurally and conceptually linked together to provide a richly descriptive, granular and standardised (MPEG-7) content model, allowing users to view the content from multi-faceted perspectives.

The research presented in this thesis aims to focus on solutions to address issues surrounding the automatic feature extraction and content modelling of video. In this context, its goal is to offer solutions focused on the combination of automatic extraction, analysis and indexing of syntactic and semantic content features for digital video streams. The research aims to explore the ability of a video feature extraction and indexing framework that links semantic features to syntactic foundations that allows both high and low level querying of video content in a more integrated manner.

The main contribution of the thesis is the introduction of a new video framework called MAC-REALM. MAC-REALM has novel approaches for content modelling for organising and solutions to facilitate automatic feature extraction. Specifically contributing novel approaches for:

- Pre-processing video content to optimise syntactic feature extraction.

- Extracting syntactic features that incorporate semantic attributes.

- Analysing and linking temporal and spatial relationships to mid and low level content features

- Modelling feature extraction into standardised content models

Other contributions include a state-of-the-art review of related literature, a reference implementation of the MAC-REALM framework and an empirical evaluation of the techniques and solutions proposed by this thesis. We will have a walkthrough of MAC-REALM that shows how the features are extracted and then indexed into a content model. The main culminations of these contributions are combined to create a novel framework for content-based video retrieval, which is able to achieve a more feature inclusive approach when compared to current solutions. Besides the aforementioned contributions, many others derived from this PhD work, including earlier work on Automatic feature extraction on an MPEG-7 content models (M. Parmar & Angelides, 2010), and Automatic feature extraction to COSMOS-7 content models (M. J. Parmar, 2007), that both dealt with content feature extraction into a content model. In addition, other related work has been previously undertaken for Classified Ranking of Semantic Content Filtered Output Using Self-organizing Neural Networks (M. Angelides, Sofokleous, & Parmar, 2006), XML-based Genetic Rules for Scene Boundary Detection in a parallel processing environment (M. J. Parmar & Angelides, 2007) and Multimedia Information Filtering (M. J. Parmar & Angelides, 2005)

1.2 Research Direction

The motivation for this thesis came from experiences gained while working at QVC, a broadcast shopping channel. They had launched an interactive service in 2000 (Minter, 1999) to maximise revenue by taking advantage of services made available from digital broadcasting and take advantage of online shopping boom that was happening at the time (Aburjanidze & Boucher, 2010). The interactive channel, as it was called, would allow customers to get extra information about the products showing on the screen. This information was prepared before the show and made available at the correct time, by synchronising the information with the product broadcast scheduling system, in the gallery. This meant that the information was only available at the time of broadcast. The sourcing of information and then cataloguing of the product is a manual task and takes a department of over 40 staff.

The system in place at QVC up until September 2001 was adequate, as the interactive service ran as an additional resource to the TV broadcast. In September 2001, BskyB launched the first major digital personal video recorder (PVR)(BSkyB, 2012). The popularity of this and other PVR's such as Freeview+(Group, 2006b), Freesat+(Humax, 2008), BT Vision(Williams, 2006), and Virgin Media's V+(Which?, 2009) changed the way people interacted with their televisions. PVR's allowed viewers greater freedom and control to consume content in a manner that suited them. It also brought around the age of on demand TV. PVR's can hold many hours of recordings and searching the content is a laborious task. Within an hour show in QVC there could be up to 15 products for sale, also the product could be added or removed from the schedule depending on need. Some viewers like to buy more products from certain guests, presenters and product lines, or a combination of the three. Now viewers could not only watch what they wanted, they could decide when they wanted to watch it. The only problem with PVR's was there was no way of actually searching the video itself, only the general description of the clip. What would be ideal would be a way of indexing the content so the user could perform a search and find clips of content that was of interest to them. The indexed content would be in the form of an associated metadata file, either in an xml based description language or machine readable binary format, that could accompany or be transmitted with the content (Wollborn, 2010).

That was 2001 and digital broadcast TV and PVR's are no longer the only means to view content. IPTV in the form of BT vision (Group, 2006a), which is a hybrid of digital terrestrial TV and IPTV, was the first commercially available IPTV service in the UK. Three major IPTV projects soon to be available are BskyB Now TV (Scott, 2012), YouView and Google TV(Goss, 2011) that will aggregate content from both digital terrestrial and satellite TV and also the Internet in the form of catch up TV services. With current treads moving to use the Internet as a broadcast medium, there will not only be an increase in the amount of content, but content will also be available to a global audience. Broadcasts in different spoken languages complicate the problem of searching over multimedia content even further (LawTo et al., 2011).

Traditional ways of viewing content in terms of location is also changing with the advent of mobile devices, such as smartphones and tablets, which allows content to be consumed ubiquitously. Virgin will be launching their exclusive web portal designed to bring their content to a range of mobile devices (Goss, 2012). With the proliferation of content and the amalgamation of services across so many domains and devices searching and pinpointing relevant content becomes more challenging.

When creating an index for multimedia content it is common to use syntactic features, such as colour, shape and texture (LawTo et al., 2011). However, this does not allow meaningful searches to be made on the content. Indexing semantic features allows the capturing of events, actions and concepts from the content, allowing more meaningful search results (LawTo et al., 2011). The search results are also provided with a context from a user's query and can lead to more accurate results (LawTo et al., 2011).

Research on extracting syntactic features from video and then creating a content model for those features has been researched (Weiming et al., 2011), as has semantic feature extraction, where the underlying concepts, events and objects are indexed (Lavee et al., 2009). As suggested in the latter's research, syntactic and semantic feature extraction are interlinked, as syntactic features provide the foundation for semantic description of the concepts and events portrayed in the content. The syntactic feature extraction of the low-level aspects of the video can be extracted easily using machine based techniques such as pixel, object and logic based extraction (Lavee et al., 2009). Semantic feature extraction uses content modelling techniques such as state models, Pattern recognition methods and semantic models (Snoek & Worring, 2009). What becomes apparent in this survey and others (Stamatia Dasiopoulou, Giannakidou, Litos, Malasioti, & Kompatsiaris, 2011; Lavee et al., 2009; Money & Agius, 2008; Weiming et al., 2011) is that there is a firm conceptual distinction between syntactic and semantic feature extraction.

Syntactic features are discernible because of their physical characteristics, such as colour, shape and texture (Kaleka, Singh, & Sharma, 2012) but some features hold a semantic facet to them. Scene segmentation is a low-level feature but has attributes of a high-level feature in that they represent a semantic event. Scenes are a collection of shots that are grouped together by how a user perceives the thematic relationship between the shots. This thematic relationship between the shots is semantic in nature (LawTo et al.). However, with regards to scenes the difference between syntactic and semantic features is not well defined. Video content feature extraction is used other domains aside from digital broadcasting domains, such as application in digital libraries, distance learning, video-on-demand and multimedia information systems (LawTo et al.).

Content models can best be described as "surrogates" for the actual physical content of the multimedia (M.C. Angelides & Agius, 2006). This means that instead of searching, filtering or browsing the content directly the content model acts as an index of the information contained in the content. The content model has to be tightly integrated with the video stream, cataloguing all the features within the content that would be of interest to the consumer. The indexing methodology must be able to allow the content features to be accessible and usable for a myriad of purposes. To this end the content model scheme must be standardised so that it can facilitate wider interoperability.

Within the content model, features must represent not only the high level features that humans can search for, such as events and relationships, but also low level features that can be searched by automated methods. Therefore, by employing querying techniques that utilise the full breadth of the information contained within the stream, more relevant content can be discovered. This detailing and structuring of the content information within the model creates a description that is both rich and granular and represents the content comprehensively.

1.3 Literature Review

The following is a literature review based on the direction of research in the fields of content feature extraction and content modelling. The state-of-the-art review surveys video content modelling, technologies and techniques facilitating automatic feature extraction.

The review begins with the pre-processing of raw media and then progresses through the different stages of syntactic feature extraction, followed by deriving the semantic features of the content and finally examining work to standardise the description of these features.

1.3.1 Raw Media

Raw media is the untreated video footage of the content. This is what most consumers want to query in a more meaningful way. They want to locate and view the content within that is relevant to them without searching manually through the whole content itself. The raw media is the formatted and encoded medium for the transport of the content to the user. Most raw media does not contain any structures for the discovery or querying of content features, either low or high level. Indeed the sole focus of most media is to enable the efficient transportation of the content to the end client. This treatment of the media is not beneficial, and is sometimes at odds with the goal of content feature extraction and modelling .

Before content feature extraction can take place the raw media, in most cases, needs to be pre-processed to optimise the effectiveness and/or efficiency of the extraction process. Typically, the pre-processing is a filtering step to remove artefacts that could cause errors in the extraction process (Y. Chen et al., 2010; Yongquan, Weili, & Shaohui, 2009), and can be used to reduce the time or complexity of processing the features (Amiri & Fathy, 2011; Chan & Wong, 2011; J. Li, Ding, Shi, & Li, 2010). The following section reviews a number of pre-processing techniques used to prepare raw media for syntactic feature extraction. It is non-exhaustive, but presents the common approaches adopted by most feature extraction systems to prepare the raw media.

As already mentioned, pre-processing is a necessary step in preparing the raw media so that the extraction process can extract it more readily. This is usually a case of normalising, converting or filtering the media in the pre-processing step. One such method of normalising is 'flattening' an image. Flattening is a pre-processing step that can help in making the syntactic feature extraction processes become more accurate. In (Yongquan et al., 2009) there are many different grey scale levels within a real time scene image. A correlation window is used to compute the grey mean of pixels across the image. To avoid a complex over segmentation of the image, Yongquan et al smooth the regions using a temporal correlation window that samples the different grey scale values and uses a 3x3 median filter to normalise values of adjacent pixels, if they are within a certain range. This reduces the region into candidate areas that are likely to be foreground and background regions. In (Y. Chen et al., 2010) the authors transform the colour space profile from the H.264 YUV colour space, into a more humanly perceptible HSV colour space. (Y. Chen et al., 2010) then quantise the continuous HSV colour values into discrete intervals as follows:

Eq. (1.1)

$$\begin{aligned} h' &= [h/\Delta h] \\ s' &= [s/\Delta s] \\ v' &= [v/\Delta v] \end{aligned}$$

(Y. Chen et al., 2010)

Here Δh, Δs and Δv denote the H, S and V dimension quantization intervals, and (h', s', v') are the quantized colour value. These pre-processing steps are performed to optimise the hue component that represents the most significant characteristic of the colour. These two steps come before the first phase of feature extraction. Figure 1.2 shows the life cycle of this system:

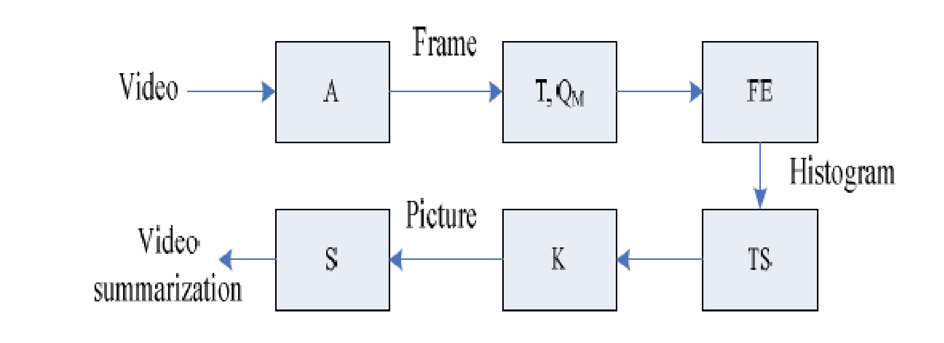


Figure 1.2: GENERATING VIDEO SUMMARIZATION A: FRAME ABSTRACTION, T: COLOR SPACE TRANSFORMATION, QM: COLOUR QUANTIZATION FE: FEATURE EXTRACTION, TS: TEMPORAL SEGMENTATION, K: KEY FRAME EXTRACTION, S: IMAGE SCALING(Y. CHEN ET AL., 2010)

Another reason for pre-processing is to reduce the computational complexity, and therefore the processing time, into an acceptable amount. This of course, must be done without impacting the effectiveness of the extraction process. One such method is to reduce the amount of information needing to be processed by removing redundant data. In (Amri & Fathy, 2010) the sampling frame rate of the video sequence is reduced by a several factors before the extraction process, it was shown that this was adequate for video clips with no fast action sequences. The method employed reduces the high computational cost for processing higher frame rates. In (Chan & Wong, 2011) the authors describe using a sampling rate at one frame per half second, or 2 frames per second (fps) for the pre-processing step. They used this sampling strategy since it assumes that for most domains, shot lengths are longer than 15 frames or half a second. In (Chan & Wong, 2011) go on to use Edge Change Ratio ECR to perform a first-pass on the video for optimal performance of the algorithm, and generate metrics for the evaluation of the genetic algorithm GA fitness function.

Another way of reducing redundancy is to remove the amount of frames by applying a simplified data technique. Before shot extraction can begin, (J. Li et al., 2010) reduces the number of shot candidate frames by using a block colour histogram difference. This method is highly effective as a computational efficiency tool, as the shot boundaries included in film programs typically amount to less than 1% of total frames; thus it is inefficient and extremely time consuming to apply boundary detection processing to detect all the frames.

Converting one codec from another codec is another method for reducing processing time. Some codecs are faster or better suited to certain feature extraction techniques. Popular codecs like MPEG- 2 are not particularly suited to feature extraction as they were not optimised for feature extraction and were focused on compressing the video signal to an absolute minimum (Haskell, Puri, Netravali, & Langdon, 1998). Newer codecs such as H.264 are better suited to video feature extraction as they have advanced features such as motion vector encoding that can be for syntactic feature extraction. In work by (Fei & Zhu, 2010) they segment objects based on motion vectors (MV) directly from H.264 formatted media. They still have to temporally normalise the raw MV to provide a uniform sample, which is ready for the segmentation process. In (Zajić, Reljin, & Reljin, 2011) for the purpose of the experiment, the introductory sequence of the film "Good Year"[2] was used. The video sequence which lasted 4 minutes, was converted from DIVX format to uncompressed AVI format, which is used for frame extraction. AVI format provides more

[2] http://www.imdb.com/title/tt0401445/

uncompressed frames and is more suitable feature extraction as there is more frames in every frame sequences to process. This improves the precision of the extraction process.

1.3.2 Syntactic Extraction

Syntactic feature extraction is the segmentation of a video signal into its constituent parts. These parts represent the different physical aspects of video content that are directly discernible from viewing the video. Each aspect has its own unique physical attributes that describes a certain physical feature of the content that is of interest to a consumer. These properties can be used in a query to identify that segment, if it fulfils the criteria of that query.

The motivation for syntactic feature extraction, or syntactic abstraction as it has been referred to, is to provide an intermediary representation of the video sequence. In this section, we concentrate on syntactic feature extraction and the different techniques used to segment features. These techniques can be grouped into three categories, pixel based, object based and logic based (Lavee et al., 2009).

Pixel based techniques are generally used for temporal segmentation, and employs the processing of colour, texture, or gradient information in the content. Object based techniques are those that identify features that are the basis for description of semantic items, such as object detection and tracking, face recognition. Unlike pixel based techniques, which define a global primitive feature such as a shot, object based events aggregate edges, colour and textures into recognisable items. Objects based techniques are not typically classified as low level extraction techniques as they describe a feature or features that have semantic connotation of identity, albeit anonymously. Logic based techniques are the observation that the world is not described by multi-dimensional parameterizations of pixel distributions, or even a set of semantic objects and their properties, but rather by a set of semantic rules and concepts, which act upon units of knowledge. Thus it aims to abstract low-level input into statements of semantic knowledge (i.e. assertions) that can be reasoned on by a rule based event model.

Both logic and, to a lesser extent, object based techniques can be described as mid-level features. The major challenge for content based retrieval is to bridge the gap between the low level syntactic features and high level semantic features (Y.-F. Huang & Tung, 2010). Mid-level features help us achieve this aim by providing a linking mechanism between the low and high level features. Mid-level features are still syntactic features that have a semantic characteristic about them. For instance a object is a syntactic feature as it has no semantic meaning. It is a generic form and has no semantic concept attached to it such as a “person” or “car” for example. It does have a semantic connotation of being a "thing" of interest that is cognitively distinct from the background. This distinction cannot be explained in purely syntactic terms; therefore it is a mid-level feature.

The choice of syntactic feature extraction is intended to isolate salient properties of the video data especially those that allow useful discrimination between interesting events. Syntactic feature extraction is thus related to the problem of feature selection. There are two categories of content based features that can be analysed in syntactic feature extraction: the global features extracted from a whole image and the local or regional features describing the chosen patches of a given image (Harikrishna, Satheesh, Sriram, & Easwarakumar, 2011). Each region is then processed to extract a set of features characterizing the visual properties including the colour, texture, motion and structure of the region. The shot-based features and the object-based features are the two approaches used to access the video sources in the database.

Syntactic feature extraction may be a transformation of the low-level input or simply a way of organizing this input. Syntactic feature extraction approaches may be designed to provide input to a particular content model or to construct informative atomic primitives that can serve as input to a general content model. In this section we will discuss several popular ideas for how to abstract video data. Along with capturing the important event-discriminating aspects of the video data other main motivations in selecting a particular syntactic feature extraction scheme are computational feasibility, and ability to complement the chosen content model.

We will examine the syntactical structure of video in a natural hierarchical analysis. We will begin by looking at the basic foundation of temporal segmentation, and look at the role it plays in deriving semantic features. This will be followed by review of spatial segmentation techniques and their importance in starting a semantic narrative of the content. Finally we will look at how the semantic gap, in terms of human perspective, has an influence on temporal segmentation when segmenting into hierarchical components.

1.3.2.1 Syntactic Media

The strategy employed for the use of low level primitives for input into the feature extraction process is key to the efficiency and effectiveness of the syntactic feature process. The syntactic media is the input for the syntactic feature process. The attributes of different types of syntactic media are used for different syntactic extraction processes. The choice of media and the way the attributes for that media are selected have a major impact on the quality of the extracted syntactic features. This then has a direct impact on the descriptions of those features within the content model. The rest of this section provides examples of different types of syntactic media and the impact they have on their extraction process.

In (Seidl, Zeppelzauer, & Breiteneder, 2010) they investigate gradual transitions in old archive footage. They were specifically looking at how historic footage required different transition detections algorithms from contemporary footage. Contemporary footage algorithms mainly used colour and luminance based techniques. Historic footage cannot use colour and would have to use texture based methods. As historic material is black and white, they use global and local luminance histograms instead of colour histograms. They also use DCT coefficients and MPEG-7 edge histograms. To be invariant to object motion, they extract the luminance histograms globally. To be more sensitive to spatial information, they also extract the same features in localised blocks of 4, 9 and 16 pixel.

Scale invariant feature transforms (SIFT) are used in computer vision processing as a feature that In (J. Li et al., 2010) they use a SIFT descriptor that uses pixel intensity as the feature to be transformed but in (Sharmila Kumari & Shekar, 2010) they have extracted SIFT descriptors for each colour plane of the RGB colour space. This is so that important visual information regarding colour is not missed. They call this approach Colour Scale Invariant Feature Transform (CSIFT).

In (Zajić et al., 2011) they extracted for each frame low-level features (colour and texture) and concatenated in the form of a feature vector. The feature vector consists of the following features: HSV Colour histogram, Colour moments, Colour layout descriptor, Structural colour descriptor, Colour correlogram, Gabor transformation features, Radial co-occurrence matrix features, Edge histogram and Wavelet texture feature. The total number of FV coordinates is N = 1369. The selected sequence is characterized by feature matrix MxN = 4512x1369. Features matrix columns were normalized with a maximum value within a column.

In (W. Li, Chen, Zhang, Shi, & Li, 2012) they produce an illumination-invariant histogram that is a robust method against illumination changes and object /camera motion without spatial information. Illumination-invariant histogram is selected as the feature vector. The normalized chromaticity is defined as:

Eq. (1.2) $$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}$$ (W. Li et al., 2012)

Histogram with 256 bins of each video frame is generated as features in normalized chromaticity colour space.

Syntactic media can be used for context features as well as content features. In (J. Chen, Ren, & Jiang, 2011) they use motion and edges as context features since both of them mainly reflect the activities inside the captured visual scenes. In this way, shot cut detection can be made adaptive to the context changes as well as content changes. When motion is high, for example, it indicates that proportional content difference is caused by motion rather than by cuts, and thus the threshold should be moved higher.

1.3.2.2 Syntactic Temporal segmentation

The first step to manage video data is to divide them into a set of meaningful and manageable units, so that the video content remains consistent in terms of camera operations and visual events. This has been the goal of a well-known research area, called video segmentation. Video can be thought of as a hierarchical syntactical structure as shown in Figure 1.3. The video itself is comprised of scenes. The scenes are logical story units that describe a singular event. The scenes can be split into shots. Shots are units of action and consist of a continuous set of frames. A scene or shot is, on occasion, represented by a selected frame called a “key frame”. This frame is representative of the main event or action of the scene or shot.

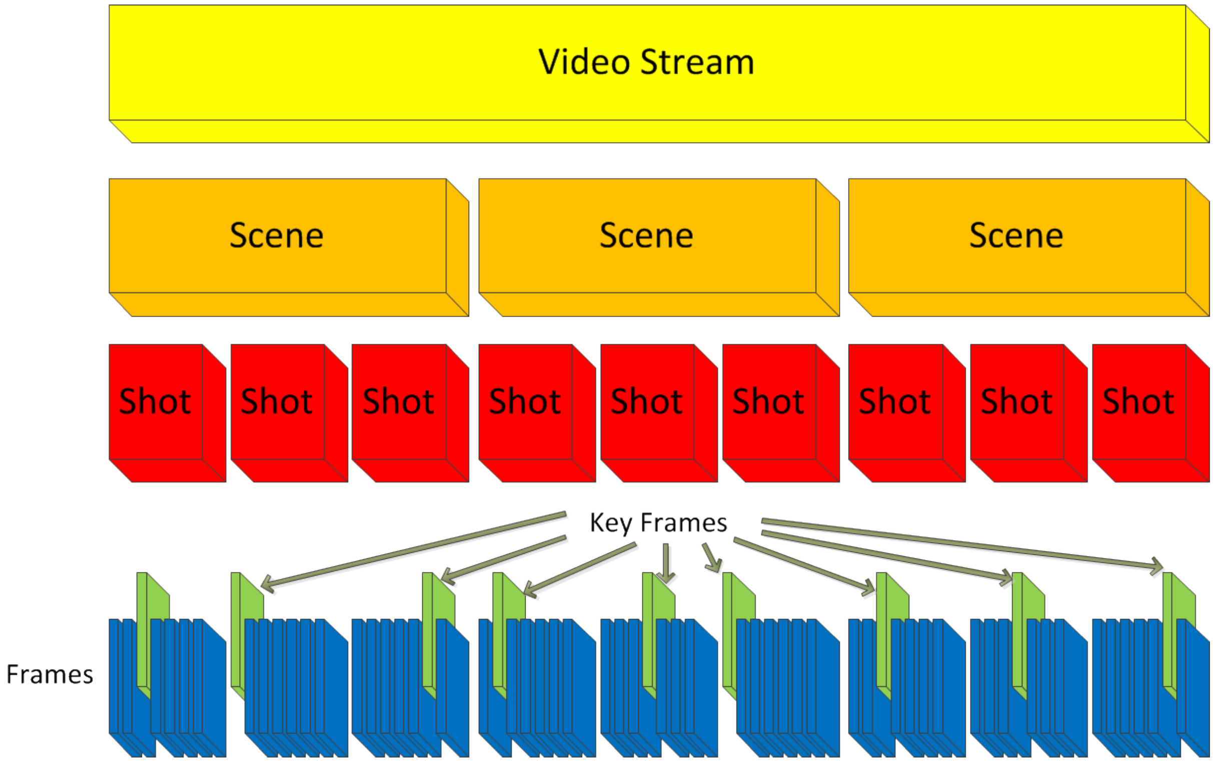


Figure 1.3: Hierarchical structure of video content

The transition from one shot to the next may be of various types: broadly categorized as abrupt change shots and gradual change shots. Abrupt change shots, also known as cut shots, denote an instantaneous transition from one shot to another. This occurs due to simplest physical concatenation of two successive shots. On the other hand, a gradual transition shot is obtained by incorporating photographic effects, usually through editing. It can be further classified as fade-out, fade-in, dissolve, and wipe shot. Fade-out is a gradual transition of a scene by diminishing overall brightness and contrast to a constant image (usually a black frame). Fade-in is a reverse transition of fade-out. Dissolve is a gradual super-imposition of two consecutive shots. In general, abrupt transitions are much more common than gradual transitions, accounting for over 99% of all transitions found in video(Krulikovska, Pavlovic, Polec, & Cernekova, 2010). As shown in Table 1.1 there is still a lot of activity in the area of shot boundary detection (SBD). The table shows the transition types detected, the syntactic feature used to detect them, whether they are compressed or not and the techniques used to detect them.

It is well known that, in case of abrupt transition, the last frame of a shot and the first frame of the following shot are uncorrelated (Mohanta, Saha, & Chanda, 2012). A cut is generated by the natural process of capturing video data through the camera. On the contrary, gradual transitions (fade-in, fade-out, and dissolve or cross-fading) are generated through editing. For example, dissolves are generated by super-imposing the boundary frames of two successive shots over a duration. In case of fade-out (or fade-in), the intensity of boundary frames at the end (or the beginning) of a shot gradually decreases (or increases) and the last (or first) frame of such transition is usually a black frame. Thus, unlike abrupt transitions, gradual transitions span over a range of frames which are correlated.

Different techniques have been proposed in the literature to address the temporal segmentation of video sequences (Haller, Krutz, & Sikora, 2009; H. Li & Ngan, 2011). Many research works have focused on the uncompressed domain (Amri & Fathy, 2010; Grana & Cucchiara, 2007; Hameed, 2009). The simplest technique employed is one based on pixel-wise difference between consecutive frames (Grana & Cucchiara, 2007) but it is very sensitive to motion of objects. To address the variation in pixel difference and mutual information due to object motion and small camera pan, zoom, and tilt, features like motion vectors (Krulikovska et al., 2010) are incorporated to measure continuity. In (Mohanta et al., 2012) motion vectors are used as localised feature statistics. By judging the shift in edge pixels in the horizontal and vertical directions a motion matrix can be built up that can identify both panning motion and zooming shots.

Greyscale or colour histogram-based features are also tried in need colour histogram and grey level references which are relatively stable though they lack spatial information. While most systems use intensity (Mohanta et al., 2012) or RGB colour histogram(C. Ma, Yu, & Huang, 2012) , some use other colour triplets, for example, YUV (Hameed, 2009) or HSV palette (Y. Chen et al., 2010; R. Tapu & T. Zaharia, 2011; Xu & Xu, 2010). When using colour histogram features, it is necessary to decode the compressed video streams firstly (C. Ma et al., 2012). Hence these methods lack of spatial information. Histogram based technique are usually based on the fact that the colour distribution across a shot is usually stable and homogenous throughout the shot. When a shot break occurs there is usually, for abrupt shot transitions, a sharp change in colour distribution occurs. Measuring the colour histogram difference is a good indicator of abrupt shot change and has provided high rate detection results(Y. Chen et al., 2010). This can be affected by fast global motion (such as action scenes and quick pan and zoom) and special effects. It is argued that different colour spaces are better for shot boundary detection (Hameed, 2009). In (Krulikovska et al., 2010) they used both RGB and YUV colour spaces and found that RGB format gave a marginally higher detection rate. There are rare colour triplets in use for SBD such as L*a*b* colour space which is used by (Küçüktunç, Güdükbay, & Ulusoy, 2010) for their SBD implementation for a content based copy detection application. The choice for this colour space is because of its practical application in this domain, as it is robust to illumination changes and quantization errors which are common when video is copied from one format to another.

Edge and texture information is another content feature description that is useful for detecting shot boundaries (Chan & Wong, 2011). In (Mohanta et al., 2012) they use an edge strength scatter matrix to distinguish between fade in/fade out, dissolve, wipe and cut shots by mapping a scatter matrix of the pre-normalized gradient magnitude of corresponding edge pixels of successive frames which reveals the type of frame transition.

Many works have used a hybrid technique in an effort to negate the disadvantages of one technique by using the strength of another. In (Grana & Cucchiara, 2007) they use a pixel based approach and histogram based approach in a unified linear transition decomposition. The iterative algorithm tries to determine optimal transition extremities and length using only these two parameters. In (Y. Chen et al., 2010) they use two algorithms for shot detection, as each one negates the disadvantages of the other. They use a combination of colour histogram difference (CHD) and edge change ratio (ECR) to identify different types of shot. CHD is used to identify abrupt change shots as the algorithm has a strong precision and recall in identifying this type of shot. Where it is weak in identifying transition shots they use ECR. ECR is not as strong in identifying abrupt shots as CHD but is extremely more effective at identifying transition shots.

(R. Tapu & T. Zaharia, 2011) use a graph partition scheme that represents each video frame as a node in a hierarchical structure that is connected with the other vertexes by edges. The weight of an edge, expresses the similarity between the corresponding nodes. They have adopted as a visual similarity measure the chi-square distance between colour histograms in the HSV colour space. To solve the invariant lighting and shading problem QR decomposition has been put forward as a solution(Amri & Fathy, 2010). They use QR decomposition to utilise three-dimensional histograms, split into 3×3 blocks in the RGB colour space of each frame as spatial features. They then use these histograms as a feature vector of each frame in the video, applying the QR decomposition to this matrix and incorporating the QR components of this matrix as temporal features along the frames. To distinguish between the shot transitions and the image differences caused by large camera or object motions, they model each shot transition by using a Gaussian model.

Now, a majority of video has deposited into compressed format So more studies on shot boundary detection are processed in compressed video streams from which the features are extracted such as discrete cosine transform coefficients (Mohanta et al., 2012) and motion vectors (Zhenyu & Zhiping, 2012). These features are extracted from the coded video bit stream. So the process of decode is omitted. The efficiency of algorithm in which feature is extracted from compressed video sequences is much better than those used in uncompressed video sequences. In (C. Ma et al., 2012) they only partially decompress the mpeg videos in order to obtain the I-frames. DC images are obtained by extracting DC coefficient of DCT coefficient in video code stream in the coarse phase of shot detection. In (Grana & Cucchiara, 2007) they design a linear transition model for SBD; their method is purely concentrated on gradual transitions with a linear behaviour. They utilized an accurate model which yields more discriminative power than with common methods.

Motion vectors (MV) are a new research area in their own right (Amel, Abdessalem, & Abdellatif, 2010). Motion is a salient feature in video, in addition to other typical image features such as colour, shape and texture.

An interesting method for video segmentation that uses geographical data to identify shots is proposed by (Wu, Liu, Wang, & Cai, 2012). This technique uses both Geographical Information System (GIS) and GPS data to segment shots from road cameras. The segmentation therefore is based on geographical metadata associated with the video file, which is generated at the time of filming.

	Cut	Gradual	Colour/Intensity	Edge/Texture/Pixel	Motion	Compressed	Uncompressed	SBD Technique(s)	Domain
Amel	x	x			x		x	MV	-
Amiri	x		RGB				x	QR-D/GTD	-
Amiri	x	x	RGB				x	GED-GTD	-
Babar	x	x	Greyscale				x	SURF	-
Bai	x		RGB				x	MuI	Video summary
Boyar	x		RGB				x	CHD	Sport
Chan	x	x		x			x	GA	-
Chen	x	x	HSV				x	HID	News
Chen	x	x	YUV/Greyscale	x	x	x		MPDT/FSM	-
De Bruyne	x			x	x	x		MCIPM	-
Dhillon	x	x	Greyscale	x			x	OC/SURF	-
Donate	x			x			x	SLAM	-
Feng	x	x	RGB				x	FCM	-
Grana		x	RGB	x			x	LTD	-
Hameed	x		YUV				x	WTAS	-
Jiang	x	x	YUV	x			x	ABS-SIFT	-
Krulikovska	x		RGB/YUV		x	x	x	MV/MoI	-
Küçüktunç	x	x	L*a*b*				x	FL	CBCD
Lee	x		RGB				x	SVD	News
Lei	x	x	HSV				x	DS	-
Li	x	x	RGB				x	ICA	Music Video
Li	x	x		x				SURF-SVM	-
Ma	x	x			x	x		DCT	-
Mendhi	x		YUV				x	SSIM	-
Mohanta	x	x	Greyscale		x		x	IHD/MM	-
Seidl		x	Greyscale				x	EF-LF	Historic Film
Sharmila Kumari	x			x			x	CSIFT	
Shekhar	x		HSV				x	LFT	-
Tuanfa	x	x		x	x		x	PI/MC	-
Wei	x	x	Greyscale				x	ST	-
Wenzhu	x	x	HSV				x	GT	-
Wu	x				x		x	GPS	GIS
Xu	x	x	x	x			x	FSHT/EC	-
Yongliang	x	x		x			x	KNN-SVM	-
Yu	x	x				x		MV	-
Zajić	x						x	MA	-
Zeinalpour-Tabrizi		x		x			x	FA	-
Zhang		x		x			x	NVF-SVM	-
Zhang	x		HSV				x	CHD	-

Table 1.1: Shot Segmentation algorithms; for hybrid systems the colours indicate which algorithm is responsible for which feature

1.3.2.3 Semantic Temporal Segmentation

An important step in the process of video structure parsing is that of segmenting the video into individual scenes or "logical units" (Mezaris, Sidiropoulos, Dimou, & Kompatsiaris, 2010; Sidiropoulos et al., 2011). Scenes are defined as "composed of one or more shots which present different views of the same event, related in time or space" (J. Hunter & Iannella, 2009). Shots describe actions or self-contained events that do not have much focus until they are put together to describe a larger story unit that are commonly called scenes. Shots have a physical boundary that is accurately detectable by computer vision processing methods, whereas scene are demarcated by semantic boundaries that are harder to detect by automatic methods. From a narrative point of view, a scene consists of a series of consecutive shots grouped together because they're related semantically, either spatially or temporally, or because they share some thematic content. Scenes are more conceptual in structure and therefore have a strong semantic dimension about them.

Video segmentation to shots and scenes are two different problems that are characterized by considerably different degrees of difficulty. State-of-the-art shot segmentation techniques, detecting the presence of video editing effects such as cuts and fades with the use of low-level visual features, have been shown in large-scale experiments (e.g., TRECVID[3]) to reach an accuracy that is close to perfect; this accuracy is deemed by the relevant community to be sufficient for any practical application (Smeaton, Over, & Doherty, 2010). Whereas scene segmentation has to take into account the semantic perspective of the content in order to temporally link shots into a scene. Due to the ambiguous nature of deciding the exact end and beginning of an event, scene segmentation is a more complex problem that has not enjoyed the same success rates as shot segmentation, nor the research focus.

Scene segmentation plays an important part in dissecting a large volume of video content into smaller semantic constituencies which are easier to digest. It is often used to create video summarisation of content into a more semantically concise form. This is often used to make a shorter trailer of the content that contains the more salient points of the content. This can be viewed by a consumer to see if the content is relevant to their requirements. Scene segmentation is also used for splitting factual, news or sports programmes into semantic units that portray a particular event. This is useful for search and personalisation of content. There is a close relationship between scene segmentation and event detection.

[3] http://www-nlpir.nist.gov/projects/tv2012/tv2012.html#med

The close relation between video scenes and the real-life events depicted in the video make scene detection a key-enabling technology for advanced applications such as event-based video indexing (Ballan, Bertini, Bimbo, Seidenari, & Serra, 2011). It also has uses in movie video summarisation (Sang & Xu, 2010), artistic video archives (Mitrović, Hartlieb, Zeppelzauer, & Zaharieva, 2010), news story classification (Aly, Doherty, Hiemstra, & Smeaton, 2010; Choroś & Pawlaczyk, 2010; Dumont & Quénot, 2012; Heejun & Jaesoo, 2011),sports video classification (Choroś & Pawlaczyk, 2010; del Fabro & Boszormenyi, 2010; Y.-F. Huang & Tung, 2010; Tjondronegoro & Chen, 2010), scene genre identification (Ellouze, Boujemaa, & Alimi, 2010; S. Zhu & Liang, 2011)

Much work has been done on scene segmentation in the last decade. They can be roughly classified into three categories.

- **Shot clustering based approach:** It is well known that video shots belong to the same scene are semantically similar. The similarities between the shots provide a basic clue for the clustering based approach. In (Choroś & Pawlaczyk, 2010) they cluster shots based on content features of shots of TV sports news broadcast. Evaluation has shown that studio shots and action shots are arranged in certain sequences within a scene that can be used to cluster the shots into scenes using a rule based methodology. In (del Fabro & Boszormenyi, 2010) they cluster shots into scene sequences by employing a distance similarity measure between shot clusters that compares motion information. Shot clusters that have similar motion histograms are clustered together in an iterative approach.

- **Boundary detection based approach:** In this approach, shot boundaries are considered as the candidates of scene boundaries and the false boundaries are removed by checking the coherence of the similarity between different shots. In (Baber, Afzulpurkar, & Bakhtyar, 2011) they detect fade and abrupt shot boundaries by frame entropy analysis and frame difference. Their hypothesis is that fade effects are usually found at the start or end of the scenes. Therefore, a fade-in is an indication of the beginning of the scene and fade-out indicates the end of the scene. (Dumont & Quénot, 2012) propose a fusion of content feature vectors that when analysed will show story segment boundaries where the multimodal vector shows a clear demarcation for most features.

- **Model based approach:** This approach views that to group N shots into K scenes is equivalent to estimating the model parameters $\{\Phi i\}K i=1$, which represent the boundaries of K

scenes. In (Chao, Changsheng, Jian, & Hanqing, 2011) they use a Hidden Semi-Markov Model (HSMM) to model the relationship between the script video alignment and video shot clusters to the hidden scene partition sequence.

Many methods have been developed to partition video scenes. Generally speaking, automatic scene boundary detection techniques can be categorized into following classes, i.e. graph based (Ayadi, Ellouze, Hamdani, & Alimi, 2012; del Fabro & Boszormenyi, 2010; Mezaris et al., 2010; Sakarya & Telatar, 2010; Sakarya, Telatar, & Alatan, 2012; Seeling, 2010; Sidiropoulos et al., 2011; Su, Bailan, Peng, & Bo, 2012; Ruxandra Tapu & Titus Zaharia, 2011), film editing technique based (Choroś & Pawlaczyk, 2010; S. Zhu & Liang, 2011), statistics learning based (Baber et al., 2011; Chao et al., 2011; Ellouze et al., 2010; S. N. Huang & Zhang, 2010; Mohanta, Saha, & Chanda, 2010; Sang & Xu, 2010; Seung-Bo, Heung-Nam, Hyunsik, & Geun-Sik, 2010; Tjondronegoro & Chen, 2010; Wilson, Divakaran, Niu, Goela, & Otsuka, 2010; Zeng, Zhang, Hu, & Li, 2010), and multi-features based (Dumont & Quénot, 2012; Ercolessi, Bredin, Sénac, & Joly, 2011; Heejun & Jaesoo, 2011; Y.-F. Huang & Tung, 2010; Hui & Cuihua, 2010; S. B. Li, Wang, & Wang, 2010; Mitrović et al., 2010; Poulisse, Patsis, & Moens, 2012).

Graph based techniques for shot detection have been very successful when employed in semantic scene segmentation. In (Sidiropoulos et al., 2011) they have proposed a technique, where the low-level and high-level features extracted from the visual and the aural channel have been used jointly. The proposed technique has been built upon the renowned method of the Scene Transition Graph (STG) for overcoming the difficulties of existing scene segmentation techniques. Firstly, a STG approximation has been introduced for reducing the computational cost, and then the uni-modal STG-based temporal segmentation technique has been extended to a method for multimodal scene segmentation. The latter has exploited the results of numerous TRECVID-type trained visual concept detectors and audio event detectors using a probabilistic merging process that merges several individual STGs while at the same time reducing the need for selecting and adjusting many STG construction parameters. Their proposed approach has been analysed using three test datasets, such as TRECVID documentary films, movies, and news-related videos. In (R. Tapu & T. Zaharia, 2011) they use a computationally efficient shot extraction method which adopts a normalized graph partition approach. This is enriched by using a non-linear, multi-resolution filtering of the similarity vectors involved. The groups are then iteratively clustered into visually similar shots, under a set of temporal constraints. Two different types of visual features are exploited; HSV colour histograms and interest points. (Sang & Xu, 2010) propose an effective method for video scene segmentation based Ncut to decompose the scene similarity graph into subgraphs (scene clusters). They generate a story flow graph (SFG) from the temporal relationships between scene clusters as nodes and transition probabilities between clusters as edges. Sub-story units are extracted by finding the cut edges of the SFG.

Scenes are just one of many film editing techniques that make up a lexicon of film grammar. This grammar itself can be used to identify scenes. The arrangement of content features and effects can be used to either cluster shots together or find the boundaries between scenes. Audio cues are just as important as video cues in detecting scene boundaries. In (Sidiropoulos et al., 2011) they jointly exploit low level features from both the visual and auditory channels. (Ercolessi et al., 2011) use speaker diarisation to segment TV series into scenes. Speaker diarisation is the process of segmenting an audio stream and clustering resulting segments in different speakers. The structure of a film is conceived before the first camera ever starts rolling. Scenes are created first by screen writers who produce a script of the screenplay. The script information itself can be combined with the footage to identify scenes. Both (S. B. Li et al., 2010; Seung-Bo et al., 2010) use the movie script, that has the scene information, and match it to the subtitle information of the footage. By synchronising the script information with the on screen subtitles they can identify the time points of the start and end of the scene boundaries.

The problem with film editing techniques is that they are used differently depending on genre and/or style of the filmmaker. This makes detection of scene boundaries using film-editing techniques, especially heavily dependent on the genre or film making style. A film editing technique in one genre or style will be totally ineffective in another. Using script subtitle synchronisation has a big drawback in that if you have large time periods without any dialogue then synchronisation will be inaccurate at best, and at worst, impossible.

The solution to the problems of relying on one set of features is to use a multi-feature based approach. For example (Chao et al., 2011) combine script names with faces in the video to negate the problems mentioned before, along with the discrepancies between the script and subtitles and the scarcity of subtitles in non-English speaking languages. In (Poulisse et al., 2012) they use a similar technique for live sports action. As there is no script they use subtitles, which are time coded already, and extract SIFT features to produce multi-content type chains that identify scene boundaries through density graphs. This still relies on textual information being available and the accuracy of the transcription of the subtitles but using the SIFT features allows similarity matching of shots and overcomes the scarcity of subtitles problem. In (Dumont & Quénot, 2012) they use numerous visual and audio features and fuse them together after applying a local temporal context window to them. The results are then analysed by various machine-learning algorithms, of which Random Forest had the best F1 score for detected scene boundary detection. A mixture of syntactic features can be used to cluster shots together for certain domains, as in (Mitrović et al., 2010). Using block based intensity histograms (BBH), Edge change ratio (ECR) and SIFT keypoints they build up an orthogonal view of visual information that represent intensity, edges and salient keypoints. This captures a larger spectrum of visual similarities that can be used in identifying shot clusters using similarity measures.

Although semantic scene segmentation is considered to be a concept based problem that requires the visual understanding of the content semantically, implying that all video streams need to be uncompressed, information from the compressed domain can be used to understand spatial relationships that can be of use in identifying scene boundaries or shot clusters. In (del Fabro & Boszormenyi, 2010) they extract the motion information from H.264/AVC compressed video that are used to create motion histograms that are one of the features that are used in the scene motion classification pattern matching.

A general problem in semantic temporal segmentation is the dependence of techniques on domain or genre. For example in sports video annotation they still suffer from two important drawbacks: 1) a definitive scope of events detection and annotation (i.e., where to start and finish the extraction) and 2) the lack of a universal set of features for detecting different events and sports. (Tjondronegoro & Chen, 2010).

1.3.2.4 Spatiotemporal segmentation

Spatial segmentations aim to group image pixels together based on attributes that define a pixel region into a semantic object. Spatiotemporal segmentation takes this one step further by adding a temporal element to the segmentation by tracking the pixels over time and defining the object in both appearance and motion. Spatiotemporal segmentation is often described as 3D segmentation because of the temporal dimension (Fei & Zhu, 2010; Grundmann, Kwatra, Mei, & Essa, 2010; Sharir & Tuytelaars, 2012; Tian, Xue, Lan, Li, & Zheng, 2011; Vazquez-Reina, Avidan, Pfister, & Miller, 2010). This should not be confused with stereo camera based object segmentation that is used in the surveillance domain, where 3D video data is used to segment the objects using depth (Ghuffar, Brosch, Pfeifer, & Gelautz, 2012; Y. Ma & Chen, 2010; Van den Bergh & Van Gool, 2012). Spatial segmentation differs from spatiotemporal segmentation in that temporal coherence of the object boundary maybe compromised when segmenting a series of contiguous frames as they are treated in isolation and redefine the object boundary for every frame (Grundmann et al., 2010).

Objects can be defined at several levels, from general geometric boundaries, such as bounding boxes(Babenko, Ming-Hsuan, & Belongie, 2011) to regional granularity (Grundmann et al., 2010). The best balance is achieved when object are segmented into regions that can be easily recognised by humans (Grundmann et al., 2010; Ladický, Sturgess, Alahari, Russell, & Torr, 2010; Ochs & Brox, 2011). These should follow a hierarchical structure based on perception. For example, a person can be segmented into arms, torso, arms and legs (Shao, Ji, Liu, & Zhang, 2012). The arm can then be split into upper arm, elbow, forearm and hand. This segmentation along semantic understanding of objects is the most natural and easily relatable.

A number of spatiotemporal segmentation algorithms have been proposed in recent years. The most popular approach employed is that of Optical flow, a time-domain motion analysis algorithm (Ghuffar et al., 2012; Lezama, Alahari, Sivic, & Laptev, 2011; Lin, Zhu, Fan, & Zhang, 2011; Ochs & Brox, 2011; Sharir & Tuytelaars, 2012; Tian et al., 2011; Van den Bergh & Van Gool, 2012). The optical flow method models the physical properties of optical flow that the moving objects change over time to subtract the moving object effectively. The basic optical flow equation is given by:

Eq. (1.3) $$I_x\mathrm{u} + I_y\mathrm{v} + \mathrm{I}_t = 0$$ (Ghuffar et al., 2012)

where I_t is the image difference between the two images, and $I_x\mathrm{u}$ and $I_y\mathrm{v}$ are image derivatives. Its advantage is that it can also segment the independent moving object under the condition of camera motion. Its vulnerabilities are to image noise, colour, and non-uniform lighting, also most of flow computation methods have large computational requirements that make them unsuitable for real time processing and are sensitive to motion discontinuities. There are other types of Motion Analysis techniques apart from Optical flow such as (Christodoulou, Kasparis, & Marques, 2011; Fei & Zhu, 2010; Porikli, Bashir, & Huifang, 2010) but are very similar in their machinations.

Conditional Random Fields (CRF) and Markov Random Fields are techniques that have recently been gaining popularity for spatiotemporal segmentation . In (Vazquez-Reina et al., 2010) they use a multiple hypothesis video segmentation technique that generates multiple pre-segmentations per frame into multiple hypothesis and finds sequences of superpixels (shown as coloured regions) that match consistently in time. Each of these sequences, called a superpixel flow, is ranked depending on its photometric consistency and considered as a possible label for segmentation. The processing windows overlap one or more frames to allow labels to propagate from one temporal window to the next. They use higher-order conditional random fields (CRFs), which they use to solve the hypothesis competition problem. They define the higher-order conditional random field on a sequence of fine grids of superpixels $S = \{S_1, \ldots S_f\}$. Each grid S_t is obtained as the superposition of the P tessellations that were generated for the enumeration of hypotheses. The mapping g_t takes superpixels v_t from one of the pre-segmentations to the superposition S_t. Each superpixel in St is represented in our CRF with a random variable that can be labelled with one of the hypotheses $\{H_1, \ldots, H_L\}$. In (Subudhi, Nanda, & Ghosh, 2011) they propose an edge-based compound MRF model for attribute modelling of video image frames followed by the maximum a posteriori probability (MAP) estimation by a hybrid algorithm (hybrid of both simulated annealing (SA) and iterated conditional mode (ICM)). The compound MRF uses spatial distribution of colour in the current frame, colour coherence in the temporal direction and edge maps in the temporal direction. The difference images obtained from the given video frames are largely affected by illumination variation and noise that propagates in the form of silhouette to the VOP. They then use an adaptive temporal segmentation scheme that reduces the effect of noise. Instead of segmenting the whole image at a time by a single threshold, they partition the input image into different windows/blocks and segment the objects in each of these windows. Then they combine the segmented objects from each window. The window size is determined by the entropy content of the considered window.

So far we have looked at techniques that track an object over a moving background. Numerous works have looked at modelling the background first and then detecting the pixels of foreground objects by differencing the current frame with the background. This approach is only effective if the camera is stationary or has a background that is unchanging. These techniques are obviously suited to the surveillance domain of CCTV (Appiah, Hunter, Dickinson, & Meng, 2010; Bai, Wang, & Sapiro, 2010; Ladický et al., 2010; Y. Ma & Chen, 2010). A wide and increasing variety of techniques for background modelling have been described. The most basic way to do this is by using a frame difference techniques such as in (Christodoulou et al., 2011). This looks at the temporal difference in pixels across frames that identify moving object pixels across a non-moving background of pixels. The algorithm utilises statistical quantities such as mean, standard deviation, and variance to define an adaptive and automatic threshold based on two-frame and three-frame differencing using automatic and adaptive statistical thresholding techniques for motion object detection. It dynamically adapts to environmental conditions by making use of the previous frame, as the current background model. However, temporal differencing works well only if the motion is small. It is common that methods only detect the outlines of regions of interest, which usually leads to generating holes inside moving entities.

The most popular method for background modelling is a unimodal approach that uses Gaussian Mixture Model's (GMM)(Q. Zhu, Xie, Gu, & Wang, 2012). It constructs a grayscale distribution model of each pixel based on the distribution information of each pixel in time domain and builds a background model of the pixels. This technique and other GMM techniques (Bai et al., 2010; Subudhi et al., 2011) are used as they have relatively low computational cost and memory requirements. This technique gives poor results when used in modelling non-stationary background scenarios like waving trees, rain and snow. In (Appiah et al., 2010) they use a multimodal approach, modelling the values of each pixel as a Mixture of Gaussian (MoG). The background is modelled with the most persistent grey scale intensity values. The equation is given as:

Eq. (1.4) $$P(X_t) = \sum_{i=1}^{K} \omega_{i,t}\, \eta(X_t, \mu_{i,t}, \sigma_{i,t})$$ (Appiah et al., 2010)

Where $\mu_{i,t}$, $\sigma_{i,t}$ and $\omega_{i,t}$ are the respective mean, standard deviation and weight parameters of the *i*th Gaussian component of pixel Xat time t, and η is a Gaussian probability density function:

Eq. (1.5) $$\eta\left(X_{t,}, \mu_{i,t}, \sigma_{i,t}\right) = \frac{1}{\sigma_{i,t}\sqrt{2\pi}} \exp\left(\frac{\left(X_t - \mu_{i,t}\right)^2}{2\sigma_{i,t}^2}\right)$$ (Appiah et al., 2010)

A new pixel value is generally represented by one of the major components of the mixture model, and is used to update the model. This technique though a more powerful alternative to GMM's requires more computing power due to its multimodal nature and therefore is unsuitable for real time performance. There are also disadvantages including the blending effect, which causes a pixel to have an intensity value which has never occurred at that position (a side-effect of smoothing).

With the advent of stereoscopic cameras and the emergence of 3D video, techniques have been developed that take advantage of the depth field to provide spatiotemporal segmentation. In (Y. Ma & Chen, 2010) they have used a stereoscopic camera to integrate depth information into the object segmentation process. They produce a 3D depth density image from the disparity map and then apply a region growing method to segment foreground objects. In (Ghuffar et al., 2012) they use motion estimation and segmentation of independently moving objects in video sequences from a time of flight range camera that can record depth. They present a motion estimation algorithm which is based on fusion of range flow and optical flow constraint equations. The flow fields are used to derive long-term point trajectories. A segmentation technique groups the trajectories according to their motion and depth similarity into spatiotemporal objects. In (Van den Bergh & Van Gool, 2012) the authors use a real-time superpixel segmentation algorithm, which employs real-time stereo and real time optical flow. The system provides superpixels that represent suggested object boundaries based on colour, depth and motion. Each outputted superpixel has a 3D location and a motion vector, and thus allows for straightforward segmentation of objects by 3D position and by motion direction. In particular, it enables reliable segmentation of persons, and of moving hands or arms.

To reduce computational expense a few works have tried to segment video without decoding the signal from its compressed state (Fei & Zhu, 2010; Khatoonabadi & Bajic, 2013; Porikli et al., 2010; Tsao, 2011). The approaches used in the compressed domain make use of the data from the compressed video bit stream, such as motion vectors (MVs), block coding modes, motion-compensated prediction residuals or their transform coefficients, etc. In practice, some, but not necessarily all, of the information from the bit stream needs to be decoded. There are two main types of cues: motion vector (MV) and discrete cosine transform (DCT) coefficients, which can be derived in the process of video coding or extracted by partially decoding the MPEG-compliant compressed videos (Fei & Zhu, 2010). These algorithms can be classified as following three classes: (i) DCT domain segmentation, which exploits texture characteristics of DCT coefficients for segmentation (Tsao, 2011); (ii) MV field segmentation. In this case, the spatial and temporal information of MVs were used for segmentation (Fei & Zhu, 2010); (iii) joint DCT and MV domain segmentation(Porikli et al., 2010). Another approach to reducing computationally efficiency is by using hardware based techniques to speed up the processing of complex algorithms. In (Appiah et al., 2010) they process a multimodal background differencing algorithm on a single Field Programmable Gate Array (FPGA) chip and four blocks of RAM. The real-time connected component labelling algorithm, also designed for FPGA implementation, run-length encodes the output of the background subtraction, and performs connected component analysis on this representation. The run-length encoding, together with other parts of the algorithm, is performed in parallel; sequential operations are minimized as the number of run-lengths are typically less than the number of pixels. The two algorithms are pipelined together for maximum efficiency.

Temporal continuity of the spatiotemporal segmentation regions can only be achieved by tracking the object boundaries over the duration of a shot. In (Vazquez-Reina et al., 2010) they extract multiple segmentation hypotheses of superpixel flows in each frame, and then search for a segmentation consistent over multiple frames. Robust unsupervised video segmentation must take into account spatial and temporal long-range relationships between pixels that can be several frames apart. Segmentation methods that track objects by propagating solutions frame-to-frame (Yongquan et al., 2009) are prone to overlook pixel relationships that span several frames.

Some of the problems faced in spatiotemporal segmentation include occlusion (Ayvaci & Soatto, 2012). Local image measurements often provide only a weak cue for the presence of object boundaries. At the same time, object appearance may significantly change over the frames of the video due to, for example, changes in the camera viewpoint, scene illumination or object orientation (Lezama et al., 2011). Due to occlusions, objects often merge and split in multiple 2D regions throughout a video (Vazquez-Reina et al., 2010).

This problem also relates to unsupervised segmentation of arbitrarily long videos that require the automatic creation, continuation and termination of labels to handle the free flow of objects entering and leaving a scene (Vazquez-Reina et al., 2010). Such events are common when dealing with natural videos with arbitrary camera and object motion. A complete solution to the problem of multiple-object video segmentation requires tracking object fragments and handling splitting or merging events.

Two or more syntactic features are used to segment objects. This hybridisation is applied in two ways; by combining techniques that use different features symbiotically to segment the object or use different features to independently segment the object and use the results from one to reinforce the other. Examples of a symbiosis technique is (Bai et al., 2010) where they use motion estimation as a probability framework of object localisation and then adapt the selection of colour model from global to localised for different parts of the object so successive frames can be easily segmented. The work from (Hu & Hsu, 2011) is an example of the second type that uses different syntactic features to extract and then reinforce object segmentation. They combine all three feature classes; colour, motion and edge information to extract foreground objects. The proposed method uses a coarse to fine segmentation approach for object segmentation. They begin by extracting the motion information of the object using the angle module rule (Carmona, Martínez-Cantos, & Mira, 2008). Then a coarse moving object motion mask is obtained using the motion and gradient variation information. Compensation for still regions in a moving object, noise elimination, morphological processing and connected component labelling are used to provide a fine moving object mask. Finally, moving object region refinement is achieved by combining the object boundary refinement with region growth/compensation performed by Sobel edge detection.

Sometimes the same feature can be used by two different techniques to reinforce each other. For example, they use pixel intensity values in (Mahesh & Kuppusamy, 2012) with both frame difference algorithm and intersection of frame algorithm to extract objects for two different scenarios. The motion segmentation process is carried out by both the frame difference algorithm and intersection method and subsequently the most common and accurate segmented objects are retrieved from both the segmented results whereas the static foreground are segmented using the intersection of consecutive frames.

In Table 1.2 the state-of-the-art spatiotemporal segmentation methods are presented. It shows which content features are used for extraction, whether it is supervised or unsupervised, The name of the algorithm and the domain of use.

Author	Colour/Intensity	Edge/Texture/Pixel	Motion	Supervised	Unsupervised	SS Technique(s)	Domain	Notes
Appiah	x			x		GMM	CCTV	MoG
Avidan		x		x		AdaBoost	-	
Ayvaci			x		x	LP	-	
Bai	x		x	x		GMM	CCTV/Sports	DCF
Brosch	x			x		CVF	-	
Christodoulou			x		x	FDT	CCTV	Pre-processing
Elminir	x	x	x		x	HLSC	-	
Fathi		x		x	x	GT	Biotracking	
Fei			x		x	MV	-	Compressed
Ghuffar			x		x	GT	Flight Cameras	Uses 3D
Grundmann	x				x	GT	-	
Hosten			x	x		GT	-	
Hu	x	x	x		x	GVD/ED/MD	-	
Ladický		x			x	CRF	CCTV	
Lezama			x		x	OF	-	GBS
Lin			x		x	MRF	CCTV	Pre-processing
Ma			x			MD	CCTV	Stereo based
Mahesh	x				x	FD/IF	Sport	Hybrid
Nagahashi	x			x		GT	-	GMM
Ochs			x		x	LTTP	-	
Phan	x			x		CA	-	Biological
Porkili			x		x	VGCDS	-	Compressed
Price	x			x		FGT	-	
Sharir		x			x	CRF	-	
Shao			x		x	MA	Sports	
Subudhi		x			x	MRF	CCTV	
Tian	x		x	x	x	GMM/MV	CCTV/Sports	Video Summary
Tsao		x			x	GT	CCTV	Compressed
Van den Bergh	x		x		x	SLIC	CCTV	Stereo based
Vazquez-Reina		x			x	CRF	-	MHVS
Yongquan		x			x	IFM	Sports	
Zhu		x		x		GMM	CCTV	

Table 1.2: Spatiotemporal Segmentation algorithms; for hybrid systems the colours indicate which algorithm is responsible for which feature

1.3.3 Semantic Relationships

The semantic information used to identify video events has two important aspects (Marios C. Angelides, 2003). They are: (a) A spatial aspect presented by a video frame, such as the location, characters and objects displayed in the video frame. (b) A temporal aspect presented by a sequence of video frames in time such as the character's actions and the object's movements presented in a sequence.

There has been much work recently on concept detection (Weiming et al., 2011). This has mostly been based on feature fusion and classifier fusion, which use syntactic feature sets for detection. The semantic feature based methods of concept detection are based on modelling relations. The two semantic features that need to be modelled are spatial and temporal relationships. Spatial relationships exist only between spatiotemporal regions and can evolve over time. Temporal relationships can be modelled between all features, syntactic or semantic, as all video features have a temporal component.

1.3.3.1 Spatial Relationships

Over the past years, the representation of spatial relationships in video has been extensively discussed (Weiming et al., 2011).

One of the most important abilities of a semantic content model should be to be able to query the position of objects in relation to other objects or their relative positioning within the shot, not just as a reference to their absolute positioning stated as coordinates (Agius & Angelides, 2005). Consumers of content can query using simple relationships between objects as "A is left of B", and also state by inference the inverse relationship "B is right of A". Relative positioning can be given using 8 point compass direction such as "North" or "South-East". Spatial relationships therefore can be an important tool in semantic querying of content.

Unlike spatial relationship in images, spatial relationships in video have a temporal dimension. Temporally consecutive frames have explicit spatial constraints with object inheritance, spatial relationships and motion information from their previous frames. Temporal trajectories of spatial relations among objects are as important as temporal object trajectories to represent object activities and reveal semantic evolution of spatial properties over time.

Unfortunately, spatial relationships have not been adequately addressed in most video indexing systems despite their obvious importance, and where they have been they have not been explicitly derived (Bas̨tan, C̨am, Gu̇du̇kbay, & Ulusoy, 2010; Kannan, Andres, & Guetl, 2010; Vrochidis et al., 2010). In such systems, indexing techniques work on modelling video by treating video shots/scenes as collections of still images, extracting relevant key-frames, and comparing their low-level features.

Spatial relationships were formalised using Allen's temporal logic as a basis (Güsgen, 1989). Spatial relationships between objects describe the relative location of objects in relation to other objects (rather than their absolute screen coordinates) within the segment. Spatial representations aren't an alternative to screen coordinates; they complement them. Sometimes when it's difficult to derive screen coordinates, a spatial relationship is the only way to model an object's presence. The spatial relationships between two objects may differ over time within the same segment. In Table 1.3, we see the spatial relationships that are defined in MPEG-7 (Manjunath, Salembier, & Sikora, 2002).

RELATION	INVERSE RELATION
SOUTH	NORTH
WEST	EAST
NORTHWEST	SOUTHEAST
SOUTHWEST	NORTHEAST
LEFT	RIGHT
RIGHT	LEFT
BELOW	ABOVE
OVER	UNDER

Table 1.3: NORMATIVE SPATIAL RELATIONSHIPS IN MPEG-7 (MANJUNATH ET AL., 2002)

Spatial relationships are a highly active field in other domains such as Content based information retrieval (Singhai & Shandilya, 2010), Human activity classification (Ryoo & Aggarwal, 2009), Robotics (Rosman & Ramamoorthy, 2011) and Surveillance (Ryoo, Lee, & Aggarwal, 2010).

1.3.3.2 Temporal Relationships

One of the most important distinctions between semantic querying of video rather than images is the temporal dimension. Semantically queries can be structured to investigate the temporal relationships not only between syntactic features, but also semantic features. Temporal relationships between these features allow the content model to express dynamism at the higher level (Agius & Angelides, 2005).

Temporal relationships were first defined meaningfully by J.F. Allen (Allen, 1983) in his quintessential paper on temporal intervals. He stated that temporal intervals should be able to be represented imprecisely using a strictly relative nomenclature. He also stated that representation should allow for the uncertainty of temporal information. Often, the exact relationship between two times is not known, but some constraints on how they could be related are known. The representation should also allow one to vary the grain of reasoning. For example, when modelling knowledge of history, one may only need to consider time in terms of days, or even years. When modelling knowledge of computer design, one may need to consider times on the order of nanoseconds or less. Finally, the model should support persistence. It should facilitate default reasoning of the type, "If I parked my car in lot A this morning, it should still be there now," even though proof is not possible (the car may have been towed or stolen). Allen's scheme for temporal relationships was expanded on by the MPEG-7 group (Manjunath et al., 2002), as shown in Table 1.4.

Binary	Inverse Binary	N-ary
Precedes	Follows	Contiguous
Co-occurs	Co-occurs	Sequential
Meets	Met by	Co-being
Overlaps	Overlapped by	Co-end
Strict during	Strict contains	Parallel
Starts	Started by	Overlapping
Finishes	Finished by	-
Contains	During	-

Table 1.4: Normative Temporal Relationships in MPEG-7 (Manjunath et al., 2002)

1.3.4 Content Modelling

Once features are extracted they need to be described so that text-based search engines can access the content descriptions. Each feature that is extracted needs to modelled into a content description that describes the syntactic and/or semantic properties of that feature. Once the content features are modelled they themselves need to be modelled into a single document, called the content model, that structures the content descriptions into a logical arrangement of content features that describes a video stream in term of what can be seen and heard, and what that means. The content model provides a content description "proxy" of the content contained within a video stream, and indexes the content to recreate the visually salient points of the content that would be of interest to users to formulate queries with.

In this section we will examine which features need to be modelled that are at the core of describing a video stream. These core features should be included in a content model as they cover the majority of features that are generally queried by most video content search applications. The next section looks at the requirements of applications that require content models in terms types of query and how state of the art content feature extraction and modelling applications fill those requirements. The last section looks at different multimedia content descriptions standards and focuses on MPEG-7 as a complete multimedia content description interface for creating content models.

1.3.4.1 Modelled Features

Content models must represent the content of a video stream in a complete and detailed manner. The content features must be described in both structurally syntactic and a semantically meaningful terms, concisely and comprehensively (Moens et al., 2012). The types of descriptors and the granularity of the description scheme have a direct impact on the usefulness of the content model to different domains and consumers. This leads to the issue of interoperability of the content model across multiple platforms and applications and vendor and propriety independence (Haslhofer and Klas, 2010). The content model must ideally be available to be used for as many purposes as possible. A content model must be structured in an explicit manner that must represent the content as a proxy that describes the information within that content in a complete and comprehensive manner. To achieve this the syntactic and semantic content descriptions that make up the content model must be integrated so that a symbiosis of structure and concepts within the content become manifest.

Extracting the content features into a content model in a structured format that unlocks the semantic meaning of the content within the perspective of consumers is the main goal of content modelling. This is commonly referred to as the semantic gap, which is the difference between what a user perceives as the meaning of the content (semantics) to what can be extracted using machine based indexing methods (syntactic) (Küçük and Yazıcı, 2011). This is one of the main problems in designing video indexing and retrieval systems that can effectively support semantic querying that can be translated and mapped to annotated semantic features. The choice of what and which content features to model can have an impact on the effectiveness and use of the resulting content model. The content model must contain features that users query regular. Looking at the type of query requirements for video identifies the core content features that need to be modelled to enable the content model to be utilised by a wide array of content based video search engines.

The content features that are modelled must include all levels of the content feature hierarchy. The content features can be categorised into groups depending on their structural and/or conceptual attributes. (Baştan et al., 2010) stated that user queries could be categorised into four categories, but also stated this list was not exhaustive. Related works (Marios C. Angelides, 2003; Inigo & Suresh, 2012; Lavee et al., 2009; Mezaris et al., 2009; Moens et al., 2012; Ren, Singh, Singh, & Zhu, 2009; Smeaton et al., 2010; Snoek & Worring, 2009; Weiming et al., 2011; Shiping Zhu & Guo, 2012) have categorised all the query types based on what type of content the query addressed. These query categories can be classed into categories based on their syntactic or semantic nature. The queries are categorised into four classes:

- Low level syntactic queries – "Query by example" that is used for features that are easily processed by automatic feature extraction, such as images or video. The queries return multimedia results that have the same similarity in structural features such as colour, shape, texture and/or motion.

- Mid-level syntactic queries – can be queried by providing examples or can use keyword based querying to find perception-based and spatiotemporal syntactic features such as scenes and objects.

- High level semantic queries – Text based queries, expressed as natural language or keyword based, that are based on the human understanding of content. These queries might also incorporate narrative and context of the requirements of the results needed. This returns results that have high level semantic concepts, such as events, actions and conceptual relationships.

- Combination of all the above – A mixture of all or some of the above query types.

From the above query examples we can see the need for a content model to support these types of query will need to have similar content features that match up to the content feature queries. The core content features proposed by (Marios C. Angelides, 2003) contain four content feature classes, spatiotemporal objects, spatial relationships, event segments and temporal relationships. These four basic categories only cover the mid-level, high-level and semantic relationships requirement for content based video querying. Another category must be added for the low level syntactic features. This category is the temporal segments. The addition of the fifth content feature group makes the content feature classification complete. The expanded content feature descriptions are below:

- Spatiotemporal objects – objects that are depicted in the content and are tangible and have a causal effect on the semantics of the content by creating or changing events

- Spatial relationships – the spatial relationship between objects and how they change over time

- Temporal segments – A video segment or clip that depicts a single action or instance of an object that is part of an event.

- Event segments – video segments or clusters of video clips that depicts events involving the objects

- Temporal relationships – the temporal ordering between the different content features

The difference between the temporal segment and the event segment is that the temporal segments action may not have a substantive semantic meaning of its own e.g. it may just be a simple action of camera movement such as a pan shot. The event segment has a definitive semantic meaning to it. The event segment could be potentially be made of many temporal segments, who's individual actions add up to an event. Conversely a temporal segment on its own could have a definitive semantic occurrence (e.g. car crashing) but this does not necessarily mean that it is an event as there could have been other actions that complete the event (e.g. car tyre blows out).

These four basic categories covers most types of syntactic and semantic content features that are to be modelled for a comprehensive, granular and richly described content model. If we take these four content feature groups and match them up to the content feature query groups we get the following in Table 1.5:

CONTENT QUERY GROUP	CONTENT FEATURE GROUP
LOW-LEVEL SYNTACTIC QUERIES	TEMPORAL SEGMENT
MID-LEVEL SYNTACTIC QUERIES	SPATIOTEMPORAL OBJECTS, EVENT SEGMENTS
HIGH-LEVEL SEMANTIC QUERIES	SPATIAL RELATIONSHIPS, TEMPORAL RELATIONSHIPS

Table 1.5: CONTENT QUERY REQUIREMENT GROUP VS. CONTENT FEATURE GROUP

The low level syntactic queries are fulfilled by the temporal segments, which depict frames and shots, the fundamental building blocks of video. Frames are used to represent a snapshot of a temporal segment or another features semantic context, they have no value within themselves as a feature, syntactically or semantically. The spatiotemporal objects represent the mid-level syntactic features needed for querying and event segments. Spatiotemporal objects are a mid-level syntactic feature as they represent a moving region of interconnected pixels that are semantically related to be described as an object. Events are usually described as "scenes" in video nomenclature. A scene is a syntactic feature that defines a temporal segment of video that is a collection of other temporal segments that are semantically related. Spatial and temporal relationships are semantic relationships that represent the interaction between features that give the event meaning. High level queries can use the semantic relationships to assess what type of event has occurred by being able to query when it occurred in relation to other features and what the interaction of the objects where.

So with the addition of the temporal segments all the video content query requirements of the four content type query groups are met by the five content feature groups. Content models that possess the five content feature groups should be able to provide results to any content based video query that is formulated with any combination of feature requirements. In the next section current state-of-the-art content modelling applications are examined to see how well they fulfil the content model requirements stated.

1.3.4.2 Content Modelling Applications

As described in the previous section content feature extraction and modelling applications need to extract and model certain core content feature descriptions that will create a content model that can fulfil the requirements of the majority of content based video queries. In Table 1.6 we have the state-of-the-art content feature extraction and modelling systems that are the main players in video content extraction and modelling. In the table the content feature query groups are compared to each system and the features it extracts and models for that feature stated. The domain of use is also noted, as is the standard used to describe the content model, if any, is used to describe the content feature is also provided.

At present supervised, semi-supervised and unsupervised content modelling prototypes and systems index content to a content model offline as it is a time consuming, laborious (in the case of supervised and semi-supervised methods) and computationally expensive process. The content models are then used as "proxy" for the original video stream. Some systems do not produce a full content model but produce descriptions of certain content features that are relevant for the needs of their domain or use. Not all content features are extracted by the system, indeed even those within each content feature category the actual content feature sets extracted can be quite different. The content features extracted largely depend on the extraction method of the content features and how the content features are to be used. The nuances of each system is described in an overview that follows the table and discusses the main function of the system and the content features, and how they’ve implanted their content modelling strategy, discussing the pros and cons of the strategy.

	Content Features				Indexing Technique/s	Type	Domain	Standard	Content Model
	Low	Mid	High	Relationship					
MKLAB	Visual	Concept			SIFT/Text	Semi-Supervised	Generic	MPEG-7	No
Dyana	Motion	Object			CSS/PC	Unsupervised	Generic	MPEG-7	No
Bilvideo-7	Shots, Keyframes,	Objects		Spatial, Temporal	CHD/Manual	Semi-Supervised	Generic	MPEG-7	Yes
DanVideo		Objects, Actors, Agents	Mood	Spatial, Temporal	Manual	Supervised	Dance	MPEG-7	Yes
OVIDIUS	Shots, Keyframes	Scenes, Speech			BOW/CHD/Speaker Diarisation/HMM	Unsupervised	Documentary	MPEG-7	Yes
SHIATSU	Shots		Concept			Unsupervised	Landscape	No	No
VERGE	Shots, Keyframes	Speech	Concepts	Temporal (Limited)	SIFT/ASR/BOW/SVM	Unsupervised	Surveillance	MPEG-7	No
Zavřel	Keyframes				MPEG7-XM	Unsupervised	Generic	MPEG-7	No
LawTo	Thumbnails	Persons, Speech	Concepts		ASR/NLP	Unsupervised	News	XML	Yes
XUNET	Shots, Keyframes		Concepts		Manual	Supervised	Movies/TV	MPEG-7	No
VisionGo	Shots, Keyframes	Speech	Concepts		Manual/ASR/ML	Semi-Supervised	News	No	No
Anthropos-7	Shots, Keyframes, Motion, 3D	Objects, Actors	Concept		Manual/Automatic	Semi-Supervised	Movies/TV	MPEG-7	Yes
GOS (HCT)	Keyframes				Automatic	Unsupervised	Generic	MPEG-7	Yes

Table 1.6: Automatic video indexing Systems

In (Baştan et al., 2010) they have developed an MPEG-7-compatible video feature extraction and annotation tool called Bilvideo-7. Low-level syntactic features are automatically extracted using a hierarchical temporal/spatiotemporal decomposition methodology. This segments the content features described into independent, but linked, descriptions that can be queried easily by two of the categories of content-based retrieval queries, low-level and mid-level syntactic features. Semantic labelling of these features is then added manually as to be able to query high-level semantics. The main low level features of interest that are modelled are shots, key segments (represented by keyframes) and objects (with backgrounds). It can formulate multimodal queries using the BilVideo-7 visual query interface that can support all three types of querying and also the fourth category, spatial and temporal relationships. Spatial and temporal relationships aren't stated explicitly in the content model but can be queried during the query processing stage. This lack of an explicit structure for spatial and temporal relationships means that a content model produced from BilVideo-7 cannot be used for querying of those relationships by other systems without pre-processing first. The major bottleneck in the system is that it has to be manually annotated. This can lead to errors in indexing, is time consuming and prone to unreliable human translation.

In (Bursuc, Zaharia, & Prêteux, 2012) they have developed an Online Video Indexing Universal System (OVIDIUS), which is an online video browsing and retrieval platform. The client is a web-based interface with the querying performed on an MPEG-7 search engine server using a content management system that extracts and stores the MPEG-7 feature descriptions. They adopt a hierarchical approach to video segmentation, i.e. video, scenes, shots, speech segments and keyframes. The main capability OVIDIUS has over other systems is that all segmentation is processed automatically. The low level semantic features are extracted using established extraction techniques, which are very reliable. For instance the shots are segmented by a colour histogram difference (CHD). Semantic annotation is added by analysing associated text from the transcription process as local semantic features and items such as title and synopsis as global features. OVIDIUS does not extract objects and therefore spatial relationships. It also does not explicitly support temporal relationships either.

In (Kannan et al., 2010) they have developed an MPEG-7 authoring and retrieval system for dance called DanVideo. They use a video annotator that has two parts, a macro and micro annotator, to index the raw media. The macro annotator is used to describe global semantic features of the video such as dancers, musicians, music, accompaniments, background, tempo of the dance steps (slow, medium, or fast), dance origin, dance type, context (live, rehearsal, professional play, competition, etc.). The micro annotator is used by the dance choreographer to annotate the dance steps in each song of the video. This annotator provides instantiations of the global descriptions with additional descriptions that describes the local conditions of the local content features. The micro annotator is also used to describe the spatial and temporal relationships. The annotations are first stored as hash tables and vectors as intermediate description data store. These descriptions are then turned into MPEG-7 D's and DS's by the MPEG-7 instance generator. DanVideo uses standard description tools from MPEG-7 and does not use the DDL, making it highly compatible with all MPEG-7 compatible systems. DanVideo has a very detailed and complete content model in regards to high level semantic features and mid-level syntactic features but does not have any low-level syntactic features. This is not a problem in the dance domain where this system is intended for use but would present a problem in other domains or generic use.

(Bartolini, Patella, & Romani, 2011) proposed a technique for automatic semantic-based hierarchical indexing of videos, called SHIATSU (Semantic Hierarchical Automatic Tagging of videos by Segmentation Using cuts). Shots are extracted using a double dynamic threshold system that implements a hybrid HSV based CHD and ECR. Both techniques are used for detecting cut shots but only ECR is used for detecting transition shots. This hybrid technique produced better recall and precision results than the reference technique. After the shots are segmented a keyframe is extracted from each shot to be annotated with semantic tags. Visual features are extracted from the keyframes and are compared using an M-Tree metric against pre-defined semantically annotated images in a knowledge base of concepts. The concepts can either be structured in hierarchical tree shaped taxonomy or in a flat structure. All keyframes once indexed are added to the knowledge base to improve accuracy and quality of the semantic tagging. SHIATSU is a fully automated system but initially requires a pre-defined knowledge base that has to be accurate for tagging to achieve precision. It also does not provide tagging of mid-level syntactic features, high-level semantic features or spatial and temporal relationships. SHIATSU showed a high accuracy rate for datasets of landscapes but not for generic content. It also does not produce a standardised content model and therefore its semantically indexed database is not easily accessible or usable.

In (Vrochidis et al., 2010) they introduce VERGE, a video interactive retrieval engine which combines indexing, analysis and retrieval techniques in various modalities (i.e. textual, visual and concept search). It extracts low level syntactic features such as shots and keyframes as the basis of its content retrieval strategy. Feature vectors are extracted into MPEG-7 visual descriptors that are concatenating to compactly represent each image in a multidimensional space. These are used for the visual part of the retrieval engine. For the mid-level syntactic features speech is transcribed through ASR and used to produce a full text index. This is used for textual part of the multi-content type query. Using the MPEG-7 feature vectors already processed from the BOW technique based on SIFT descriptors, a set of SVM classifiers is initially trained to represent each shot against each high level visual concept. They then iterate the results over a second set of SVM classifiers to fuse the results and produce a normalised score for each shot per high level concept. This final stage represents the high level semantic part of the video retrieval engine. Visual and textual information is then fused together by applying a manually assisted linear fusion. VERGE also supports a simple temporal querying functionality that returns temporally adjacent shots. VERGE is a great example of combining all three low, mid and high level features into a multi-content type video retrieval platform. Unfortunately it does not produce an explicit content model that allows other systems to take advantage of such an integrated and granular approach. It also doesn't support object descriptions, and therefore spatial relationships. Its limited functionality of temporal relationship queries does facilitate proper querying between all available features comprehensively as well.

In (Zavřel, Batko, & Zezula, 2010) they extract MPEG-7 visual descriptors using the MPEG-7 reference implementation library and it's summarisation client. These are then used to compare videos against each other using a "query by example" method. It extracts keyframes based on the change of specific parameters. Five MPEG-7 global visual descriptors – colour structure (CS), colour layout (CL), scalable colour (SC), edge histogram (EH), and homogeneous texture (HT) are extracted from each key frame. Keyframes are matched against each other by comparing the five descriptors using a weighted distance function. The similarity between video clips is computed from how many matching keyframes they have and how well they match. There are no mid-level or high-level features extracted or compared against, and comparison is strictly done on a keyframe level.

A scalable video search engine based on audio content indexing and topic segmentation is described in (LawTo et al., 2011). It segments news podcasts by topic, in both audio and video formats, by transcribing the audio. Using a multi-lingual state-of-the-art transcription system the audio stream is annotated into an xml file. The transcribed audio is then partitioned into speech segments, and after determining gender, the segments are clustered for each speaker. The raw text output is then segmented into topically homogenous segments that relate to a singular news story or topic. Natural language processing (NLP) are applied to each segment to extract named entities and multi-word terms. The time codes of the terms are recorded with them. A thumbnail from each segment is also extracted but is only a reference image and plays no part in processing. This system does not use any meaningful low-level syntactic features in its content model or query formulation. Although it does identify objects by audio signal i.e. person speaking, it cannot map spatial relationships as there is no visual features. Temporal relationships are also not mapped even though time codes are extracted.

XUNET (Quan & Zhiwei, 2011) is a distributed video retrieval system that supports semantic querying through graph annotation and NLP functions. Shots are segmented manually and a keyframe is extracted. MPEG-7 descriptions are used but describe only temporal attributes of the shot. Low-level syntactic features are extracted from the keyframes to MPEG-7 descriptors. Manual and semi-automatic annotation of the semantic information is added to each shot. The semi-automatic annotation utilises the script of the movie along with its time code references to match up the text with the correct shot. Although the system does support MPEG-7 descriptors the system stores them in a relational database as structured data. This negates the interoperability of the MPEG-7 content descriptions. No mid-level syntactic features are described as the high-level semantic concepts are directly related to the low level features. This reduces the granularity of the descriptions of the content. Temporal and spatial relationships are also not supported.

VisionGo (Luan, Zheng, Wang, & Chua, 2011) is a video retrieval engine for news stories that explores the role that relevance feedback can have to improve video retrieval results for CBVR. They try and bridge the semantic gap by using relevance feedback on an initial set of content based video results. The initial query is multi-content type and therefore the engine employs low, mid and high level features. The low-level features are represented by keyframes from manually segmented shots. The features extracted from the keyframes are 27-dimension colour moment features (including 1st, 2nd, and 3rd moments) obtained at a 3 x 3 block, 80-dimension normalized local edge histogram texture feature, eight directional motion features and one global motion feature, which result in a 116-feature vector for each keyframe. Speech is the mid-level feature that is extracted using ASR. From this they extract known named entities (NE) such as time, date, location, subjects and activities from text at story level. NEs have been found to be good descriptors especially for news. They use machine learning to train detectors to assign pre-defined high level concepts to the shots. The pre-defined concepts are split into concept genres: (a) objects like cars, buildings; (b) audio-genre like cheering, silence, music; (c) shot-genre in news like political, weather, financial; (d) person-related features like face, people walking, people marching; and (e) scenes like desert, vegetation, and sky. This framework has proven highly accurate as the relevance feedback provides refinement and trains learning classifiers to better select more relevant results in the future. Although the low-level features are represented in adequate detail the mid-level feature are only represented by speech. Also there is no automatic feature extraction for shots. The features are represented by system specific description structures and metadata and therefore are unusable by other systems.

Anthropos-7 is a content description interface framework based on the MPEG-7 standard (Tsingalis, Vretos, Nikolaidis, & Pitas, 2012). Anthropos-7 was created as reduced content description set to make the indexing and use of the description schemes more manageable, as MPEG-7's myriad of tools was too extensive. They describe several new description schemes sculpted from MPEG-7 DDL for low-level and mid-level syntactic features, as well as high-level semantic features. For the low-level features they propose a ShotType DS and a TakeType DS. Both describe contiguous shots but only TakeType DS can be overlapped temporally. The ShotType DS has the option of containing keyframes or not. Another low level feature description is the Correspondence DS that is used for multi-view camera set ups as in the case of stereoscopic cameras used in the production of 3D movies/TV. The mid-level syntactic features are ActorAppearanceType/ObjectAppearanceType DS, ActorInstanceType/ObjectInstanceType DS and SceneType DS. SceneType DS deals with the hierarchical scene segments that contains ShotType DS and a TakeType DS. ActorAppearanceType/ ObjectAppearanceType DS describe the temporal appearance of actors/objects, and describes the motion of the actor/object using a Motion DS. ActorInstanceType/ ObjectInstanceType DS describes the actor/object within the keyframe. This contains the BodyPartsType DS that describes the anatomy of the actor. This approach does simplify the amount of descriptors required to describe movie/TV content but because these tools are created explicitly from MPEG-7 DDL, and are not standard descriptors it may not be totally or even partially compatible with other MPEG-7 systems without modification. It also does not address spatial and temporal relationships explicitly.

Graphic Object Searcher (GOS) is video keyframe retrieval query interface that exploits a Hierarchical Cellular Tree (HCT) algorithm to index and search large video databases (Ventura, Martos, Giró-i-Nieto, Vilaplana, & Marqués, 2012). The video is segmented into representative keyframes using a keyframe extractor. The HCT partitions stores them within cells based on their similarity to each other. GOS extracts 4 MPEG-7 visual descriptors from the keyframes: (i) Colour Structure Descriptor, (ii) Dominant Colour Descriptor, (iii) Colour Layout Descriptor, and (iv) Texture Edge Histogram Descriptor. These are not embedded into a standard MPEG-7 content model but are referenced to the keyframe they were extracted from. This makes using the descriptors by another MPEG-7 video retrieval system difficult. The keyframes are the only feature used for the entire system without any other low, mid or high level features supported. Spatial and temporal relationships are not supported explicitly either.

1.3.4.3 Content Modelling Tools

Content modelling is based on the choice of the correct standard of metadata to use. The correct choice of metadata interoperability will allow uniform access to media objects in multiple autonomous and heterogeneous information systems (R. Tapu & T. Zaharia, 2011). There are three main metadata building blocks: The language for defining the metadata scheme, the element definitions of metadata scheme and the metadata instance that contains content values of the metadata description (R. Tapu & T. Zaharia, 2011). Several types of structural and semantic heterogeneities must be resolved in each of these building blocks before metadata interoperability is achieved. Standardised metadata schemes are one way of achieving this by establishing an agreement by means of consensus building from all areas of technical expertise, such as content producers; content aggregators; content distributors; post production services and consumers, both commercial and non-profit.

Metadata itself is just another type of data that acts as a descriptive intermediary that represents the essence of the content. The metadata can be generalised into four categories, extended from the work done by (Moens et al., 2012) to apply specifically to multimedia content. These are:

- Syntactic metadata – provides a description of the content structure
- Semantic metadata - provides a description of the contents meaning.
- Technical metadata – provides technical information on technical aspects of the material and the material carrier. Examples of this are file type, date of creation and encoding used.
- Administrative metadata – describes metadata that includes creation and legal aspects associated with the metadata. Creation type metadata can include date of production, type of camera used and director. The legal aspects are concerned with intellectual property rights such as copyright and distribution policy.

A number of multimedia content modelling metadata frameworks have been proposed in recent years. These frameworks have been initiated for different purposes and therefore have different function and feature sets. They all however try to model the content by linking semantics to the syntactic features in some form or manner. Table 1.7 lists state-of-the-art multimedia content modelling standards currently available. The list is non-exhaustive but relevant in that we are only looking at standards that deal with XML, as these standards can be classed as standard interoperable, or both audio and visual features. For more information please see Appendix-A – International Multimedia Metadata Standards. For this literature review we are to focus on MPEG-7, as this is the most common and suited for the purpose of generic content modelling.

Name	Encoding	Used for	Domain	Industry
MPEG-7	XML, RDF, OWL	Archive, publish	Generic	Generic
AAF	Non-XML	content creation	Broadcast	Content creation
M3O	XML	Archive preservation	Media library	Media Distribution
MXF	Non-XML	production	content creation	broadcast
SMIL 3.0	XML, RDF	publish, distribution, presentation, interaction	generic	Web, mobile applications
SVG	XML	publish, presentation	generic	Web, mobile applications
IPTC-G2	XML	publish	News, sports, events	News &sports agencies
MPEG-21	XML, Non-XML	annotate, publish, distribute	generic	generic
EBU P/Meta (V2.2)	XML, Non-XML	publish	generic	broadcast
Dublin Core	XML, RDF	publish	generic	generic
TV-Anytime	XML	distribute	Electronic Program Guides	broadcast
XMP	XML, RDF	annotate, publish, distribute	generic	generic

Table 1.7: Multimedia Metadata Standards for content modelling

The MPEG-7 standard, formally named "Multimedia Content Description" (Manjunath et al., 2002) aims to be an overall standard for describing any multimedia content. MPEG-7 standardizes so-called "description tools" for multimedia content: Descriptors (Ds), Description Schemes (DSs) and the relationships between them. Descriptors are used to represent specific features of the content, generally low-level features such as visual (e.g. texture, camera motion) or audio (e.g. melody), while description schemes refer to more abstract description entities (usually a set of related descriptors). These description tools as well as their relationships are represented using the Description Definition Language (DDL), a core part of the language. At its inception MPEG-7 the W3C XML Schema was recommended as the most appropriate schema for the MPEG-7 DDL, adding a few extensions (array and matrix datatypes) in order to satisfy specific MPEG-7 requirements. Also the facility to describe MPEG-7 descriptions as either XML or in a binary format called BiMs was introduced for real time transmission of the descriptions in live environments. Now the standard is being translated into Web Ontology Language (OWL) (Chrisa

Tsinaraki, Polydoros, & Christodoulakis, 2004) and Resource Description Framework (RDF) (Jane Hunter, 2005) to allow interoperability with other semantic web ontologies such as those mentioned in Table 1.7. A number of works are, at present, experimenting with MPEG-7 by converting the MPEG-7 XML Schema definitions into MPEG-7 RDF Schema definitions (RDFS), which will illicit the use of machine understandable MPEG-7 content descriptions that will be accessible in a semantic web environment (S. Dasiopoulou, Tzouvaras, Kompatsiaris, & Strintzis, 2010). This conversion is still to be ratified under W3C proposals and the MPEG-7 standard itself (W3C, 2007).

MPEG-7's comprehensiveness results from the fact that the standard has been designed for a broad range of applications and thus employs very general and widely applicable concepts. The standard contains a large set of tools for diverse types of annotations on different semantic levels (the set of MPEG-7 XML Schemas define 1182 elements, 417 attributes and 377 complex types). The flexibility is very much based on the structuring tools and allows the description to be modular and on different levels of abstraction. MPEG-7 supports fine grained description, and it provides the possibility to attach descriptors to arbitrary segments on any level of detail of the description. The possibility to extend MPEG-7 according to the conformance guidelines defined in part 7 provides further flexibility. In fact a proposal for Synthetic Audio-visual Description Scheme, Method and System for MPEG-7 has been recommended on just that premise (Q. Huang, Ostermann, Puri, & Rajendran, 2009). Two main problems arise in the practical use of MPEG 7 from its flexibility and comprehensiveness: complexity and limited interoperability. The complexity is a result of the use of generic concepts, which allow deep hierarchical structures, the high number of different descriptors and description schemes, and their flexible inner structure, i.e. the variability concerning types of descriptors and their cardinalities. This causes sometimes hesitance in using the standard. The interoperability problem is a result of the ambiguities that exist because of the flexible definition of many elements in the standard (e.g. the generic structuring tools). There can be several options to structure and organize descriptions which are similar or even identical in terms of content, and they result in conformant, yet incompatible descriptions. The description tools are defined using DDL. Their semantics is described textually in the standard documents.

Due to the wide application, the semantics of the description tools are often very general. Several works have already pointed out the lack of formal semantics of the standard that could extend the traditional text descriptions into machine understandable ones (S. Dasiopoulou et al., 2010; Gibbon, Liu, Basso, & Shahraray, 2011; C. Tsinaraki & Christodoulakis, 2011). Even querying MPEG-7 documents through XQuery is not straightforward, as much multimedia information is vector-based and not able to support similarity measurement and measurement results scoring and ranking (Xue, Li, Wu, & Xiong, 2009b). A method used to try and bridge these gaps are by using profiles and levels

Profiles and levels have been proposed as a means to reduce the complexity of MPEG-7 descriptions (Daylamani Zad & Agius, 2010; Höffernig, Hausenblas, Bailer, & Troncy, 2010). Like in other MPEG standards, profiles are subsets of the standard that cover certain functionalities, while levels are flavours of profiles with different complexity. In MPEG-7, profiles are subsets of description tools for certain application areas; levels have not yet been used. The proposed process of the definition of a profile consists of three steps: 1) The selection of tools supported in the profile, i.e. the subset of descriptors and description schemes that are used in description that conform to the profile, 2) The definition of constraints on these tools, such as restrictions on the cardinality of elements and on the use of attributes, and finally 3) Definition of constraints on the semantics of the tools, which describe their use in the profile more precisely.

The result of tool selection and the definition of tool constraints are formalized using the MPEG-7 DDL and result in an XML schema like the full standard. Several profiles have been under consideration for standardization and four profiles have been standardized (they constitute part 9 of the standard, with their XML schemas being defined in part 11):

1) Simple Metadata Profile (SMP). Allows describing single instances of multimedia content or simple collections. The profile contains tools for global metadata in textual form only. The proposed Simple Bibliographic Profile is a subset of SMP. Mappings from ID3, 3GPP and EXIF to SMP have been defined.

2) User Description Profile (UDP). Its functionality consists of tools for describing user preferences and usage history for the personalization of multimedia content delivery.

3) Core Description Profile (CDP). Allows describing image, audio, video and audio-visual content as well as collections of multimedia content. Tools for the description of relationships between content, media information, creation information, usage information and semantic information are included. The CDP does not include the visual and audio description tools defined in parts 3 and 4.

4) AudioVisual Description Profile (AVDP) is based on version 2 (2004) of MPEG-7, and includes all low-level visual and audio descriptors defined in parts 3 (visual) and 4 (audio) of the standard. The constraints on description tools in AVDP concern those defined in part 5 (Multimedia Description Schemes) of the standard, restricting AVDP documents only to complete content descriptions and summaries. A number of constraints are aimed at improving interoperability, by limiting the degree of freedom in choosing and combining description tools, and enforcing the use of elements and attributes that fix the semantics of elements in the description.

The adopted profiles will not be sufficient for a number of applications. If an application requires additional description tools, a new profile must be specified. It will thus be necessary to define further profiles for specific application areas. For interoperability it is crucial, that the definitions of these profiles are published, to check conformance to a certain profile and define mappings between the profiles. It has to be noted, that all of the adopted profiles just define the subset of description tools to be included and some tool constraints; none of the profile definitions includes constraints on the semantics of the tools that clarify how they are to be used in the profile.

Apart from the standardized ones, a profile for the detailed description of single audio-visual content entities called Detailed Audio-visual Profile (DAVP) (Bailer & Schallauer, 2006) was proposed but was superseded by AVDP . The profile includes many of the MDS tools, such as a wide range of structuring tools, as well as tools for the description of media, creation and production information and textual and semantic annotation, and for summarization. In contrast to the adopted profiles, DAVP includes the tools for audio and visual feature description, which was one motivation for the definition of the profile. The other motivation was to define a profile the supports interoperability between systems using MPEG-7 by avoiding possible ambiguities and clarifying the use of the description tools in the profile. The DAVP definition thus includes a set of semantic constraints, which play a crucial role in the profile definition. Due to the lack of formal semantics in DDL, these constraints are only described in textual form in the profile definition.

In addition to the profiles, revisions were made to both MPEG-7 parts 3 (visual descriptors) and 5 (multimedia description schemes). MPEG-7 Part 3 was revised in 2004, 2006, 2009 and 2010 (MPEG, 2010), with the addition of visual extensions, perceptual 3d shape descriptor, image signature tools and video signature tools. MPEG-7 Part 5 has been revised in 2003, 2004, 2008 and

2012 (MPEG, 2012a), with additions of new basic elements, additional Linguistic Description Tools, extensions to the user interactions descriptions tools for compatibility with MPEG-21 DIA, improvements to geographic descriptor and social metadata descriptors respectively.

1.4 Common research threads and challenges

	Automatic Feature Extraction				**Content Modelling**	
	pre-processing & Syntactic media	**temporal segmentation**	**spatiotemporal segmentation**	**semantic relationships**	**Content feature and modelling**	**Content model**
Problems Identified	raw media is not optimised for feature extraction Raw media processing requires high computational expense	The need to index video content into temporally syntactic and logical story units Identifying different syntactic attributes of temporal segments	to segment a video stream into spatiotemporal regions of foreground objects and background two tier problem: 1) initially segmenting the object and 2) tracking the object consistently over time	to formalise the spatial and temporal relationships between features to analyse and model semantic relationships for respective content features	combination of features that best describe the content in a video stream modelling of the content features so that they can be queried at different levels of content structure and media	standardising content model descriptions to be read by any compliant application To be able to describe the content in a multi-faceted content representation ability to create new content descriptions
Problems Solved	Filtering of media increases effectiveness and efficiency of feature extraction processes removing data redundancy reduces computational expense	Segmenting content into hierarchical structure that represents story units at different levels of detail Algorithms to identify the different types of transitions Clustering of shots into scenes using machine driven methods within certain domains or joint audio and visual cues for generic video	techniques have been developed for modelling the background and then modelling the changes into spatiotemporal regions techniques developed for foreground based on tracking moving pixels and over segmentation Techniques developed for tracking spatiotemporal regions	spatial and temporal relationships have been standardised and are complete spatial and temporal relationships are being modelled by a small number of those systems that have been reviewed	to search and describe the structure of video, syntactic features are modelled to search and describe the concepts within the video, semantic features are modelled integrating syntactic features with semantic relationships	MPEG-7 standard and other metadata standard created MPEG-7 describes content in syntactic, semantic, technical and administrative data MPEG-7 Description definition language used to create new descriptions when needed
Unresolved Problems	The pre-processing strategy is not considered important to the overall goal of application as elements can have other uses has a limited view of optimising for one feature extraction method and does not support a multi-feature extraction use	reliably identify shots with different transition types clustering of shots into scenes using only syntactic visual cues for generic video	generic segmenting and tracking of Spatiotemporal regions is poor for non-modelled background techniques Techniques for generic segmenting cannot track objects consistently	both spatial and temporal relationships are not being modelled explicitly and are ambiguous in description as different applications use different formulations for modelling the relationships	extracting the content features in a machine driven manner that reduces the need for human intervention reducing the semantic gap by establishing the relationships between syntactic and semantic features by linking them through their semantic relationships	creation of application specific tools makes some MPEG-7 content models "less" standard then others

Table 1.8: RESEARCH TOPICS - IDENTIFIED, SOLVED AND UNRESOLVED IN AUTOMATIC FEATURE EXTRACTION AND CONTENT MODELLING

1.5 Literature Review Discussion

As shown in Table 1.8 the literature review has covered a broad swath of research that is related to extracting content features and how those features can be modelled to represent the content both syntactically and semantically. We began by looking at how raw media is processed with a view to feature extraction. We then examined work on extracting low level features, namely shots, scenes and spatiotemporal segmentation, followed by looking how these features could be semantically linked. The focus of this was spatial and temporal relationships. We concluded the literature review by looking at how the semantic and syntactic features are indexed into a content model. This began with the reasoning behind which content model features are the most commonly modelled. This was followed by a state-of-the-art review of feature extraction systems that model features for retrieval purposes. The review ends with an examination of the MPEG-7 standard for content description and its advantages and disadvantages in being able to produce a generic content model that can be used by any compliant MPEG-7 system.

The importance of pre-processing raw media is to A) make feature extraction more effective by either filtering or converting the media so the salient points of the features of interest are easier to extract and B) to reduce processing time to within acceptable levels by reducing computational complexity. From the reviewed literature we can see this is a definite benefit in having a defined media preparation stage as the resultant media conversion improves feature extraction by many factors. Most feature extraction systems have not employed an active strategy in defining a raw media pre-processing methodology. They have seen it as an implicit factor of extracting only a certain feature and do not apply a more broad philosophy to the system as a whole. Such a method could be more beneficial as the pre-processing could be applied more methodically in order to improve feature extraction for more features and improve on reducing processing times.

The choice of low-level syntactic primitives used for syntactic feature extraction is an important factor in the success and effectiveness of extracting the desired content features. To extract a certain syntactic feature the chosen primitive feature type is integral to the feature extraction process. The choice of syntactic primitive feature itself is influenced by its physical properties and attributes. For instance, when wishing to extract shots by identifying shot boundaries there are a number of techniques available. If using a colour histogram difference technique the choice to use colour histograms is implicit, but the choice of colour space to be used is not. The choice of colour space has a direct bearing on the detection rate. Similarly the choice of low-level syntactic primitive should be made with a view to reusability and polymorphism of use in a multi-content feature extracting environment. Most systems assign one primitive to one process, increasing computational expense and waste and not fully leveraging the benefits that a more multi-content centric approach could yield.

Temporal video segmentation is a fundamental building block of all syntactic and semantic feature extraction systems. The ability to segment video into a temporal hierarchy is imperative to the construction of a video content model. Central to temporal video segmentation is shot segmentation. This has been an on-going topic of research for many years, and a popular one as well. Many techniques have been formulated over the years and all address the same problem. This is mainly to not only identify the shot boundaries but the type of boundary between it e.g. abrupt or transition. Very few techniques have equal success at identifying both types, and if they do they do not have the precision and recall of techniques that identify one or the other. The type of shot boundary can be semantically significant, as it can indicate the start of a semantic event. For example the presence of a fade in is usually an indication that a new scene has started. The relative entropy of the shot can also indicate genre, for instance action scenes usually have fast moving panning shots or a lot of camera shake. Identifying such features and attributes of shots is important in linking semantic meaning to the underlying syntactic features.

Whereas shot segmentation is a purely syntactic derivative, scene segmentation is dependent on the semantic relationship between shots. A scene describes a collection of shots that are temporally related to describing a semantic event as a narrative unit. Due to the semantic nature of scenes there are no effective machine readable techniques that can be used to directly identify scenes generically. There are syntactic feature techniques that are genre specific that identify possible scene boundaries by certain syntactic "landmarks" but these rely too heavily on format and content within the content stream being standardised with little change. More generic techniques have used either film grammar or machine learning techniques to either cluster shots, detect boundaries or model shots into scenes. These techniques though do not enjoy a high level of precision and recall such as shot segmentation. Scenes are an implicit structure required for content modelling, as they are a bridge between the physical content of the media and the meaning of the content. Scenes can be described for this reason as a mid-level syntactic feature; by the way they temporally group shots into semantic events. Scene segmentation needs techniques that 1) are genre independent and 2) is more semantically correct in boundary definition. The second point needs the technique to have an understanding of the semantics of the content. This requires knowledge of the events going on within each shot and how they relate to other shots that could be semantically grouped with them.

Another important step in syntactic feature is spatiotemporal segmentation. Spatiotemporal segmentation is another important step in modelling video content. They are integral in defining events by establishing the interaction between objects. Similar to scene segmentation, spatiotemporal segmentation defines the boundaries of semantically meaningful objects. Spatiotemporal segmentation has similar problems to scene segmentation in the fact that delimiting the borders of an object is a subjective process based in semantics. Due to its semantic nature the spatiotemporal objects can be classified as mid-level syntactic features. Also due to its temporal nature the segmentation evolves over time, this is what differentiates it from image segmentation. Techniques for spatiotemporal segmentation are centred on grouping pixels based on changes in colour, texture or motion. Although there has been relative success with unsupervised techniques these are limited by certain conditions that must be met for the segmentation to be successful. Most techniques employ a learning phase or training data to establish a base line for segmentation. These methods have proven more successful in the current state-of-the-art research.

The relationship between features is as important as the features themselves, as the relationships allow the content to be queried in a more meaningful way that is natural to consumers. Spatial and temporal relationships answer the two out of the four major categories of querying which is the "where" and "when", the other two being "who" and "what" (Agius & Angelides, 2005). The spatial relationship between objects is important to the querying of events as it allows users to query a particular arrangement of objects, or change in arrangement, that might indicate a particular event or action. Temporal relationships are the basis of querying the occurrence of events in relation to other events. The ability to query temporally is a powerful tool as it not only queries syntactic or semantic features homogenously but is also able to find the relationships between heterogeneous features. This allows the content model to be queried in a multi-faceted manner for all the features contained within. To have both of these relationships to be stated explicitly modelled means that the content can be uniformly queried from any system with the results being the same regardless of method.

The features that are to be modelled play an important part in the effectiveness of the content model. Many different video indexing and retrieval systems use various content feature sets that are usually sculpted to fit the purpose of their querying methodology. Some use more features than others but they can be grouped into five syntactic and semantic content classes, as shown in Figure 1.4.

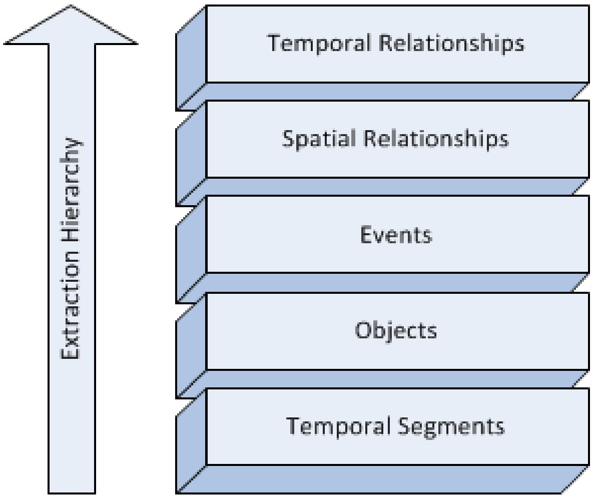


Figure 1.4: FIVE CLASSES OF SYNTACTIC AND SEMANTIC CONTENT FEATURES

The five classes of syntactic and semantic content features, as depicted in Figure 1.4, also includes an extraction hierarchy.

The hierarchy shows the direction the content features need to be extracted in order to support the content feature classes above. Temporal segments are extracted first as they are the smallest unit of video feature extractable and consists of frames and shots. Once the shots and keyframes are extracted the objects within each unit is extracted. From this we can extract the events as we now can analyse the interplay of the objects and segment the video into appropriate action clusters that become an event. From the objects position the spatial relationship of the object, to both its environment and other objects, can be calculated. Once we have the other four classes of content feature, we can then define the temporal relationships between them.

Four of the five classes represent the semantic “who” (objects), “where” (spatial relationships), “what” (events) and “when” (temporal relationships) that are the foundations of all semantic content queries. This is of course is only a semantic content feature taxonomy, and does not support syntactic content feature integration. To our knowledge, no syntactic and semantic content feature integration exists in the current literature. Such an integration of syntactic and semantic content feature classification could help in reducing the semantic gap. The semantic gap is the gap between syntactic feature representation and the high level semantics they represent (Wang, Mohamad, & Ismail, 2010). By directly mapping semantic features to supporting syntactic features, the semantic gap can greatly be reduced. If video content indexing systems applied this classification to the content features they extract they could create content models that would be universally compatible with all similar video content retrieval systems and the results from a query on one system would have identical results on another system if the same query were used.

From the state-of-the-art video indexing and retrieval systems we ascertain that the different content modelling systems use different content features depending on the needs of the original motivation that it was created for. To the best of the author's knowledge there is no other system, which models the same five classes of content feature and incorporates some type of hierarchical extraction. Even the systems that can retrieve spatial and temporal relationships do not explicitly state them as a reusable feature. This can lead to erroneous results when querying one systems content model on another. For example, if we are looking at spatial relationships one system could use the centroid of an object as the midpoint to calculate the spatial relationship of the object, but another system might use the closest edge of the object. This could lead to both systems giving a different spatial relationship when posed with the same query. This defeats the point of standardisation of content models, as the retrieval mechanism is allowed to add bias.

MPEG-7 is the de facto standard for the description of multimedia content does not explicitly state how content should be created or consumed. Its function is to provide standardised metadata that any MPEG-7 compatible system can access and use for its purpose. As already mentioned a content model should have four classes of semantic content features modelled so that any system can uniformly access and use the data to answer a query. MPEG-7 provides tools for describing both semantic and syntactic features. Most systems reviewed use a genre specific implementation of these MPEG-7 description tools. This leads to two problems. The first is the mapping of syntactic to semantic features is incomplete outside of the intended use. This does not reduce the semantic gap for generic use of the features. The second is that they tend to use the MPEG-7 DDL to craft genre specific description schemes that, although compatible with MPEG-7 description schemes, will be unusable by other systems.

From the reviewed literature we can ascertain these gaps in the current state-of-the-art research in automatic feature extraction and content modelling:

1. A defined method of pre-processing raw media would increase feature extraction and reduce computation time for the extraction of all content features.

2. Semantic temporal segmentation is ambiguous in nature and requires a user perspective to address the segmentation in a generic environment.

3. Syntactic feature extraction and modelling should be directly mapped to support semantic features in order to reduce the semantic gap.

4. Spatial and temporal relationships should be explicitly stated.

5. A complete framework that binds together the above mentioned point into a systematic architecture that converts the content into syntactic and semantic content descriptions. It's functionality should incorporate the following:

   a. The features should be described in a granular manner that allows use of the feature description to be accessed in a local or global manner in relation to the other feature sets.

   b. The content, the feature extraction algorithms and content descriptions should be modular in their integration into the framework as the need to extract different feature sets or improve on the extraction of existing feature sets should be upgradeable for future work. The separation of these three components allows reuse and multi-purposing by other applications.

   c. The content descriptions should be universally accessible content model so that it can be used in the widest array of video content search applications.

   d. The content descriptions should be richly detailed in both syntactic and semantic content description. The structure of the content descriptions should be granular so that video search applications can retrieve different levels of content description detail depending on their specific requirements.

1.6 Research aims, objectives and modelling techniques

The aim of this thesis is to provide a framework that will extract and model video content features that allows for content within the video to be searched, by any video retrieval application that is standards compliant. The video stream will be processed to extract and derive both syntactic and semantic content features and the resulting output will be formed into content model. The content model will integrate syntactic and semantic content to facilitate multi-content type query formulation. The content model will also link the syntactic features to semantic relationships to provide a foundation for other high level video modelling applications, which employ concept detection processes, through spatial and temporal modelling.

The objectives of this thesis are:

(O 1) To design an abstract framework that transcodes video stream content features into content descriptions. The framework must extract both syntactic content and semantic relationship descriptions and interlink them, in order to take a step closer to 'bridging' the semantic gap between syntactic and semantic features. To achieve these goals the framework must incorporate the following:

(O 1.1) A pre-processing method that increases the feature extraction potential of the video stream, by filtering the media to improve extraction accuracy and reduce computational expense of the whole framework.

(O 1.2) To extract syntactic features through machine driven processes, to substantially reduce the time of manually segmenting these features. The low-level features will be extracted through unsupervised techniques. The mid-level features will be extracted by semi-supervised techniques. These techniques will employ semantic understanding of the structure of the features through user feedback.

(O 1.3) Through analysis of the syntactic features, semantic relationships will be derived and linked to these features. The semantic relationships will exploit the temporal and spatial dynamism in the content. It will allow querying of the semantic relationships in an unambiguous and explicit structure. In addition, the video search and high level concept detection applications produce results that are uniform and consistent across different platforms.

(O 2) To integrate the syntactic and semantic descriptions into a content model that is accessible to the widest range of applications. The content model must:

(O 2.1) Create an accessible content model that adheres to multimedia content description standard. The content model must also be independent of proprietary restrictions and backward compatible with earlier versions, thus making it available to the widest range of relevant content based video search applications.

(O 2.2) The content descriptions will be organised into a hierarchical structure that interlinks all the content descriptions, regardless of modal type. This will allow querying of the content using a multi-content type approach consisting of syntactic and semantic elements.

(O 2.3) The content model will be granular in the structuring and detail of the content descriptions. This will allow video search applications to use "coarse to fine" search approaches making result retrieval more efficient and focused.

(O 3) A prototype of the framework must be developed as a proof of concept. The prototype must implement all the objectives stated in objective 1 and 2. It must incorporate a modular framework that allows the component of the prototype to become extensible for future components and also allow updating of the existing modules. The prototype should incorporate all the modules into a holistic framework, where the components can be added, removed, reused and modified independently, thus increasing efficiency and potentially reducing processing time, while allowing for custom video processing pipeline to be created.

From the objectives we formulate a research modelling methodology that will be used to provide an experimental "proof of concept" framework, which consists of content media, extraction and modelling components. Below are the research modelling methods that will be used:

RM 1. Algorithms will be developed for each sub-objective of the framework. The algorithms will be modelled to implement the functions of each of the of the sub-objectives:

RM 1.1 To develop a filtering and optimisation algorithm that can increase the effectiveness of the feature extraction process by increasing the saliency of syntactic features, while also reducing computational expense by removing data redundancy. This algorithm will be implemented into a proof-of-concept prototype and will be evaluated mathematically.

RM 1.2 To develop a syntactic feature extraction algorithm that extracts syntactic features from a video stream. The algorithm will extract syntactic features in a hierarchical process to reduce computational expense by extracting features to take advantage of the linear dependency of the features in relation to each other. This algorithm will be implemented into a proof-of-concept prototype and will be tested using benchmarks test that are used widely by the research community.

RM 1.3 To develop an algorithm that analyses syntactic and semantic features and then links the features through spatial and temporal relationships. This algorithm will be implemented into a proof-of-concept prototype and will be analysed and evaluated against groundtruth samples.

RM 2. To use create a content model that is standards compliant, namely MPEG-7 and that is:

RM 2.1 Hierarchically structured to allow the content to be granular in description to allow video search applications to access the detail of content they require or employing a "coarse to fine" filtering approach for relevant content

RM 2.2 To model the content description so that they are available to the widest range of applications. The content model must be unbiased to any particular specification and use so it must be structured to eliminate any ambiguity that can arise from vendor or proprietary use of its descriptions.

RM 2.3 The syntactic and semantic content descriptions should be integrated and linked to facilitate multi-content type filtering and search queries. This will allow the content to be search using more naturally arranged queries that incorporate both syntactic and semantic features that are intertwined.

RM 3. To build a "proof of concept" prototype, namely the MAC-REALM Framework, which incorporates all the developed algorithms mentioned in (1). The framework will be built as a modular and extensible development platform that allows the components to be updated or changed, or for the framework to be extended for future components or functions.

1.7 Theses Outline

The remainder of the thesis is organised as follows, Chapter 2 proposes MAC-REALM, an abstract video content extraction and modelling framework that comprises of three horizontal layers and four vertical planes, in its architecture. The three layers are the content layer, application layer and MPEG-7 layer. These describe the different stages of content as an input/output scenario that translates content into different states during the conversion of the content media into content descriptions. The four planes are comprised of 1) a raw media plane, 2) an extraction plane, 3) an analysis and linkage plane and 4) a modelling plane. These planes describe the conversion of the video stream into a content model.

Chapter 3 proposes a MAC-REALM proof of concept prototype application that implements the abstract framework in chapter 2. Using a reusable code base the prototype is developed into a modular platform. An overview of MAC-REALM is presented showing the content extraction and modelling process as a custom video processing pipeline that converts the raw video into a content model. This is followed by a detailed description of the components, sectioned plane by plane.

Chapter 4 begins with a step-by-step walkthrough of the MAC-REALM prototype showing its functions and user interaction. This is followed by a performance evaluation that uses benchmark testing, where available, to examine the effectiveness of the frameworks extraction and modelling techniques in regard to their objectives of the content feature extraction and modelling framework. Finally a MAC-REALM evaluation is then presented, which discussed the walkthrough and results in the context of the framework itself.

Chapter 5 concludes the thesis with a summary of the chapters, followed by a discussion of research contributions against research objectives. Lastly, we look at future work that can be undertaken based on the research in this thesis.

CHAPTER 2: THE MAC-REALM FRAMEWORK

This chapter presents the MAC-REALM system (MPEG-7, Application and Content layers with Raw media; Extraction; Analysis and Linkage; and Modelling planes) an abstract modular cross-functional framework that is able to extract video content features into an MPEG-7 content model using a mixture of automated heuristic techniques. By combining several content and feature extraction techniques, as well as content analysis and modelling a system is created that indexes a video stream in terms of objects, shots, scenes and the spatial and temporal relationships between them and integrates them into a tightly integrated syntactic and semantic content model.

The chapter is organised as follows. Section 2.1 presents MAC-REALM framework and discusses the role and function of the framework. Section 2.2 discusses the role automatic feature extraction has to play in MAC-REALM. Section 2.3 examines the content modelling strategy behind MAC-REALM and its feature selection and modelling strategy. Section 2.4 presents the design requirements for the MAC-REALM framework. Section 2.5 presents a detailed high level overview of the MAC-REALM architecture and provides a walkthrough of the custom video processing pipeline and the role the function modules serve in the process. Section 2.6 introduces the three layers of content conversion and creation of MAC-REALM. Section 2.7 provides a run through of each of the functional planes of MAC-REALM that convert the video stream into a content model. Finally, section 2.8 summarises the chapter.

2.1 MAC-REALM Framework

In chapter 1 we reviewed feature extraction and content modelling. Both aspects are integral to producing a video extraction and content modelling framework. Most video content extraction systems segment the content and do not process it any further, as the segmentation was the primary purpose. This wastes the potential of the information to be reused for other purposes, such as video search and concept detection. In addition, most video content extraction applications concentrate on one particular feature of the content. This is inadequate at describing the video content, as it contains a wealth of other content features. The more features extracted leads to more information that can be modelled, and the more useful the content model becomes. The content model should consist of syntactic and semantic content so that the content is described both structurally and conceptually. These have to be integrated so that the content can be searched or mined in a more semantically meaningful way.

The only way to achieve the goal of modelling content features from the video stream is to produce a framework were the flow of control follows a path of processing the raw media into a content model, whilst transforming the content features into content descriptions. As the content passes through the framework, it will be refined into more complex and meaningful content descriptions. The strength of the framework is that each stage of its process is designed to provide a complete set of content features and descriptions that can be reused or extended to capture even more content features and descriptions. The framework as a whole will provide a content model that will have a syntactic content description base that is semantically linked spatially and temporally, reducing the semantic gap between those sets of syntactic and semantic features.

The first function of MAC-REALM is to extract and segment the video stream into content features. Content feature extraction is a complex computational task, and requires a number of different algorithms and approaches to extract the different types of features. These features once extracted can be structured into a metadata format that can act as a data exchange mechanism that can be used by content based video search applications. The relationship between content feature extraction and modelling techniques is integral to producing a content model that is rich in description, granular, universally accessible and multi-faceted.

The second function of the MAC-REALM framework is producing a content model that integrates and couples syntactic content features to semantic content features to reduce the semantic gap associated between low-level and high level content features. From section 1.3.4.1, which describes which content features to model according to the requirements of content based video queries, any video content modelling system must incorporate five content features in order to produce a content model that is capable of supporting syntactic and semantic queries. These features must be coupled on two levels, a semantic level and a syntactic level. On the syntactic level the semantic attributes of the mid-level features must be properly defined to make the syntactic features more semantically correct e.g. Scene boundaries are defined by the cognitive perception that an event has taken place from start to finish or a person riding a horse are two different objects and yet the saddle of the horse can be attributed as part of the horse object. On a semantic level all content features, syntactic and semantic, must be have a mechanism which allows all the features to be queried or compared against each other through some sort of semantic relationship. For instance ways of semantically querying what shots have certain spatial relationships. Shots are totally syntactic features and the query would have to include an objects parameter in its formulation to act as a proxy for comparison between the features for the query to be answered.

Providing another mechanism for greater search flexibility would yield more possible ways to query different feature sets.

MAC-REALM's goal is to provide a framework that that extracts syntactic and semantic content features from a video stream and then models them into a content model that integrates the features so that the semantic gap is reduced for multi-content type queries. The way the content features are extracted is directly related to the way the content descriptions are modelled. The segmentation of the content features is designed to extract features the features in a hierarchical extraction process. This hierarchical extraction process is then mimicked in the modelling of the content model so that the syntactic and semantic content features are closely coupled. The resulting richly and granularly detailed content model is structured to facilitate multi-content type content type search from compliant content based video search applications.

2.2 Automated feature extraction

The role of automatic content feature extraction is to provide the content features that would take an inordinate amount of time to manually extract or the complexities of extraction require precise segmentation that can only be provided by computer analysis. Syntactic features are usually the most difficult to extract manually in terms of manual processing as the features are very rich in detail and quantity and the intricacies of capturing them can lead to errors and omissions. Semantic features on the other hand are more accurately extracted by humans, as they perceive the complex conceptual intricacies of the semantic features better than any computer analysis can at present. Mid-level syntactic features have semantic attributes that help define the boundaries of that feature, whereas low-level features are purely signal based entities in that their boundaries are structure based and can be represented in purely physical characteristics e.g. a frame is a still image of a point in time of a video and a shot is a contiguous set of frames that are all visually similar. The complexities of extracting each type of feature must be addressed, as the subsequent content descriptions will be incomplete or incorrect if these nuances in definition are not captured properly.

MAC-REALM will extract two types of content feature, syntactic (low and mid-level) and semantic high level features. The purpose of this is to fully represent the content features in the video stream completely when converting the features into a content model. The way these content features are extracted will have a direct impact on the quality of the content model. Once the content features are extracted, they are translated into content descriptions that will become integrated to produce a content model. The method of extracting the content features therefore will not only extract the features but will also define the attributes and characteristics of captured content feature. These content feature characteristics can be coupled with characteristics from other content features if captured correctly. When capturing the characteristics of mid-level attributes correctly, from a user perspective, this can create better linkage between syntactic and semantic content features and help reduce the semantic gap.

The first features that need to be extracted are the syntactic low level features, as these will become the input media for the extraction of the syntactic mid-level and semantic content features. The syntactic low level content features are extracted by automatic methods that do not require any human interaction. As being signal based entities they are prime candidates for automatic extraction as they will be extracted more precisely and efficiently then manual methods of extraction. The low level features are generally used for "query by example" methods of content based video search, where the need for minuet levels of detail are examined and compared by the search engine. Having low-level features described completely in a content model actually makes the process of search longer then if comparing the extracted feature directly. In this instance, the content model will act as an indexing system for these features and private a direct reference to the feature that best fits what the query was trying to find. This way the overhead of converting the search query is reduced. The low-level features are the building blocks of the content model and the higher level type content features will be extracted from them. The low-level syntactic features are directly used to extract the mid-level syntactic features as they are closely coupled in structure. Then the syntactic low and mid-level features are then analysed to create the semantic relationships between them.

Syntactic mid-level features are more complex to extract but are of higher value as they are content features that are queried more directly by users then low-level features. The problem in extracting syntactic mid-level features is that semantic attributes of the feature make the segmentation process troublesome as machine driven methods find it hard to capture the semantic aspects of the features. For this reason most unsupervised methods are restrictive to one domain or genre, where the complexities of the semantics can be modelled accurately and applied to syntactic structures. These methods are of limited use, as the semantics have to be remodelled for every new domain or genre. Where domain restriction is removed, semi-supervised techniques are used as they can introduce semantic definition by directly using human perception. The user interaction is used to initiate or provide feedback to the process to allow it to more accurately capture the semantic attributes of the content feature, which in turn provides better definition to the segmented boundary. This better definition of the semantic attributes of the mid-level features is key to addressing the problem of the semantic gap and how to bridge it.

The semantic features of the media cannot be directly extracted from the video stream like the syntactic features were. Semantic features are in essence a cognitive feature that requires human perception to be able to be perceived correctly. What can be extracted are the semantic relationships between the content features. This does not require cognition of the features and only requires semantic analysis of the content features. The features are analysed from a specific semantic focus that compares the features against each other and derives a semantic link between those features. The semantic relationships can be automated, as they do not require any understanding of the content just that the relationship between features has specific meaning. The relationship between the features can be closely coupled to the syntactic feature. The semantic relationships themselves are key to defining the event occurring within the media stream. Thus through defining the semantic relationships and tightly integrating them with the syntactic features, the syntactic features themselves become more closely coupled to the events they represent, reducing the semantic gap between syntactic and semantic features.

The importance of which features are extracted and how they are extracted are very important to the content model. The extraction process directly influences the quality of the content descriptions and the detail they provide. MAC-REALM will extract features primarily for the reason of modelling syntactic and semantic content features into a closely coupled integration of those features in order to produce a content model that reduces the semantic gap. The way the features are extracted will be in a "bottom up" manner, where the lowest level features are extracted first. The low level features will provide a foundation for the higher content features to me built upon. This will help build when creating the content model as the hierarchical structure of the content will have been implicit in its creation.

2.3 Content Modelling

In section 1.3.4.1 the five content classes that need to be extracted and represented for a content model to be able to answer all types of queries is explained. From those content classes five types of feature were identified, 1) segments that represent and action or instance of an object 2) objects within the media stream, 3) the spatial relationships between those objects, 4) events that occur involving those objects and 5) the temporal relationships between all those features. From these five content types we have five content features that can be modelled to represent them. The content class to content feature mapping is shown in Table 2.1.

Content Class	Content feature
temporal segments	shots
objects	Moving regions
Events	Scenes
Spatial relationships	Spatial Relationships
Temporal relationships	Temporal Relationships

Table 2.1 Content class mapped to content features

Shots represent the temporal segments as they are a set of contiguous frames over a period of time within the video stream. Shots can represent an action, or they can represent an instance of an object. Objects are represented my moving regions as they represent an object completing an action. An inanimate object is treated as background as it is not taking part in an action or event. An object that was once moving and has become inanimate will still be represented by a moving region, as it once moved and may move again. Scenes represent events because an event is a collection of actions that together form a semantic event; much like a scene is a collection of semantically related shots. Spatial and temporal relationships cannot have a physical manifestation as they exist exclusively as high level semantic concepts.

The features selected for content modelling are important to the ability of the content model to be granular in description. The content descriptions must be integrated in a hierarchical manner for each feature, which has a more complex structure and detail. Features that are of a higher content type might have a lower content type feature nested within them as the higher content feature might consist of the lower feature. For instance, Shots would naturally be nested in scenes as scenes consist of shots. Most features will have sub-features nested within them giving the features extra detail. Two examples are that shots contain not only there start time and duration but will also contain their transition type or objects will contain their object coordinates and colour distribution. This nesting allows the search to be granular by locating first the main content feature and then being able to query further the detail of that object.

MAC-REALM will use the MPEG-7 standard to encode the content features into content descriptions. The subsequent content descriptions will themselves be integrated to produce an MPEG-7 compliant content model. The selection of which MPEG-7 tools to describe the content features affects the accessibility of the content model to content based video search applications. The selection of the correct feature set allows the media to be searched by many different video content search and retrieval systems, independent of the use or domain of the system in question. As MPEG-7 has been revised on numerous occasions and the introduction of profiles has added to the fracturing of the standard. The profiles use subsets of tools that are used within certain domains or functions. MAC-REALM will use tools that are used by all profiles of MPEG-7 and are backward compatible with the version 1 of standard.

Another important aspect of the creation of the content model is the way these features are modelled. The modelling of the features has a direct impact on the interoperability between the content model and the integration of its features so that it can better enable multi-content type content type querying and reduce the semantic gap. MAC-REALM will integrate the syntactic and semantic content features together in closely coupled structure, where the linking shows the interdependence of the features. In Figure 2.1, the outline for the mapping scheme is presented. It shows the linking of the semantic content to syntactic features. The objects are represented as moving regions, as foreground objects can be distinguished by the action they play. From the moving regions, we can then determine the spatial relationships between the objects and their positioning. To represent events the video stream is initially segmented into shots. The shots are then grouped together to form scenes. The scenes represent the events, as they are, by definition, a cluster of semantically related shots, which are linked to a portrayal of a common semantic theme or concept.

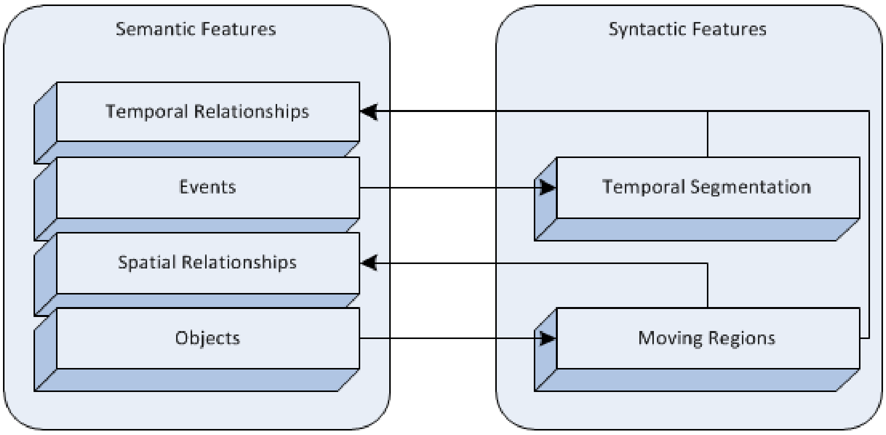


Figure 2.1MAPPING OF SEMANTIC CONTENT FEATURES TO SYNTACTIC FEATURES

Once all the moving regions, spatial relationships and temporal segments have been extracted or derived the temporal relationships between all these features is calculated. All features, whether syntactic or semantic, have a temporal characteristic, as video is a temporally evolving medium for content. It is this temporal component that is the basis of the linking mechanism between all the content features. This provides a powerful heterogeneous platform for search and retrieval in a syntactic/semantic environment.

2.4 Design Requirements for MAC-REALM

The main aim of MAC-REALM is to extract syntactic and semantic features from a video stream and then model them so that the content model can be used by many video search applications that are compliant with MPEG-7. It will model the features so that the same query in each application should retrieve the same results. It will achieve this by removing ambiguity caused by the differing ways that applications interpret relationships between content features. It will also allow the content to be queried both syntactically and semantically in a manner that is familiar to the way video is structured and perceived by consumers.

The framework will extract three types of feature; low and mid-level syntactic and high level semantic relationships. Five feature components will represent these three types of feature. The first feature to be modelled is the low-level feature of shots. Shots form the foundation blocks of the content model. The other content features will use the shots as the reference features to build upon for the hierarchical structure of the content model.

The mid-level features will be the objects and scenes. Unlike the low level features, they are semantically derived syntactic features. They cannot be extracted by purely machine driven processes, as they require a level of semantic "recognition", as well as comprehension, to their syntactic boundaries. The objects require a two fold approach; first they are segmented from the background, similar to image segmentation, and then they need to be tracked for the duration of the shot. Scenes have do not have a generic syntactic marker that can be used to segment them. They are usually demarcated with specific film grammar techniques or domain specific graphic or effect transition (see section 1.3.2.3). Each video stream will have its own formulation of syntactic features that will identify where the scene boundaries are.

The high level semantic relationships consist of two components, the spatial and temporal relationships. The spatial relationships will be modelled in two ways, absolutely and relatively. This will allow the position of objects to be queried or analysed with respect to their global position and their position to each other. Unlike the spatial relationships that are modelled for only one feature, the temporal relationships are modelled between all content features. The modelling of temporal relationships between all features, both syntactic and semantic, makes the querying and analysis of the content multi-dimensional and allows syntactic and semantic content descriptions to be queried temporally by direct comparison. This not only allows polymorphic querying of the content, but can also be used by temporal concept learning methods to model concepts to features that are not exclusively syntactic or semantic, such as scenes.

To model these five feature components we must extract them from the video stream. The raw signal must be pre-processed before extraction can take place. This is to improve the efficiency and effectiveness of the extraction process. In the extraction process, the syntactic features that form the foundation of the content model are segmented. The features are then analysed together both spatially and temporally and linked to form semantic relationships. The syntactic and semantic features are then modelled into an MPEG-7 compliant content model that is made available to all MPEG-7 video search engines.

Therefore the design requirements of MAC-REALM Framework, based on the research objectives and methods in section 1.6, are:

1. A method for pre-processing the raw media that optimises the potential of the video stream for feature extraction and reduce computational overhead. The method should be based on 1) filtering the raw media so that the feature extraction is more accurate and precise and 2) redundancy of data should employed so that only the minimum amount of salient data is processed during feature extraction.

2. To extract the low level and mid-level syntactic features from the filtered media. The low level syntactic features will be extracted through unsupervised machine techniques. The mid-level features will need semi-supervised techniques as they have semantic attributes that can only be defined through user input. The processes will be optimised for efficiency and effectiveness in reducing computational expense and accuracy of features extracted.

3. To derive from the low level and mid-level features the semantic relationships between them, both temporally, and where possible spatially. These must be explicitly expressed, as to avoid ambiguity caused when different applications use the same query but not the same semantic relationship formulation.

4. To model the syntactic and semantic features into a content model that interlinks the content so that the semantic gap between syntactic and semantic features is reduced. The content model must be structured to describe the content in a multi-faceted, richly detailed and granular manner.

5. To integrate the four design requirements into a framework that uses each design goal to process the content from raw media into a syntactically and semantically complete content model. At each design goal, the process must be based around turning the content features

into content descriptions that can be used directly, integrated or extended. The four design requirements allows for custom video processing pipelines to be created. Processing pipeline are the arrangement of a sequence of customer or modified modules.

2.5 MAC-REALM Architecture

The MAC-REALM Framework comprises four planes and three layers: the raw media plane, the extraction plane, the analysis and linkage plane and the modelling plane. The three layers are the MPEG-7 layer, the application layer and the content layer. Each layer is described in more detail in section 2.6, whilst each plane is described in more detail in section 2.7.

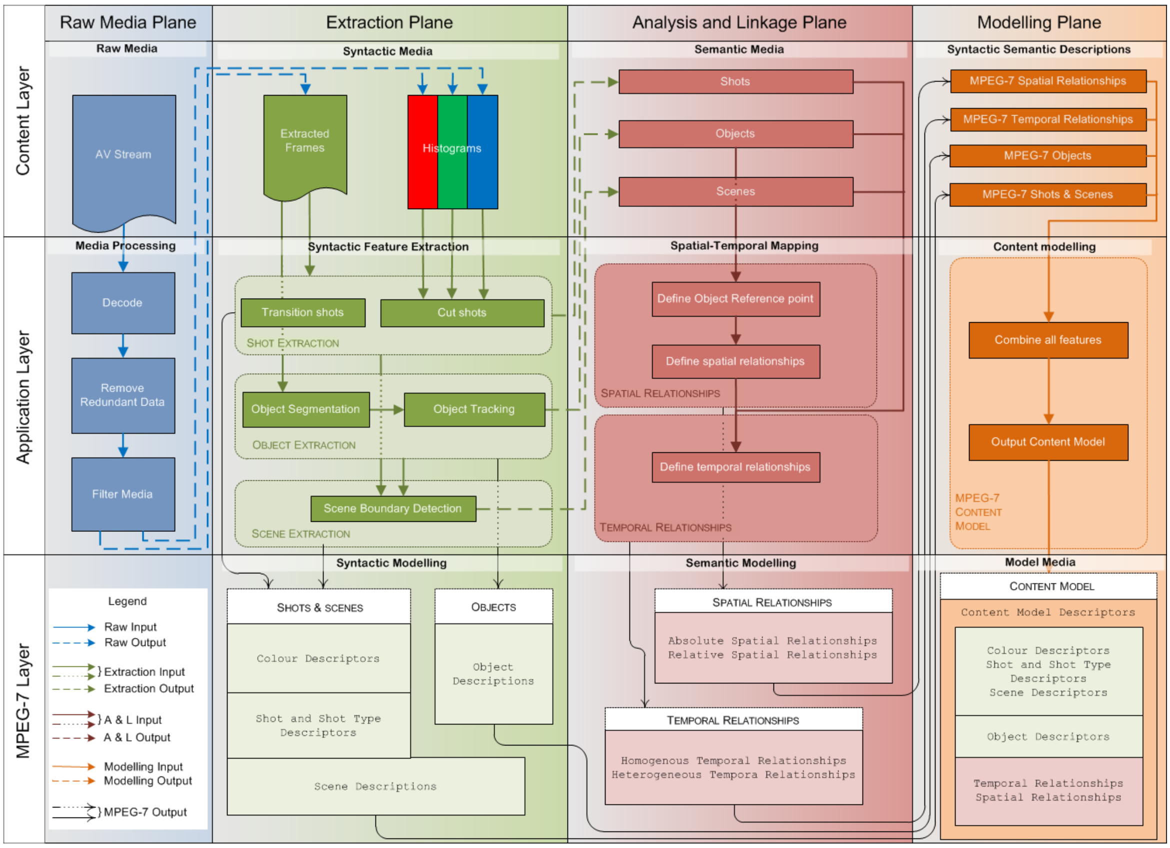


Figure 2.2: MAC-REALM DESIGN FRAMEWORK

In Figure 2.2 we show the MAC-REALM design framework. It shows the flow of the content processing through the planes and the content transformation through the layers. Where MAC-REALM intersects between layers and planes we have stages of processes of content or processed content. Each stage is responsible for the content conversion process at that intersection. The flow of content media between stages goes left to right and down then back up in the next plane.

We begin with the raw media, which is the video stream that will be extracted into content features and then into content descriptions, and finally represented by a MPEG-7 content model.

The pre-processing stage processes the raw video streams into syntactic media that is optimised for feature extraction. The pre-processing stage removes redundant data by eliminating chunks of data that is only incrementally different to each other by a small margin, as to be insignificant in change. The media is then filtered to emphasis the content feature properties that are used for feature extraction.

The syntactic media stage stores the filtered frames and histograms from the pre-processing stage, ready for the syntactic feature extraction stage. The syntactic feature extraction stage processes the syntactic media into syntactic content features. Three processes are part of the syntactic feature extraction stage, the shot, object and scene extraction processes. The shot processes extracts cut and transition shots. The object sub-process segments the objects and then tracks them. The scene process detects and segments scene boundaries. The segmented syntactic content features are then sent to two places the first is the semantic media stage for storage and the second is to the syntactic modelling stage. The syntactic modelling stage is where content features, once converted, are stored as MPEG-7 syntactic feature descriptions.

The semantic media stage is where the temporal and spatiotemporal syntactic features are stored ready for processing by the spatial-temporal mapping stage. The spatial-temporal mapping stage consists of two processes, the spatial and temporal relationships process. The spatial relationship process analyses the spatiotemporal objects and maps the spatial relationships between them. The temporal relationship process then analyses all content features created and maps all the temporal relationships between them. Once the semantic content features are converted to MPEG-7 content descriptions they are stored in the semantic modelling stage.

All the MPEG-7 syntactic and semantic content descriptions are stored in the syntactic and semantic descriptions stage. The syntactic and semantic descriptions are analysed and then integrated into a MPEG-7 content model. The content model is then serialised and stored in the modelled media stage.

2.6 MAC Layers

The three layers of MAC-REALM relate to the processing of the content that each plane goes through to convert its content media into modelled MPEG-7 content descriptions. The layers are the **M**PEG-7 layer, the **A**pplication layer and the **C**ontent layer. The content layer stores the media to be processed. The application layer processes the media and outputs syntactic or semantic descriptions of that media. In the MPEG-7 layer the syntactic or semantic description is modelled into MPEG-7 content descriptions. The layers and the flow of content between them are shown in Figure 2.3.

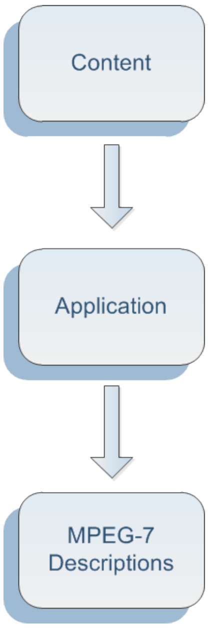


Figure 2.3: MAC-REALM LAYERS

Whereas the planes describe the transformation of the video stream into a content model, the layers describe the process of the content being transformed and translated from media into content descriptions of the media.

2.6.1 Content Layer

The content layer contains the media for MAC-REALM. The type of media contained changes as you move down the planes from left to right. As the media moves from raw to modelled state the content and content features become more advanced and the content descriptions are of a higher type. In each plane, the content media can be used as supplementary media for the MPEG-7 content model. This could prove useful for adapting the media to a user's usage environment as discussed in (Sofokleous & Angelides, 2008).

The first plane that has a content layer is the raw media plane. The raw media that is to finally be extracted and then converted into a content model, is an input at this point. The raw media will usually be a compressed digital video asset. The unprocessed media is usually compressed using some video coding standard or technique such as MPEG-1, 2, or 4, Quicktime, AVI or some other popular format for video encoding. The automatic feature extraction techniques cannot process the audio-visual stream in its compressed form, as the techniques require the visual components in uncompressed form for feature extraction to become possible.

The next plane with a content layer is the extraction plane. The extraction plane content layer has the pre-processed syntactic media that has been optimised for feature extraction. From these basic elements the syntactic features will be extracted, i.e. shots, objects and scenes. The extracted frames stored here can be used for digital item adaptation in conjunction with the content model to provide low-level content representation for scenarios where more complex media is not feasible. They can also be transformed into MPEG-7 BiMs (Heuer, Hutter, & Niedermeier, 2010) that use less bandwidth and storage space than the MPEG-7 XML making it useful for making the syntactic features adaptable for mobile devices and those with limited storage (M. Angelides & Sofokleous, 2013).

In the analysis and linkage plane, the content layer contains the semantic media that is the input for the spatial-temporal mapping process. The three type of feature stored here are the extracted syntactic shots, objects and scenes. The scenes, shots and objects are stored as java objects and can be reused to produce different metadata formats of the syntactic content features if another xml-based metadata exchange format is required.

The modelling plane's content layer is where the MPEG-7 syntactic and semantic content features are stored. The syntactic semantic media section aggregates all the syntactic and semantic descriptions before they are multiplexed together. Here we can see how the feature sets described earlier share characteristics and can be used in the multiplexing process to produce a more meaningful content model then if we were to use these feature sets in isolation. The MPEG-7 descriptions from the previous planes can be modelled independently of each other or multiplexed together in certain combinations in order to keep bandwidth and storage requirements to a minimum.

2.6.2 Application Layer

The application layer is the processing layer for MAC-REALM and processes the content media into MPEG-7 descriptions. The processing of the content media becomes more complex content feature-wise as MAC-REALM goes across the planes. The application layer has two tasks; 1) to process all content description into either syntactic or semantic content features and 2) convert these content features into MPEG-7 descriptions.

The choice of processing engine for each layer is selected on the suitability of the techniques to produce the content descriptions that will be modelled into MPEG-7 descriptions. In the extraction plane, we convert the syntactic media into temporal and spatiotemporal segments. In the analysis and linkage plane, the semantic media is processed into semantic relationships. The spatiotemporal regions are analysed and then the spatial relationships of and between them are linked. All syntactic and semantic features are temporally analysed and the temporal relationships between them modelled. In the modelling plane, we have the syntactic and semantic MPEG-7 descriptions. Here they are analysed and integrated into a fully compliant MPEG-7 content model that can be used by any MPEG-7 compliant application.

The raw media plane is the only exception to the processing paradigm of MAC-REALM layers as it is a pre-process and does not produce any MPEG-7 descriptions. Instead, it filters and optimises the syntactic media for syntactic feature extraction.

Each processing plane has an MPEG-7 XML binding engine. This converts and serialises all processed content features into MPEG-7 content descriptions. All the content descriptions that are serialised are well formed and are complete, and the MPEG-7 schema is used to validate them during their serialisation.

2.6.3 MPEG-7

The MPEG-7 layer stores the MPEG-7 content descriptions as they are created for each plane. In the syntactic and semantic content feature extraction planes, the MPEG-7 content descriptions for those planes are stored. In the modelling plane they are combined to finally create a content model of the syntactic features and semantic relationships. The MPEG-7 descriptions for each plane are complete and can be extracted and used to build customised content models for specific uses and domains if necessary.

The descriptions are produced by the application layer, as a product of its feature extraction process. Each plane produces MPEG-7 content descriptions that are related to the features extracted by that plane. The MPEG-7 content descriptions described come from both the content and application layer e.g. Frames from the content layer and shots from the application layer.

In the syntactic feature extraction layer we have, from the content layer, the keyframes that are described by the VisualDescriptor DS and ColorSpaceDescriptor DS (Ohm et al., 2003). These MPEG-7 content descriptions describe the colour distribution of the keyframe image using a continuous RGB value. The ColorSpaceDescriptor uses the RGB values extracted for all three bands during pre-processing. These values are quantised value of the three bands, each represented by 256 bins. From the application layer, the extracted shots scene and objects are described. The shots are represented by GlobalTransition DS, EvolutionType DS and Shot DS. Together these describe the length of the shot and the type of transition that precedes it. If it is a gradual transition, it will also state the length of the transition. The VideoSegmentTemporalDecompositionType DS is used to cluster the MPEG-7 shot segments into clusters that represent a scene. It only has duration attribute, as it takes its start time and transition descriptions from the first shot segment. This structuring of the temporal segment types allows a tight integration of the shots and scene, which reproduces there natural relationship. The objects are described using the MovingRegion DS and supporting descriptors and description schemes SpatialMaskType, SubRegion, Polygon and Coords. The Coords describes the silhouette of the object using Cartesian coordinates that have their origin in the bottom left corner of the keyframe and describe the position of pixel points. The objects are linked to the shots and scenes by referencing their unique reference as the prefix to the MovingRegion DS id reference. Using this to link all relevant temporal and spatiotemporal segments together allows a tight integration of the features on a structural level. This aids both concept detection through spatial and temporal inference and collaborative search techniques in video retrieval.

In the Analysis and linkage layer we have the MPEG-7 semantic features. From the content layer of this plane the id's references for all the syntactic features is retrieved and used to model them into Node DS's. These node DS's are used to instantiate reference nodes that are structured into both spatial and temporal MPEG-7 semantic graph. The use of nodes makes features they represent polymorphic in their proxy representation, as they can be referenced to each other temporally without the restriction of type and usage that is a limiting factor when comparing heterogeneous feature types within MPEG-7. The temporal relationships of all the content features extracted and derived from the video stream are described using the TemporalRelationship CS and modelled into a semantic graph using the nodes. With the polymorphic properties of the temporal relationships all the content features can be queried and analysed from multiple viewpoints and any combination of low and high level queries can be formulated without structural and conceptual constraints. The spatial relationships of the objects are described using the SpatialRelationship CS and modelled into a semantic graph using the nodes.

In the modelling layer, the MPEG-7 descriptions from both the Extraction plane and the Analysis and Linkage plane are integrated into a fully compliant MPEG-7 content model. The MPEG-7 descriptions in the content layer are the descriptions that were generated in the previous two layers. Within the application layer, these are taken and multiplexed into a content model, using the MPEG-7 schema to validate. The hierarchical structure of the final MPEG-7 content model is layered so that all elements can be referenced from a search query either independently or as a combination of features. This allows the content to be searched using multi-content type forms of queries. Along with the polymorphic properties of the temporal relationships, it makes it ideal for use in a generic and universal multimedia search space that is domain and purpose independent.

The complete MPEG-7 layer contains finished MPEG-7 content descriptions that describe the content and the results of the application layer of each plane. In this layer the descriptions from the extraction planes can be repurposed to be either integrated to the content that it was derived from in the content layer or they can be integrated together to provide a content model that is a comprehensive description of the content. If they are repurposed they can provide feature specific content descriptions that can be used for specific purposes that are focused on those features. If they are integrated into a richly detailed and multi-faceted content model they can be used to search the content using any combination of feature sets or used by concept detectors to infer new concepts to feature through inference that was not available due to structural or conceptual limitations.

2.7 REALM Planes

In Figure 2.4 we can see the basic flow chart diagram for the content extraction and modelling framework. This is the REALM processing model and is represented in the framework as planes. The Planes are; **R**aw media, **E**xtraction (of syntactic features), **A**nalysis and **L**inkage (of semantic relationships) and **M**odelling of the content features.

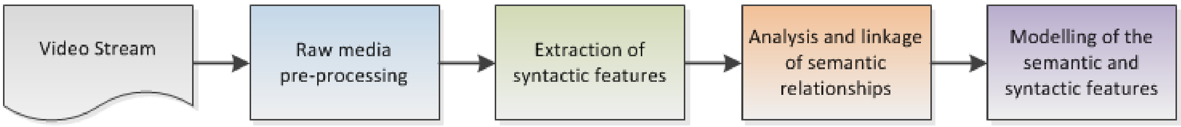

Video Stream
Raw media pre-processing
Extraction of syntactic features
Analysis and linkage of semantic relationships
Modelling of the semantic and syntactic features

Figure 2.4: VIDEO EXTRACTION AND MODELLING PROCESS (REALM)

The diagram shows how the video is processed through each stage beginning with pre-processing of the raw media. The syntactic features are extracted from the filtered media and the semantic features are derived and linked to those features. Finally, both syntactic and semantic content features are modelled into a standard content description document that can be read by any compliant video search and retrieval system. Each stage of processing where the content is converted into another content type is represented by a plane within the MAC-REALM framework.

2.7.1 Raw media

The video stream has to be pre-processed to filter the media so that it makes extracting the features more effective and efficient. The video stream also has to have feature redundancy techniques applied to remove the non-salient content data that adds no value to the extraction process and increases processing time.

Before filtering, we must initially decode any compressed video into an uncompressed state, where each frame becomes available. During the initial decoding, we perform a redundancy operation whilst we decode all the frames. There is usually between 24 to 30 frames per second for any video footage. Experiments have shown that only two frames per second is adequate for shot segmentation (Chan & Wong, 2011). Two keyframes are picked per second for use in the extraction process. The keyframes that are chosen are at the beginning and middle frames of every second. The timestamp for each frame is extracted and stored as a reference point that will be used in successive processing stages in the framework.

To know what filtering techniques we need we must first look at the features that are to be extracted and what it is required to extract them. The syntactic content features we need to extract from the raw media directly are the keyframes, shots and objects. Each feature needs different filtering techniques applied to improve its particular segmentation process. Shots need to have the lighting source in the target video clip to be even and without any abrupt changes e.g. flashing light sequences such as lightning. Objects, depending on the technique used for extraction, also need the light source to be even throughout the shot for the segmentation to be effective as the outline of the objects becomes obscured in dimly lit scenes. Both also need the removal of distortion or "noise" that can affect the segmentation process.

The way to negate the effect of such lighting changes is to use a colour space that is tolerant of such changes and can reduce their impact on shot segmentation and spatiotemporal extraction. To reduce the effect of lightning changes the video needs to be converted into the RGB colour model, if not in RGB already. RGB is shown to reduce the effect of lightning changes and improve invariance to shadows (Kristensen, Nilsson, & Öwall, 2006). YC_bC_r is marginally better than RGB for lighting and noise invariance but as RGB is commonly used by most codecs and recording equipment, the time taken to convert RGB to YC_bC_r is not worth the processing overhead for such a small gain. Converting to YC_bC_r from another colour space also adds noise in the conversion process. YC_bC_r is preferred in object extraction for as the illumination is limited to the Y band, but RGB is better for colour based shot segmentation algorithms as YC_bC_r is too insensitive for colour changes to be recognised. Therefore RGB colour space has more advantages then YC_bC_r for both shot and object extraction. After the conversion is completed the RGB values are extracted and histograms of each frame are stored for use in the content layer of the extraction plane.

Noise is removed by performing a flattening function over each of the extracted frames. Noise comes in the form of pixel "particulates" that are usually formed as artefacts left over from the decoding process as information was lost during the compression of the original video stream. The technique from (Yongquan et al., 2009) is adopted. A median filter is used over each keyframe to reduce the noise of each pixel by smoothing the pixel using the adjacent pixels. The noise reduction removes pixel-fine artefacts from the frame that could cause erroneous segmentation boundaries for both object and shot boundary detection.

The brightness and contrast are then adjusted to compensate for bad lighting levels. Once the adjustment is performed, the prominent features of the video stream become much more visible and thereby make the extraction processes more reliable.

2.7.2 Extraction of syntactic features

In section 2.3 the features that are needed to produce a universally compliant and accessible content model were identified. Temporal and spatiotemporal segmentation were the key syntactic features that will provide the foundation for the content model. The temporal content features will consist of one low level syntactic feature and two mid-level syntactic features. The low level syntactic feature will be shots, and the mid-level syntactic features will be the scenes and objects. The mid-level syntactic features have a conceptual structure and are harder to extract directly from the video stream. To facilitate this better the low level syntactic features are extracted first and then used as the basis for extraction of the mid-level syntactic features.

The temporal segmentation of the video into shots and scenes provides the foundation of the content model. The shots are the basic building blocks for the content model. Each shot will be represented by a keyframe extracted from the pre-processing stage within MAC-REALM. The spatiotemporal segmentation of the video stream will begin after the shot extraction process in the temporal segmentation component. After the spatiotemporal segmentation has taken place the scenes will be extracted.

Once all the features are extracted, they are described by MPEG-7 syntactic content description schemes. The syntactic feature extraction process for this plane is shown in Figure 2.5.

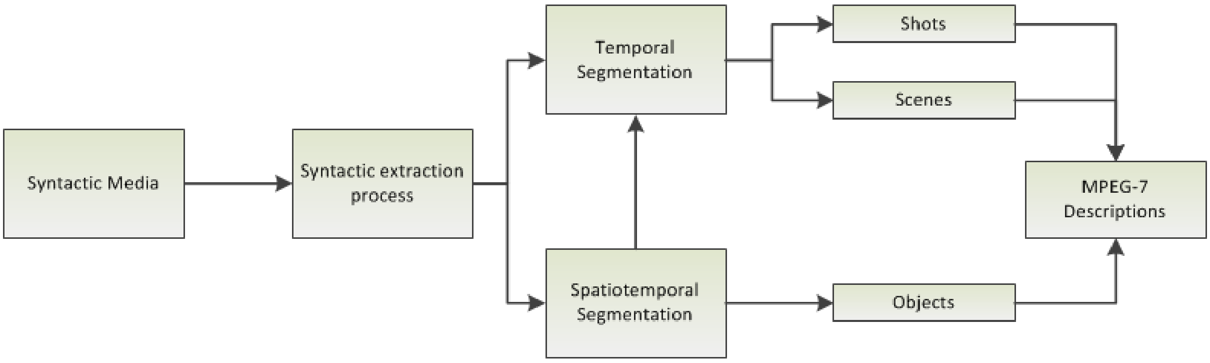


Figure 2.5: SYNTACTIC FEATURE EXTRACTION PROCESS

2.7.2.1 Shot extraction

Shots are the elemental unit of video storytelling. They are a continuous temporally uninterrupted sequence of frames taken by a single camera. They do not have any semantic characteristics of their own, but can have syntactic attributes that are significant for other features semantically. Shot segmentation is the first process that yields a content feature within MAC-REALM. The extracted shots will become the reference structure for all the other features, for both syntactic and semantic features. The shots will become the input and basic unit of the content model. Objects will be extracted from shots and scene will be a group of contiguous shots that are semantically related. The semantic relationships are derived from these content features; ergo they are derived from units of shots.

For shot segmentation we need to identify two features, the first is the boundary between shots and the second is the type of transition between the shots. The importance of the type of boundary is usually an indication of a semantic event change. Normal abrupt cut transition shots are normally associated with non-semantic changes, whilst gradual transition type shots are usually an indicator of a new semantic narrative within the video stream. When a semantic change does occur with an abrupt cut shots it is usually referred to as an 'establishing shot', which is a shot that is semantically neutral before a change in the narrative of the video. An establishing shot is usually a still or slow panning shot that is a visual break between events. These shots are usually short in duration and the syntactic low level features do not show much change. The other type of shot is a gradual transition and is usually associated with a semantic event change. These are usually indicated with a wipe, dissolve or fade type transition. These visual cues are important for establishing semantic event boundaries and are therefore important to any content extraction and modelling framework for video.

There has been varying success with different algorithms on each type of shot. Some algorithms can do one or the other very well but are incapable or have bad success rates for the other type of shot. Others have been adapted to do both but have limited success in achieving better results than using individual methods for each type. The MAC-REALM shot extraction technique is based on the research from (X. Chen & Liu, 2010) that uses a hybrid algorithm of two different shot extraction techniques. The shot extraction techniques use a combination of shot algorithms that complement each other by eliminating the weakness of the other. Each algorithm specialises in identifying either an abrupt transition or a gradual transition. The abrupt shot technique uses colour histogram difference and the gradual transition technique uses edge change ratio. Both algorithms are extremely effective and identifying the type of shot, they have been selected for.

In (X. Chen & Liu, 2010) they use fuzzy subset-hood theory for abrupt transition and fade out/in (FOI) transition shots. They begin using a binarysation process to assign frames to one shot or another. They convert each frame into greyscale and assign a value of either 1 or 0 to pixels depending on their shade. If they are not matched, they use an arbitrary threshold to approximate pixel difference until a match can be achieved. They then use an inclusion degree feature that determines if two frames belong to the same shot.

The binarysation process, which is a type of frame differencing algorithm, has been proven a computationally expensive and inefficient at shot segmentation (Gargi, Kasturi, & Strayer, 2000). MAC-REALM uses Colour Histogram Difference for abrupt transition detection and ECR for FOI/Dissolve transitions. This improves the performance of the overall extraction process for both types of shot. Each is well suited to its particular type of transition and each achieves good precision and recall rates. We reduce the complexity of the calculation using one step processes for both abrupt and gradual transition shots. The reduced complexity does not mean reduced performance. Results should be comparable in precision and recall as other similar techniques.

Colour histogram difference (CHD) is good at identifying abrupt transition shots due to the sharp change in the colour distribution of disjointed frame belonging to two different shots. Frames from the same shot tend to have a close fit to each other in terms of colour distribution. This is due to all the frames coming from one particular camera motion and therefore all frames are contiguous with little variation. Figure 2.6 is a visual representation of how CHD detects a shot. An illustration of frames represents the shot as it approaches the shot change boundary. Above the frames is shown a histogram line graph of the frames, over time, to illustrate the colour distribution change between shots and the moment the shot change occurs.

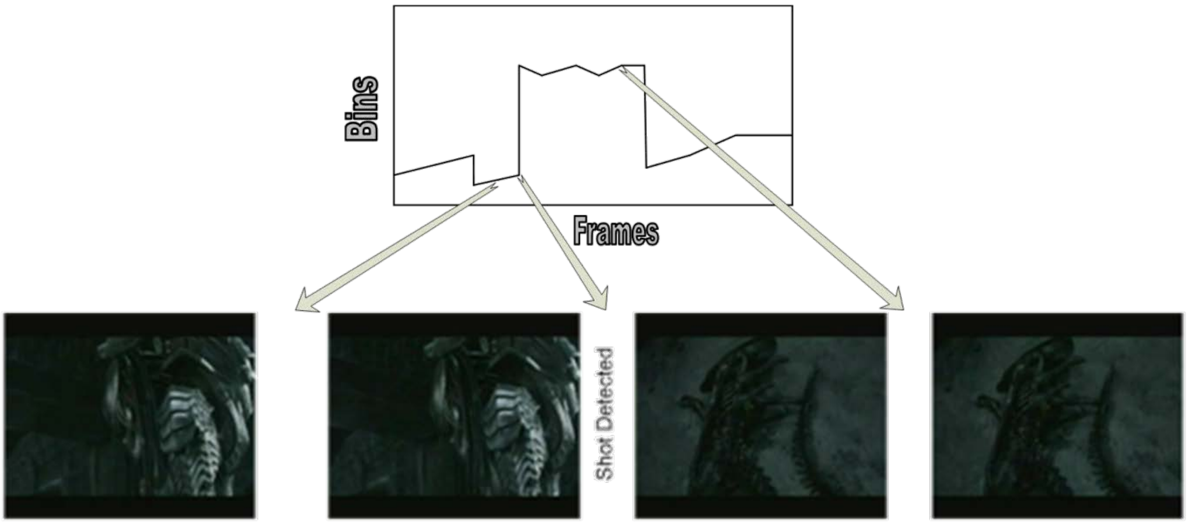


Figure 2.6: COLOUR DISTRIBUTION CHANGE BETWEEN SHOTS

There is some colour fluctuation between frames and the colour distribution is never uniform. Instead, the distribution falls within a certain range. Therefore, a threshold has to be set that allows for minor fluctuations between frames from the same shot. The threshold must be sensitive enough to distinguish between shots that have low light levels or are uniform in colour distribution.

MAC-REALM proposes an adaptive threshold that measures the fluctuation of the colour distribution over a certain period of frames and then takes the mean difference between those frames and multiplies them by a certain factor. That factor for triggering a shot change will be calculated using the following adaptive technique. Taking the mean of the fluctuation and using that as the basis of the threshold value negates any outliers of the colour distribution. Sensitivity to changing or low level lighting conditions is also minimised. To reduce the effect of uniform colour distribution over shots, the CHD threshold is performed over three bands (red, green and blue) and the adaptive threshold is worked out for each individual band. Using this method means that the colour distribution for each shot would have to be uniform for all three colour bands to miss the shot boundary.

To identify gradual transitions, the change in integrity of the edges changes within each consecutive frame image over time has proven to be one of the best methods. There are three main types of gradual transition: dissolve, fade in/out and wipe. MAC-REALM concentrates on finding only dissolve and fade in/out type transitions. Wipe transitions are not focused on as they are a rare feature. The edge change ratio (ECR) method is a very good technique for transition shots. It indicates the measure of the integrity of the edges for both types of transition shot. For dissolve shots, the edges are strong-weak-strong but in fade in/out the edges are either weak-strong or strong-weak respectively. Figure 2.7 shows the integrity of edges over a certain period of frames for both dissolve and FOI. Shot A is a fade in shot and shot b is a dissolve shot. The accompanying graph of edge integrity vs. frames is shown above for each type of shot. Each particular type of gradual transition has its own type of graph curve.

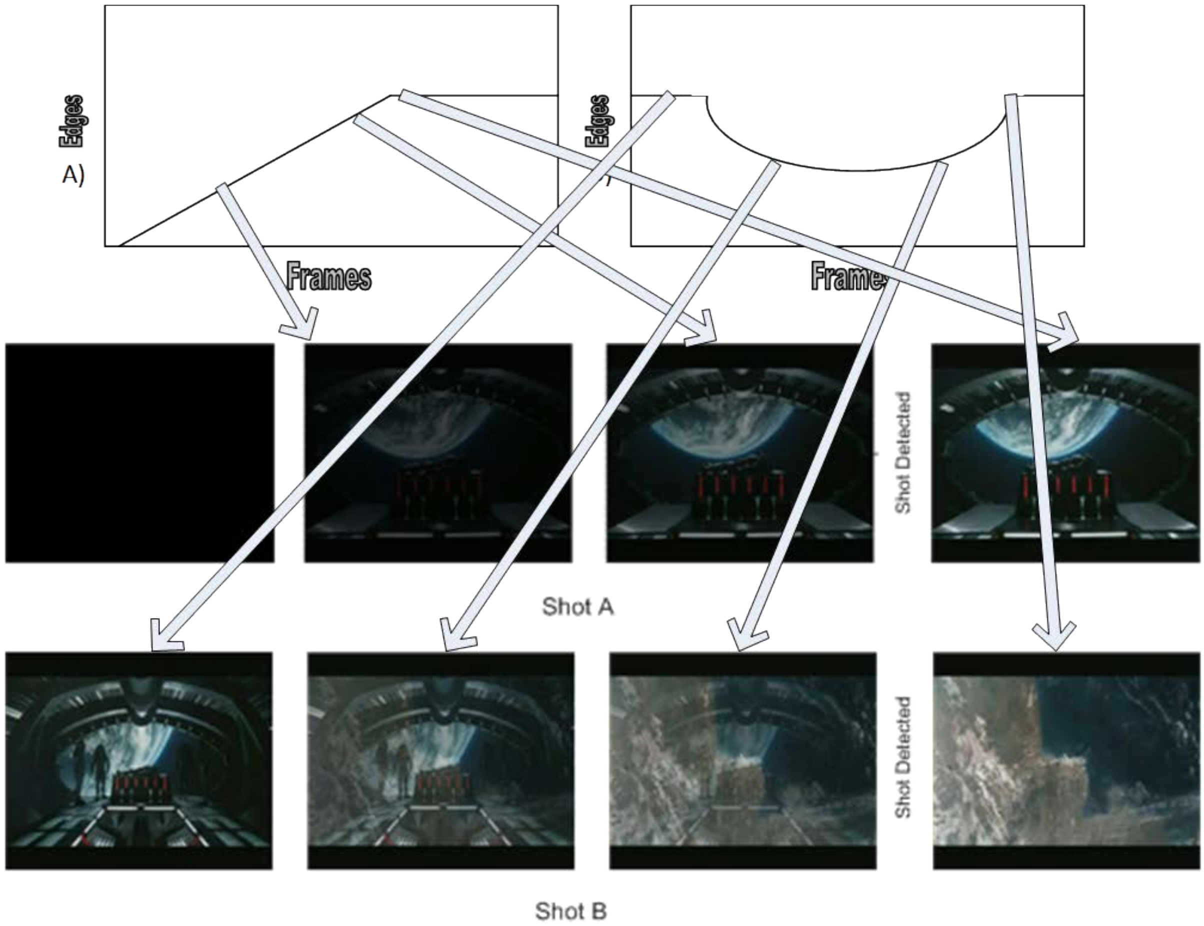


Figure 2.7: ECR SHOT DETECTION: A) FADE IN B) DISSOLVE

MAC-REALM uses the same adaptive threshold technique to detect the FOI and dissolve transitions. The difference is that it is the fluctuation of the of the edge complexity that is used instead of the colour distribution, and the average is taken over a sliding window of frames rather than two consecutive frames. The fluctuation of the edge integrity over a certain period of frames is measured. The change in edge complexity over time is measured over a fixed window of frames. If the edge complexity has a certain gradient, it can be matched as either a dissolve or a FOI.

2.7.2.2 Object extraction

Objects are one of the most fundamental building blocks of a content model. Each shot represents a unit of action. These actions build up to events. For the events and actions to take place, they must be performed by objects. This is the reason why they are important and form one of the major components of content modelling.

For spatiotemporal segmentation, or object extraction as it is more commonly known, a set of frames must be segmented into foreground and background regions. To achieve segmentation a frame, which is an image snapshot of action, must be divided into a number of disjoint regions such that the features of each region are consistent with each other i.e. belong to an object. Since images generally contain many objects, which can be obfuscated by clutter, it is often not possible to define a unique segmentation.

Another problem with objects as mid-level features is taking into account the semantic perspective needed to segment them. Objects need to be perceived cognitively to establish their boundaries. Even though they are directly extracted from the syntactic information in the video, and therefore syntactic in structure, they have a semantic connotation in that an object is a matter of perspective and cannot be reduced to any syntactic key feature points. This is complicated further by objects consisting of different parts, producing a hierarchical structure of connected "sub-objects", for example as a person can be split into limbs. In addition, there is no correlation between the low-level syntactic features the object consists of and the object itself. Movement, Colour, shape and texture cannot be relied upon to distinguish the object solely.

Deciding which regions belong to what object, and what is foreground and what is background is the main problem of spatiotemporal segmentation. A degree of user interaction is required for the most accurate methods in generic situations (see section 1.3.2.4). In other words, the segmentation problem can be ill posed when working in an unsupervised framework. Interactive algorithms allow the user to label a few pixels as either object or background, thereby making the segmentation problem well posed. In addition, there is the problem of tracking the spatiotemporal segmented regions once the initial segmentation is performed. It proves computationally very expensive and inefficient if the segmentation technique was used for the same technique for each frame. It could also lead to irregular segmentation of objects over time as the segmentation process reinitialises for every frame.

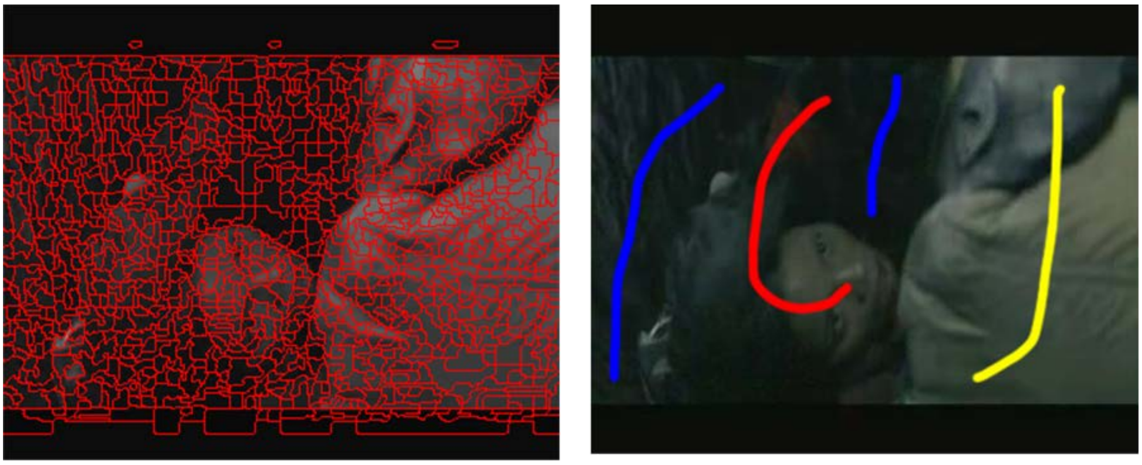
To solve both problems two algorithms are used. One algorithm is used to segment the initial frame of the shot and then a second to track the segmented region through the shot. The advantage of using both techniques is that gives the most effective and efficient form of extracting objects for the purposed of MAC-REALM. This allows for a highly accurate segmentation of the object, which is then tracked in a coherent and efficient manner.

To segment the initial frame and initiate the object instantiation we use the technique from (Noma, Graciano, Cesar Jr, Consularo, & Bloch, 2012). They use an interactive attribute relation graph (ARG) segmentation technique to segment an image into foreground and background regions. It uses a watershed technique to oversegment the image into regions of spatial and colour homogeneity, known as the input graph. A user defined input graph is then overlaid the over segmented image, known as a model graph, and is used to mark regions on the input graph. Examples of the input and model graph are shown in Figure 2.8. These marks are used to instantiate a region-merging algorithm that is based on discrete search using deformed graphs to efficiently evaluate the spatial information. The advantages of using the ARG technique to segment objects is that:

a) It reduces the problem of clutter in the image and focuses on regions of interest improving the definition of the segmentation

b) It reduces the merging of objects that have similar visual properties and are touching e.g. two people in shot together wrapped around each other

c) It the user input from one image can be reused on multiple similar images, reducing the supervision of the process.



(A) Input Graph: Over segmented watershed image

(B) Model Graph: User defined image showing two objects (red & yellow strokes) and background blue strokes

Figure 2.8: Examples of A)Input graph and B) Model graph

To track the spatiotemporal region MAC-REALM uses Hausdorff matching SVD covariance descriptors from work by (Guo, Xu, Ma, & Huang, 2010). Hausdorff distance measurement is a widely used tracking algorithm (Z. Liu, Shen, Feng, & Hu, 2012). The reason for using this particular variant of the method is that it is robust against rotation and scale change, a problem for the Hausdorff tracking method. It also has proven to have a lower computational expense than other Hausdorff tracking algorithms. These factors make it ideal for tracking objects in MAC-REALM as it reduces the time of processing whilst providing accurate tracking.

The hybrid object extraction and tracking technique for MAC-REALM is a computationally efficient and effective way of segmenting and tracking objects. The shots extracted from the shot extraction process form the basis of the spatiotemporal segmentation process. A frame from the syntactic feature content layer that represents the key point in which the object first appears clearly in the shot is used for the initial segmentation frame. The frames extracted are at one second intervals, which represent an adequate interval for sampling the change in object spatial behaviour. At this frame rate most action changes are caught but computational expense is reduced. Each frame is then segmented, using the ARG technique, into a number of disjoint regions such that that the features of each region are consistent with each other. These regions are then tracked by the Hausdorff spatiotemporal region tracking algorithm tracking for the duration of the shot. The reuse of earlier features which have already been extracted, and using a separate computationally less expensive tracking algorithm instead of using the segmentation algorithm for all images, reduces the processing time of the object extraction and tracking

2.7.2.3 Scene extraction

Once objects are segmented and tracked, along with the extracted shots, they are used as the input features for the segmentation of the scenes. The scenes are an important syntactic feature as far as laying a foundation for semantic features. Scenes are another complex syntactic feature and, like objects, can be described as Mid-level content features. Scenes are syntactic features that are semantically defined. They are a collection of actions constituting shots that when combined describe a single event. To segment a video into scenes, shots must be clustered together based on a common semantic theme.

Scenes are a type of cinematic grammar that is used by film and video creators to create story units. Just like grammar in a book, the video must consist of self-contained sections that describe the a story "unit" that is part of the plot of a book or play. Shots and objects in video are akin to scenes and actors in the structure of a script, indeed shots and objects are all scripted in a screenplay, and a storyboard formed of how they visually play out. Scenes themselves are like acts, which play out a particular story plot of a play. The structure of a scene does not itself play a part in the semantic themes of a scene, but its arrangement and use define it. MAC-REALM exploits this grammatical relationship between video syntactic features and their correlating semantic themes to cluster shots together into scenes.

Defining a common semantic theme for groups of shots is beyond present state-of-the-art scene segmentation. As reviewed in section 1.3.2.3 most scene segmentation algorithms use specific syntactic cues that can only be relied upon within a certain domain or use user input or training data to initiate segmentation for techniques that try and extracts scenes generically.

For MAC-REALM we have chosen to use a scene boundary detection technique based on work from (Marios C Angelides & Kevin Lo, 2005). They proposed a genetic programming experiment that uses video and audio features to formulate rules that would identify the start of a scene boundary. To reduce the computational complexity of analysing both video and audio streams MAC-REALM will only formulate rules using video features. In Chapter 4, we will show this approach has positive impact on the performance of the algorithm and increases the efficiency of the overall scene segmentation process.

The genetic programming method is preferred for scene segmentation because it takes into account the semantics perception associated with identifying scene boundaries and applies them abstractly to the syntactic features. MAC-REALM applies a multi-content type syntactic approach to defining the scene boundaries. Rules are evolved consisting of multiple syntactic features that have an association with the start of a scene change. These rules state the relationship between certain syntactic features and their visual cues are good at identifying scene boundaries. Using training data, the rules are evaluated on their fitness to identify scene boundaries. Those scenes that are better at identifying the boundaries are evolved further. This process is repeated until a rule is formulated which identifies a certain percentage of scene boundaries, or the closest matching rule after a certain number of cycles is achieved.

Using this particular method of scene segmentation allows MAC-REALM to maintain its genre and domain independence as this method can handle generic content and can achieve a good degree of accuracy for scene segmentation compared with similar methods (see chapter 4).

2.7.3 Analysis and Linkage of semantic relationships

Semantic querying is built upon the main categories of "what and when" and "who and where". What and when refers to events and the temporal relationships between them. Who and what refer to objects and where they are, to their environment and to other objects. The semantic analysis and linkage plane establishes the "when" and "where" that is so vital in semantic search and concept detection.

Although semantic relationships have been critical to semantic search and retrieval they have also been noted as playing an important part in concept detection methods(Weiming et al., 2011). The spatial and temporal relationships between content features play an important part in determining the relationships between concepts. From these relationships knowledge of the actions and events can be learnt, and concepts that share similar themes can be grouped and new concepts inferred for the content. This makes accurate modelling of spatial and temporal relationships very important in discovering and learning concepts and ontologies.

The MAC-REALM analysis and linkage plane is responsible for modelling the relationships between low and mid-level features extracted in the previous plane. As shown in section Table 1.6 all video indexing and modelling systems do not explicitly model the semantic relationships between the content features. They treat the matter of spatial and temporal relationships as a post process to modelling that is done in an ad-hoc manner. This can lead to ambiguity as different methods for processing spatial and temporal relationships can lead to them being interpreted with different meanings.

For uniformity between query results, and for improving concept detection through spatial and temporal concept modelling, having explicitly modelled spatial and temporal relationships is a necessity. This would allow the formation of consistent results and concept detection using semantic ontologies over all applications that used the content model.

In Figure 2.9 we see the processes of the analysis and linkage plane. The semantic media from the content layer is processed to produce spatial and temporal relationships. The temporal relationships between spatial relationships and other features are also modelled. They are all finally converted into MPEG-7 content descriptions.

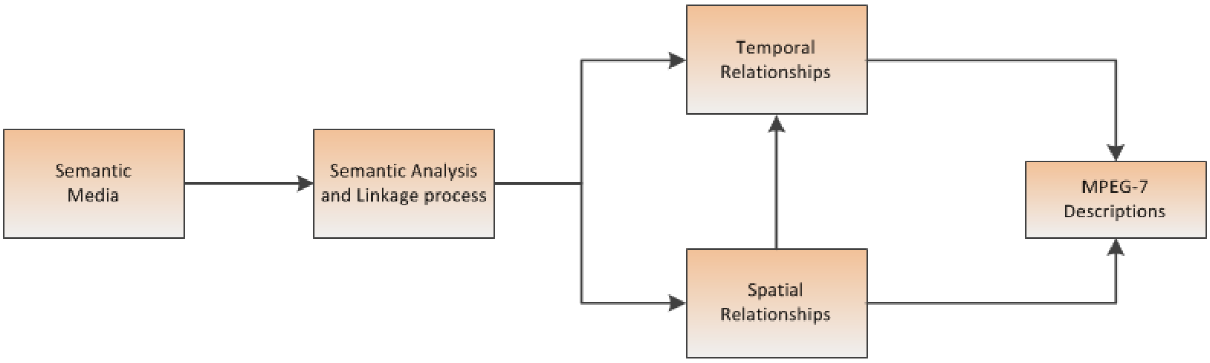


Figure 2.9: SEMANTIC RELATIONSHIP ANALYSIS AND LINKAGE PROCESS

2.7.3.1 Spatial relationships

As discussed in section 1.3.3.1, spatial relationships have not been given much attention in recent studies. Though spatial relationships have been formalised, there has been no attempt at unifying the processes from which they have been derived. In section 1.3.4.2, we see that video content extraction and indexing applications approach the problem as a post process ad-hoc methodology problem.

The main problem that stops spatial relationships being uniform in through content based video search applications is where the reference point for basing the spatial reference is calculated. None of the systems reviewed in section 1.3.4.2 stated the quantitative methods used for defining the spatial relationships. This is not an inconvenience if the content model is used exclusively for the purpose or application it was designed for. It becomes a problem though when other applications use the content model and then use a different reference point for the spatial relationships. This could lead to a different interpretation of spatial relationships and therefore different results if queried with the same criteria.

To find a solution to this particular problem we look at the possible ways of how to define the reference points for objects, for both absolute and relative spatial relationships. With absolute relationships, the reference point or points must accurately depict the relationship between the object and the global position it occupies within the frame. In relative relationships, the reference point or points must define accurately the position of the objects in relation to each other in real and perceived terms. Both sets of reference points should also ideally align themselves to the same philosophy of definition of referencing as not to induce any problems from querying on both types of relationship.

With absolute relationship, the initial reaction is to use the centre point of the object. The reason for this is the natural way absolute positions are judged by humans. The main problem here is defining the centre point of a non-uniform body. The irregular shape may mean that the centre of the body may not be obvious as irregular protrusions may make defining the centre more complex. What is needed is a technique that takes the most natural and actual representation of the centre of the mass. MAC-REALM for this reason uses the centroid of a mass technique (Marghitu, 2012). The centroid of a mass finds the arithmetical mean of all the points in a two dimensional object. If we use the edge silhouette of the object for the points, we can determine the centre of mass. Setting the absolute relationship on the centre of mass allows the definition of the relationship to not only be based on the true centre but also the perceived centre of the object as its central mass is centred around that point.

For the relative position, we find the centroid of the mass for both objects. Their relative positioning is then based on the calculation between points. This technique works well because it is accurately used to depict relationships where one object might be larger than the other. If we used the nearest point between both objects, this might give an inaccurate reference point as the mass of the object might be located in another region or dispersed over a great area.

Using the centre of mass as the defining technique for both relationships means that querying and comparisons of spatial co-occurrence can be modelled with consistency throughout. Both sets of relationships will provide a consistent approach to absolute or relative spatial relationship queries.

2.7.3.2 Temporal relationships

In section 1.3.3.2 and 1.3.4.1, it was shown that temporal relationships are fundamentally important to the areas of video extraction and content modelling. Video is a temporally defined media and therefore having the ability to search it temporally is an important aspect for all video search applications.

It has also been used for concept detection, but has only been considered by a few and is not used as widespread as spatial relationships. This is surprising as the temporal relationships have the advantage of being able to model all syntactic and semantic content features and spatial relationships can only be used for spatiotemporal regions. Using temporal relationships between features increases the concept detection probabilities throughout a video, and seems an intuitive answer to not only modelling concepts based on objects but more accurately to modelling concepts based on events.

The ability to model temporal relationships between both syntactic and semantic content features is unique. As shown in Table 1.6 most systems that use temporal relationships do not model them explicitly; only the semantic features are modelled, which also misses the opportunity to model the syntactic features. All features in video have temporal properties that can be used to find similarities and associations between them, and can also be used to compare features temporally against each other. These can then be used by applications to model or infer concepts between syntactic and semantic features, which can be used to reduce the semantic gap. The dynamism of temporal relationships forms intra (between the same content type e.g. shot and scene) or inter (between the different content type e.g. shot and spatial relationships) temporal relationship links between features based on proximity and co-occurrence. Because of its flexibility in being structurally independent of feature types it can be used by applications to detect and infer concept relationships between features of different types and domains.

Modelling the temporal relationships explicitly is important for completeness of the relationship between all features. Most video content extraction and indexing applications only model the temporal relationships between concepts and ignore the relationships between them and their underlying syntactic foundations. This limited view of temporal relationships does not exploit the potential that modelling all the features will have on being able to search the content temporally in a content-type independent manner.

Below in Table 2.2 we show the relationship combinations between syntactic and semantic features. The temporal relationships between these feature sets can be described as either intra-temporal or inter-temporal relationships. Where the feature sets are homogenous in structure (e.g. both syntactic) or in concept (e.g. both semantic) they are described as intra-temporal relationships, where the feature sets are heterogeneous they are described as inter-temporal relationships.

	SHOTS	SCENES	OBJECTS	SPATIAL RELATIONSHIPS	TEMPORAL RELATIONSHIPS
SHOTS	INTRA	INTRA	INTRA	INTER	INTER
SCENES	INTRA	INTRA	INTRA	INTER	INTER
OBJECTS	INTRA	INTRA	INTRA	INTER	INTER
SPATIAL RELATIONSHIPS	INTER	INTER	INTER	INTRA	INTRA
TEMPORAL RELATIONSHIPS	INTER	INTER	INTER	INTRA	INTRA

Table 2.2: INTRA/INTER RELATIONSHIPS BETWEEN SYNTACTIC AND SEMANTIC RELATIONSHIPS

The relationship between homogenous entities are described as intra as the attributes between them have a direct correlation to each other in content type and can be compared in a similarly structured content queries. Entities that do not share similar attributes in content type are considered inter-temporal. They cannot be queried together semantically as one feature does not have any semantic meaning. Due to the heterogeneity of the content features involved in inter-temporal relationships, the temporal relationship is the only feature that can be queried semantically between them. This allows querying of temporal relationships between these feature sets, allowing for multi modal querying on a semantic level, whereas before it was only possible on a logical level.

The temporal processing is a basic chronological comparison exercise. For all syntactic and semantic features, the timestamp of when they begin and end is stored. This is then used by the temporal processor to calculate each relationship. Once all the relationships are processed, they are explicitly stated and referenced using the unique id of the features involved. Any application that can use the content model can then analyse and compare the relationships for any feature against all other features. There is no need for the application to calculate or query the content model for the availability of temporal relationships, all are available and all possible combinations of relationships and features are included.

2.7.4 Modelling

Once we have modelled all the syntactic and semantic content features we need to integrate them into a content model that can be used by MPEG-7 compliant applications. The content model must be integrated so that all the syntactic and semantic content features are interlinked and can be searched by queries that are formulated using different content requirements.

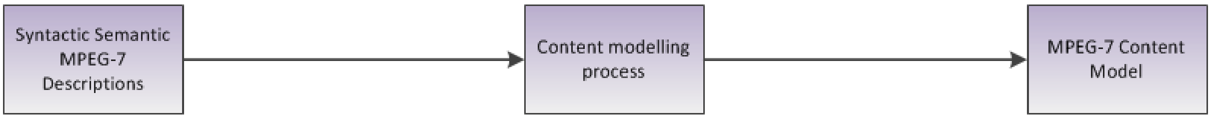

Syntactic Semantic MPEG-7 Descriptions
Content modelling process
MPEG-7 Content Model

Figure 2.10: MODELLING PROCESS

To help bridge the semantic gap between the underlying syntactic foundations and the semantics of the content the extracted content description need to integrate the syntactic and semantic features into a unified content model. This would help to address the problem of the many types of video content query that can be formulated (see section 1.3.4.1). A content model should be able to handle queries with impartiality to the domain or genre of the querying application. The content features must be accessible to as many applications as possible.

MAC-REALM proposes a solution to the semantic gap problem by modelling the extracted syntactic and derived semantic features into a hierarchical MPEG-7 compliant content model. The content model uses the syntactic features as the foundation blocks of the content model, and then uses MPEG-7 semantic graphs to link the semantic relationships to the syntactic foundations. The use of the MPEG-7 graph description scheme in defining temporal relationships allows the content model to establish a semantic multi-feature linking mechanism between all features regardless of content type. The content model only uses standard MPEG-7 tools to model the features. This way any MPEG-7 compliant application can readily interpret the content model with no ambiguity of the content semantic.

In section 1.3.4.1, the four main types of semantic feature categories were identified that should feature in all content models. These were spatiotemporal objects, the spatial relationships between them, events depicted within the content and the temporal relationships between all the features. The four categories describe these features in semantic terms only. The fifth type of feature was temporal segments, but these are syntactic and already represented. The semantic categories alone are not adequate query formulation can also be a mixture of syntactic and semantic features. We have to revise these features into an integrated content description framework, with all features integrated into a layered hierarchy that supports multi-content type querying. In Figure 2.11 the mapping of the four semantic categories to MPEG-7 content descriptions to produce the MAC-REALM content model is shown. The spatial and temporal relationships are high level features that are directly mapped to the content model as they are. Events are represented as mid-level temporal segments (i.e. scenes). This integrates it with the other low-level type temporal segments (i.e. shots). The spatiotemporal objects are described using moving regions.

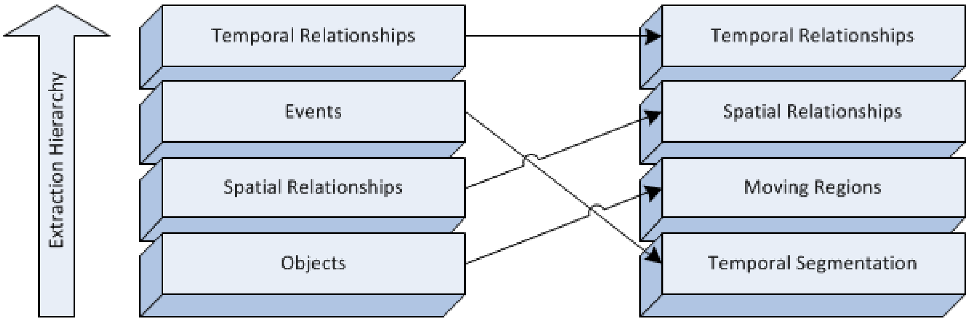


Figure 2.11: TRANSLATION OF SEMANTIC FEATURE CATEGORIES TO A SYNTACTIC SEMANTIC CONTENT MODEL

From Table 1.6 we found that most video content extraction and indexing applications do not explicitly extract and create a content model that has all five video content feature types categories. MAC-REALM takes a step closer to 'bridging' the 'semantic gap' by incorporating a combination of low, mid and high content features to provide a foundation that can be used for searching the content. It can also be used for concept detection providing a framework for concept discovery by learning spatial and temporal concept modelling using the spatial and temporal relationships.

Although users formulate their queries on a semantic level, they formulate the queries from the basis of a syntactic foundation. The foundations of an integrated content model needs to be built on two feature sets; syntactic features that support and help define concepts, and semantic relationships that can be used to infer and model concepts. MAC-REALM uses the shots as the skeleton of the content model as it is the syntactic foundation of the content. The spatiotemporal moving regions that represent the objects are then associated with each shot that they belong to. The scenes are created from the shots, providing a very close coupling between them and the shots. The close coupling between the scenes and shots leads to a relationship between the spatiotemporal objects and scenes. Finally semantic graphs are created, first for the spatial relationships for the objects and then for the temporal relationships between all the features. This interlinking and integration of features makes the content model searchable from a multi-content type perspective, and rich and granular in description.

MAC-REALM begins by modelling the temporally segmented syntactic features that have been extracted, namely the shots and scenes. The shots are embedded within the scenes they originate from as well as the shot transitions that precede each shot. Within each shot we have the description of the colour histogram of the shot, calculated from the aggregation of colour samples from extracted frames within the shot. The shot and transition descriptions are linked by their time attribute. Therefore, if a shot is the start of a scene then the associated shot transition is also the transition for the start of the scene. Each scene and shot is given an id reference. The scene id reference just states the scene number in reference to its position numerically to other scenes. The shot id reference however incorporates the scene it originates from as well as the numerical position of the shot. Shots that do not belong to a scene do not have a scene number, just a shot number. The shot numbering is carried through to the next shot, regardless of what scene it originates from, in order to show the position of orphaned shots in relation to other shots.

After the scenes and shots have been modelled, the spatiotemporal moving regions are modelled. Each spatiotemporal moving region has a unique id reference that is used to link it to the scene/shot it originates from. Within the unique identifier of the moving region is an "object" reference id assigned that uses the frame number of the shot. The object number relates to the number of objects within the shot, with the frame number showing exactly when the object appeared in the shot. This helps to identify and link the spatiotemporal moving region to the temporal segment it originated from.

The semantic modelling phase is split into three parts; relationship nodes, spatial relationships and temporal relationships. The relationship nodes' are responsible for the polymorphic nature of the semantic relationships. Nodes take a feature and assign a unique identifier for each feature that describes only the features in arbitrary terms, whilst removing the syntactic or semantic attribute descriptions. This allows for building temporal relationships between all features, without becoming encumbered with syntactic or semantic description that would complicate content based video search and retrieval tasks.

Nodes are assigned to every feature instantiation of shots, scenes, objects and spatial relationships. These are then used by either (in the case of object nodes) the spatial relationships or temporal relationships when identifying the source and target of the entities described in the relationship.

Spatial relationships use the nodes to describe the spatial relationship between objects, taking into account the change of that relationship over time. Each object is tracked and when it's spatial relationship changes, a new node is created that identifies that change. All spatial relationships are modelled into nodes themselves, for the purpose of defining their temporal relationships.

The temporal relationships of all the features are then modelled using the node identifiers for source and target of the relationships. Each intra-temporal and inter-temporal relationship is mapped for all the feature sets. Temporal relationships are not modelled into nodes because of their semantically finite nature.

2.8 Summary

Chapter 2 proposes a design of a content feature extraction and modelling framework called MAC-REALM. The framework is introduced and the motivations behind the requirements of MAC-REALM are examined. The following two sections examine automatic content feature extraction and content modelling design requirements in further detail. These are then stated as formal design requirements that elaborate on the requirements from the objectives in chapter 1.

The MAC-REALM Framework is presented as an architecture that incorporates the design requirements into function components that are linked by a custom video processing pipeline.

Content passed through the pipeline and is converted from content media to content descriptions in layers of different content feature levels as the video stream is translated into a content model.

The design of the content, application and MPEG-7 layers is then looked at. For the content layer we describe the media to content description conversion for each plane. The content layer stores the media for each plane that will be processed. The application layer converts the content for each plane into content descriptions that are relevant for that planes function. The MPEG-7 layer is where the content description are modelled into MPEG-7 content descriptions. An in depth view is given of the planes and how they are to perform their function. The choices of the processing strategy for each component are discussed in reference to the function it performs in the MAC-REALM framework. Where applicable the sub-processed are discussed and the techniques employed are focused on in their own sections.

CHAPTER 3: PROTOTYPING MAC-REALM

In chapter 2 we proposed MAC-REALM, a cross-functional framework that is able to extract video content features and model them into a MPEG-7 content model that tightly integrates syntactic and semantic content features. The extraction process takes place over two function planes with another function plane responsible for modelling the content into a content model. Before the extraction of features can begin the content is pre-processed to optimise the extraction potential of the video stream. This chapter presents the implementation of MAC-REALM, and its three-layer, four plane architecture.

In this chapter the design for MAC-REALM is implemented into a proof of concept prototype. A modular framework is developed and the component modules for each plane are added to provide the functions of MAC-REALM as described in the design requirements (section 2.4.). The MAC-REALM prototype is developed using an iterative prototyping methodology. Existing codebase is repurposed and modified to implement the function components of the framework. The components are self-contained modules that are loosely coupled modular framework that used custom video processing pipeline to pass content between the components. This implementation strategy allows the modules to be updated or extended without altering the functionality of the framework as a whole. This development strategy allows the prototype to be maintainable and extendible for future development of the platform.

The chapter is organised as follows. Section 3.1 presents the implementation requirements of the MAC-REALM prototype, and then introduces an overview of the custom video processing pipeline between the modules within the framework. Section 3.2 discusses the Raw Media plane and shows how the AV stream is decoded and filtered for feature extraction. Section 3.3 presents the Extraction plane and discusses the multi-tiered automated/intelligent heuristic processing that automatically extracts syntactic features and then models these features into MPEG-7 visual tools. Section 3.4 discusses the Analysis and Linkage plane that derives semantic relationships, both temporally and spatially, of the syntactic content features extracted and the semantic relationships themselves. Finally, in section 3.5 the Modelling plane is presented and how the syntactic and semantic features are combined together to provide an MPEG-7 content model that enables granular search and facilitates multi-content type video search.

3.1 MAC-REALM Framework

The MAC-REALM design requirements in chapter 2 outlined the requirement for a video content feature extraction and modelling framework. These design requirements were derived from the research objectives and methods in chapter 1. From these requirements the aim of the MAC-REALM implementation is as follows:

1. The framework will convert the media into a content model through an extraction process that segments and then models syntactic and semantic content features. The integrated content model will be MPEG-7 compliant. The framework will consist of functional planes arranged in custom video processing pipeline that will:
    a. Pre-process the raw media to increase the potential of the extraction of the content media and reduce the processing during the extraction phases of MAC-REALM.
    b. Extract syntactic features from the video stream. The feature will be extracted in a hierarchical process of extraction that will:
        i. Extract shots and identify the type of shot transition.
        ii. Extract objects and track them
        iii. Identify scene boundaries
    c. Derive from those features explicit spatial and temporal relationships. These semantic relationships will specifically implement:
        i. A reference algorithm that defines the centre point of objects to provide uniformity in spatial relationship definition across platforms.
        ii. To model temporal relationships of all syntactic and semantic features to facilitate semantic multimodal search for all combinations of content type.
    d. Create a content model that integrates the syntactic and semantic content feature descriptions into:
        i. A hierarchical structure to allow the content to be searched granularly.
        ii. A content model that is coded to be accessible to a wide a range of MPEG-7 compliant applications, regardless of the profile or version.
        iii. An interlinking structure of syntactic and semantic content features that are modelled to reduce the semantic gap and emphasis the relationships between the heterogeneous features.

For the prototype implementation of the MAC-REALM framework, it was decided to integrate existing framework platforms if possible and reuse other codebases. This section will provide an overview of MAC-REALM prototype, implemented with the listed requirements.

MAC-REALM is designed as a framework that allows the components that it consists of to be added, amended or replaced by other components. Therefore what was required was a platform for implementation that was both extensible and modular. For these reasons MAC-REALM was designed on the NetBeans platform[4]. NetBeans is a generic platform for swing applications and is written in Java. It provides a modular platform for designing complex desktop applications such as MAC-REALM that require GUI environment that has multiple screens for different functions. The programming platform has many features such as ready-made modules and tools designed to streamline the development process. The interaction between all the components is handled by NetBeans and does not require any complex coding. NetBeans offers many advantages that are useful to the implementation of a prototype for MAC-REALM.

To begin with NetBeans employs a module system where each logical component of MAC-REALM can be created and then be deployed into the MAC-REALM container. The MAC-REALM container uses a bootstrap module that the different module functions are registered to, this dictates the order the modules are available and how they are integrated into the MAC-REALM container to provide the complete MAC-REALM prototype.

The communication between the modules is another key advantage of NetBeans allows the different modules to interact through a look-up service that provides a generic communications mechanism that allows all the modules to correctly transmit and receive data from each other. The look up service facilitates the exchange between not just between native java data structures but also MPEG-7 XML based description schemes through JAXB[5]. The lookup service also can handle other non-native API's such as C++ and MATLAB, making it an important part of the ability of MAC-REALM to be an extensible framework that is independent of propriety restrictions.

The usefulness of NetBeans in providing an ideal coding environment as test bed to MAC-REALM is in the ability to manipulate the modules and the coding level during runtime. This allows deployment, debugging and testing of any module of MAC-REALM without having to halt

[4] https://netbeans.org/features/platform/features.html

[5] https://jaxb.java.net/

the other components. The NetBeans Platform provides a virtual file system, which is a hierarchical registry for storing user settings, comparable to the Windows Registry on Microsoft Windows systems. It also includes a unified API providing stream-oriented access to flat and hierarchical structures, such as disk-based files on local or remote servers, memory-based files, and even XML documents.

Figure 3.1 depicts the implementation of the MAC-REALM Framework. The diagram highlights component intersections between the layers and the planes. Within each component, features or processes are shown within them. In the content layer, we can see all the features that are created and stored as we go down the planes. For the processes in the application layer the algorithm(s) that are implemented are shown within each sub-component. The MPEG-7 layer contains the MPEG-7 description schemes that are used to describe the modelled syntactic and semantic features.

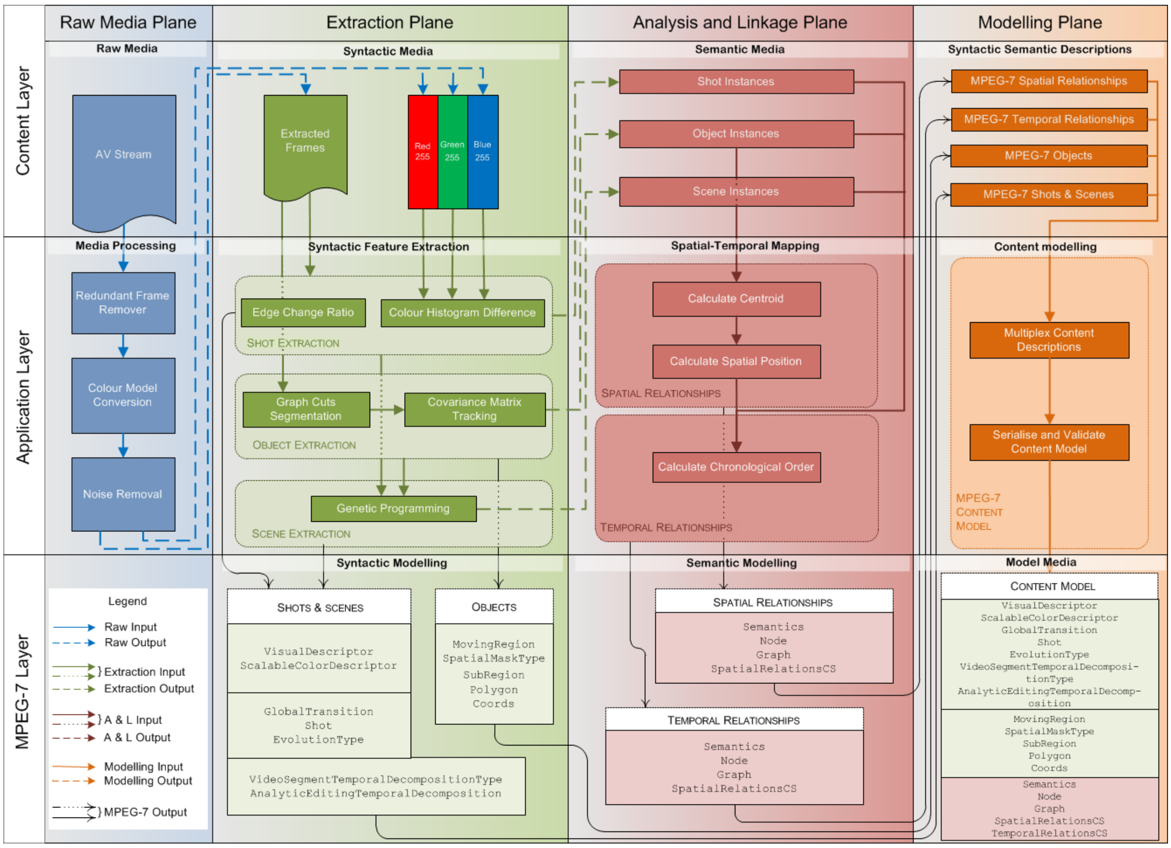


Figure 3.1: OVERVIEW OF MAC-REALM IMPLEMENTATION

The ***Raw Media Plane*** is where the video stream is input into MAC-REALM as a compressed digital footage. The video is then decoded using libVLC[6] (an open source video multimedia framework) and the all the frames are extracted. The *Media Processing* module then processes the extracted frames to remove redundant data. If the colour space of the video is not RGB, it is converted to RGB. MAC-REALM implements the colour space converter, frame redundancy reducer and noise filter using Java Media Framework (JMF)[7] and Java Advanced Imaging (JAI)[8]. Noise is removed from the frames by applying a morphological filter. Once the noise removal is complete the histograms for the corresponding frames are finally extracted, ready for the next plane.

In the ***Extraction plane*** the *Syntactic Media* module stores the frames with their corresponding histograms, for use by the *Syntactic Feature Extraction* module. The *Syntactic Feature Extraction* module extracts shots, objects and scenes, from the information stored. All of the low-level and mid-level syntactic features that are to be modelled are extracted in this plane (Section 3.2.2). The *Syntactic Feature Extraction* module consists of three sub-modules namely *Shot Extraction*, *Object Extraction* and *Scene Extraction.* The *Shot Extraction* sub-module is based on a new algorithm that is implemented by combining two existing algorithms in a novel arrangement, Colour Histogram Difference (CHD) and Edge Change Ratio (ECR), each algorithm is responsible for detecting different shot types (Section 3.2.2.1). The *Object Extraction* sub-module has an algorithm implemented that uses the outputs from both Graph Cuts Segmentation and Covariance Matrix Tracking algorithms (Section 3.2.2.2) to extract objects and then track them for the duration of the shot. The final sub-module of the *Syntactic Feature Extraction* module is *Scene Extraction. Scene Extraction* is implemented using a modified genetic programming algorithm that evolves a rule that can identify scene boundaries by certain syntactic feature markers (Section 3.2.2.3). The *Syntactic Feature Extraction* module passes the extracted shots, objects and scenes to the *Semantic Media* module in the *Analysis and Linkage Plane.* The last module in the *Syntactic Feature Extraction* module is the *Syntactic Modelling* module. Here the extracted shots, objects and scenes are modelled into MPEG-7 syntactic feature description schemes (section 3.2.3).

The shots, objects and scenes are then ready to be processed in the ***Analysis and Linkage Plane*** where they are stored in the semantic media layer (Section 3.3.1). The spatial and temporal relationships are mapped in the *Spatial-Temporal Mapping* module, which has two sub-modules. The

[6] http://www.videolan.org/vlc/libvlc.html

[7] http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html

[8] http://www.oracle.com/technetwork/java/javase/tech/jai-142803.html

Spatial Relationships sub-module defines both absolute and relative spatial relationships, and the inverse of the relative spatial relationships (Section 3.3.2.1). The spatial relationship sub-module calculates the centre of mass of each object to provide a uniform point of reference for measuring the spatial relationships. The centre of mass reference point defines the resulting spatial relationships with an accurate focus that mimics human perception of the bearing of the object/s. The temporal relationships for all the syntactic and semantic content features are mapped in the *Temporal Relationship* sub-module (Section 3.3.2.2). The semantic relationships are then modelled into MPEG-7 semantic content descriptions in the *Semantic Modelling* module (Section 3.3.3).

The MPEG-7 syntactic and semantic features are retrieved from the *Syntactic Modelling* and *Semantic Modelling* modules and placed in the *Syntactic Semantic Descriptions* module, as detailed in section 3.4.1. The syntactic and semantic MPEG-7 descriptions are then interlinked together within the *Content Modelling* module, as detailed in section 3.4.2. The descriptions are set into a MPEG-7 document shell and presented as a complete MPEG-7 content model in the *Model Media* module, as detailed in section 3.4.3.

From the diagram we can see that MAC-REALM goes from Layer to layer and from plane to plane. Using the NetBeans platform as the development platform for implementing MAC-REALM satisfies the design and implementation requirements for MAC-REALM to be modular an extensible. NetBeans offers a platform to build a framework that is extensible and modular through a loose coupled architecture offering high cohesion, but low coupling of components, offers pluggability of different technologies, and platform independence. The following two sections we look at how the MAC layers and the REALM planes are implemented to satisfy their design requirements.

3.2 Raw media plane

The need for an integrated method to pre-processing the raw media to optimise the feature extraction process was discussed in chapter 1. Digitised video comes in many formats, each with their own subtle variations for encoding the video. Some formats are better than others for feature extraction. The requirements of pre-processing, as stated in chapter 2, are that the video is to be optimised for effective and efficient feature extraction whilst reducing computational expense.

The algorithm presented in this section presents a holistic pre-processing method for MAC-REALM. It begins with decoding the video and removing redundant data. To reduce the computational expense we extract keyframes from intervals that reduces data redundancy whilst not impacting on the effectiveness of the feature extraction process. Once the frames are extracted the colour conversion begins, converting the colour space into RGB, which is the colour space most suited for the subsequent feature extraction processes The keyframes are then filtered to remove noise to improve the salient features that are most important to feature extraction.

In Figure 3.2 we have a flow chart that represents the raw media plane process, followed by a detailed description of the processes.

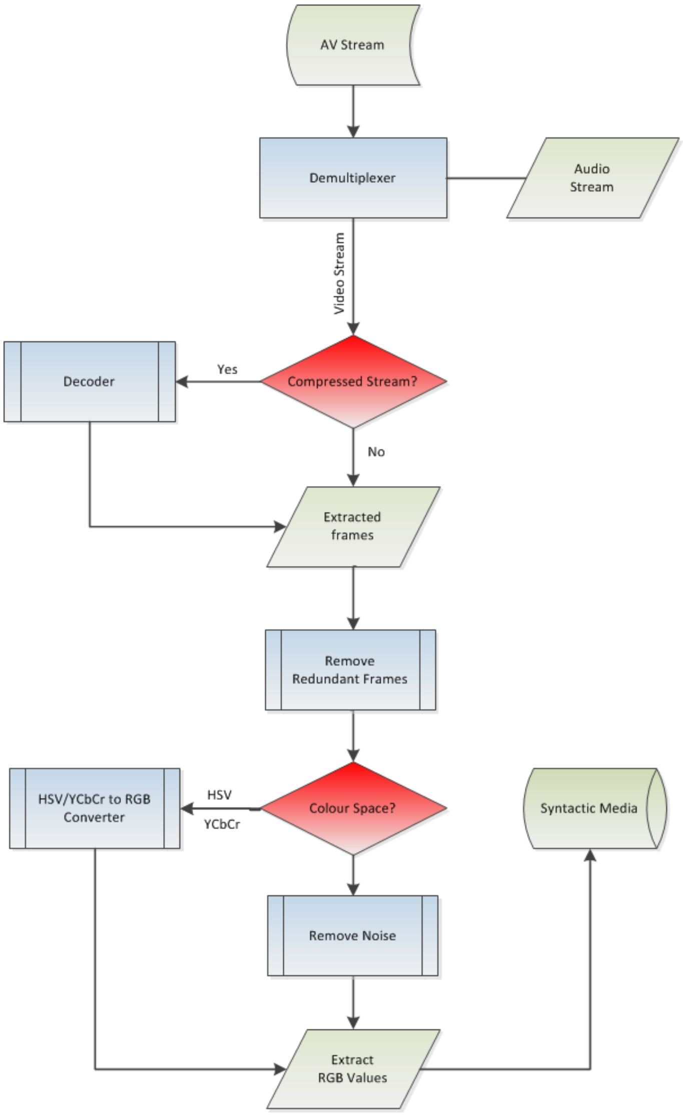


Figure 3.2: RAW MEDIA PLANE PROCESS

The extraction process begins by separating out the video component from the audio. The plug-in libVLC is used to decode the video and demultiplex AV content from multiple formats. Time stamps are also extracted during this operation and are used to sync the features and calculate the temporal relationships in latter operations.

Once video stream frames are input the frame extraction process begins. They are extracted using Java Media Framework (JMF) plug-in along with Java Advanced Imaging (JAI). If the video stream comes from a format that is using an uncompressed video (PAL, NTSC, AVI, DV, etc.) Java Advanced Imaging (JAI) can just grab the frame and buffer it in memory as an image. The images are grabbed at 1 frames per second (fps). Due to the many frame rates possible, and in the future, the fps has to be calculated on a per video basis. This frame rate is calculated dynamically by the equation:

Eq. (3.1)
$$fps = \frac{1}{Frames_{max}/Time_{max}}$$

Where $Frames_{max}$ is the total amount of frames in the video and $Time_{max}$ is the total time in seconds of the video. We use MediaInfo[9] plugin to get the total number of frames and the total time of the video.

The extracted keyframe is then normalised for efficient processing. We use linear normalisation to regulate the resolution of the image. After the frame is normalised the image is filtered to remove noise from the decoding processing. Using a morphological opening/closing gradient filter the noise within the image is reduced and the keyframe is “flattened”. The images are then stored for in the syntactic media component.

We then follow analyse the colour space of the images and determine if they are using RGB or HSV/ YCbCr. If the images are RGB we extract the RGB values into three separate bins representing each band, with a value between 0 and 255. If the colour space is either HSV or YCbCr then we send it to the colour space converter. For conversion of the colour space we use Java Colour class which has standard functions for HSV to RGB and for YCbCr we use JAI that supports that colour profile. Once the RGB colours are obtained they can be stored in the syntactic media layer in the extraction plane.

3.3 Extraction plane

The design requirements for a content model were based on syntactic feature extraction of the video to produce the foundation for the content model. The foundation of any content model is based on syntactic features, notably temporal segment content descriptions. The three syntactic features identified for extraction were shot, objects and scenes. Shots are a low level syntactic

[9] http://mediaarea.net/en/MediaInfo

feature that can be identified through unsupervised machine based methods. Objects and scenes are mid-level syntactic features, meaning that they have conceptual attributes that require semi-supervised machine based algorithms to identify them. Once the features are extracted, they require modelling into MPEG-7 syntactic content description schemes. The syntactic feature extraction process is implemented so the extraction process implicitly interlinks the syntactic content features together. This close coupling of features will be replicated in the content model to build foundation that will then integrate semantic relationships in a tightly integrated content model that will help bridge the semantic gap. In this section the algorithm and techniques used to extract the shots, objects and scenes, and then model them into MPEG-7 syntactic descriptions are described in detail.

In the application layer for this plane, we have three different feature extraction engines; Shot Extraction, Object extraction and Scene extraction. In shot extraction, the normalised images are used to detect transition type shot boundaries, whilst the RGB histograms are used to detect the cut type shot boundaries. The shot boundaries are then used by the object extraction process as the demarcation points to start extracting objects. Object extraction uses the extracted frames from the content layer to extract objects from the first frame of a shot and then track the objects in all subsequent frames of the shot. The output from the shot and object extraction is then used by scene extraction to extract the scene boundaries of the content. The following sections are a discussion of each of the syntactic feature extraction processes, and the modelling of those features into MPEG-7 syntactic description schemes in the MPEG-7 layer.

This plane is split into three layers; 1) Syntactic Media, 2) Syntactic Feature Extraction and 3) Syntactic Modelling. The first two sections use java along with JMF to process and extract the lower level features of the syntactic elements of the content. In the third section we use parse the resultant extracted features, which are still java data structures into corresponding MPEG-7 description schemes that represent the low level features.

3.3.1 Syntactic Media

At this layer, the syntactic media is parsed into java data structures, for both images and RGB values. The reference id for each frame and RGB bin is the timestamps that were extracted in the raw media plane. The RGB values are input into the shot extraction process and the extracted images are used in both the shot extraction process and the object extraction process.

3.3.2 Syntactic Feature extraction

We have three distinct processes within syntactic feature extraction section; 1) Shot extraction, 2) Object Extraction and 3) Scene extraction. The processes are not independent as each preceding process produces features that are then used as input for the processes that follow. This was an implicit occurrence of using features that share the same characteristics in their structural composition. The rest of this section describes each process and it's relation to other processes, and how this all comes together to produce the syntactic model of the content.

3.3.2.1 Shot Extraction

As previously discussed in Chapter 2, a shot extraction algorithm needs to be robust (i.e. gives high recall and precise and thus reducing missed and false positive shots), while keeping computational expense to a minimum. MAC-REALM's shot extraction algorithm combines two separate shot detection algorithms, ECR and CHD. They combine them to produce a new algorithm, which negates the weaknesses of both algorithms. The algorithms are modified to increase system performance by reducing computational expense, whilst not impacting on the overall effectiveness of the algorithm.

Depending on the type of genre, cinematography and shot boundary, some techniques offer several advantages when it comes to identifying one type of shot but will display disadvantages when it comes to identifying other types. To negate these disadvantages the techniques have been fused together to negate their disadvantages whilst exploiting their strengths. The hybrid technique consists of CHD for detecting abrupt cut type shots and ECR for transition type shots. This hybrid approach offers several advantages over using the techniques individually.

Figure 3.3 show the shot extraction algorithm diagram. The shot extraction process consists of three sub-processes; 1) Edge Change Ratio, 2) Colour Histogram Difference and 3) the shot fusion processes. The shot extraction implementation is explained in more detail the following sections.

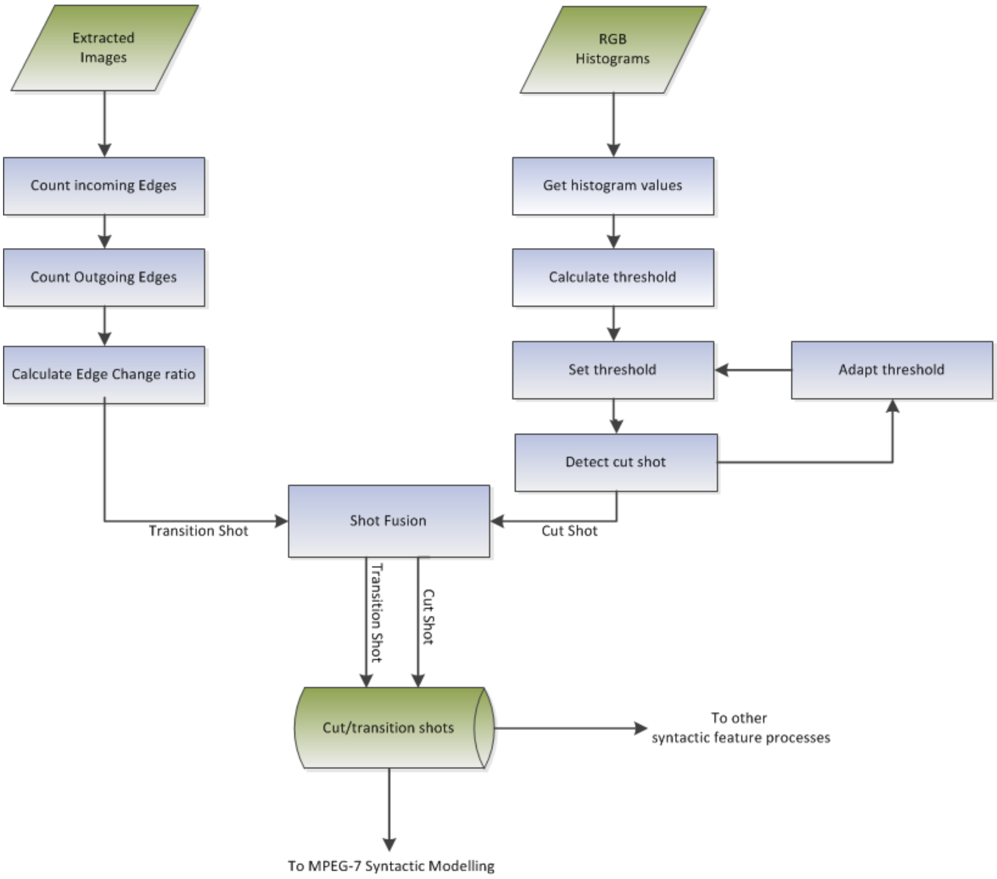


Figure 3.3: SHOT EXTRACTION PROCESS

CHD provides a robust method for detecting cut shots. CHD work by detecting colour discontinuities between frames over a certain threshold that indicate that a shot has been detected. This measure is denoted by CHD_i , where i is a frame of the shot, and is related to the difference or discontinuity between frame i and $i + k$ where $k > 1$. The absolute difference between frames is used to compute the value of CHD_i:

1.1 $$CHD_i = \frac{1}{N} \cdot \sum_{r=0}^{2^n-1} \sum_{g=0}^{2^n-1} \sum_{b=0}^{2^n-1} |p_i(r,g,b) - p_{i+k}(r,g,b)|$$ (Lienhart, 2001)

Where $p_i(r,g,b)$ is the colour histogram of a frame i with 2^{n-1} bins per histogram being considered. The RGB values for each frame are retrieved from the syntactic media component. Once the histograms are retrieved the colour histogram difference between each frame is calculated. For each frame stored in the syntactic media component, the colour value for each colour band is stored (RGB_n). To compute the histogram difference the formula (A. Jacobs, A.

Miene, GT Ioannidis, & O. Herzog, 2004) has been adapted for use. The CHD between each frame is calculated, giving an initial threshold value using the equation:

1.2 $$\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$$ (A Jacobs, A Miene, GT Ioannidis, & O Herzog, 2004)

The threshold is adaptive as it is constantly revaluated as it works through the series of frames. When it detects a shot it resets the threshold and calculates a new threshold based on the first successive frames in the new shot. This method works to make the threshold maxima sensitive to the localised colour differences within the shot. The thresholding technique is more suitable for MAC-REALM as it uses less processing time. This is because the square difference method used in the original work used a calculation over five contiguous frames, as they tried to find transition shots as well.

To stop false positives caused by flashing lights we simply omitted frames where $R_n, G_n, B_n \geq 250$ and the next frame $\mathrm{n} + 1$, where $R_n, G_n, B_n < 250$, would be used instead to determine if there was a cut shot. Figure 3.4 is the pseudo code for the CHD process.

```
1. Get histograms for frame $n = 1$ and $n + k$, $RGB_n$ and $RGB_{n+k}$
2. Calculate initial $\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$
3. For frames $n$ to $\forall n$
      a. if $RGB_n > 250$ then skip $RGB_n$
      b. if $RGB_{n+k} > RGB_n + \Delta_{RGB}$
           i. then mark $n$ as start of shot
          ii. Set $n = n + 1$
         iii. calculate new $\Delta_{RGB} = 2 \times (RGB_{n+k} - RGB_n)$
4. Mark last frame as end of shot
```

Figure 3.4: CHD PSEUDO CODE

The CHD technique is effective for abrupt cut shots were there is a sharp colour difference between two shots due to a sudden change of all colour pixel values. If there is a transition shot in which the colour pixel values between the two shots change gradually and smoothly, CHD will not pick up the change and will miss the shot change.

To counteract this disadvantage ECR is used to detect the transition shots (Lienhart, 2001). ECR is used to identify abrupt shot transitions by comparing consecutive frames. MAC-REALM extends the work by producing edge transition graphs over a 10 frame sliding window. Within the

10 frames we can identify the types of gradual transition from the shape of the transition graphs. The equation for this is given by:

1.3
$$ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}\right)$$
(Lienhart, 2001)

In fade shots the amount of hard edges of objects increases from zero or decreases to zero over time. Fade in's, having increasing visible edges, lead to a positive slopped graph. Fade outs, having decreasing visible edges as the shot gradually fades to black, create a negative slopped graph. Dissolve shots on the other hand produce a concave hyperbolic graph as the pre-dissolve edges dissolve and the post-dissolve edges form. These have been illustrated in Figure 3.5.

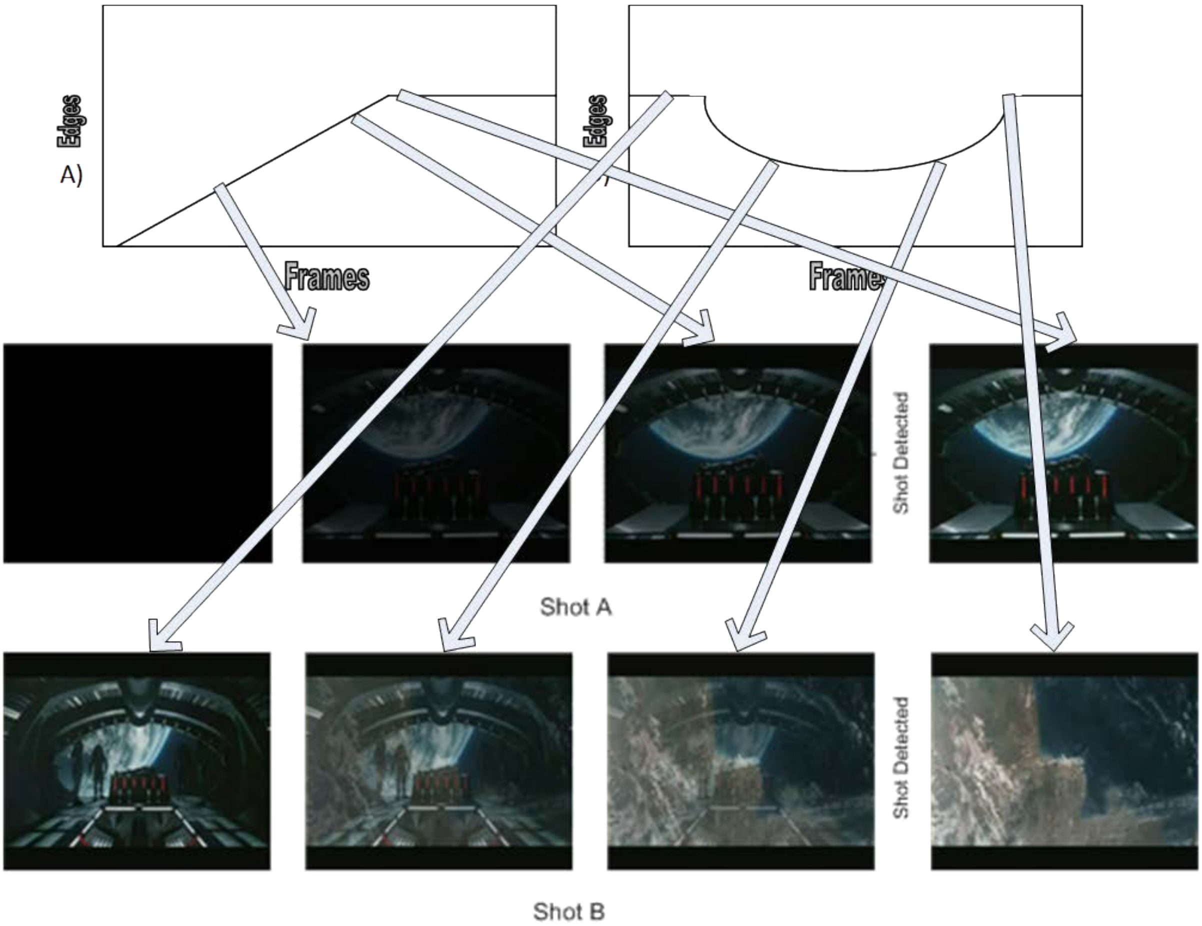


Figure 3.5: ECR SHOT DETECTION: A) FADE IN B) DISSOLVE

The algorithm charts the edge change ratio over the 3 frame sliding window. The sliding window allows the computational complexity of the algorithm to only be greater than using the original method at the beginning of the process. This is due to the edge count for the first ten frames is first calculated and then after that only the 3rd frame is processed for a new edge count every operation. The algorithm for the sliding window is given below in Figure 3.6.

```
1. For Frames $n = 1$ to $\forall n$
2. Perform Canny edge detection(Canny, 1986)
3. Then for every frame $\mathrm{n} + \mathrm{k}$, where $1 \leq k \leq 3$
    a. Count the number of $P_n^{in}$ and $P_{n+k}^{out}$ pixels.
    b. Dilate the edges and invert the images.
        i. Store dilated & inverted image $n$ in $DI_{n-k}^{out}$
        ii. Store dilated & inverted image $n + k$ in $DI_n^{in}$
    c. Perform bitwise AND operation
        i. For every pixel (i,j)
            1. $n$ && $di_{n+k}$
            2. $n + k$ && $di_n$
    d. Count the number of entering and exiting edge
       pixels in the images to obtain $X_n^{in}$ and $X_{n-k}^{out}$

    e. Calculate the $ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}\right)$

4. Compare edge transition plot to FOI and dissolve patterns
   and see if a gradual transition is present

5. Increment n by one and repeat step 3
```

Figure 3.6: ECR PSEUDO CODE

Once shot detection has been performed for both cut and transition shots, using CHD and ECR respectively, both techniques are used in shot fusion to provide ancillary confirmation of the preciseness of the other shot techniques detection accuracy. The ECR technique picked up both types of shot, which we will call $ECR_n^{cut}$and ECR_n^{trans}, which represent cut and transition shots respectively. The ECR_n^{cut} is used as a confirmation on the CHD cut shots, CHD_n. If it is confirmed, then the probability of cut shot is considered high. If not, then the cut shot is considered a medium probability of being correct. If there is an ECR_{cut} but no corresponding CHD_n cut the probability of a shot is considered low.

For transition shots, ECR_{trans} confirmation of a shot change is given by the first and last frame of a transition, ECR_n^{trans} and $ECR_{n+k}^{trans}$and comparing them against the corresponding frames from the CHD process,CHD_n and $CHD_{n+k}$to check to see if a shot change has occurred. If the histograms of CHD_n and CHD_{n+k} are compared sequentially and a cut shot is found we can deduce that ECR_{n}^{trans} and ECR_{n+k}^{trans} are the start and finish of a transition shot. The algorithm is presented in Figure 3.7.

```
1. For $\forall CHD_{n}^{cut}$
     a. If $CHD_{n}^{cut}$ = $ECR_{n}^{cut}$ then $CUT_{prob}$ is high
     b. If $CHD_{n}^{cut}$ != $ECR_{n}^{cut}$ then $CUT_{prob}$ is average
2. If $ECR_{n}^{cut}$ !=  $CHD_{n}^{cut}$ then $CUT_{prob}$ is low
3. For $\forall(ECR_{n}^{trans},ECR_{n+k}^{trans})$
     a. Get $CHD_{n}^{cut}$ and $CHD_{n+k}^{cut}$
     b. If $(CHD_{n}^{cut},CHD_{n+k}^{cut})$ is cut then
          i. Mark $ECR_{n}^{trans}$ as start of transition shot
         ii. Mark $ECR_{n+k}^{trans}$ as end of transition shot
```

Figure 3.7: SHOT AMALGAMATION PROCESS PSEUDO CODE

Using both the modified CHD and ECR implementations significantly improves precision and recall of both abrupt and gradual transition shots. The modification of the ECR algorithm makes it effective at identifying gradual transition shots and does not need more computational expense then the original that compared only two frames. The CHD is reduced in computational efficiency by reducing amount of frames processed for the threshold value. The reductions in computational expense lower the processing time, which provides a more feasible overall time span for processing in MAC-REALM. This is done whilst keeping the shot extraction at a performance level that is close to other similar approaches as shown in chapter 4.

3.3.2.2 Object Extraction

Object segmentation for video is a two-task process of segmentation and tracking, as described in chapter 2. The first task is to segment the foreground objects from the background. The segmentation must also be able to differentiate and group multiple objects correctly, even when they are overlapping. The second task is to track the object(s) over time as they move. The tracking must be consistent and maintain the integrity of the object boundary from the initial segmentation.

MAC-REALM approaches the two-step problem with a unified two-phase algorithm. In the first phase it uses graph cut theory to segment the initial frame, which can segment multiple objects. The second phase tracks the objects segmented from the first phase, maintaining the integrity of the object silhouette, even if tracking multiple objects.

Object extraction, or image segmentation as it is more commonly known, refers to the problem of dividing an image into a number of disjoint regions such that the features of each region are consistent with each other. Since images generally contain many objects that are further surrounded by clutter, it is often not possible to define a unique segmentation. In other words, the segmentation problem can be ill posed when working in an unsupervised framework. Interactive algorithms allow the user to label a few pixels as either object or background, thereby making the segmentation problem well posed.

Another problem is once the image is segmented, how is it tracked through the shot? As the shot moves on from the original frame the objects shape and position will change. The objects shape will change for non-rigid bodies as they move, even rigid bodies can change their shape through the effect of perspective. The position of objects changes over time. The position can change slowly such as an interview or rapidly such as action sequences. The fast change sequences poise a problem, as it is hard to find continuity with consecutive frames as the position of the object could have drastically altered.

For these reasons segmenting and then tracking the object in MAC-REALM is treated as a two phase problem. The initial phase is the segmenting of frame into background and foreground objects. This is followed by the tracking phase, where the region(s) of interest (ROI) are then analysed and their features are used as the initial reference point for tracking the object.

Object extraction in MAC-REALM uses graph theory to segment an image. A graph based segmentation approach from (Noma et al., 2012) is used. Graph based image-segmentation is a fast and efficient method of generating a set of segments from an image. They supersede old edge-based approaches as they not only consider local pixel-based features, but also look at global similarities within the image.

Object extraction is performed using a semi-automated procedure that segments based on structural pattern recognition to extract objects from their background. The object extraction process begins by creating two attributed relational graphs (ARG's). ARG's are very useful at not only model initialisation but also providing information on image structure. The first graph is an over segmented image using a watershed algorithm. The second image is a user defined input image that has different coloured stroke marks for different objects and the background. The first graph known as the input graph is processed against the second user defined graph, the model graph. The model graph is used to prime segmentation by providing and approximation of the objects core. From the initial stroke marks the regions are expanded, by merging the interconnected regions based on colour similarities and structural consistency. The background strokes are used to grow the background regions in the same manner. Once all regions have been assigned to either objects or background the segmentation stops. This method was chosen as it is a very fast, and deals with the problem of image clutter by using user feedback to determine the initial ROI.

Once we have identified the region we have to track it across several frames so we can track the objects movements and spatial orientation for the duration that they appear. The algorithm described for segmentation was conceived for the use with still images. Using the same algorithm to segment the rest of the frames in the shot would lead to two problems. The first is the continuity of the object outline. The silhouette would become unstable, as the algorithm would segment each frame of the shot individually. This would cause the outline to fluctuate as the segmentation information from the prior frame is ignored. The second problem would be that the stroke marks used in the initial frame could become inaccurate as the shot progresses through the frames. What is required is a second algorithm that takes the ROI and tracks the pixels, using the information from the previous frame as the starting point for tracking.

In order to solve the problem of tracking in the second phase we use the region covariance technique implemented by (Tuzel, Porikli, & Meer, 2006). The tracking is initialised by extracting feature vectors from the ROI's of the keyframe. From the vectors a covariance matrix is built of the feature vectors for the ROI's of each frame. The covariance matrix is measure of how much two variables vary against each other. This is used to track the adjacent pixels next to each other in the ROI. The covariance becomes more positive for each pair of values that differ from their mean in the same direction, and becomes more negative with each pair of values that differ from their mean in opposite directions. The covariance descriptor method can use any set of features (intensity, colour, gradients, filter response). For MAC-REALM, colour and intensity has been chosen for the covariance descriptors. These were selected as they are convenient features to extract as the colour histogram extraction algorithm used to provide the colour histograms can be used, and with a small modification can also be used to extract intensity image as an alpha value.

The tracking algorithm is suitable for MAC-REALM as it has many advantages over other techniques:

a) It is robust against lighting changes and moving camera motion

b) It can track non-rigid bodies as they change

c) It can track fast moving object even if there is a large gap in position since the last consecutive frame

d) The algorithm is very fast at computing covariance as it uses integral images which are intermediate image representations used for fast calculation of region sums.

Using a two phase approach to segmenting and tracking the objects makes the overall result more precise, robust and fast then just using a single technique. The image segmentation phase segments the initial keyframe into ROI's in a fast and precise manner. The user defined strokes eliminate the confusion of image clutter and provide a template for the region growing algorithm. Once the keyframe is segmented into ROI's the tracking algorithm then tracks them through covariance matrices of extracted feature vectors of the ROI's. The result is that objects can be reliably segmented and then tracked with minimal input from the user. It can handle multiple objects and objects that are similar in size and colour.

3.3.2.3 Scene Extraction

Scene segmentation clusters shots together into semantically themed scenes. Syntactic queues that identify the scene boundaries are hard to identify as different cinematography is applied to different genres and even between different film makers. Within genres and specific footage, rules can be produced for identifying scene boundaries with a high level of precision and recall (see section 1.3.2.3 Semantic Temporal Segmentation). However, these rules are limited to their own genre and cannot be applied universally. What is required is an algorithm that can formulate rules for any video clip that is supplied.

MAC-REALM uses a Genetic Programming (GP) approach based on work from (Marios C. Angelides & Lo, 2005) that evolves rules from a set of pre-defined features that are good indicators of scene boundaries in general film production. MAC-REALM has improved upon this by selecting different features which are better indicators and that also reduce the computational expense of processing the footage to formulate the rules. As shown in chapter 4 this approach gives higher precision in identifying scene boundaries, whilst reducing processing time overall for MAC-REALM.

The scene boundary detection is a semi-automatic process that detects boundaries by using a trained GP algorithm that identifies low level feature combinations that identifies scene boundaries. Due to scene boundaries having a semantic definition, the boundary must be perceived semantically. This means a user must train the GP with a small video clip of preidentified scene boundaries. The GP then formulates rules that identify certain feature sets, i.e. histogram difference, object number, shot transition type and shot duration that identify the scene boundaries. It uses a fitness function based on how well the rule correctly identifies scene boundaries from the training clip. Input for scene extraction is sourced from both shot and object extraction processed, as well as the content layer. The shot duration and transition type is sourced from shot extraction, whilst the number of objects present in a shot is sourced from the object extraction engine. The histogram values are sourced from the content layer via the shot extraction engine.

In the original work the features that were used to create the rules were multimodal i.e. 2 video features and 2 audio features. MAC-REALM has instead used four video features, and foregone the audio features. This has been done for two reasons. The first is that although audio features are a good indicator of scene change, they are only as good as the analysis of the features extracted. Voice recognition has to be used, as well as other audio recognition algorithms, as a scene change is usually indicated by a change in actors or environments. In the original work they used speech and audio breaks to formulate the rules. Although these are adequate, they in themselves do not provide accuracy to the start of a scene. In MAC-REALM they have replaced them with video features that correlate more strongly with a scene change and therefore give a higher degree of accuracy of rules evolved that will identify a scene boundary. The second is that it reduces computational complexity and therefore allows processing time to be kept to a minimum. All the features used by MAC-REALM to create the rules have already been extracted during the previous processes.

The two new video features that replace the audio features are the number of objects in the shot and the global histogram difference of the shot. These two features are a very good indicator of a scene change. The number of objects in a shot are a good indicator to the start of a scene as they usually have a low or fixed number for the establishing shot of the scene. The histogram difference can provide a good measurement for a scene change as the colour distribution for shots belonging to the same scene are more similar to each other than the colour distribution from a shot from another scene. These two new features are a better indicator of scene change than audio breaks. So the feature set to be used as the main parameters of the GP algorithm includes:

- *Shot Duration* – the length of a shot in seconds till the start of the next shot,

- *Histogram difference* – the change in the mean histogram values of the RGB values of the first frame of a shot to a specified preceding/subsequent shot,

- *Transition Effect* – what transition effect there is between the shot, gradual or cut transitions, and

- *Number of Objects* – how many identified objects are there in the shot.

The low level features described have already been automatically extracted during earlier stages of feature extraction. The shot duration, histogram difference and transition effect are sourced from the shot extraction stage, whilst the object extraction stage provides the number of objects.

The goal of the GP algorithm is to discover rules that determine scene boundaries (shot detection and feature extraction are not included). The GP algorithm takes as input a series of shots S1, S2, … SN, and their corresponding features. The choice of features directly affects the result. If not enough features are selected, an optimal rule may never evolve (rules evolve by reproduction, crossover and mutation). On the other hand, if there are too many features, the search space will become inoperably large and seriously affect the processing time of the system. We attribute the following five features with each shot: transitional effect, number of objects, shot duration and histogram difference.

There are two types of transitional effects: abrupt change (cut) and gradual change (dissolve, fade and wipe). We look at the number of objects in the starting frame of a shot and count how many, if any, objects there are. Shot duration is measured using the W3C time code from the ISO 8601 Standard (Wolf & Wicksteed, 1998). With respect to the histogram difference, two key frames are extracted from each shot. The first one is extracted from the beginning of the shot and the second one from the end. The first key frame's colour histogram difference with the second key frame from the previous shot is computed using the same formula as the one used in shot boundary detection.

The tree root must be either AND or OR
The left child of a TE, HD, SD or NO must be a sp
The middle child of a TE or NO must be an ops1
The middle child of a HD or SD must be an ops2
The right child of a TE or NO must be a bv
The right child of a HD or SD must be a pi

Figure 3.8: **Formal symbol syntax**

The function set of the algorithm can be defined as $\mathrm{F} = \{\mathrm{SD}, \mathrm{HD}, \mathrm{TE}, \mathrm{NO}, \mathrm{AND}, \mathrm{OR}\}$, Where SD, HD, TE, NO are the four features; Shot duration, Histogram difference, Transition effect, Number of Objects and AND, OR are Boolean operators. The terminal sets comprise of $\mathrm{T} = \{\mathrm{sp}, \mathrm{bv}, \mathrm{pi}, \mathrm{op1}, \mathrm{op2}\}$, where $\mathrm{sp} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$is the position of the shot to be compared against the current shot (C), $\mathrm{bv} = \{\mathrm{true}, \mathrm{false}\}$, pi is a positive integer in the range of 1 – 126789, $\mathrm{op1} = \{=, \neq\}$ for Boolean operations and $\mathrm{op2} = \{<, \geq\}$ for arithmetic operations. The formal symbol syntax is shown in Figure 3.8.

The Scene boundary rules use the grammar provided by reverse polish notation (Visser, 2011). This is convenient because as a last-in-first-out (LIFO) stack is used implementing the *stackbuffer* method in java. It also makes calculations much more efficient by reducing the complexity of the calculations as all brackets and parentheses are eliminated.

An example of a scene boundary rule is provided in Figure 3.9. For simplification reasons only two features are present in this particular example, where transition effect and shot duration are assigned the identifiers a and b respectively.

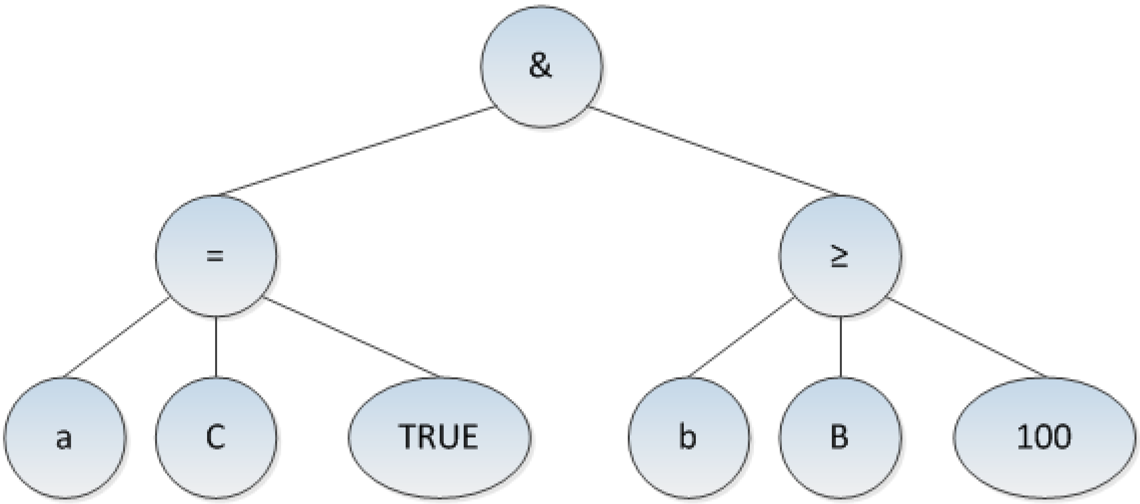


Figure 3.9: EXAMPLE OF A SCENE BOUNDARY RULE

Computation starts with evaluating the rule against each of the shots. We ignore shot 1 because it does not have a preceding shot, hence making the = operator incomputable. Starting from shot 2, the = operator on the right returns TRUE because shot 2 has a gradual transitional effect. The operator on the left also returns TRUE because shot 1 has a duration of 120 frames. The final result of the rule is TRUE because TRUE AND TRUE = TRUE. Similarly, the result for shot 3, 4, 5, 6, 7 and 8 are FALSE, FALSE, TRUE, TRUE, FALSE and FALSE respectively (see Table 3.1).

SHOT	RESULT	SHOT	RESULT
2	TRUE (CORRECT)	6	TRUE (CORRECT)
3	FALSE (CORRECT)	7	FALSE (CORRECT)
4	FALSE (WRONG)	8	FALSE (WRONG)
5	TRUE (WRONG)		

Table 3.1 : THE RESULT OF THE RULE ON EACH SHOT

The result is correct for shot 2, 3 6 and 7. Using the fitness function, to calculate:

$$f = \frac{4}{7} = 0.57 \tag{1.4}$$

After finalising the syntax for the rules, the initial population has to be created to evolve the rules from. The initial population of rules are grown using three different GP strategies. These increase the diversity of the rules and helps evolve more varied and healthier children that are more resistant to convergence of the population. There are three popular generative methods in classic GP: full, grow and ramped half-and-half (Torres et al., 2009). The full generative method creates a population with full trees (the left tree in Figure 3.10). The grow method; on the other hand, generate the initial population with trees that are variably shaped (the right tree in Figure 3.10).

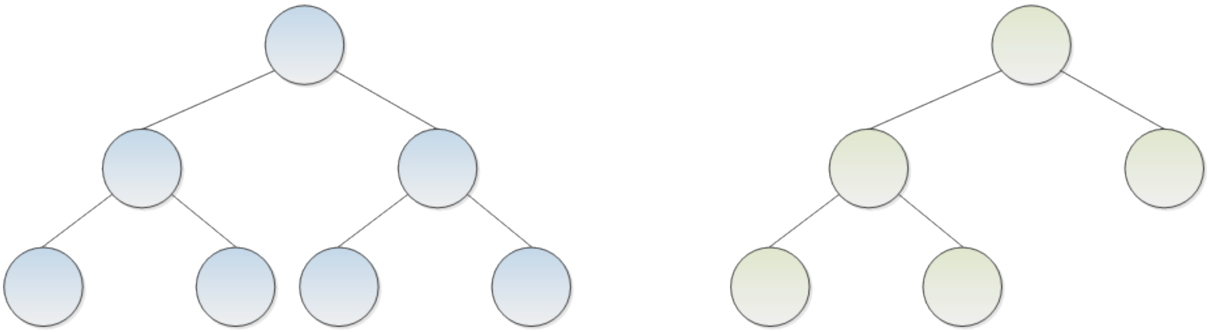


Figure 3.10: TREES GENERATED BY FULL AND GROW METHOD (MARIOS C ANGELIDES & KEVIN LO, 2005)

The ramped half-and-half generative method is a combination of the full method and the grow method. The ramped half and half method has a depth limit of five to achieve a reasonable level of diversity. Half of the trees are generated by the full method and half of the tress are created by the grow method.

After the rules have been generated, they need to be assessed to see which are better at identifying scene boundaries then others. A fitness function is used that identifies the rules that are more proficient at finding scene boundaries. The fitness function is given by the equation:

1.5 $$f = \frac{Nc}{Nt}$$ (Marios C Angelides & Kevin Lo, 2005)

Nc is the number of correctly identified scene boundaries, and Nt is the total number of shots. The fitness function gives a score between 0 and 1, with 1 representing the optimal solution. The fitness function evaluates the quality of a rule, i.e. the rule's performance in determining scene boundaries. What follows is an example that shows how a fitness value is calculated. The example works on the testing data in Table 3.2.

	SCENE BOUNDARY	TRANSITION EFFECT	SHOT DURATION
SHOT 1	NO	FALSE	120
SHOT 2	**YES**	TRUE	80
SHOT 3	NO	FALSE	220
SHOT 4	**YES**	FALSE	140
SHOT 5	NO	TRUE	200
SHOT 6	**YES**	TRUE	800
SHOT 7	NO	FALSE	10
SHOT 8	**YES**	TRUE	200

Table 3.2 : AN EXAMPLE WITH EIGHT SHOTS AND THEIR CORRESPONDING SET OF FEATURES (MARIOS C ANGELIDES & KEVIN LO, 2005)

Once the fitness of all the rules is assessed a new generation of rules is created that are better adapted to identifying scene boundaries. These rules must carry over the best traits from the existing rules for producing better ones in the next evolution. MAC-REALM uses a method of cloning, mutation, crossover and introducing new rules to facilitate this. To begin with the top 10% are copied over to the next generation, whilst the bottom 70% are discarded. The top 30% are mutated to provide new rules. The top 30% are then used in a crossover operation to provide another set of new rules. The last 30% of rules are generated using the same methods as the initial population. This technique of creating new generations allow the properties of the best rules to be favoured in the next cycle of evolution whilst making sure that the population stays diverse enough to stop convergence. Ensuring that suitable divergence is assured is paramount, or the algorithm could converge to early on a less than optimal solution.

The algorithm is iterative and will stop either when an optimal rule is obtained (i.e. the fitness value fo the rule matches the target fitness value) or the maximum pre-determined number of generations is reached. The optimal rules fitness value limit has been set at a minimum of 95%.

The maximum number of generations that will be generated is set at 300. Table 3.3 lists the key steps of the GP scene boundary detection algorithm:

Step	Instruction
1	Generation = 0
2	Create initial population with size p
3	Apply *fitness function* to evaluate the fitness value of each rule
4	Sort the rules according to their fitness value in descending order
5	If termination criterion met (best fitness value > 0.95 or Generation > max generation k), output the best rule and exit. Else go to step 6
6	The worst fit rules (the worst 70%) are discarded
7	Generation = Generation + 1
8	Perform *reproduction* operation (Top 10%)
9	Perform *crossover* operation (Top 30%)
10	Perform *mutation* operation (Top 30%)
11	Create new rules (30%)
12	Go to step 3 unless a) generations = 300 b) A rule has 95% fitness score
13	End

Table 3.3 : Major steps of the GP scene boundary detection algorithm (Marios C Angelides & Kevin Lo, 2005)

The GP scene boundary detection algorithm is a suitable in MAC-REALM as it is good at detecting the scene boundaries of generic video clips. Most scene boundary detection algorithms are limited to certain domains as they apply rules that are specific to a genre (see section 1.3.2.3). The GP algorithm is suitable for generic footage as it builds the rule explicitly for any footage that has a clip of video data where the scene boundaries are identified. The algorithm formulates the rule as feature vectors based around video features that are good indicators of scene boundaries regardless of the domain or content. As the rules are judged on a fitness function that uses the training data as ground truth, a rule can be generated that takes into account the abstract semantic nature of the scene boundary. This makes the scene boundaries identified very close to the semantic perspective of users.

3.3.3 Syntactic Modelling

Once the shots, objects and scenes have been extracted they need to be modelled into MPEG-7 syntactic content descriptions. There are a myriad of ways to describe content in MPEG-7, and these can be used to model the same features but in a different manner to facilitate different functionality or use. MPEG-7 also allows customised descriptions that can be created for a particular purpose within the target application.

Within MAC-REALM, we use standard pre-defined MPEG-7 syntactic content descriptions to describe the features. This makes the content descriptions accessible to all MPEG-7 applications, as there is no ambiguity that can be associated with customised schemes written for a specific profile or application. The selection of the descriptions schemes has been based on two criteria a) the ability to describe the feature comprehensively and concisely and b) the ability to interlink the DS's together into a multi-faceted description structure.

Modelling the syntactic extracted features into MPEG-7 is done in two parts. We model the scenes and shots together as one feature set as they both share exactly the same attributes as lower level features. They are only distinct on a semantic level.

Objects have temporal characteristics but also have spatial attributes and are modelled separately in MPEG-7. Their temporal attributes are used as a referencing mechanism associated with the scenes and shots they exist in, and these are used as their reference id's.

3.3.3.1 Scene and shot descriptions

For describing the scenes and shots, the VideoSegment DS is used. Scenes and shots are similar as they have the same physical attributes i.e. Start time and duration, so temporally they are integrated in the modelling process. Scenes are described using the VideoSegment DS and are given an ID to uniquely identify them. The physical location of the media is defined by the MediaLocator DS and can locate media from either a local or remote source using the MediaUri D. The physical media is then given a unique id using the Video DS tag.

Scenes are created using the VideoSegmentTemporalDecompositionType DS to segment the scenes temporally. A scene is embedded into the root of the Video DS using the VideoSegmentTemporalDecompositionType DS. Using the MediaTime DS within the VideoSegment DS, the start of the scenes is stated by its timestamp and its duration using the MediaTimepoint D and MediaDuration D respectively. A unique id is given to the scene in the VideoSegment DS. After the scene has been described the shots that comprise the scene are contained within the AnalyticEditingTemporalDecomposition DS. The DS describes a temporal decomposition of the segment into one or more sub-segments that correspond to shots or global transitions. The shots are listed in the scene description using the Shot DS and GlobalTransition DS. The GlobalTransition DS and Shot DS come in pairs with the GlobalTransition DS appearing before the Shot DS it is describing. The GlobalTransition DS describes the edit of the shot boundary, i.e whether it is a cut or a transition. The GlobalTransition DS has an "evolutionReliability" attribute that shows the confidence in the transition state. The

EvolutionType CS is a classification scheme that identifies what type of transition is described in the Shot DS. The Shot DS contains an “id” attribute that contains the unique identifier of the shot.

```
<VideoSegmentTemporalDecomposition>
        <VideoSegment id = "AVP-SCENE-1">
                <MediaTime>
                        <MediaTimepoint>36.36</MediaTimepoint>
                        <MediaDuration>80.72</MediaDuration>
                </MediaTime>
                <AnalyticEditingTemporalDecomposition>
                        <GlobalTransition evolutionReliability="false">
                                <MediaTime>
                                        <MediaTimepoint>36.36</MediaTimepoint>
                                </MediaTime>
                                <EvolutionType ref="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
                        </GlobalTransition>
                        <Shot id = "AVP-SCENE-1-SHOT-29">
                                <MediaTime>
                                        <MediaTimepoint>36.36</MediaTimepoint>
                                        <MediaDuration>6.3600006</MediaDuration>
                                </MediaTime>
                                <VisualDescriptorxsi:type="GoFGoPColorType"
aggregation="Intersection">
                                        <ScalableColornumOfCoeff="16"
numOfBitplanesDiscarded="0">
                                                <Coeff>264712</Coeff>
                                        </ScalableColor>
                                </VisualDescriptor>
                        </Shot>
<GlobalTransition evolutionReliability="false">
<MediaTime>
        <MediaTimepoint>42.72</MediaTimepoint>
                               </MediaTime>
        <EvolutionType ref="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
        </GlobalTransition>
        <Shot id = "AVP-SCENE-1-SHOT-30">
        <MediaTime>
<MediaTimepoint>42.72</MediaTimepoint>
        <MediaDuration>3.5999985</MediaDuration>
        </MediaTime>
        <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Intersection">
        <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
        <Coeff>463375</Coeff>
        </ScalableColor>
        </VisualDescriptor>
        </Shot>
```

Figure 3.11: EXAMPLE OF DESCRIPTION OF SCENES & SHOTS USING MPEG-7

For video segment identification using the histogram values extracted in the shot extraction process the VisualDescriptor DS is used. The type of the VisualDescriptor DS is set to “GoFGoPColorType“, which aggregates the colour distribution across a number of frames in a shot. Then the ScalableColorDescriptor DS is used to model the colour distribution. This can then be used to locate shot segments based on cinematography (e.g. a search for a warm toned scene or shots) or query-by-example (e.g. basing a search on histogram values from an image). An example of a snippet of the MPEG-7 descriptions for scenes and shots is given in Figure 3.11

3.3.3.2 Object representation

Objects are described by modelling them using the MovingRegion DS tool. A unique id is given to the object using the attribute tag in the MovingRegion DS. The id tag for the object references' what scene (e.g. AVP-SCENE-0), shot (e.g. SHOT-25), and frame number (e.g. 711), the object appears in, as well as the number of the object in relation to other objects in the frame (e.g. OBJECT-1). The frame number is added due to the fact that an object may not appear in a shot in the first frame.

```
<MovingRegion id = "AVP-SCENE-0-SHOT-25-OBJECT-1-711">
    <Mask xsi:type = "SpatialMaskType">
       <SubRegion>
          <Polygon>
                     <Coords mpeg7:dim="2:5">191 290 153 154 155 156 157 158 159 160 161
162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 199 221
222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243
267 268 269 270 276 277 278 279 286 287 288 289 290 292 293 312 313 314 315 316 320 321
322 323 324 325 ....</Coords>                           <Coords mpeg7:dim="2:5">35 35 36
36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36
36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36
36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36 36
36 36 36 36 36 36 36 36 36 36 36 37 37 37 37 37 37 38 38 38 38 38 38 38 38 38
          <Polygon>
       </SubRegion>
    </Mask>
</MovingRegion>
```

Figure 3.12: EXAMPLE OF OBJECT DESCRIPTION USING MPEG-7

The object boundary that was extracted during the object extraction process is referenced by the Mask DS. The Mask DS is typecast to "SpatialMaskType". The SpatialMask D describes a mask in 2-D space. The SpatialMask D is used by the MovingRegion DS to describe the boundary of a region within the video frame using a polygon. The spatial mask type is comprised of an unbounded set of subregions using the SubRegion D, where each sub-region is described using the Polygon D. The Polygon D demarcates the silhoulette of the object as Cartesian coordinates using the Coords D. The first row of Coords D references the x coordinates and the second the y coordinates. An example of MPEG-7 object descriptions is given in Figure 3.12.

3.4 Analysis and Linkage plane

The analysis and linkage plane consists of three layers, the first is the semantic media layer, which is the content layer for this plane and contains the shots, objects and scenes that have been extracted from the extraction plane. The second layer is the spatiotemporal mapping layer, which is the application layer of the plane. Here the semantic media is analysed and the feature vectors processed and their semantic relationships created and linked. The spatial relationships are calculated for the objects, relative to both other objects and their global position in the frame. The temporal relationships are then processed for all the syntactic and semantic feature vectors. Once all temporal and spatial relationships are created they are converted into MPEG-7 Semantic descriptions using the Semantic Graph DS to provide the linking mechanism between the relationships between all the features.

3.4.1 Semantic Media

In this layer, the semantic media is parsed into java data structures that represent the shots, objects and scenes that were extracted from the previous phase. The scenes and shots are represented by timestamps, whereas the objects are represented as Cartesian coordinates in 2D space with timestamps as their reference id.

3.4.2 Spatial and Temporal Mapping

In chapter 1 the importance of semantic relationships to content modelling was identified as a major feature that should be explicitly defined in all content models. The relationships between features are the "glue" of a content model and helps contextualise the interactions between features that is all important in semantic or multi-content type querying. Spatial and temporal relationships fulfil two of these criteria that is related to the "Where and When". Spatial relationships deal with the question of "where", by stating the position of objects to their position, globally and with relation to each other. The "when" deals with the temporal relationships of all the features, both syntactic and semantic, and their chronological ordering in relation to each other.

There are two distinct processes within the spatiotemporal mapping section; 1) Spatial relationships and 2) Temporal relationships. The first semantic relationship to be defined is the spatial relationship mapping of objects. This is followed by the temporal relationship mapping of the shots, scenes, objects and spatial relationships.

The rest of this section describes the processes for mapping of spatial relationships, both absolute and relative. This section is then followed by how the temporal relationships are formulated between scenes, shots and objects.

3.4.2.1 Spatial relationships

Spatial relationships are a problem within content modelling as they are not explicitly modelled. Indeed, from the literature review in chapter 1 it can be seen that it is left to the target application to calculate the relationships in any manner they see fit. Having spatial relationships to be arbitrarily defined can lead to the problem of ambiguity, as the method to calculate the position of objects varies between systems. This can lead to different applications giving different spatial relationships for the same content, which can lead to inaccurate query results.

The implementation requirements stated that the spatial relationships needed to be explicitly modelled into content descriptions. The explicitly stated spatial relationships need to fulfil two criteria:

a) They need to calculate the reference point for the centre of the object in a consistent and natural manner that makes the resulting measurements logical and intuitive in relation to queries

b) Relationships must be given for both the global positions of the individual objects and also for the relative positions of the objects to each other, stating their inverse relationships as well. MAC-REALM uses a centroid algorithm that can work out the centre of an irregular shaped lamina, and then calculates the absolute and relative positions of the object using standard techniques.

The modelling of the spatial relationships begins with finding the centroid of the object(s). The centroid of the object is used as the reference point to measure the position of the objects. To find the centroid of an object the boundary of the object must be defined. The object silhouettes are retrieved from the semantic media module and are used as the object boundaries. They are analysed frame-by-frame, with each objects edge boundary used as the Cartesian coordinates as the input for the centroid algorithm.

The spatial relationships are defined within two classification types; 1) absolute and 2) relative. Absolute spatial relationships are stated using the points of the compass, as stated Table 1.3, and are precise about location in terms of description. This is used for objects independently and gives a spatial orientation that is dependent on its global orientation within the frame. Relative spatial relationships are stated in terms of the objects position in relation with another object. Examples of this are object 1 is ***above*** object 2 and object 1 is on the ***right*** of the screen.

Absolute spatial relationships

To calculate absolute spatial relationships we split the screen according to the visual composition rule known as the "rule of thirds" (L. Liu, Chen, Wolf, & Cohen-Or, 2010). The screen is split into 9 different sections and forms a 3x3 matrix. Each position in the matrix has an absolute spatial relationship attached to it depending on its absolute spatial orientation within the matrix P, such that:

Eq. (3.11) $$P_{(i,j)} = \begin{Bmatrix} NW & N & NE \\ W & C & E \\ SW & S & SE \end{Bmatrix}$$

The absolute position of an object is given by the placement of the centroid within the matrices boundary. If the object is in the middle of the matrix it's position is given as "centre".

Relative spatial relationships

To calculate the relative spatial relationships of objects two methods for different cases are employed; 1) generalised position and 2) relative to another object. In the first case two matrices are used, one for vertical positions and the other for horizontal positions. This is done to capture the spatial relationship of an object in both planes. An object is only counted as having a single position (in one plane) when it is in a central position, or "neutral position in the other. Where both sets V and H are 3x3 matrices and have members:

Eq. (3.12)

$$V_{(i,j)} = \begin{Bmatrix} T & T & T \\ 0 & 0 & 0 \\ B & B & B \end{Bmatrix}$$

$$H_{(i,j)} = \begin{Bmatrix} L & 0 & R \\ L & 0 & R \\ L & 0 & R \end{Bmatrix}$$

Where $T = top, B = bottom, L = left$ and $R = right$. The combined relative position in both planes can be calculated by adding the two matrices together as can be seen in equation 3.13. There is no "centred" position within the relative spatial positions as there is no relative centre as the both objects positions are arbitrary.

Eq. (3.13) $$(V + H)_{(i,j)} = \begin{Bmatrix} TL & T & TR \\ L & 0 & R \\ BL & B & BR \end{Bmatrix}$$

For the relative positioning between two objects, from object A to object B, the cardinal point positions of one object from another are taken in degrees. North is taken to be $\theta = (0°, 360°)$. For any cardinal point, any $22.5°$ angled section from that point, both clockwise and anti-clockwise, can be considered as having the same bearing as that cardinal point. Each cardinal point is given the same $45°$ segment around its point. The half cardinal points are given to the boundaries of each main segment. This arrangement gives the best mapping cognitively to what users perceive when thy think of direction. Therefore the cardinal point sections are represented by:

Eq. (3.14)

$$B = \begin{pmatrix} 315° < 45° & = & N \\ 45° & = & NE \\ 45° \leq 135° & = & E \\ 135° & = & SE \\ 135° \leq 225° & = & S \\ 225° & = & SW \\ 225° \leq 315° & = & W \\ 315° & = & SW \end{pmatrix}$$

The inverse bearing is given by:

Eq. (3.15)

$$B^{inverse} = \begin{pmatrix} if \;\; B \leq 180° \;\; \therefore B + 180° \\ if \;\; B > 180° \;\; \therefore B - 180° \end{pmatrix}$$

3.4.2.2 Temporal relationships

In chapter 1 it was discussed that temporal relationships are one of the most important content features that can be queried, as most searches in video will have an element of when an event happens or object appears. As with spatial relationships it has been shown that temporal relationships are not explicitly stated but are modelled as a post-process to query input. Without an explicit structure interlinking the events and objects the risk of "content discovery", the ability to discover new features and concepts that might be of interest but were not inferred in the original query, are limited.

Temporal relationships are important as they provide an all-important linking mechanism between syntactic and semantic features. This allows a much more integrated approach to content discovery that makes connections between the physical structure of the video and the meaning of the content.

MAC-REALM explicit media structure enables temporal relationships between syntactic features (scenes, shots and objects) to be determined through a partial temporal ordering of these entities. A partial ordering < can be defined on a set of features as follows:

Eq. (3.16)

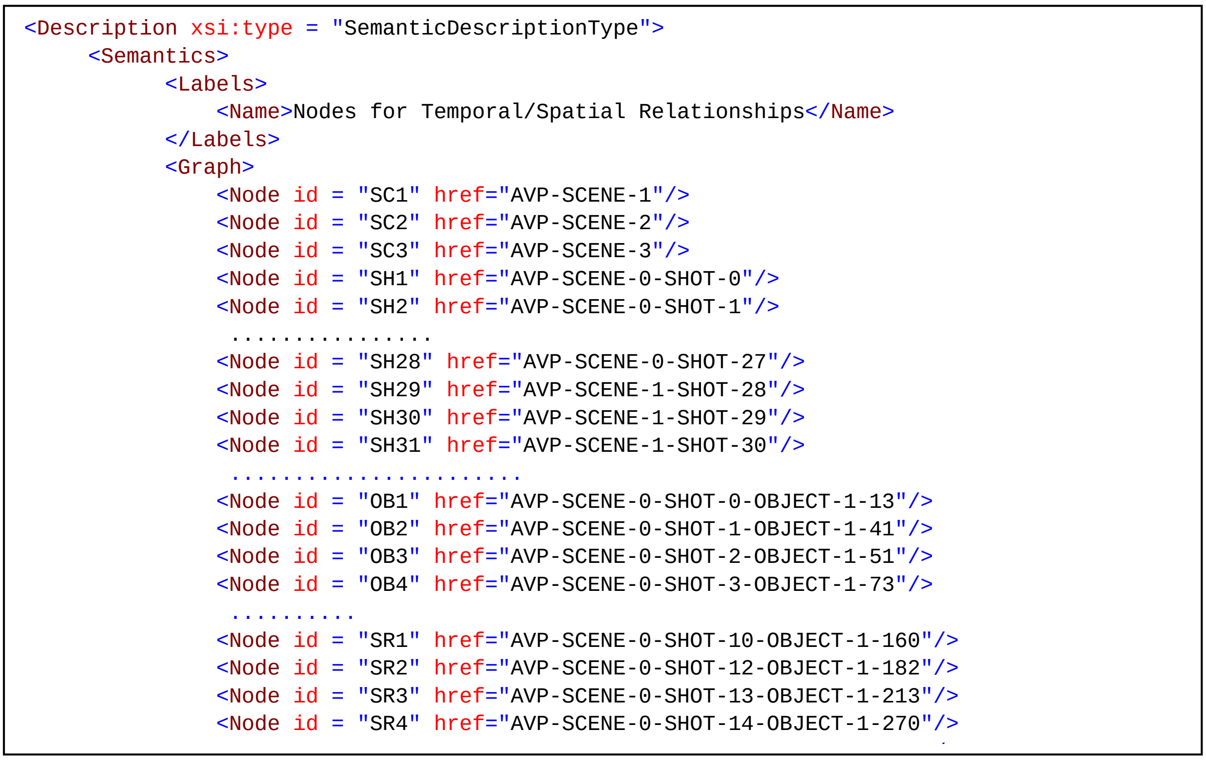
$$[i_1, j_1] < \ [i_2, j_2]$$

$$if\ i_1 \leq \ i_2 \ \therefore \ j_1 \leq \ j_2$$

Where $i_1 =$ start of syntactic features and $i_2 =$ end of syntactic features thus syntactic features can be ordered according to each associated features i value such that the feature denoted by $[i_1, j_1]$ precedes the feature denoted by $[i_2, j_2]$. Partial ordering enables MAC-REALM to determine which content features occur before or after which other content features and which intersect or occur simultaneously ('simultaneously" is defined as $[i_1, j_1] \subseteq \ [i_2, j_2]$).). For example, once partially ordered, to determine if a syntactic feature, A, occurs before or after a given group of syntactic features, B, the i value of A is compared to the i value of the first syntactic feature within B. If it is smaller, then A occurs before B. However, if the j value of A is greater than the j value of the last syntactic feature within B, then A occurs afterwards. Similarly two syntactic features, $S_1 = [i_1, j_1]$ and $S_2 = [i_2, j_2]$, can be compared to determine if they intersect, which will be in one of five ways; (1) $i_1 = \ i_2$ and $j_1 < \ j_2$, (2) $i_1 < \ i_2$ and $j_1 = \ j_2$, (3) $i_1 < \ i_2$ and $j_1 < \ j_2$, (4) $i_1 = \ i_2$ and $j_1 = \ j_2$ and (5) $i_1 < \ i_2$ and $j_1 > \ j_2$. With these temporal ordering rules, it is possible during querying to, for example, determine the next group of syntactic features within a given set of syntactic features that occur simultaneously. This takes place as follows. Once all the time stamps for the start and finish of each syntactic feature is collected, those constituent syntactic features that occur within a specific parent syntactic feature would be partially ordered. The next group of syntactic features is determined by taking the first syntactic feature and then adding to the group those syntactic features whose j values are not greater than the j value of the first syntactic feature. These syntactic features are thus those that occur simultaneously with the first syntactic feature.

3.4.3 Semantic Modelling

The semantic relationships produced during the content analysis and linkage phase need to be referenced in order to show not just the relationships between the spatial and temporal relationships of the low level features (i.e. scenes, shots and objects), but also the temporal relationships of the spatial relationships between themselves and the low level features. Using the `SemanticDescriptionType` DS a referencing system can be constructed that allows for the flexibility and grammar needed to achieve such a referencing system, and also use MPEG-7 classification schemes to define the relationships between entities. Using the node graph structure allows both syntactic and semantic features to be named in a manner that is independent of their abstract type and attributes, and thus makes stating the relationships between features of heterogeneous origin uniform and standard.



```
<Description xsi:type = "SemanticDescriptionType">
     <Semantics>
           <Labels>
               <Name>Nodes for Temporal/Spatial Relationships</Name>
           </Labels>
           <Graph>
               <Node id = "SC1" href="AVP-SCENE-1"/>
               <Node id = "SC2" href="AVP-SCENE-2"/>
               <Node id = "SC3" href="AVP-SCENE-3"/>
               <Node id = "SH1" href="AVP-SCENE-0-SHOT-0"/>
               <Node id = "SH2" href="AVP-SCENE-0-SHOT-1"/>
                ..............
               <Node id = "SH28" href="AVP-SCENE-0-SHOT-27"/>
               <Node id = "SH29" href="AVP-SCENE-1-SHOT-28"/>
               <Node id = "SH30" href="AVP-SCENE-1-SHOT-29"/>
               <Node id = "SH31" href="AVP-SCENE-1-SHOT-30"/>
                ......................
               <Node id = "OB1" href="AVP-SCENE-0-SHOT-0-OBJECT-1-13"/>
               <Node id = "OB2" href="AVP-SCENE-0-SHOT-1-OBJECT-1-41"/>
               <Node id = "OB3" href="AVP-SCENE-0-SHOT-2-OBJECT-1-51"/>
               <Node id = "OB4" href="AVP-SCENE-0-SHOT-3-OBJECT-1-73"/>
                ..........
               <Node id = "SR1" href="AVP-SCENE-0-SHOT-10-OBJECT-1-160"/>
               <Node id = "SR2" href="AVP-SCENE-0-SHOT-12-OBJECT-1-182"/>
               <Node id = "SR3" href="AVP-SCENE-0-SHOT-13-OBJECT-1-213"/>
               <Node id = "SR4" href="AVP-SCENE-0-SHOT-14-OBJECT-1-270"/>
```

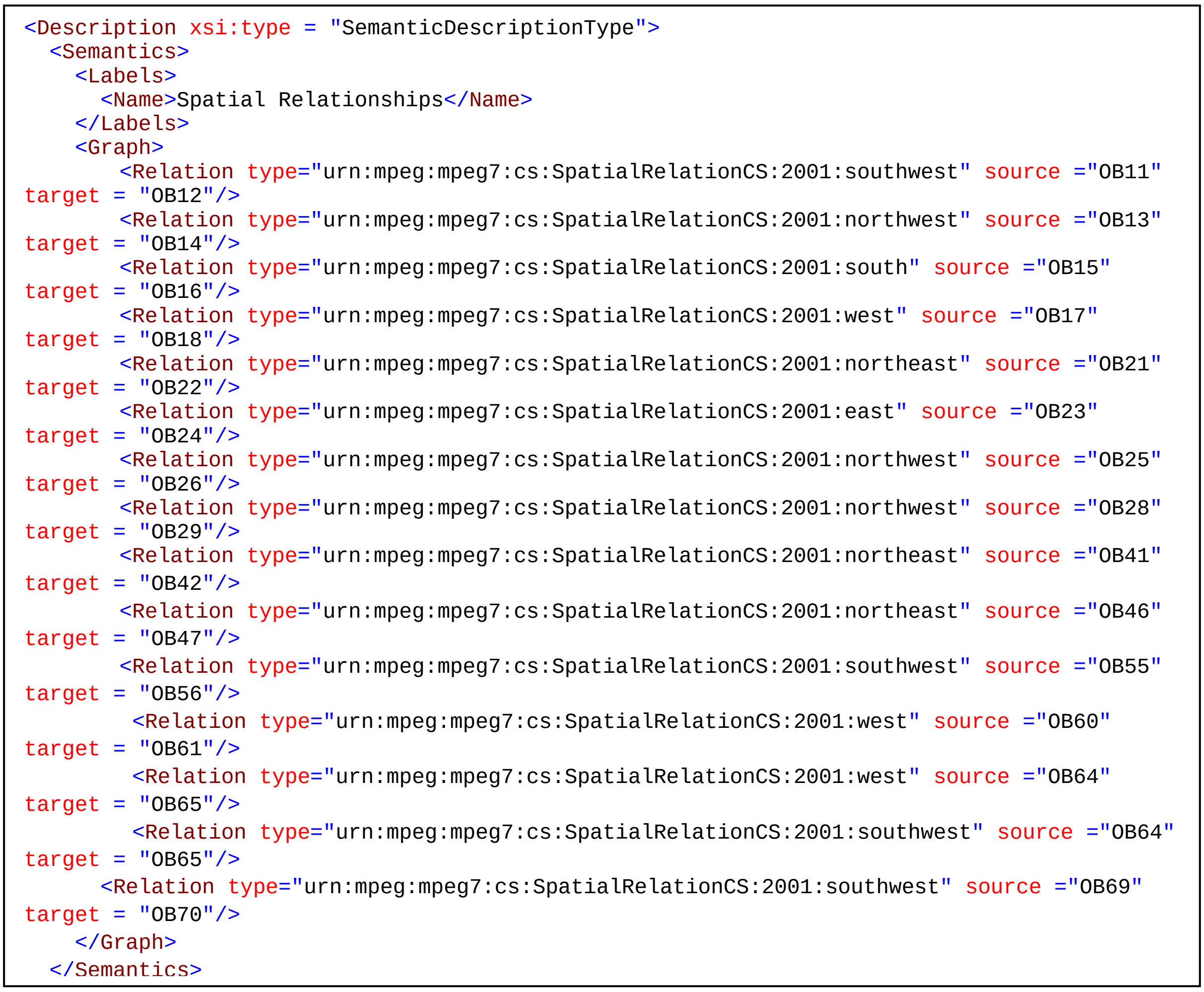
Figure 3.13: EXAMPLE OF LOW AND HIGH LEVEL FEATURES BEING REFERENCED IN MPEG-7

Using the SemanticDescriptionType DS a referencing system is created using the Graph DS. The Graph DS describes language-independent terms for use in multimedia descriptions and schemes for classifying a domain using a set of such terms. The ClassificationScheme DS describes a vocabulary for classifying a subject area as a set of terms organized into a hierarchy. A term defined in a classification scheme is used in a description with the TermUse or ControlledTermUse datatypes.

In the instance of referencing all low and high level features with a homogenous referencing system, the Graph DS allows us to create nodes that identify each feature set using the Node D tool. This tool allows us to assign a unique id tag to all low level and high level features that is related to the temporal instance of that feature. This keeps complexity to a minimum by not over complicating the linking mechanism when a relationship between a high level and low level feature is described. The low level features are referenced using their id tag from their original feature description. In the case of the spatial relationships the id tag from their original feature of the first object is used. This is done because it is known that for the spatial relationships only need a time reference point to be identified with, as the spatial relationship will be compared in terms of its temporal relationship to other features. The spatial relationship is linked to the other objects in the spatial relationship graph presented later. An example of scenes being modelled into nodes is given in Figure 3.13.

3.4.3.1 Spatial relationships

The spatial relations are modelled using the SpatialRelation CS, which defines all the spatial relationships that are describable in MPEG-7. Typecasting the Description DS to "SemanticDescriptionType" allows for the description of the spatial relationships between objects. Using the Semantics DS, objects are stated and the the spatial relationships described between them. The spatial relations graph is labelled using the Label DS within this element. The Graph DS is then used to describe the spatial relationships between those objects. The Relation DS is used to describe the spatial relationship between two objects. The spatial relationship is stated using the SpatialRelation CS, which defines the relationship in terms of a source node applied to a target node. The node structuring allows for a flexible and clearer way of describing relationships then if stating them directly. An example of MPEG-7 SpatialRelationship CS is given in Figure 3.14 that shows the relative spatial relationships between objects.

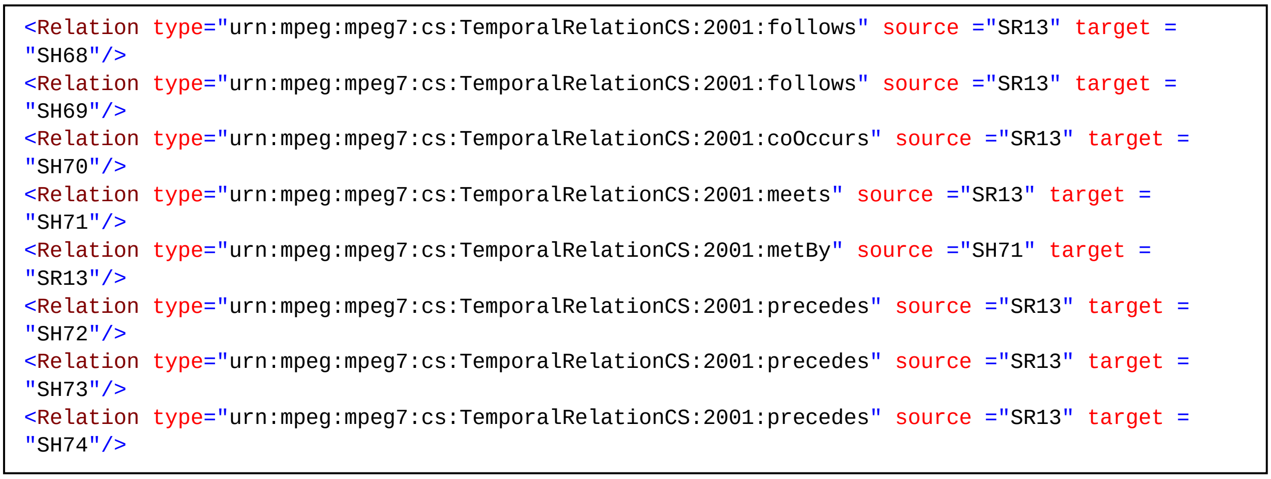


```
<Description xsi:type = "SemanticDescriptionType">
  <Semantics>
    <Labels>
      <Name>Spatial Relationships</Name>
    </Labels>
    <Graph>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB11"
target = "OB12"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB13"
target = "OB14"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:south" source ="OB15"
target = "OB16"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB17"
target = "OB18"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB21"
target = "OB22"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:east" source ="OB23"
target = "OB24"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB25"
target = "OB26"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB28"
target = "OB29"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB41"
target = "OB42"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB46"
target = "OB47"/>
       <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB55"
target = "OB56"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB60"
target = "OB61"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB64"
target = "OB65"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB64"
target = "OB65"/>
      <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB69"
target = "OB70"/>
    </Graph>
  </Semantics>
```

Figure 3.14: EXAMPLE OF SPATIALRELATIONSHIP CS IN MPEG-7

3.4.3.2 Temporal Relationships

Temporal relationships are modelled in much the same manner as spatial. Once again we typecast the Description DS to "SemanticDescriptionType" to indicate the following graph is describing high level features (i.e. semantic content). The graph is labelled using the Label DS to identify it as a temporal relationship graph. In a similar manner as before the Relation D within the Graph DS is used to describe the relationships. The difference is that now the MPEG-7 TemporalRelation CS is used to typecast the graph as containing temporal relationships. Using the aforementioned referencing system the temporal relationships are described between both homogeneous and heterogeneous content type feature sets. In Figure 3.16 an example of the variety of different types of temporal relationship is shown between homogeneous content type feature sets. The nodes in Figure 3.16 that represent scenes and shots content descriptions. In Figure 3.15 an example is given of temporal relationships modelled between heterogeneous content feature types, the nodes represent shot and spatial relationships. From the two examples of homogeneous and heterogeneous content feature types it can be seen that the nodes provide a proxy representation of the features. This abstract representation of the feature sets allows temporal comparison between them facilitating the requirement of multi-content type search possible in a semantic context.



```
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SR13" target =
"SH68"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SR13" target =
"SH69"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:coOccurs" source ="SR13" target =
"SH70"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SR13" target =
"SH71"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SH71" target =
"SR13"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH72"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH73"/>
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SR13" target =
"SH74"/>
```

Figure 3.15: EXAMPLE OF TEMPORAL RELATIONSHIPS BETWEEN HETEROGENEOUS CONTENT TYPE FEATURE SETS

```
<Description xsi:type = "SemanticDescriptionType">
  <Semantics>
    <Labels>
      <Name>Temporal Relationships</Name>
    </Labels>
    <Graph>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SC1" target =
"SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SC2" target =
"SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SC1"
target = "SC3"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC2" target
= "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:meets" source ="SC2" target =
"SC3"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:metBy" source ="SC3" target =
"SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC3" target
= "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC3" target
= "SC2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC1" target
= "SH1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows" source ="SC1" target
= "SH2"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1"
target = "SH29"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:starts" source ="SC1" target
= "SH29"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:startedBy" source ="SH29"
target = "SC1"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1" target
= "SH30"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictDuring" source ="SC1"
target = "SH30"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictContains" source ="SH30"
target = "SC1"/>
        <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:contains" source ="SC1"
target = "SH31"/>
      <Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:strictDuring" source ="SC1"
target = "SH31"/>
```

Figure 3.16: EXAMPLE OF A VARIETY OF MPEG-7 TEMPORAL RELATIONSHIPS WITHIN A HOMOGENEOUS FEATURE SET

3.5 Modelling plane

As discussed in chapter 1 the combining of all the content descriptions into a single content model document is the primary goal of MAC-REALM. Only then is the full potential of the content descriptions achieved, as the content model gives all the features a context and relationship to the structure and meaning of the content. As stated in chapter 2 the final content model document should link all the features together through a hierarchical structure and should provide mechanisms for all the features to be interlinked into a flat structure to optimise search capabilities and content discovery.

MAC-REALM modelling algorithm achieves this by layering the content features to the root node so that the top-level container for each feature is only one link away from any other top-level container. This makes the content more easily searchable and content discovery multi-faceted as the features are not rigidly structured in a nested tall structure. To date, there is no other content modelling scheme that has used such a method to modelling as they have been created with a specific purpose. MAC-REALM content model facilitates generic use through the structure and interlinking of its content descriptions.

The modelling plane has three distinct sections; 1) Syntactic Semantic Descriptions, 2) Content Modelling and 3) Model media. The first section is the MPEG-7 output from the extraction plane and analysis and linkage plane. These descriptions were modelled in the MPEG-7 layer. The second section parses all the MPEG-7 descriptions and then combines them into one DOM. The DOM is then serialised to the final stage, which is the model media. This is the final process of MAC-REALM, and outputs a MPEG-7 document model that can be used by any search/filtering application that is MPEG-7 compliant.

3.5.1 Syntactic Semantic Descriptions

In this content layer section, both the syntactic and semantic content descriptions are stored, and expressed as MPEG-7 descriptions. Syntactically there is shots/scenes and objects, and semantically there is spatial and temporal relationships. These descriptions on their own are still very useful and can be published as is. This would be useful to MPEG-7 compliant devices where storage, bandwidth or processing power is constrained and only certain aspects of the media content are of interest.

3.5.2 Content Modelling

In chapter 1 it was shown that the four categories of features needed to be included for a comprehensively described content model. These features were a) temporal segments b) objects, c) the spatial relationships between them, c) events and the e) all the temporal relationships between them. In MAC-REALM these are represented by a) shots b) moving regions for objects, b) spatial relationships, c) scenes and d) temporal relationships. These features are integrated together These features can be split into two different groups based on content type:

1) Syntactic (Structural) description schemes – these describe the low level and mid-level syntactic content descriptions. The syntactic content descriptions are built around the notions of segment description schemes that represent temporal or spatiotemporal aspects of the multimedia content. These description schemes utilise a hierarchical organisation that can produce an indices for searching the multimedia content. The

common reference point for video is the media time DS. These segments can then be further described in terms of colour, shape, etc.

2) Semantic (Conceptual) description schemes – These describe the higher level semantic content descriptions. The semantic content description entities are described through graph structures that provide a method for defining the semantic relationships between content features. The graph structure creates an abstract relationship between entities to form a conceptual narrative that is abstractly linked to the structural foundations of the multimedia content.

The syntactic and semantic content description schemes are linked by two methods that allow the multimedia content to be integrated so that the semantic gap can be bridged. The first method allows the syntactic and semantic content features to be linked on a semantic level. The semantic linking mechanism is provided by modelling all the features into nodes that represent all content features abstractly. The nodes are then used as proxy representations for the features and the temporal relationships between all these features is modelled allowing direct comparison between all content features, regardless of content type. This is shown in the temporal relationships section 3.3.3.2.

The second method for facilitating multi-content type search is to model the syntactic and semantic content descriptions together into a content model with all the features interlinked through their logical dependencies. When the features were extracted they were extracted using a hierarchical input/output extraction process where each feature extracted was the input for the next feature. Due to the extraction process all the features are intrinsically and implicitly linked together as the features share many attributes in common.

The scenes and shots have the media time to link them together and are naturally nested within each other, as scenes consist of shots. The objects are created from the shots and are linked to them through their id reference attributes. All the syntactic features are then modelled into nodes. The object nodes are used to model the spatial relationships, which implicitly links the spatial relationships to the shots and scenes through inheritance of attributes. The nodes of all the features are then modelled into temporal relationships, providing semantic linking of all the features. The arrangement of the linking mechanisms throughout the content makes joint syntactic and logical content based video search more effective as one search parameter can be applied to any amount of content features simultaneously.

MAC-REALM unifies the syntactic and semantic content using these linking mechanisms. The framework is set by defining the top level elements that state this MPEG-7 document relates to content description of the structural and conceptual content of video. Within this modelling structure both types of content can be defined and link together, specifically usingMPEG-7 part 5 MDS descriptions. The first structural elements to be defined are the top level elements, as these are the skeleton of the content model and establish MPEG-7 compliance. The Multimedia DS, which is typecast to video, is the anchor element for both the syntactic and semantic content description schemes. There are two description schemes that relate directly to the Multimedia DS, as they describe two global values associated with the video; MediaLocator DS and MediaTime DS. The MediaLocator DS contains the MediaURI D which describes the physical location of the media. The MediaTime DS uses the MediaTimePoint DS and MediaDuration DS to describe the global start time of the media and its duration respectively.

Within the structural description schemes there are three main description schemes anchored to the Multimedia DS; AnalyticEditingTemporalDecomposition DS (container element for Shots DS), VideoSegmentTemporalDecomposition DS (scenes) and MovingRegion DS (objects). The VideoSegmentTemporalDecomposition DS defines scenes through VideoSegment DS child nodes. Within the VideoSegment DS, which represents the scenes directly, there is anchored the AnalyticEditingTemporalDecomposition DS that contains the shots for that particular scene. The AnalyticEditingTemporalDecomposition DS can also be a direct node from the Multimedia DS that can represent shots that do not belong to a scene. The AnalyticEditingTemporal-Decomposition DS can only be instantiated once as a top level structural type, unlike the VideoSegmentTemporalDecomposition DS, but can appear many times under the VideoSegment DS, on a one-on-one basis per instantiation of the VideoSegment DS.

The MovingRegion DS, which represents objects, is rooted to the Multimedia DS and treated as a structural top level type. Each object is represented by its own instantiation of a MovingRegion DS. Each object can appear as an instantiation of the MovingRegion DS as a top level node as many times as is necessary. Alternatively if there aren't any objects, then there will be no MovingRegion DS instantiations.

The semantic relationships of the content model are all anchored to the Multimedia DS through the SemanticDescriptionType DS, which is cast through the abstract Description DS. Whereas the structural components had used time as the basis of their hierarchical structure, within the SemanticDescriptionType DS the graph structure of semantic relations is used. The Graph DS is the only child node of the Semantic DS and is instantiated for both spatial and temporal relationships. The Shot DS, VideoSegmentTemporalDecomposition DS and the MovingRegion DS are all model into Node D's. As stated, the Node D allows the freedom to make relationships not between different content type features and allows the content model to detail the intricacies of relationships between syntactic and semantic content features. In Figure 3.17 an entity diagram is presented of the unified content model showing the relationships between the top level document nodes and the syntactic and semantic nodes within them.

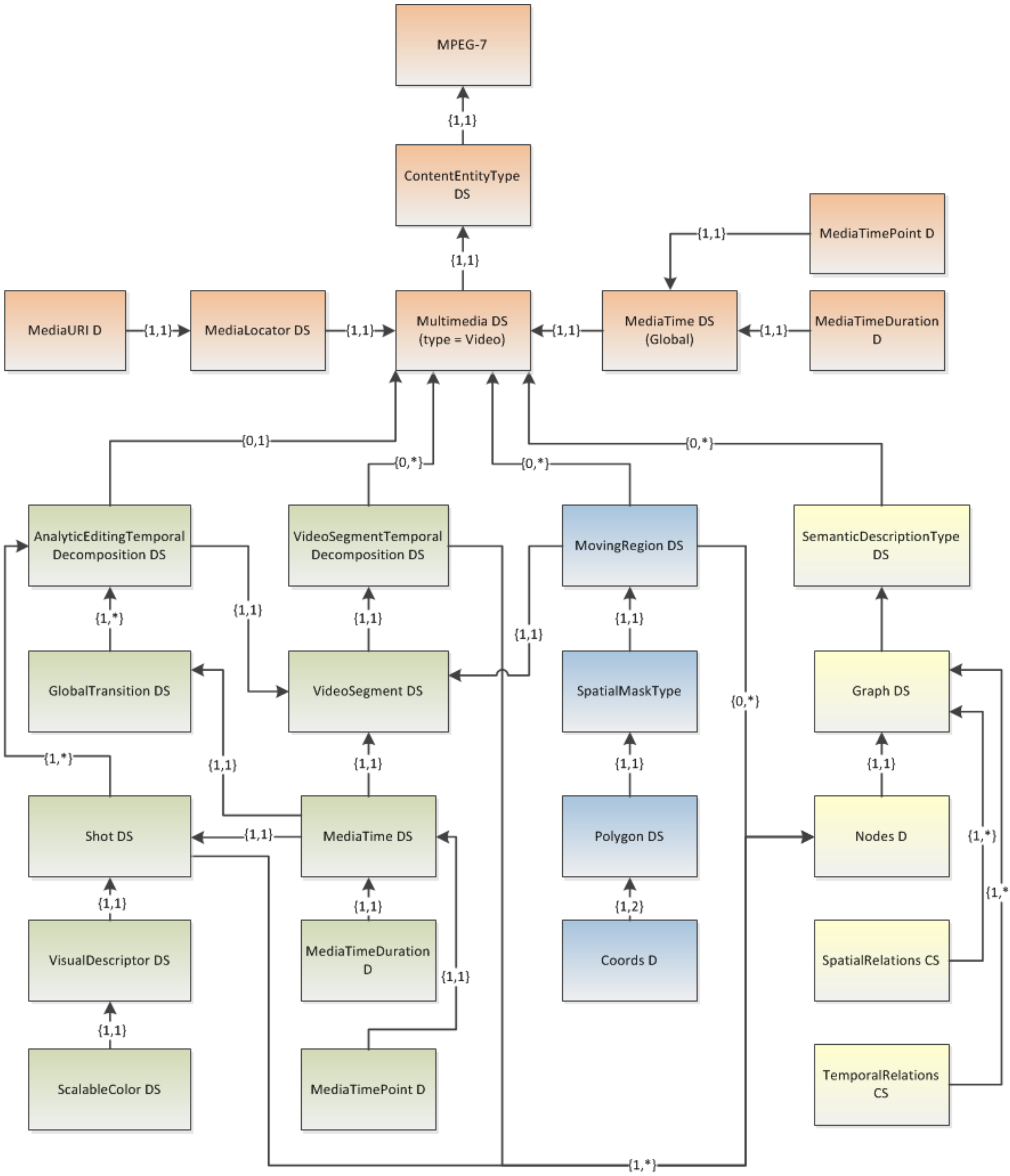


Figure 3.17: UNIFIED MPEG-7 CONTENT MODEL ENTITY DIAGRAM

3.5.3 Model Media

The linking mechanism between the syntactic and semantic description schemes is based on the structural hierarchy of the syntactic features. The serial numbering of scenes and shots provide the linking mechanism foundation for the content model. Objects are referenced using the shot id reference they come from and spatial and temporal relationships use the node reference to associate by proxy with the content features.

In Figure 3.18 is an example of the MAC-REALM content model that has been simplified so that all the content descriptions can be placed within it. The content model begins with the top level description schemes that provide the anchor for any MPEG-7 standard content model. Using the attribute declaration "type" within the Multimedia DS, the element is typecast as "VideoType". The child element of the Multimedia DS is the Video DS that encompasses all the syntactic and semantic content description schemes and relates them to the description of video. The Video DS contains the MediaLocator DS, and the MediaTime DS, stating the physical location and global time properties of the media respectively.

After the content model container is initialised the first content features to be added are the shots that do not belong to a scene. These orphan shots are modelled within the top level AnalyticEditingTemporalDecomposition DS element. Within this element the Shot DS's are modelled with an id reference that consists of the name of the movie, a scene number of '0' and the shot number, all delimited by a hyphen e.g. "AVP-SCENE-0-SHOT-1". Next the scenes are modelled, which are explicitly stated within VideoSegmentTemporalDecomposition DS via a single child VideoSegment DS. The VideoSegment DS id attribute is given the scene number of the clip in the same manner as the orphan shots but with the shot information omitted, e.g. "AVP-SCENE-1". Within the VideoSegment DS the shots for the scene are represented via a child AnalyticEditingTemporalDecomposition DS. The Shot DS's are referenced as before but have a scene number that references the scene number from the corresponding parent VideoSegment DS. This way the scenes, shots and orphan shots can all be searched using the same search parameters.

```
<Mpeg7xmlns="urn:mpeg:mpeg7:schema:2001"xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"xmlns:schemaLocation="urn:mpeg:mpeg7:schema:2001:Mpeg7-
2001.xsd">
  <Description xsi:type="ContentEntityType">
    <Multimedia xsi:type="VideoType">
      <Video>
         <MediaLocator>
           <MediaURI>F:/AVP/Video/AVP_test.mpg</MediaURI>
         </MediaLocator>
         <MediaTime>
           <MediaTimePoint> PT1H5M3S0N25F </MediaTimePoint>
           <MediaDuration>PT25M35S20N25F</MediaDuration>
         </MediaTime>
         <AnalyticEditingTemporalDecomposition>
          …………………. .
            <Shot id = "AVP-SCENE-0-SHOT-1">
             …………………….
         </AnalyticEditingTemporalDecomposition>
         <VideoSegmentTemporalDecomposition>
            <VideoSegment id = "AVP-SCENE-1">
            <AnalyticEditingTemporalDecomposition>
               …………………. .
              <Shot id = "AVP-SCENE-1-SHOT-1">
               …………………….
            </AnalyticEditingTemporalDecomposition>
         </VideoSegmentTemporalDecomposition>
         <VideoSegmentTemporalDecomposition>
           <VideoSegment id = "AVP-SCENE-2">
         </VideoSegmentTemporalDecomposition>
            …………………. .
         <MovingRegion id = "AVP-SCENE-0-SHOT-1-OBJECT-1-41">
          …………………. .
         </MovingRegion>
          …………………. .
         <MovingRegion id = "AVP-SCENE-3-SHOT-13-OBJECT-2-786">
          …………………. .
        </MovingRegion>
        <Description xsi:type = "SemanticDescriptionType">
          <Semantics>
             …………………. .
            <Graph>
              <Node id="SC1" href="AVP-SCENE-1"/>
               …………………. .
              <Node id="SH203" href="AVP-SCENE-3-SHOT-1"/>
               …………………. .
              <Node id = "OB50" href="AVP-SCENE-2-SHOT-55-OBJECT-1-3463"/>
               …………………. .
              <Node id = "SR13" href="AVP-SCENE-2-SHOT-69-OBJECT-1-5908"/>
            </Graph>
          </Semantics>
          <Semantics>
            …………………. .
            <Graph>
              <Relation  type  =  "urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west"  source  ="OB17"
target = "OB18"/>
              …………………. .
            </Graph>
          </Semantics>
          <Semantics>
             …………………. .
            <Graph>
             <Relation  type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:follows"  source  ="SH180"
target = "SR15"/>
            </Graph>
          </Semantics>
        </Video>
     </Multimedia>
   </Description>
</Mpeg7>
```

Figure 3.18: UNIFIED MPEG-7 CONTENT MODEL SKELETON

The id reference for scenes and shots is extended for objects and used in the MovingRegion DS to include the object number and frame the object first appears in, e.g. "AVP-SCENE-0-SHOT-1-OBJECT-1-41". This provides the link between scenes/shots and objects.

This naming convention for scenes, shots and objects is then used as the input for the Semantics DS relations structure, via the Graph DS. As explained in section 3.3.3, the referencing mechanism employed allows for the relationships between heterogeneous and homogenous content features to be explored and stated without the attributes associated with content feature or content type. This abstract approach to syntactic and semantic content type linkage opens up the opportunity to explore relationships between syntactic and semantic information that are more informative. They provide a more cognitive approach to the content model that is more holistic and true to the content and the myriad ways a user perceives and searches content.

3.6 Summary

Chapter 3 presents MAC-REALM and the implementation of its four planes, three layers architecture. The implementation requirements are presented based on the research methods and design requirements stated in earlier chapters. An overview of the MAC-REALM prototype is shown, and how the custom video processing pipeline passes through the planes and the interaction of the components of each plane play a part in transforming the content. Next each plane is presented individually and the function of each component within the plane detailed.

The raw media plane removes redundant data by removing frames incrementally, converts the colour space to be more beneficial to extraction and finally the noise is removed from the frames using morphological filtering to stop impurities affect the performance of the extraction process. The filtered frames and the colour histograms are sent to the syntactic media component in the extraction plane to await processing.

The extraction plane integrates two shot detection algorithms to detect two different types of shot transition, abrupt cut and gradual transition. The shots are then used for object extraction, where a two-phase approach is used. Graph cuts segmentation is used to extract the object/s from the background, and then covariance matrix tracking is used to track the pixels across the shot. Both shots and objects, along with the colour histograms are used by the scene segmentation algorithm as the input of features that will be used by the GP Algorithm to evolve rules that will identify a scene boundary. The resulting shots, objects and scenes extracted in this plane are then used as input for the next plane. They are also modelled and serialised into syntactic MPEG-7 content descriptions.

The analysis and linkage plane analysis and links the content features together to form spatial and temporal relationships between them. Before the spatial relationships are defined the centroid of each object is defined to provide the reference point of measurement. Then the absolute, relative and inverse relative spatial relationships are calculated. Then all the temporal relationships between all the content features are mapped and modelled. The spatial and temporal relationship are then serialised into MPEG-7 semantic content descriptions.

The MPEG-7 syntactic and semantic features are then integrated together in the modelling plane. They are combined in a hierarchical structure that uses a MPEG-7 content model wrapper to interlink the syntactic and semantic content features into a tightly coupled integrated content model that is capable of granular search and facilitates multi-content type search.

CHAPTER 4: EVALUATING MAC-REALM

In chapter 3 the implementation of the MAC-REALM prototype is presented as a four planed, three-layered modular framework that implements a custom video processing pipeline to convert video into a MPEG-7 content model.

In this chapter provides a walkthrough MAC-REALM, undertakes an empirical performance evaluation of the sub components of MAC-REALM and how well each component completes its task. Finally, an evaluation of MAC-REALM framework is provided that discusses the walkthrough and performance evaluation results in the context of the research objectives, and draws con

In order to conduct a performance evaluation of the initial reference implementation of MAC-REALM, the objectives outlined in Chapter 1 are used as a benchmark for testing:

(E 1) To design an abstract framework that translates a video stream into content descriptions. The framework will extract and integrate syntactic and semantic content descriptions into a content model to reduce the semantic gap. This is proven through benchmark testing and evaluation of the framework functionality. The benchmark test will also prove the sub-objectives of:

(E 1.1) Creating an algorithm that reduces computational expense and filters the media to improve extraction accuracy of features. This objective is evaluated in two parts. First the computational saving is evaluated through mathematical proof that shows the reduction in computational expense. The improvement of feature extraction accuracy is evaluated by feature extraction metrics. These feature extraction metrics are part of the evaluation process in point 2 of this section.

(E 1.2) Detecting and accurately extracting low-level and mid-level syntactic features from the video stream. The extraction techniques that need evaluating are for shots, scenes and objects. The features are evaluated using common and standard benchmark tests that look at the accuracy and detection rates of MAC-REALM for each feature. This will investigate whether MAC-REALM extracts syntactic that can then be modelled into MPEG-7 syntactic content features.

(E 1.3) Automatically creating semantic relationships between extracted features, in a suitable manner for video searching to be possible. There are no standard benchmarking tests for the derivation of spatial or temporal relationships. An analysis is provided of the spatial and temporal relationships that quantifies the features and shows the relationship between the content features and the semantic relationships. For spatial relationships a benchmark test has been implemented to test the precision of the derived relationships compared to a user's view of the relationships to see if the method shown in chapter 3 for calculating the position of the objects.

(E 2) To implement the MAC-REALM framework into a proof of concept prototype. The functionality of MAC-REALM must be proven through a walkthrough of the proof of concept prototype to show how the functionality of the framework was implemented.

(E 3) To combine syntactic and semantic descriptions into a compliant content model that can be used by other applications in a standardised manner. The content model will be validated and shown to the best of effort to:

(E 3.1) Prove to be MPEG-7 compliant. A standardised and popular MPEG-7 validation tool is used to validate the resulting MAC-REALM content model and to check if it has any errors or is using illegal syntax or structure.

(E 3.2) Show the syntactic and semantic content descriptions that are valid to enable multi-content type content based video queries.

(E 3.3) Validate the hierarchically detailed structure of the content model to show that it enables granular content based video search in "coarse to fine" detail.

4.1 Walkthrough of MAC-REALM

THE MAC-REALM prototype is designed with the interface corresponding to the systems' architecture described in chapter 2. The prototype implements the four plane and three layer architecture that converts a raw media stream into a content model. Each module can be experimented and tested and results clearly viewed for evaluation. It also allows for easier development of future features and extensibility of MAC-REALM as further modules can be added as plug-ins, without major alteration to the MAC-REALM base platform.

MAC-REALM consists of four main tabs each corresponding to the 4 planes of the MAC-REALM architecture: Raw Media, Extraction, Analysis and Linkage and Modelling. Each main tab is then split into subsequent sub-tabs that each represents one of the MAC layers: Content, Application and MPEG-7. Section-tabs are provided for different processes for extracting/linking features when a sub tab has multiple processes. These section tabs correspond with the sub-sections indicated in the application layer of the system architecture.

4.1.1 Raw Media Plane

The initial sub-tab of the first screen of MAC-REALM allows the user to input an AV stream to initialise the raw media extraction process. In Figure 1.0Figure 2.1Figure 3.1Figure 4.0 the "Load" button is seen that the user presses to bring up a file chooser pop up box. The user then navigates through the directories to the media clip of their choice. If the user picks a format that is not recognised a pop up box appears telling them that they cannot use this clip as the format is unrecognised and must select another media clip that is compatible.

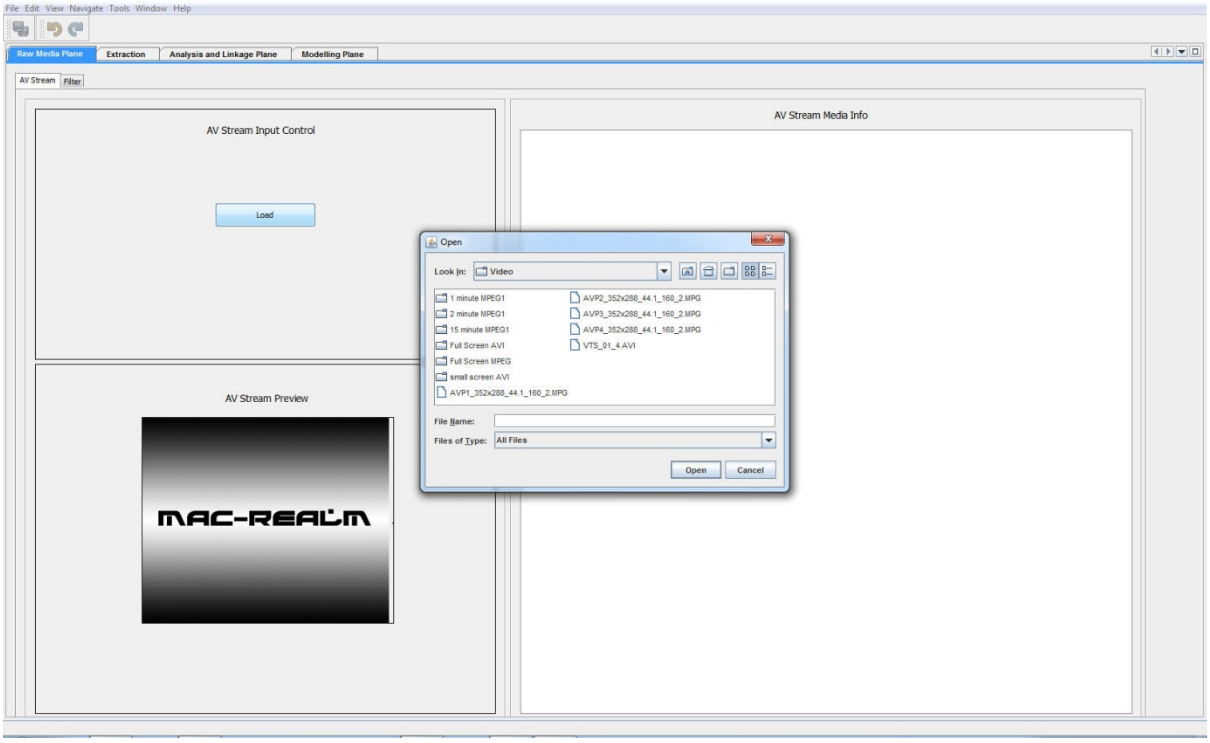
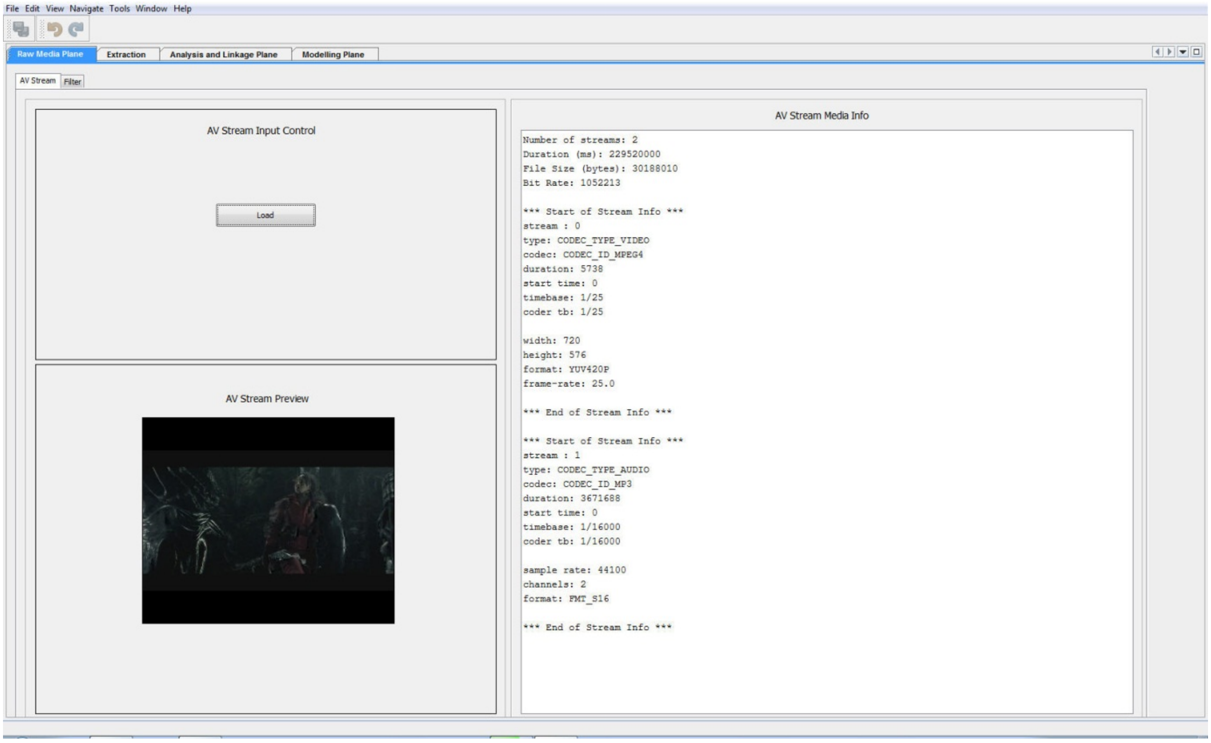


Figure 4.1: RAW MEDIA PLANE – AV STREAM SCREEN (CHOOSING AV FILE)

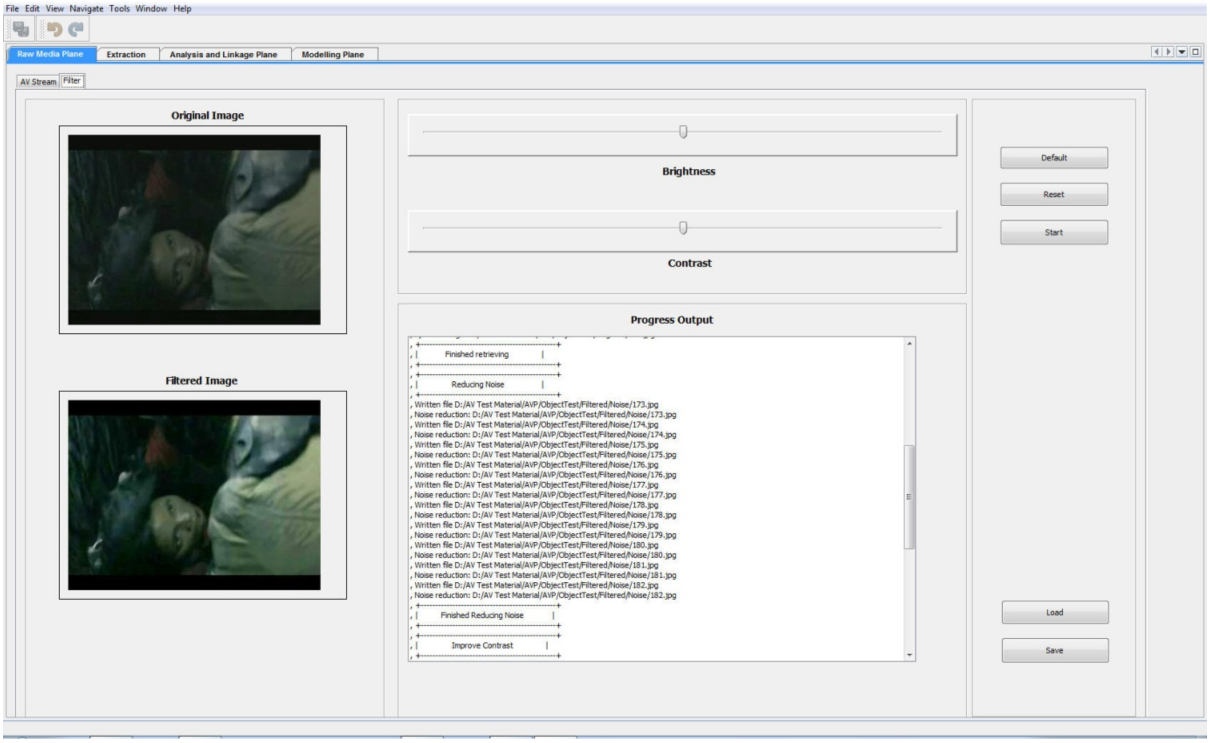
Once the input stream is accepted the media clip is played in the lower left preview panel. The AV stream media info also shows information about the clip's physical structure. The AV stream info is split into two main sections, the general media attributes and the stream attributes. Depending on the number of streams there might be more than one stream attributes section. The general media attributes described are always consistent. In Figure 4.1 it can be seen as: number of streams present, duration (in milliseconds) of media clip, file size (in bytes) and bit rate. The individual stream attributes will differ depending on the type of stream, video or audio. For both video and audio the codec type and the codec itself, the start time of the media, timebase and coder timebase are all potentially available. For video there is also width and height, format and frame rate.



```
Number of streams: 2
Duration (ms): 229520000
File Size (bytes): 30188010
Bit Rate: 1052213

*** Start of Stream Info ***
stream : 0
type: CODEC_TYPE_VIDEO
codec: CODEC_ID_MPEG4
duration: 5738
start time: 0
timebase: 1/25
coder tb: 1/25

width: 720
height: 576
format: YUV420P
frame-rate: 25.0

*** End of Stream Info ***

*** Start of Stream Info ***
stream : 1
type: CODEC_TYPE_AUDIO
codec: CODEC_ID_MP3
duration: 3671688
start time: 0
timebase: 1/16000
coder tb: 1/16000

sample rate: 44100
channels: 2
format: FMT_S16

*** End of Stream Info ***
```

Figure 4.2: RAW MEDIA PLANE – AV STREAM SCREEN (MEDIA INFO)

Once the video has finished the input of the video, and the media information has been retrieved, the filtering process of the raw media plane begins. In Figure 4.2 the filter panel is shown. The filter panel is set into four parts: the image preview panel, filter settings control panel, main filter control panel and process output panel.



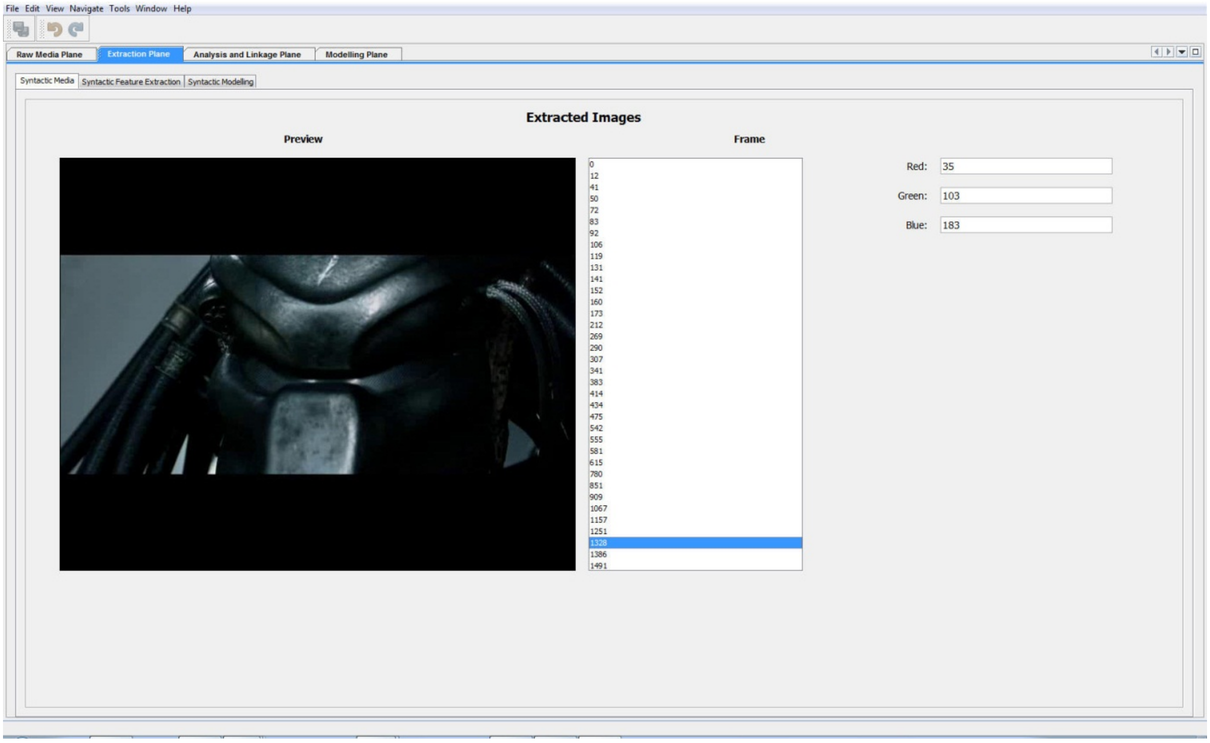
```
, +------------------------------------+
, |        Finished retrieving         |
, +------------------------------------+
, +------------------------------------+
, |          Reducing Noise            |
, +------------------------------------+
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/173.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/173.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/174.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/174.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/175.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/175.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/176.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/176.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/177.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/177.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/178.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/178.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/179.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/179.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/180.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/180.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/181.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/181.jpg
, Written file D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/182.jpg
, Noise reduction: D:/AV Test Material/AVP/ObjectTest/Filtered/Noise/182.jpg
, +------------------------------------+
, |      Finished Reducing Noise       |
, +------------------------------------+
, +------------------------------------+
, |         Improve Contrast           |
, +------------------------------------+
```

Figure 4.3: RAW MEDIA PLANE – FILTER SCREEN

The image preview panel shows an extracted frame in two preview panels, the original image and the filtered image. The original image preview is the image without any filtering process applied, whilst the filtered image preview shows in real time the changes being made in the filter settings control panel. The filter setting control panel has 2 slider controls: one to alter the brightness the other to alter the contrast. Using these sliders the user can improve the contrast and brightness of a video. Once the user has selected settings they may save them using the save button in the right panel. Alternatively the user can load previous settings and use them if they wish. The reset button resets the brightness and contrast slide controls to their original positions. Once the user is happy with the contrast and brightness they may press the start button for the process of filtering all images to begin. The process can be viewed in real time in the progress output text area. This shows the frames (as image file) being retrieved, the noise in each frame being reduced and then the contrast and brightness being adjusted.

4.1.2 Extraction Plane

In the first tab of the extraction plane (Figure 4.3) we can see the content that was extracted from the plane before in the "syntactic media" tab. Here we can view both normalized frames and the colour histograms associated with those frames. This is done clicking on a frame number in the list box.


File Edit View Navigate Tools Window Help
Raw Media Plane
Extraction Plane
Analysis and Linkage Plane
Modelling Plane
Syntactic Media
Syntactic Feature Extraction
Syntactic Modelling

Extracted Images

Preview

Frame

0
12
41
50
72
83
92
106
119
131
141
152
160
173
212
269
290
307
341
383
414
434
475
542
555
581
615
780
851
909
1067
1157
1251
1328
1386
1491

Red: 35

Green: 103

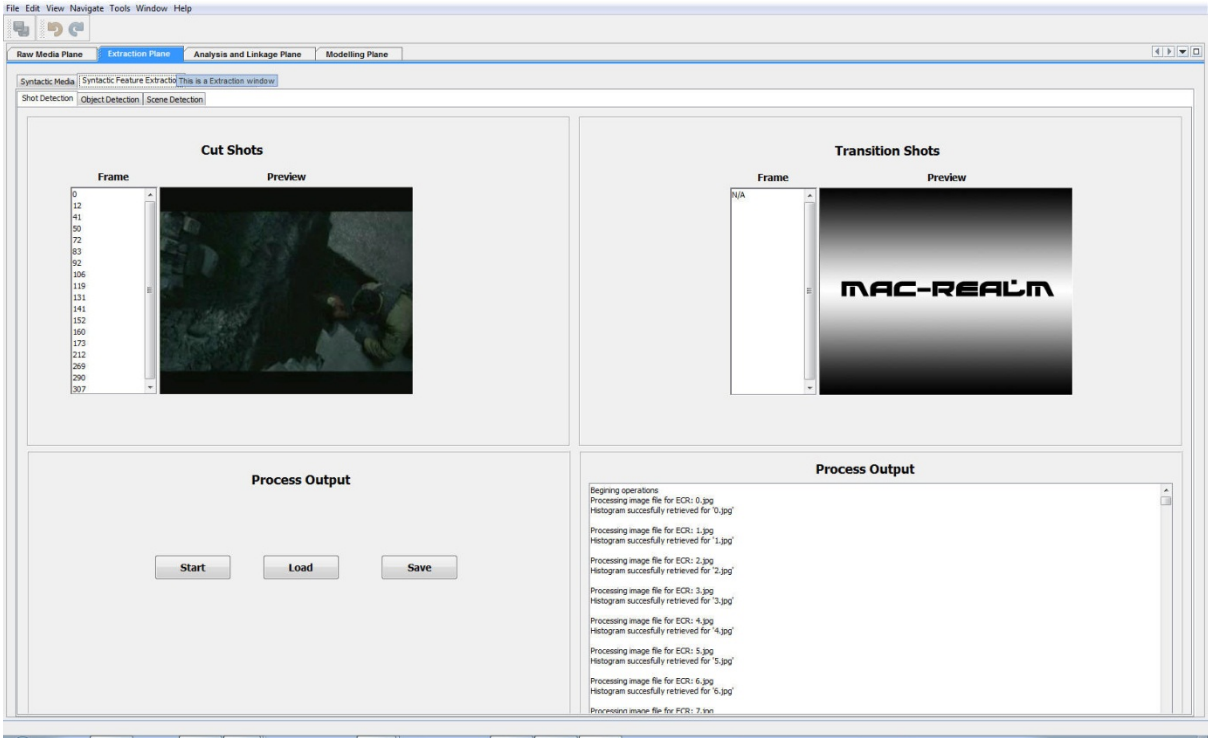
Blue: 183

Figure 4.4: EXTRACTION PLANE – SYNTACTIC MEDIA SCREEN

The next tab holds the application layer for the extraction plane. This has three sub tabs: 1) shot detection, 2) object detection and 3) scene detection.

The cuts shots sub-tab consists of four panels: the cuts shots preview panel, the transition shots preview panel, the control panel and the process output text area. The control panel has three buttons. The start button starts the shot detection for both cut and transition shots. The save button lets the user save the CHD's and ECR's of every frame in a text file. The load button lets the user load previously saved frames data text file. This was used for testing purposes so we would not have to go through the process from the start again and again. The test file button allows the user to load a file that has the cut/transition shots for the clip marked out manually. This is used to calculate the detected/false positive/missed scores for the detected shots. The results are shown in the text fields underneath.

Once the cut and transition shots are detected they are previewed in the cut and transition shot preview panels respectively. The cut shots preview panel allows the user to browse through the first frame of every cut shot that has been detected. This is done by selecting the frame in the list box on the right of each preview panel. The transition shots preview panel performs the same function but for transition shots. If no shots for either are found the preview panels show a cover image and a blank list box, as seen in Figure 4.4.



```
0
12
41
50
72
83
92
106
119
131
141
152
160
173
212
269
290
307
```

```
N/A
```

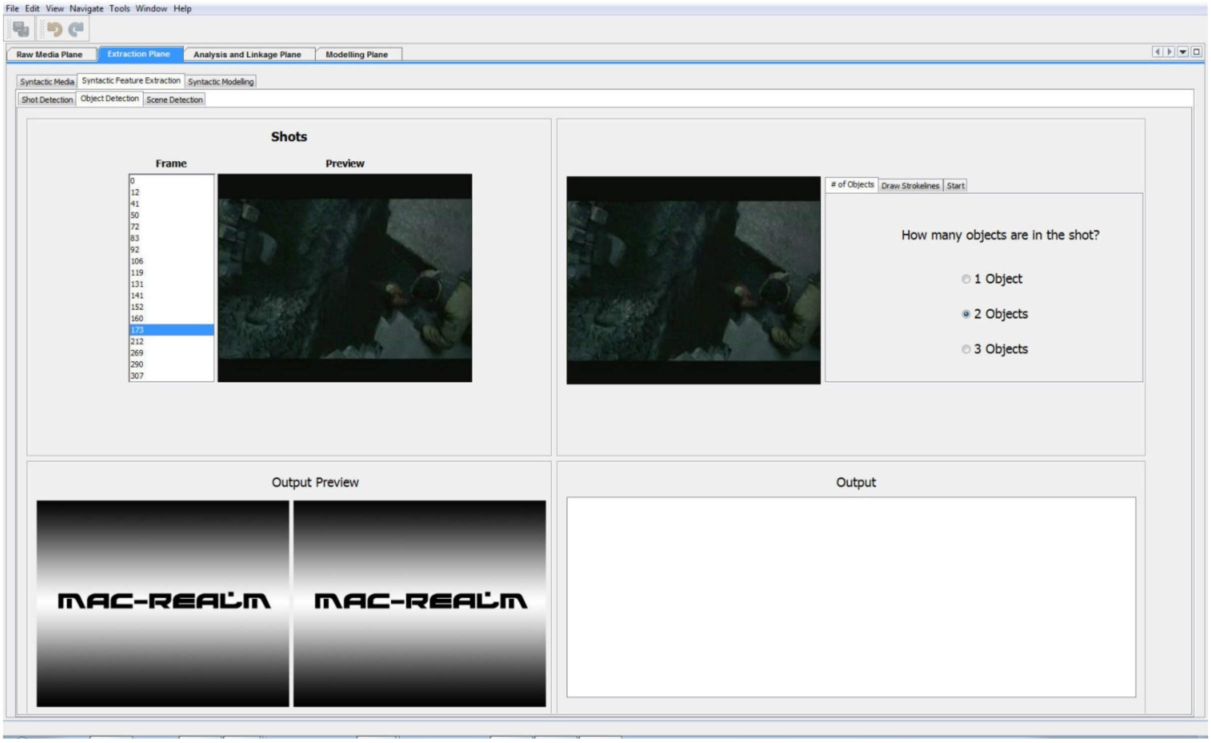
```
Begining operations
Processing image file for ECR: 0.jpg
Histogram succesfully retrieved for '0.jpg'

Processing image file for ECR: 1.jpg
Histogram succesfully retrieved for '1.jpg'

Processing image file for ECR: 2.jpg
Histogram succesfully retrieved for '2.jpg'

Processing image file for ECR: 3.jpg
Histogram succesfully retrieved for '3.jpg'

Processing image file for ECR: 4.jpg
Histogram succesfully retrieved for '4.jpg'

Processing image file for ECR: 5.jpg
Histogram succesfully retrieved for '5.jpg'

Processing image file for ECR: 6.jpg
Histogram succesfully retrieved for '6.jpg'

Processing image file for ECR: 7.jpg
```

Figure 4.5: EXTRACTION PLANE – SHOT DETECTION SCREEN

The process output text area panel for shot detection shows the process of detection in real time of both the transition and cut shots for monitoring purposes during tests.

After the shot detection tab we have the next sub-tab of the application layer for the extraction plane, the object detection panel. This is split into four separate panels: the shot selector panel, the user input panel, object preview panel and process output panel. The user selects a shot from the shot list box whose key frame is shown in the preview panel to the right of the list box. Once selected the user then begins using the user input panel. This has three sub tabs that can only be used in the order they are presented and must be completed for object detection to start. To the left of the sub tabs we have the image canvas panel that allows the user to draw stroke lines directly onto the image.

In Figure 4.5 we see the first step in the object detection process, namely the user input that is used to determine the starting points for model initialisation of the segmentation graphs. The user must first select the amount of objects in the frame to a maximum of three. The actual amount of objects that can be handled is unlimited. Three was picked as a suitable number for test purposes as this cuts down on computation time, storage and analysis of results. It also reduces the overhead in coding and application.



Frame
0
12
41
50
72
83
92
106
119
131
141
152
160
173
212
269
290
307

How many objects are in the shot?

○ 1 Object

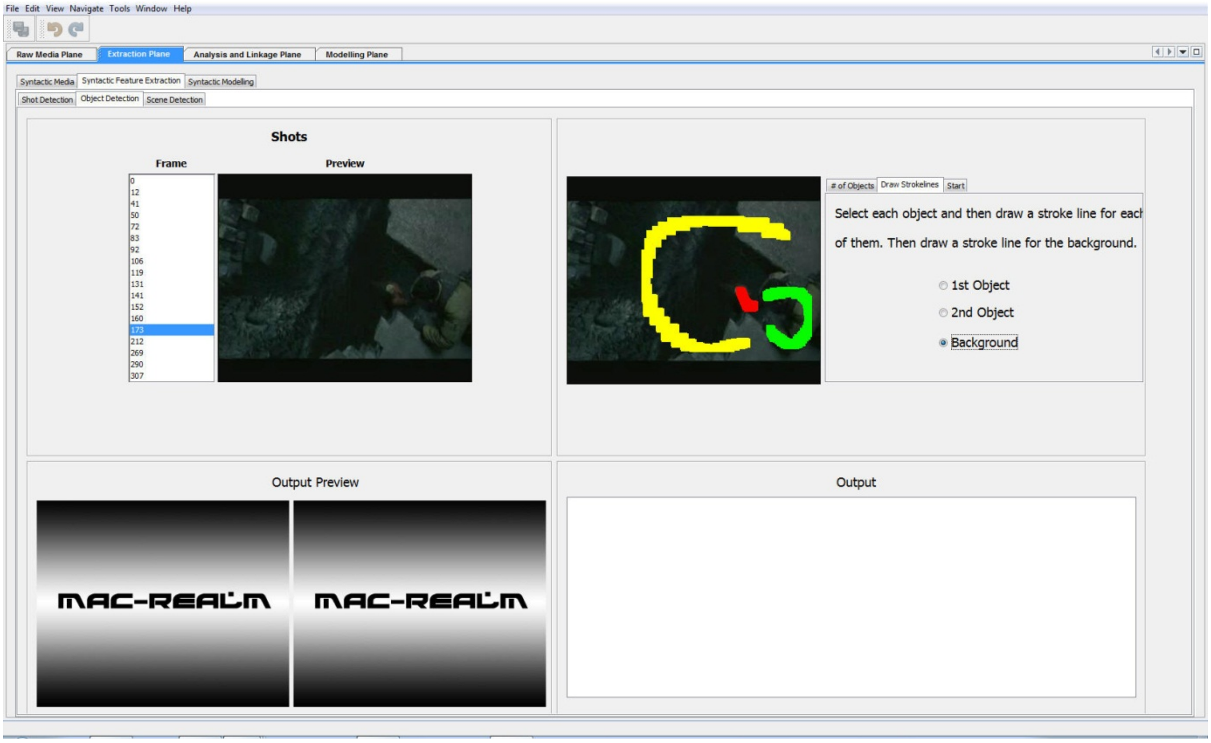
● 2 Objects

○ 3 Objects

Figure 4.6: EXTRACTION PLANE – OBJECT DETECTION SCREEN (NUMBER OF OBJECTS)

Once we have identified the amount of objects in the screen we move onto the next sub tab in the user input panel that allows the user to provide input that initialises the segmentation of the image into foreground objects and background. In Figure 4.6 we see that the next tab has three check boxes, two for the 1st and 2nd object and one for the background. This corresponds to the choice made by the user in Figure 4.5. If the user had selected a different amount of objects then this would have been reflected in the amount of choices for objects displayed. The user selects a tick box and then draws a stroke line for that selection on the image canvas panel. The user will select each tick box and draw a stroke line on the image that corresponds to the selection. Each item is given a different colour to distinguish it from the other selections.

The colours used for strokes are highly saturated primary colours so they can be easily distinguished from each other and the image background. This also stops false positives occurring by having colours too similar to colours already present in the background image. The drawback to this is that if the background image has highly saturated primary colours within it this would cause incorrect model initialisation of the segmented graphs causing erroneous results. This could easily be rectified by a colour analysis algorithm that would see what the prevalent colours where in the background image and then use only colours that are not in the background image. It is a rarity and only really occurs in animation where objects are artificially created.



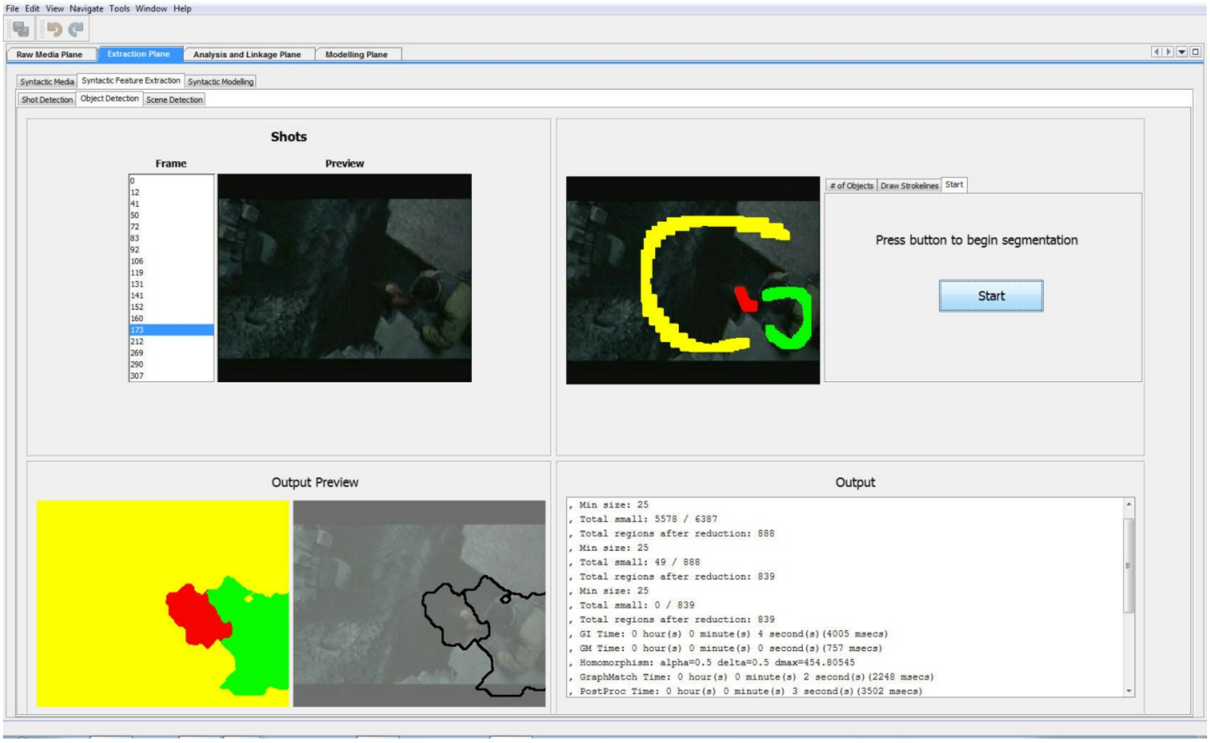
Select each object and then draw a stroke line for each of them. Then draw a stroke line for the background.

○ 1st Object
○ 2nd Object
● Background

Figure 4.7: EXTRACTION PLANE – OBJECT DETECTION SCREEN (DRAW STROKE LINES)

In figure 4.7 we see the last sub tab. This has a start button that begins the segmentation of the key frame into objects. Once pressed, we see the beginning of the automated object extraction engine begin. The process output panel shows the process of image segmentation in real time. This shows the image being initially segmented into regions by the watershed algorithm and then the region reduction being performed by applying the user input to the model image.

Once the image is segmented and the regions have been reduced to their minimal outlines the resulting object maps are shown in the output preview panel. In Figure 4.7 we see this in action. In the output process box we see the steps performed during initial image segmentation and subsequent region reduction in detail. In the output preview panel we see two images. The first image on the left of the output preview panel is the coloured object map of the extraction process. It shows the outlines of the objects according to the colour(s) used by the user when drawing stroke lines for each of the objects in the user input panel. The background colour is marked by the colour used in the user input panel as well. Next to that we have the preview image of the outline map of the objects. This shows a black outline drawn onto the original key frame and lets the user see how accurate the segmentation process was by providing a defined silhouette of the object.



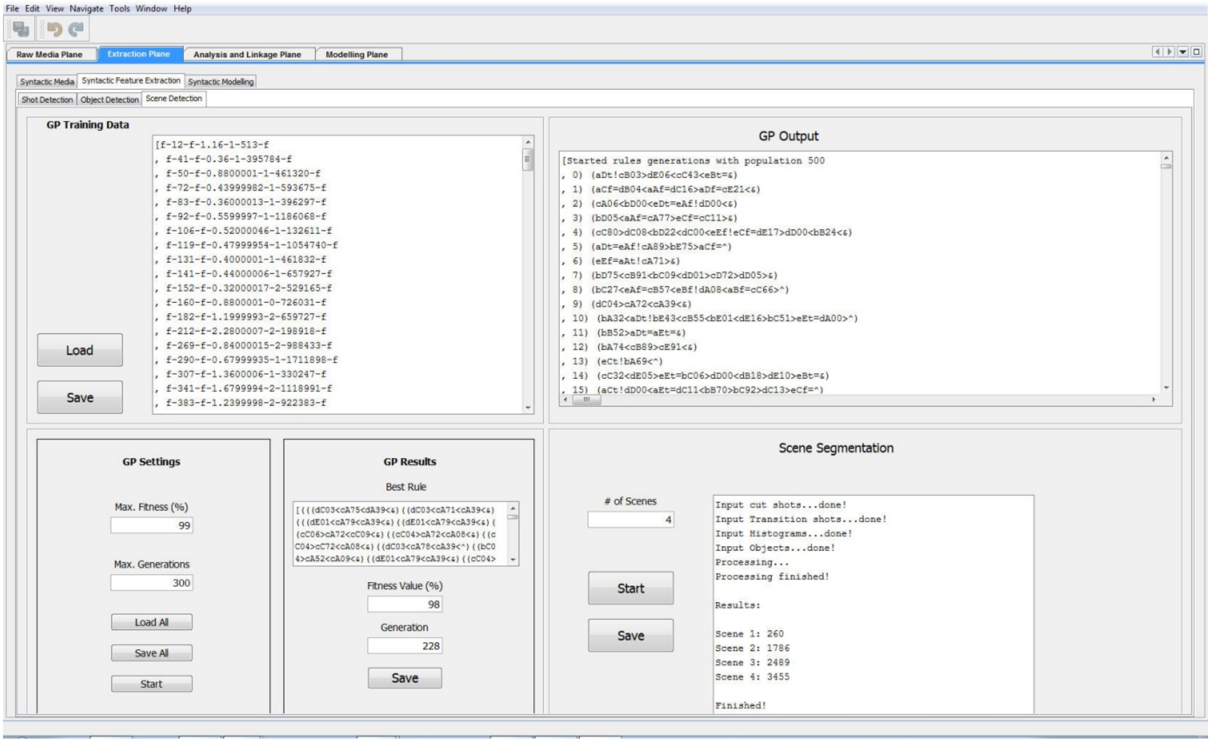
```
, Min size: 25
, Total small: 5578 / 6387
, Total regions after reduction: 888
, Min size: 25
, Total small: 49 / 888
, Total regions after reduction: 839
, Min size: 25
, Total small: 0 / 839
, Total regions after reduction: 839
, GI Time: 0 hour(s) 0 minute(s) 4 second(s)(4005 msecs)
, GM Time: 0 hour(s) 0 minute(s) 0 second(s)(757 msecs)
, Homomorphism: alpha=0.5 delta=0.5 dmax=454.80545
, GraphMatch Time: 0 hour(s) 0 minute(s) 2 second(s)(2248 msecs)
, PostProc Time: 0 hour(s) 0 minute(s) 3 second(s)(3502 msecs)
```

Figure 4.8: Extraction Plane – Object detection screen (objects detected)

The third tab is the final part of the application layer for extraction plane, the scene detection sub panel and is shown in Figure 4.8. This is split into four separate panels: the GP training data panel, the GP output panel, GP settings and results panel and scene segmentation panel.

We begin with the GP training data panel. Using the load button we select the training data file (a file with a small video sample clip with scene boundaries identified) that will be used by the fitness function to test and ascertain which rules are fit, and therefore should be allowed to evolve to the next generation. Once the file is loaded it can be viewed in the text area of the GP training data panel. Each line of the training data file represents a frame number, duration, histogram and number of objects within a shot.

Below this panel we have the GP settings and results panel. In the stings part of the panel we can adjust the maximum fitness of the rules so that if achieved the process will stop and present the rule. We also have a maximum generations setting that allows us to set the maximum amount of generations that will be cycled before termination of the GP process if the maximum fitness is not achieved. Underneath these two text fields we have the start button that begins the process of GP to find the best rule for scene segmentation.



GP Training Data

```
[f-12-f-1.16-1-513-f
, f-41-f-0.36-1-395784-f
, f-50-f-0.8800001-1-461320-f
, f-72-f-0.43999982-1-593675-f
, f-83-f-0.36000013-1-396297-f
, f-92-f-0.5599997-1-1186068-f
, f-106-f-0.52000046-1-132611-f
, f-119-f-0.47999954-1-1054740-f
, f-131-f-0.4000001-1-461832-f
, f-141-f-0.44000006-1-657927-f
, f-152-f-0.32000017-2-529165-f
, f-160-f-0.8800001-0-726031-f
, f-182-f-1.1999993-2-659727-f
, f-212-f-2.2800007-2-198918-f
, f-269-f-0.84000015-2-988433-f
, f-290-f-0.67999935-1-1711898-f
, f-307-f-1.3600006-1-330247-f
, f-341-f-1.6799994-2-1118991-f
, f-383-f-1.2399998-2-922383-f
```

GP Output

```
[Started rules generations with population 500
, 0)  (aDt!cB03>dE06<cC43<eBt=&)
, 1)  (aCf=dB04<aAf=dC16>aDf=cE21<&)
, 2)  (cA06<bD00<eDt=eAf!dD00<&)
, 3)  (bD05<aAf=cA77>eCf=cC11>&)
, 4)  (cC80>dC08<bD22<dC00<eEf!eCf=dE17>dD00<bB24<&)
, 5)  (aDt=eAf!cA89>bE75>aCf=^)
, 6)  (eEf=aAt!cA71>&)
, 7)  (bD75<cB91<bC09<dD01>cD72>dD05>&)
, 8)  (bC27<eAf=cB57<eBf!dA08<aBf=cC66>^)
, 9)  (dC04>cA72<cA39<&)
, 10)  (bA32<aDt!bE43<cB55<bE01<dE16>bC51>eEt=dA00>^)
, 11)  (bB52>aDt=aEt=&)
, 12)  (bA74<cB89>cE91<&)
, 13)  (eCt!bA69<^)
, 14)  (cC32<dE05>eEt=bC06>dD00<dB18>dE10>eBt=&)
, 15)  (aCt!dD00<aEt=dC11<bB70>bC92>dC13>eCf=^)
```

GP Settings

Max. Fitness (%): 99

Max. Generations: 300

GP Results

Best Rule

```
[(((dC03<cA75<dA39<&)((dC03<cA71<cA39<&)
(((dE01<cA79<cA39<&)((dE01<cA79<cA39<&)(
(cC06>cA72<cC09<&)((cC04>cA72<cA08<&)((c
C04>cC72<cA08<&)((dC03<cA78<cA39<^)((bC0
4>cA52<cA09<&)((dE01<cA79<cA39<&)((cC04>
```

Fitness Value (%): 98

Generation: 228

Scene Segmentation

# of Scenes: 4

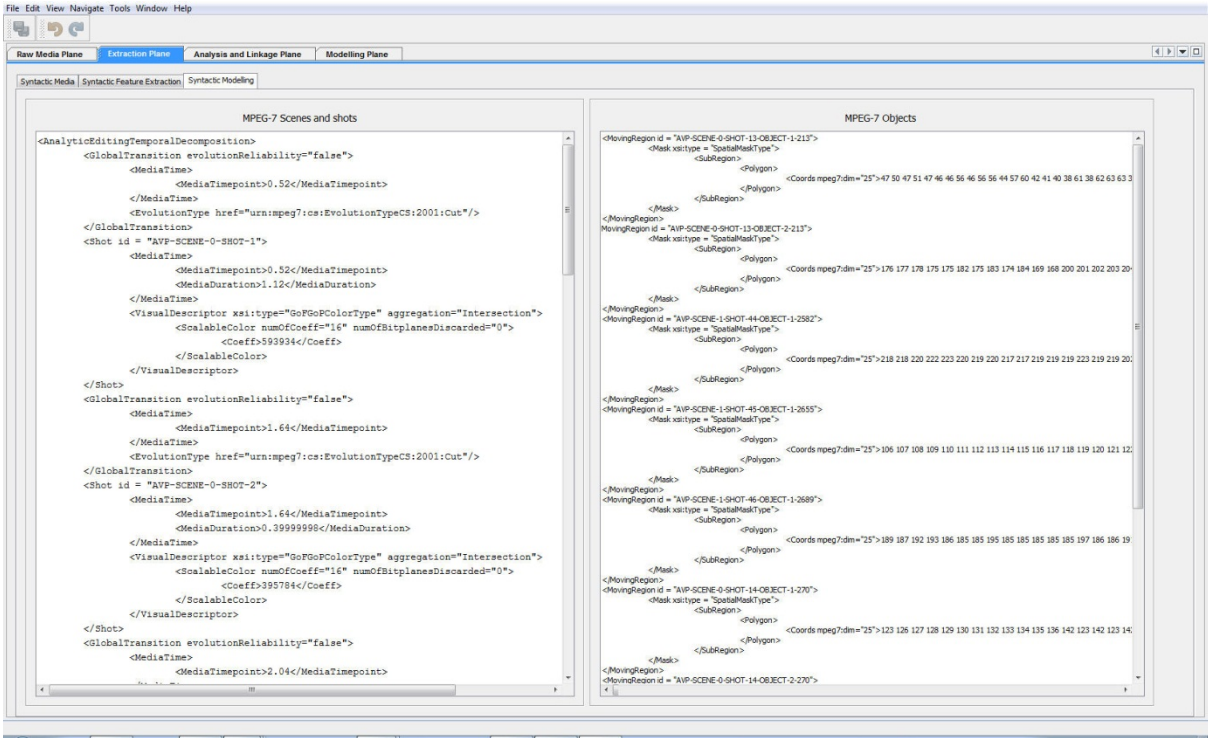
```
Input cut shots...done!
Input Transition shots...done!
Input Histograms...done!
Input Objects...done!
Processing...
Processing finished!

Results:

Scene 1: 260
Scene 2: 1786
Scene 3: 2489
Scene 4: 3455

Finished!
```

Figure 4.9: EXTRACTION PLANE – SCENE DETECTION SCREEN

Once started the process of finding the best rule is shown in the GP output panel. This shows the rules as they are generated and then shows the populations of new rules as they are evolved. Next to each rule is its fitness value as calculated from the fitness function. Once the best rule is found it is shown in the GP results section. Here we see the rule with its fitness value and the generation it was evolved in.

Once we have a best evolved rule that will identify scene breaks we are ready to begin the process of segmenting the target video clip into scenes. We use the start button in the scene segmentation panel to begin the segmentation process. We can see the process being applied in the preview text area of the scene segmentation panel. Once the scenes are identified the amount of scenes are shown in a text field with the actual start times of the scenes shown in the preview text area. These can be saved to a text file using the save button.

The final tab of the extraction plane, shown in Figure 4.9, is the MPEG-7 syntactic descriptions schemes layer that contains all the MPEG-7 descriptions of the scenes, shots and objects. We have two panels: one panel for the MPEG-7 scenes and shots, whilst the other has the DS' for the MPEG-7 objects within the scenes and shots. As the MPEG-7 description schemes are indented to xml syntax both panels are scrollable both vertically and horizontally so that they can be viewed in their completeness.



```
<AnalyticEditingTemporalDecomposition>
        <GlobalTransition evolutionReliability="false">
                <MediaTime>
                        <MediaTimepoint>0.52</MediaTimepoint>
                </MediaTime>
                <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
        </GlobalTransition>
        <Shot id = "AVP-SCENE-0-SHOT-1">
                <MediaTime>
                        <MediaTimepoint>0.52</MediaTimepoint>
                        <MediaDuration>1.12</MediaDuration>
                </MediaTime>
                <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Intersection">
                        <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
                                <Coeff>593934</Coeff>
                        </ScalableColor>
                </VisualDescriptor>
        </Shot>
        <GlobalTransition evolutionReliability="false">
                <MediaTime>
                        <MediaTimepoint>1.64</MediaTimepoint>
                </MediaTime>
                <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
        </GlobalTransition>
        <Shot id = "AVP-SCENE-0-SHOT-2">
                <MediaTime>
                        <MediaTimepoint>1.64</MediaTimepoint>
                        <MediaDuration>0.39999998</MediaDuration>
                </MediaTime>
                <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Intersection">
                        <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
                                <Coeff>395784</Coeff>
                        </ScalableColor>
                </VisualDescriptor>
        </Shot>
        <GlobalTransition evolutionReliability="false">
                <MediaTime>
                        <MediaTimepoint>2.04</MediaTimepoint>
```

```
<MovingRegion id = "AVP-SCENE-0-SHOT-13-OBJECT-1-213">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">47 50 47 51 47 46 46 56 46 56 56 44 57 60 42 41 40 38 61 38 62 63 63 3
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
MovingRegion id = "AVP-SCENE-0-SHOT-13-OBJECT-2-213">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">176 177 178 175 175 182 175 183 174 184 169 168 200 201 202 203 20
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-44-OBJECT-1-2582">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">218 218 220 222 223 220 219 220 217 217 219 219 219 223 219 219 20
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-45-OBJECT-1-2655">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 12
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-46-OBJECT-1-2689">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">189 187 192 193 186 185 185 195 185 185 185 185 185 197 186 186 19
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-0-SHOT-14-OBJECT-1-270">
        <Mask xsi:type = "SpatialMaskType">
                <SubRegion>
                        <Polygon>
                                <Coords mpeg7:dim="25">123 126 127 128 129 130 131 132 133 134 135 136 142 123 142 123 14
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-0-SHOT-14-OBJECT-2-270">
```

Figure 4.10: EXTRACTION PLANE – MPEG-7 LAYER SCREEN

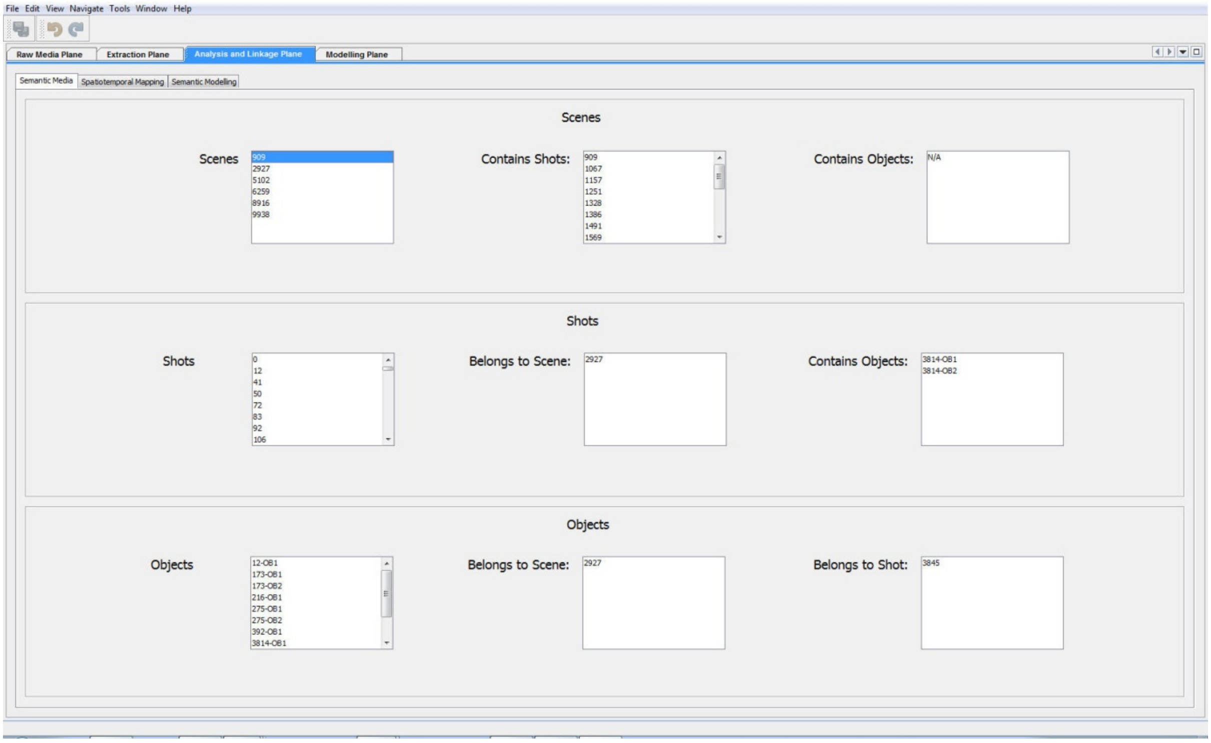
4.1.3 Analysis and linkage Plane

The first tab of the Analysis and linkage plane is the content pane which shows us the list of syntactic features that were extracted from the extraction plane. This screen allows us to analyse the implicit relationship between syntactic features that will be analysed for their semantic relationships.

In Figure 4.10 we can see that the content pane consists of three panels, one each for each of the syntactic features extracted: scenes, shots and objects. Each panel has three list boxes all of which represent the three syntactic features extracted. The first list box is selectable. If you select a feature instance from the first list box of any feature panel you will be shown the features it is related to in the other two feature list boxes within the panel.

For instance if we look at the shots panel we can see that a shot that starts with the frame number 3814 is selected. The next list box shows the scene (in this case scene starting with the frame number 2927). The list box after that shows what objects are contained with this shot.

The application layer of the analysis and linkage plane consists of two sub tabs. These two sub tabs are the spatial relationships and temporal relationships processing tabs.



Scenes

Scenes: 909, 2927, 5102, 6259, 8916, 9938

Contains Shots: 909, 1067, 1157, 1251, 1328, 1386, 1491, 1569

Contains Objects: N/A

Shots

Shots: 0, 12, 41, 50, 72, 83, 92, 106

Belongs to Scene: 2927

Contains Objects: 3814-OB1, 3814-OB2

Objects

Objects: 12-OB1, 173-OB1, 173-OB2, 216-OB1, 275-OB1, 275-OB2, 392-OB1, 3814-OB1

Belongs to Scene: 2927

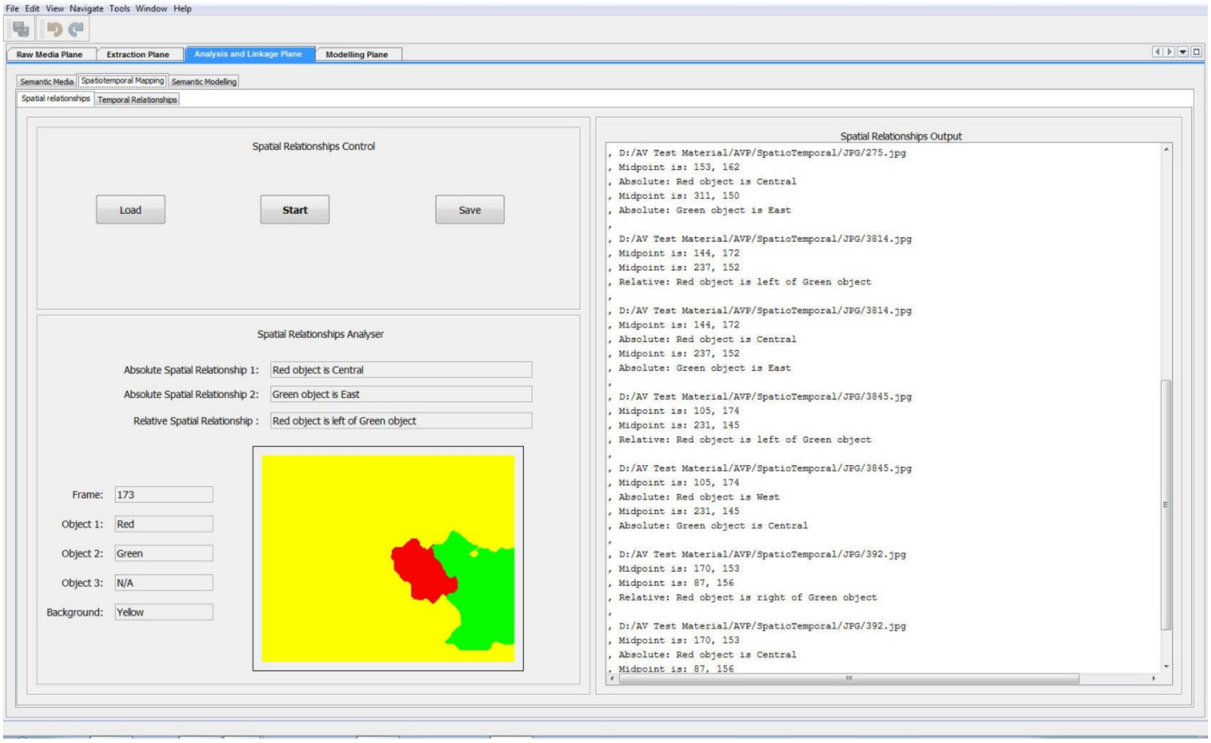
Belongs to Shot: 3845

Figure 4.11: ANALYSIS AND LINKAGE PLANE – CONTENT SCREEN

We begin with looking at the spatial relationship analyser tab. In Figure 4.11 we see that the spatial relationship tab is made of three panels: the spatial relationship's control panel, spatial relationship's analyser panel and the spatial relationship output panel.

The spatial relationship control panel has a start button that starts the process analysing the objects, finding the centroid of each objects and then calculating their absolute spatial relationship within the frame or, if two objects or more are present, then the relative spatial relationship between them. The save and load buttons on the control panel are for saving the results to a text file or loading from a text file that has results from an earlier experiment to be analysed.

The spatial relationship analyser has a drop down box that contains the frame numbers for all the shots that contain objects. By selecting a frame we show in preview panel to the right of the drop down box a colour coded object map that is produced during the object extraction phase. The text fields underneath the drop down box shows which object is of what colour. Above both of these are the absolute and relative spatial relationships of the objects depicted. This depends on the amount of objects. If there is only one object then there can only be an absolute relationship but for two or more objects there will be a suitable number of absolute and relative spatial relationships.



Absolute Spatial Relationship 1: Red object is Central

Absolute Spatial Relationship 2: Green object is East

Relative Spatial Relationship : Red object is left of Green object

Frame: 173

Object 1: Red

Object 2: Green

Object 3: N/A

Background: Yellow

Spatial Relationships Output

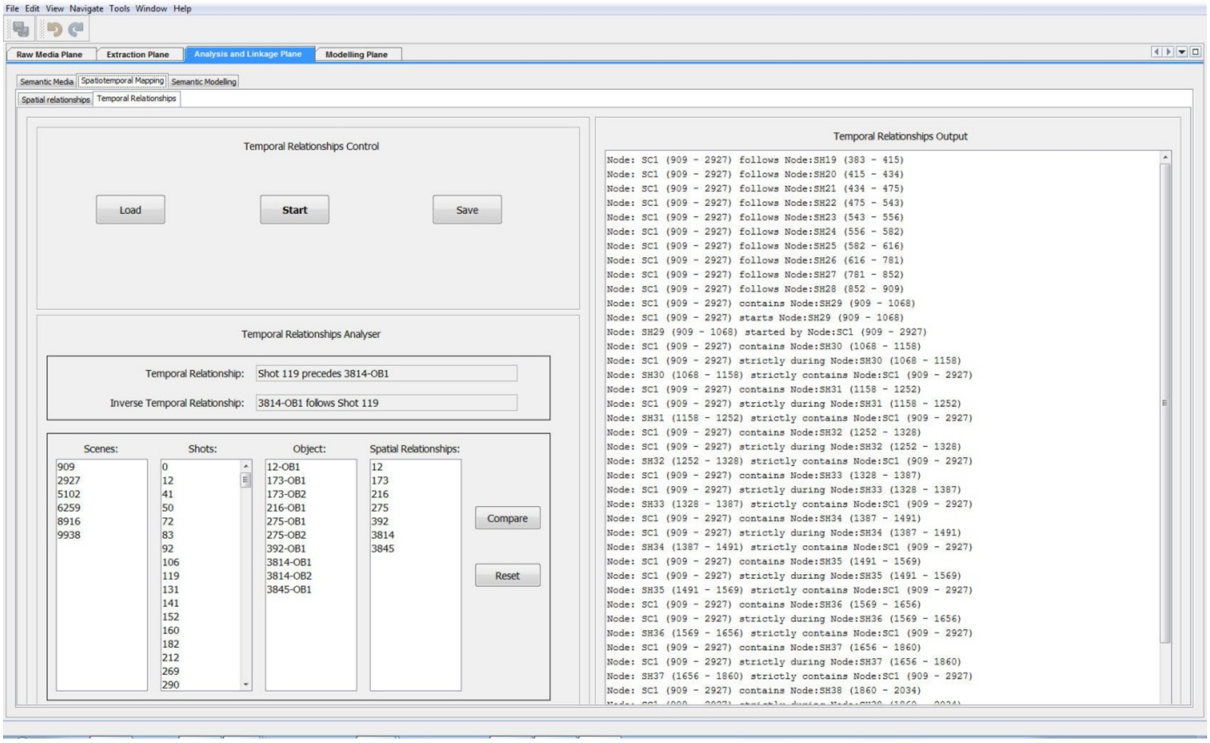
```
, D:/AV Test Material/AVP/SpatioTemporal/JPG/275.jpg
, Midpoint is: 153, 162
, Absolute: Red object is Central
, Midpoint is: 311, 150
, Absolute: Green object is East
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/3814.jpg
, Midpoint is: 144, 172
, Midpoint is: 237, 152
, Relative: Red object is left of Green object
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/3814.jpg
, Midpoint is: 144, 172
, Absolute: Red object is Central
, Midpoint is: 237, 152
, Absolute: Green object is East
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/3845.jpg
, Midpoint is: 105, 174
, Midpoint is: 231, 145
, Relative: Red object is left of Green object
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/3845.jpg
, Midpoint is: 105, 174
, Absolute: Red object is West
, Midpoint is: 231, 145
, Absolute: Green object is Central
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/392.jpg
, Midpoint is: 170, 153
, Midpoint is: 87, 156
, Relative: Red object is right of Green object
,
, D:/AV Test Material/AVP/SpatioTemporal/JPG/392.jpg
, Midpoint is: 170, 153
, Absolute: Red object is Central
, Midpoint is: 87, 156
```

Figure 4.12: ANALYSIS AND LINKAGE PLANE – APPLICATION LAYER (SPATIAL RELATIONSHIPS SCREEN)

In the spatial relationship output panel we can see the spatial relationships being processed for each object or set of objects in real time. This shows the key frame of the shot that contains the object(s), the centroid of the object(s) and the absolute, and if containing more than one object, the relative spatial relationships.

In Figure 4.12 we see the temporal relationship analyser tab that analyses the intra-temporal relationships between homogenous feature sets as well as the intra-temporal relationships between the heterogeneous feature sets. We see that the temporal relationship analyser tab is made of three panels: the temporal relationship's control panel, temporal relationship's analyser panel and the temporal relationship output panel.

The temporal relationship control panel has a start button that starts the process of analysing the temporal attributes of the syntactic and semantic features and finding all the temporal relationships between them. The save and load buttons on the control panel perform the same function as the save and control buttons on the spatial relationship analyser control panel.



Temporal Relationship: Shot 119 precedes 3814-OB1

Inverse Temporal Relationship: 3814-OB1 follows Shot 119

Scenes:
909
2927
5102
6259
8916
9938

Shots:
0
12
41
50
72
83
92
106
119
131
141
152
160
182
212
269
290

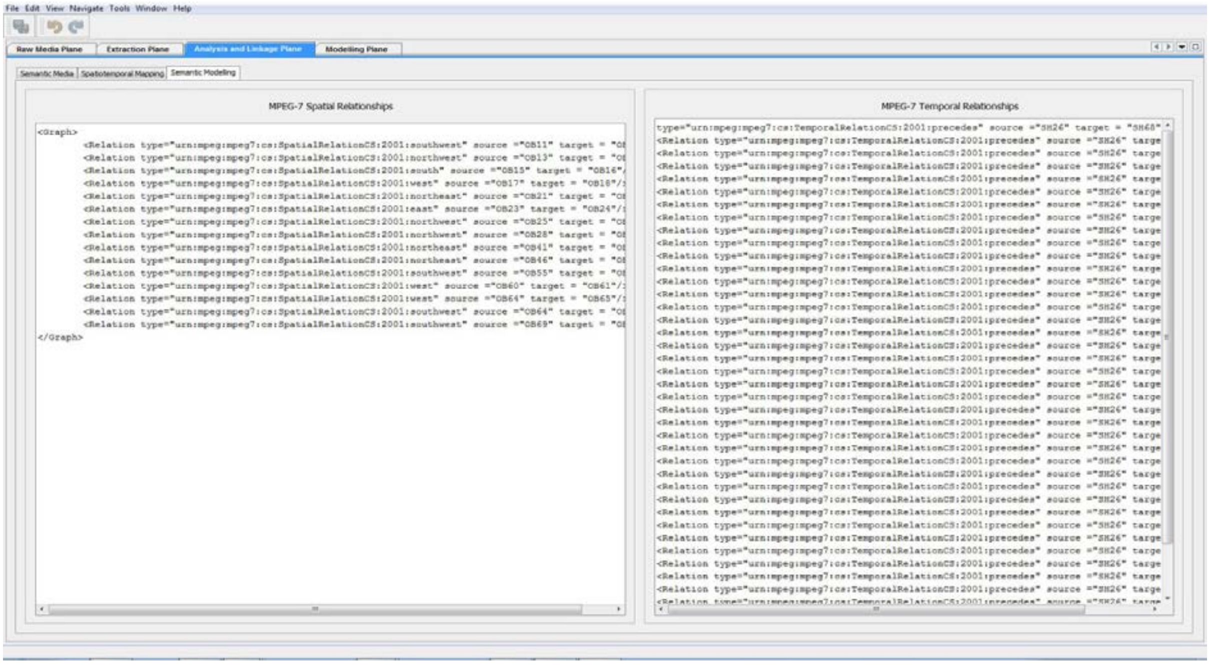
Object:
12-OB1
173-OB1
173-OB2
216-OB1
275-OB1
275-OB2
392-OB1
3814-OB1
3814-OB2
3845-OB1

Spatial Relationships:
12
173
216
275
392
3814
3845

```
Node: SC1 (909 - 2927) follows Node:SH19 (383 - 415)
Node: SC1 (909 - 2927) follows Node:SH20 (415 - 434)
Node: SC1 (909 - 2927) follows Node:SH21 (434 - 475)
Node: SC1 (909 - 2927) follows Node:SH22 (475 - 543)
Node: SC1 (909 - 2927) follows Node:SH23 (543 - 556)
Node: SC1 (909 - 2927) follows Node:SH24 (556 - 582)
Node: SC1 (909 - 2927) follows Node:SH25 (582 - 616)
Node: SC1 (909 - 2927) follows Node:SH26 (616 - 781)
Node: SC1 (909 - 2927) follows Node:SH27 (781 - 852)
Node: SC1 (909 - 2927) follows Node:SH28 (852 - 909)
Node: SC1 (909 - 2927) contains Node:SH29 (909 - 1068)
Node: SC1 (909 - 2927) starts Node:SH29 (909 - 1068)
Node: SH29 (909 - 1068) started by Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH30 (1068 - 1158)
Node: SC1 (909 - 2927) strictly during Node:SH30 (1068 - 1158)
Node: SH30 (1068 - 1158) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH31 (1158 - 1252)
Node: SC1 (909 - 2927) strictly during Node:SH31 (1158 - 1252)
Node: SH31 (1158 - 1252) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH32 (1252 - 1328)
Node: SC1 (909 - 2927) strictly during Node:SH32 (1252 - 1328)
Node: SH32 (1252 - 1328) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH33 (1328 - 1387)
Node: SC1 (909 - 2927) strictly during Node:SH33 (1328 - 1387)
Node: SH33 (1328 - 1387) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH34 (1387 - 1491)
Node: SC1 (909 - 2927) strictly during Node:SH34 (1387 - 1491)
Node: SH34 (1387 - 1491) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH35 (1491 - 1569)
Node: SC1 (909 - 2927) strictly during Node:SH35 (1491 - 1569)
Node: SH35 (1491 - 1569) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH36 (1569 - 1656)
Node: SC1 (909 - 2927) strictly during Node:SH36 (1569 - 1656)
Node: SH36 (1569 - 1656) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH37 (1656 - 1860)
Node: SC1 (909 - 2927) strictly during Node:SH37 (1656 - 1860)
Node: SH37 (1656 - 1860) strictly contains Node:SC1 (909 - 2927)
Node: SC1 (909 - 2927) contains Node:SH38 (1860 - 2034)
```

Figure 4.13: ANALYSIS AND LINKAGE PLANE – APPLICATION LAYER (TEMPORAL RELATIONSHIPS)

Once we have all the intra and inter temporal relationships we can view them in the temporal relationship analyser panel. Here we have four list boxes, each containing a different content feature set. Selecting two content features from any two boxes will show their temporal relationship as well as their inverse relationship.


File Edit View Navigate Tools Window Help
Raw Media Plane
Extraction Plane
Analysis and Linkage Plane
Modelling Plane
Semantic Media
Spatiotemporal Mapping
Semantic Modelling
MPEG-7 Spatial Relationships
MPEG-7 Temporal Relationships

```
<Graph>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB11" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB13" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:south" source ="OB15" target = "OB16"/
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB17" target = "OB18"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB21" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:east" source ="OB23" target = "OB24"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB25" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northwest" source ="OB28" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB41" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:northeast" source ="OB46" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB55" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB60" target = "OB61"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:west" source ="OB64" target = "OB65"/>
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB64" target = "OB
        <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:southwest" source ="OB69" target = "OB
</Graph>
```

```
type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" target = "SH68"
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
<Relation type="urn:mpeg:mpeg7:cs:TemporalRelationCS:2001:precedes" source ="SH26" targe
```

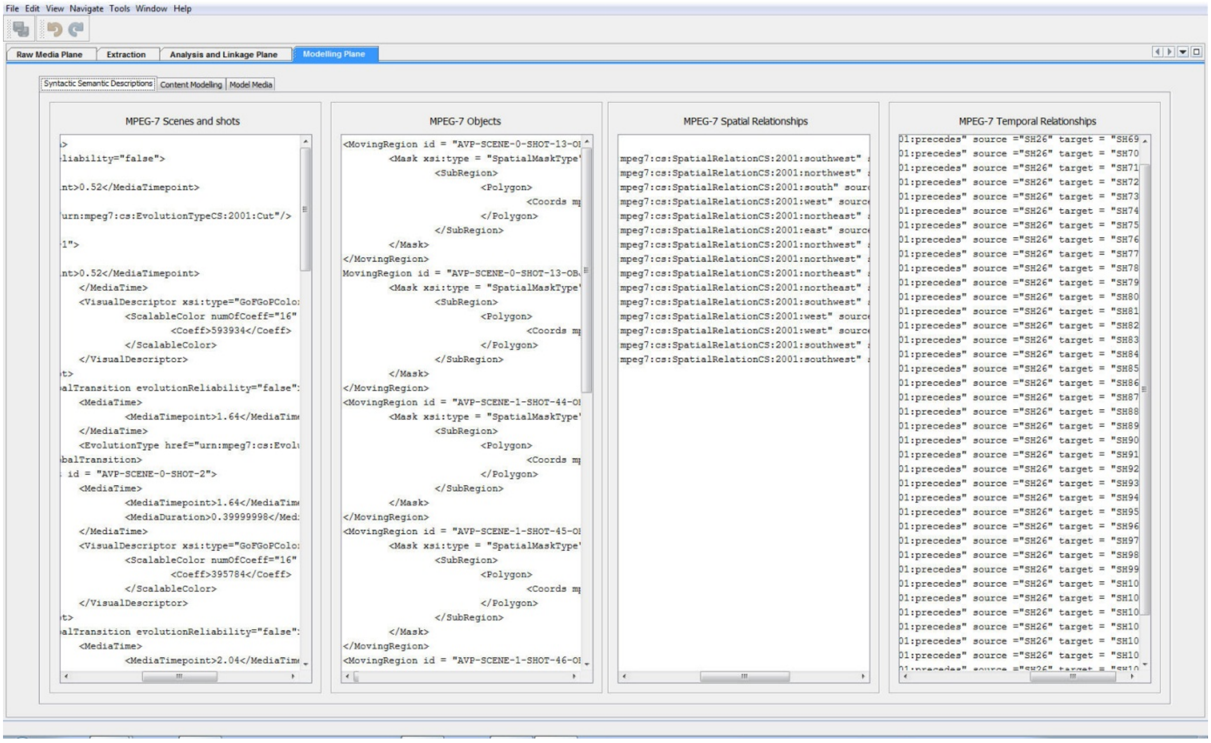
Figure 4.14: ANALYSIS AND LINKAGE PLANE – MPEG-7 LAYER SCREEN

In Figure 4.13 we see the final main tab of the analysis and linkage plane. We have the MPEG-7 semantic descriptions schemes layer that contains all the spatial and temporal relationships. We have two panels: one panel for the MPEG-7 spatial relationships and one containing the MPEG-7 temporal relationships. As before both panels are scrollable both vertically and horizontally so that they can be viewed in their completeness.

4.1.4 Modelling Plane

The content main tab presents us with all the separate MPEG-7 content features, syntactic content features that have been extracted and semantic content features that have been derived from those syntactic content features.

In Figure 4.14 we see the tab consists of four panels, two panels are for the syntactic features of scenes andshots, and objects and the other two are for the semantic features of the spatial and temporal relationships. The descriptions here are MPEG-7 compliant but these are still fragments that need to be modelled together.



File Edit View Navigate Tools Window Help
Raw Media Plane
Extraction
Analysis and Linkage Plane
Modelling Plane
Syntactic Semantic Descriptions
Content Modelling
Model Media
MPEG-7 Scenes and shots
MPEG-7 Objects
MPEG-7 Spatial Relationships
MPEG-7 Temporal Relationships

```
>
liability="false">

nt>0.52</MediaTimepoint>

urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>

1">

nt>0.52</MediaTimepoint>
   </MediaTime>
   <VisualDescriptor xsi:type="GoFGoPColo
           <ScalableColor numOfCoeff="16"
                   <Coeff>593934</Coeff>
           </ScalableColor>
   </VisualDescriptor>
t>
alTransition evolutionReliability="false"
   <MediaTime>
           <MediaTimepoint>1.64</MediaTim
   </MediaTime>
   <EvolutionType href="urn:mpeg7:cs:Evol
balTransition>
 id = "AVP-SCENE-0-SHOT-2">
   <MediaTime>
           <MediaTimepoint>1.64</MediaTim
           <MediaDuration>0.39999998</Med
   </MediaTime>
   <VisualDescriptor xsi:type="GoFGoPColo
           <ScalableColor numOfCoeff="16"
                   <Coeff>395784</Coeff>
           </ScalableColor>
   </VisualDescriptor>
t>
alTransition evolutionReliability="false"
   <MediaTime>
           <MediaTimepoint>2.04</MediaTim
```

```
<MovingRegion id = "AVP-SCENE-0-SHOT-13-OI
        <Mask xsi:type = "SpatialMaskType'
                <SubRegion>
                        <Polygon>
                                <Coords mp
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
MovingRegion id = "AVP-SCENE-0-SHOT-13-OB
        <Mask xsi:type = "SpatialMaskType'
                <SubRegion>
                        <Polygon>
                                <Coords mp
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-44-OI
        <Mask xsi:type = "SpatialMaskType'
                <SubRegion>
                        <Polygon>
                                <Coords mp
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-45-OI
        <Mask xsi:type = "SpatialMaskType'
                <SubRegion>
                        <Polygon>
                                <Coords mp
                        </Polygon>
                </SubRegion>
        </Mask>
</MovingRegion>
<MovingRegion id = "AVP-SCENE-1-SHOT-46-OI
```

```

mpeg7:cs:SpatialRelationCS:2001:southwest" 
mpeg7:cs:SpatialRelationCS:2001:northwest" 
mpeg7:cs:SpatialRelationCS:2001:south" sour
mpeg7:cs:SpatialRelationCS:2001:west" source
mpeg7:cs:SpatialRelationCS:2001:northeast" 
mpeg7:cs:SpatialRelationCS:2001:east" source
mpeg7:cs:SpatialRelationCS:2001:northwest" 
mpeg7:cs:SpatialRelationCS:2001:northwest" 
mpeg7:cs:SpatialRelationCS:2001:northeast" 
mpeg7:cs:SpatialRelationCS:2001:northeast" 
mpeg7:cs:SpatialRelationCS:2001:southwest" 
mpeg7:cs:SpatialRelationCS:2001:west" source
mpeg7:cs:SpatialRelationCS:2001:west" source
mpeg7:cs:SpatialRelationCS:2001:southwest" 
mpeg7:cs:SpatialRelationCS:2001:southwest" 
```

```
01:precedes" source ="SH26" target = "SH69
01:precedes" source ="SH26" target = "SH70
01:precedes" source ="SH26" target = "SH71
01:precedes" source ="SH26" target = "SH72
01:precedes" source ="SH26" target = "SH73
01:precedes" source ="SH26" target = "SH74
01:precedes" source ="SH26" target = "SH75
01:precedes" source ="SH26" target = "SH76
01:precedes" source ="SH26" target = "SH77
01:precedes" source ="SH26" target = "SH78
01:precedes" source ="SH26" target = "SH79
01:precedes" source ="SH26" target = "SH80
01:precedes" source ="SH26" target = "SH81
01:precedes" source ="SH26" target = "SH82
01:precedes" source ="SH26" target = "SH83
01:precedes" source ="SH26" target = "SH84
01:precedes" source ="SH26" target = "SH85
01:precedes" source ="SH26" target = "SH86
01:precedes" source ="SH26" target = "SH87
01:precedes" source ="SH26" target = "SH88
01:precedes" source ="SH26" target = "SH89
01:precedes" source ="SH26" target = "SH90
01:precedes" source ="SH26" target = "SH91
01:precedes" source ="SH26" target = "SH92
01:precedes" source ="SH26" target = "SH93
01:precedes" source ="SH26" target = "SH94
01:precedes" source ="SH26" target = "SH95
01:precedes" source ="SH26" target = "SH96
01:precedes" source ="SH26" target = "SH97
01:precedes" source ="SH26" target = "SH98
01:precedes" source ="SH26" target = "SH99
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
01:precedes" source ="SH26" target = "SH10
```

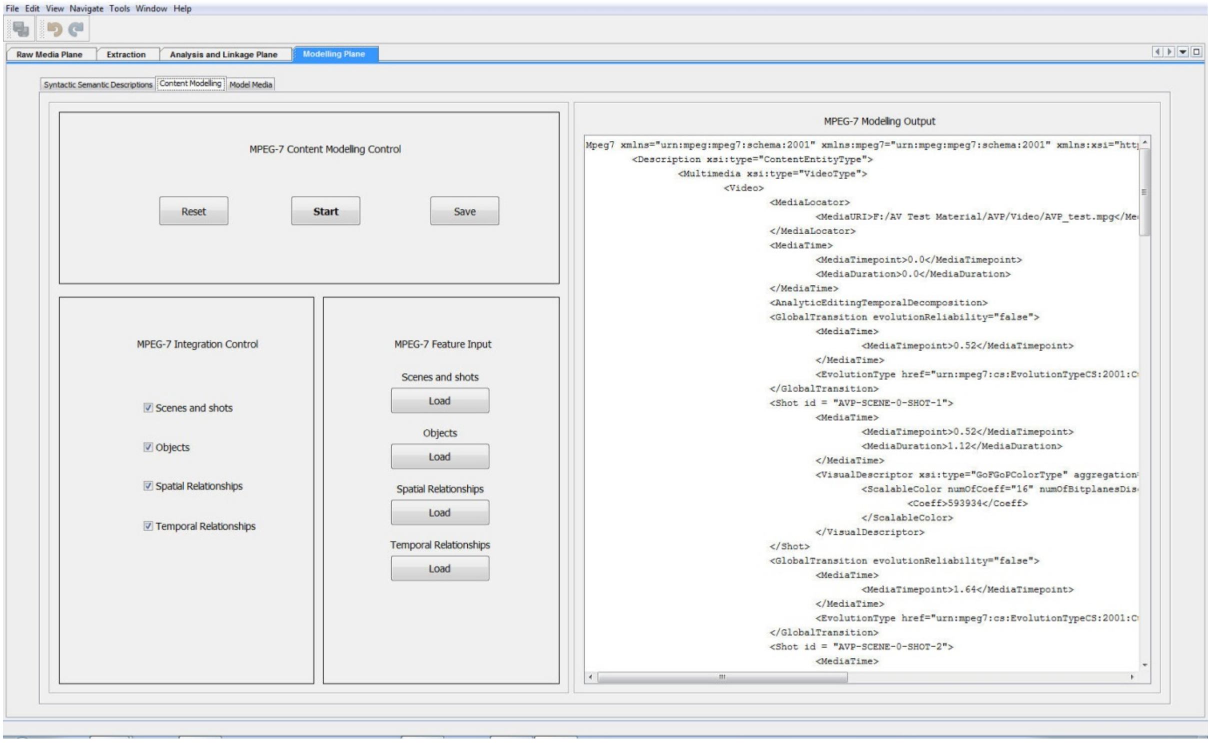
Figure 4.15: MODELLING PLANE – CONTENT LAYER SCREEN

The application tab of the modelling plane is the integration screen used for combining all the syntactic and semantic content features into one coherent MPEG-7 document that can be parsed by any MPEG-7 compliant consumer application.

In Figure 4.15 we can see that the application tab consists of four panels: the MPEG-7 content modelling control panel, the MPEG-7 feature selector panel, the MPEG-7 feature input/output panel and the MPEG-7 modelling output panel. The first three panels are all used together to provide settings, inputs and outputs to produce the final content model.

The MPEG-7 feature input/output panel allows the user to input previously saved MPEG-7 content features from earlier experiments from MPEG-7 files. It also allows the user to save the MPEG-7 content features to MPEG-7 files that will include in the file name the feature as well as a time stamp indicating when they were created. This allows the user to return and integrate a new feature in a different way.

The MPEG-7 feature selector panel allows the user to select which feature they would like to integrate into the complete MPEG-7 content model. The panel has four check boxes that represent the four feature sets displayed in the main content tab. By checking these boxes they will be included in the complete MPEG-7 content model. A fifth check box toggles the input from the present experiment to the input from the MPEG-7 feature input/output panel instead. When this is checked the other input check boxes are disabled as all the input must come from the feature input/output panel.

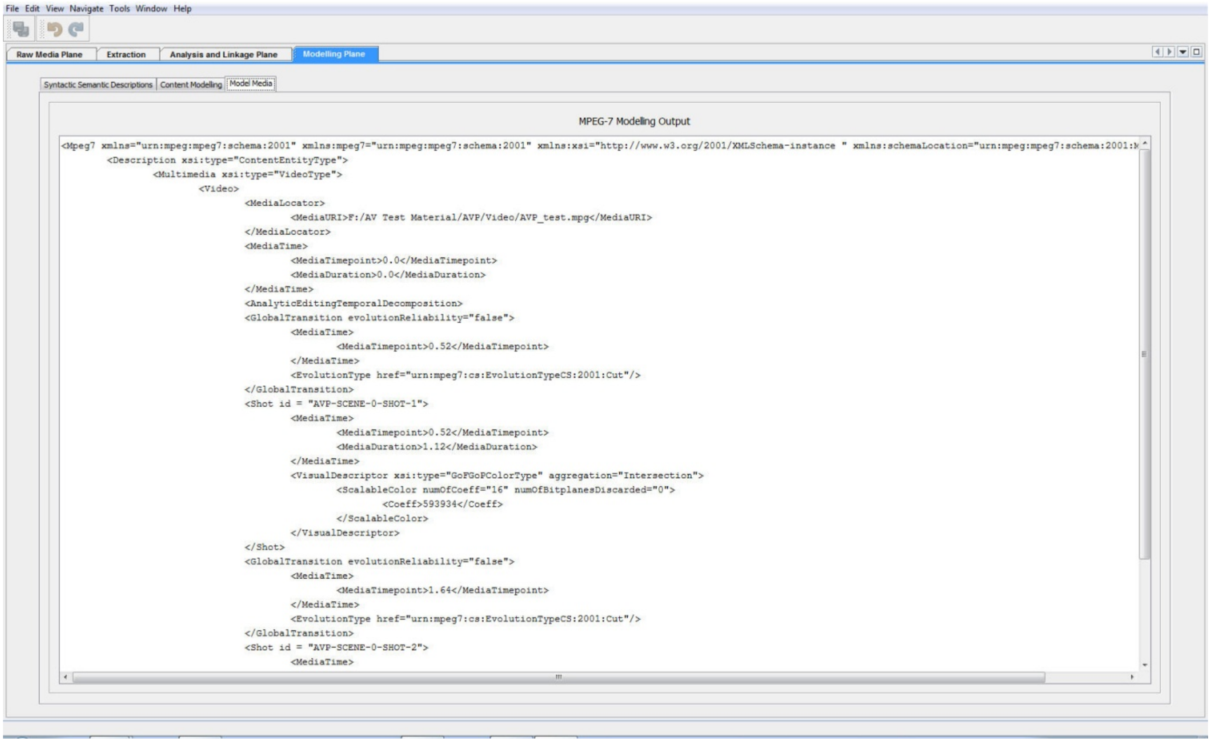


```
Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="htt
        <Description xsi:type="ContentEntityType">
                <Multimedia xsi:type="VideoType">
                        <Video>
                                <MediaLocator>
                                        <MediaURI>F:/AV Test Material/AVP/Video/AVP_test.mpg</Me
                                </MediaLocator>
                                <MediaTime>
                                        <MediaTimepoint>0.0</MediaTimepoint>
                                        <MediaDuration>0.0</MediaDuration>
                                </MediaTime>
                                <AnalyticEditingTemporalDecomposition>
                                <GlobalTransition evolutionReliability="false">
                                        <MediaTime>
                                                <MediaTimepoint>0.52</MediaTimepoint>
                                        </MediaTime>
                                        <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:C
                                </GlobalTransition>
                                <Shot id = "AVP-SCENE-0-SHOT-1">
                                        <MediaTime>
                                                <MediaTimepoint>0.52</MediaTimepoint>
                                                <MediaDuration>1.12</MediaDuration>
                                        </MediaTime>
                                        <VisualDescriptor xsi:type="GoFGoPColorType" aggregation
                                                <ScalableColor numOfCoeff="16" numOfBitplanesDis
                                                        <Coeff>593934</Coeff>
                                                </ScalableColor>
                                        </VisualDescriptor>
                                </Shot>
                                <GlobalTransition evolutionReliability="false">
                                        <MediaTime>
                                                <MediaTimepoint>1.64</MediaTimepoint>
                                        </MediaTime>
                                        <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:C
                                </GlobalTransition>
                                <Shot id = "AVP-SCENE-0-SHOT-2">
                                        <MediaTime>
```

Figure 4.16: MODELLING PLANE – APPLICATION LAYER SCREEN

The MPEG-7 content modelling control panel has three buttons: start, reset and save. Once all features to be included in the complete MPEG-7 content model have been selected the start button is pressed to begin the modelling integration process. The reset button is used to reset all the inputs back to an unselected state so that a new set of features can be input and modelled. The save button saves the completed MPEG-7 content model to an MPEG-7 file with the name of the original media clip and a time stamp for date of creation.

In Figure 4.16 we can see the final tab presents us with the final MPEG-7 Layer of the modelling plane. Here we can view the complete MPEG-7 content model. Here all the features that were selected and processed in the application tab can be seen linked together to produce a content model that is rich and granular in description and is personalised to the users requirements.


File Edit View Navigate Tools Window Help
Raw Media Plane
Extraction
Analysis and Linkage Plane
Modelling Plane
Syntactic Semantic Descriptions
Content Modelling
Model Media
MPEG-7 Modeling Output

```
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001" xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance " xmlns:schemaLocation="urn:mpeg:mpeg7:schema:2001:
        <Description xsi:type="ContentEntityType">
                <Multimedia xsi:type="VideoType">
                        <Video>
                                <MediaLocator>
                                        <MediaURI>F:/AV Test Material/AVP/Video/AVP_test.mpg</MediaURI>
                                </MediaLocator>
                                <MediaTime>
                                        <MediaTimepoint>0.0</MediaTimepoint>
                                        <MediaDuration>0.0</MediaDuration>
                                </MediaTime>
                                <AnalyticEditingTemporalDecomposition>
                                <GlobalTransition evolutionReliability="false">
                                        <MediaTime>
                                                <MediaTimepoint>0.52</MediaTimepoint>
                                        </MediaTime>
                                        <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
                                </GlobalTransition>
                                <Shot id = "AVP-SCENE-0-SHOT-1">
                                        <MediaTime>
                                                <MediaTimepoint>0.52</MediaTimepoint>
                                                <MediaDuration>1.12</MediaDuration>
                                        </MediaTime>
                                        <VisualDescriptor xsi:type="GoFGoPColorType" aggregation="Intersection">
                                                <ScalableColor numOfCoeff="16" numOfBitplanesDiscarded="0">
                                                        <Coeff>593934</Coeff>
                                                </ScalableColor>
                                        </VisualDescriptor>
                                </Shot>
                                <GlobalTransition evolutionReliability="false">
                                        <MediaTime>
                                                <MediaTimepoint>1.64</MediaTimepoint>
                                        </MediaTime>
                                        <EvolutionType href="urn:mpeg7:cs:EvolutionTypeCS:2001:Cut"/>
                                </GlobalTransition>
                                <Shot id = "AVP-SCENE-0-SHOT-2">
                                        <MediaTime>
```

Figure 4.17: Modelling Plane – MPEG-7 Layer screen

4.2 Performance Evaluation

The performance evaluation is arranged in two sections. The first describes the testbed and performance benchmarks to be used in evaluating MAC-REALM. The second section presents and discusses the results gathered.

4.2.1 Testbed

In order to answer the questions above, we ran tests for different clips of Alien vs. Predator (AVP) (W.S. Anderson, 2004) through MAC-REALM. We then looked at how accurately MAC-

REALM extracted the syntactic features from the content. AVP was used because it has objects on screen that are invisible. This provides for the hardest test footage to test MAC-REALM with.

The test data consisted of four clips of AVP. The video was digitised in MPEG-1 format at a frame rate of 25 fps (total of ~720,000 frames) and a resolution of 352*288 pixels (commonly known as the CIF standard (Richardson, 2010)). This was accomplished using a Pentium PC and Adobe Premiere Pro. For ease of manipulation, and to keep file sizes manageable, the video was cut and digitised into 4 segments of 20 minutes each.

To provide an authoritative guide to the test set, the locations and types of shot, scene, and program boundaries were manually analysed to give a series of detailed log files, each representing a 20-minute video segment. This collection of log files is referred to as the *groundtruth*, and represents a time consuming process that requires manual processing. The groundtruth allows us to compare the results generated by our detection algorithms to a ground truth. It also enables us to calculate statistics such as the number of frames and shot boundaries found in each content type. As noted above, the groundtruth contains extremely detailed semantic information.

All of the tests conducted with MAC-REALM were carried out using Windows 7 Professional operating system and a standard PC, consisting of the following hardware:

- Intel i7 3.4 GHz CPU (with Hyper Threading)
- 16 GB DDR2 2400MHz DDR3-RAM (Tested Latency: 10-12-12-31)
- 120GB SSD Primary hard drive (550 MB/s read, 510 MB/s write)
- 2TB secondary hard drive @ 5400rpm

4.2.2 Benchmark Tests

Benchmark testing methodology is employed to evaluate MAC-REALM with regards to the objectives outlined at the start of section 4.2, uses standard benchmark techniques when possible. When there are no standard tests available, tests have been devised using other common techniques used by others in related research areas.

4.2.2.1 Computational expense and improving accuracy of extraction

The raw media plane pre-processes the media with two objectives in mind: To reduce computational expense, and to filter the media to improve the extraction of syntactic features. The computational expense is evaluated first by using a mathematical proof that is explained in chapter 3. The evaluation of the extraction of syntactic features is examined by testing the filtering technique used for the MAC-REALM pre-processing method.

MAC-REALM reduces the amount of redundant data by using one frame per second of the video for feature extraction. Using only one frame and discarding the rest, reduces the processing by a factor equal to the frame rate. From the equation in chapter three a new formula is derived to calculate the computational reduction in processing. Eq. (4.1) shows the reduction in processing is directly proportional to the reduction in frame rate..

Eq. (4.1) $$Computational\ reduction = \frac{1}{fps}$$

where fps is the frames per second of the video frame rate. From this mathematical proof we can see that the computational expense of processing per frame is reduced by a factor that is equal to the fps of the video compared to the case where the full complement of frames is used.

The media is filtered in two different ways: 1) the colour space is converted into RGB, if not already in that colour space, and 2) noise is removed from the image by using morphological noise removal. The colour space is chosen as it gives the best trade-off between performance and processing load. This has been proved in the seminal paper by (Koprinska & Carrato, 2001) which thoroughly surveyed temporal segmentation techniques. Three global colour histogram based methods for temporal segmentation were tested using six different colour spaces: $\boldsymbol{RGB}$, $\boldsymbol{HSV}$, $\boldsymbol{YIQ}$, L*a*b*,L*u*v* and Munsell (Westland, Laycock, Cheung, Henry, & Mahyar, 2012). The $\boldsymbol{RGB}$ histogram of a frame is computed as three sets of 256 bins. The other five histograms are represented as a 2-dimensional distribution over the two non-intensity based dimensions of the colour spaces, namely: $\boldsymbol{H}$ and $\boldsymbol{S}$ for the $\boldsymbol{HSV}$, $\boldsymbol{I}$ and $\boldsymbol{Q}$ for the $\boldsymbol{YIQ}$, a* and b* for the L*a*b*,u* and v* for the L*u*v* and hue and chroma components for the Munsell space. The number of bins is 1600 (40×40) for the L*a*b*,L*u*v* and $\boldsymbol{YIQ}$ histograms and 1800 (60 hues×30 saturations/chromas) for the $\boldsymbol{HSV}$ and Munsell space histograms.

The study found that in terms of overall classification accuracy $\boldsymbol{YIQ}$, L*a*b* and Munsell colour coordinate spaces performed well, followed by HSV, L*u*v* and $\boldsymbol{RGB}$. In terms of computational cost of conversion from $\boldsymbol{RGB}$, the $\boldsymbol{HSV}$ and $\boldsymbol{YIQ}$ are the least expensive, followed by L*a*b*, L*u*v* and the Munsell space. Seeing that $\boldsymbol{RGB}$ and $\boldsymbol{HSV}$ had similar computational expense and had good performance they were chosen as the best all round choice. The computational expense is a factor as the size of the testbed, which is a desktop PC, has limited processing power compared to workstation or batch server configurations. RGB is settled upon out of the two as most video clips are in that format and would require no conversion. HSV has a small advantage over RGB in spatiotemporal segmentation but this is due to its explicit handling of shadow and illumination changes. The morphological noise removal and flattening negates this advantage by eliminating the luminance variations and improving the chromaticity of the image.

The accuracy and detection rates evaluated for this method are in the shot, object and scene segmentation results.

4.2.2.2 Shot Boundary

To evaluate the shot boundary technique we used the evaluation tool from TRECVid[10]. This is a popular benchmark test for many shot boundary algorithms that have been tested by the TRECVid community (Smeaton et al., 2010). This is a suitable benchmark to test the performance of MAC-REALM's Shot Boundary Detection algorithm.

The tool allows evaluation of detected scene boundaries against a groundtruth of any video clip. The results are split into three sections, total number of cuts detected, number of abrupt transitions and number of gradual transitions. For each shot the start frame of the transition is given followed by the end frame of the transition. For abrupt transitions the start and end frame numbers are consecutive and relate to the last pre-transition and first post-transition frames so that it has an effective length of two frames (rather than zero). For the gradual transitions the start frame would be the beginning of the transition and the end frame number would be the number the transition completed. A single frame overlap between the detected transitions and the reference transition was all that was required being the only detection criteria as this made the detection independent of the accuracy of the detected boundaries. Short gradual transitions (of less than 1 to 5 frames) are considered as abrupt cuts. In the ground truth samples the short gradual transitions (SGT) were expanded by 5 frames in each direction before matching against submitted transitions to accommodate differences in frame numbering from the SBD sample. The reason for this is that the number SGT's has increased over the years to become a substantial percentage of the shot transition count, and the majority of them are 1 frame long (Table 4.0).

[10] http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/

	2003	2004	2005
% of all transitions	2	10	14
% of all graduals	7	24	35
% of SGT's = 1 frame	41	88	83

Table 4.1: SHORT GRADUAL TRANSITIONS(SMEATON ET AL., 2010)

To classify the base metrics for the MAC-REALM SBD algorithm we use recall and precision (Han, Kamber, & Pei, 2011) to evaluate MAC-REALM's performance. Precision is the proportion of correct shot boundaries identified by MAC-REALM to the total number of shot boundaries identified by MAC-REALM. Precision is defined as:

Eq. (4.2) $$Precision = \frac{TP}{TP + FP}$$

where TP is 'true positives' or shots that have been correctly identified and FP is 'false positives' or where a shot boundary has been identified but is incorrect.

Recall is the proportion of shot boundaries correctly identified by MAC-REALM to the total number of shot boundaries present. Recall is expressed as:

Eq. (4.3) $$Recall = \frac{TP}{TP + FN} = \frac{TP}{P}$$

where TP is the number of shots correctly identified by MAC-REALM and FN is the number of 'false negatives' or shot boundaries that were missed. $TP + FN$ is equal to P which is the total number of groundtruth shots.

Both precision and recall are taken into account to provide an overall score for the effectiveness and efficiency of the shot extraction technique. This measure is called the $f1$ score and is defined as:

Eq. (4.4) $$f1 = \frac{2\,.precision.recall}{precison + recall}$$

Ideally, the $f1$ score should equal 1. This would indicate that we have identified all existing shot boundaries correctly, without identifying any false boundaries.

4.2.2.3 Object Detection

To evaluate the object detection (OD) we use two different benchmark tests, one for the segmentation of the object and another for the tracking of the object. The reason for this is because there is no specific evaluation test for the combined criteria of object detection and tracking. The object segmentation metric is from (Feng, Song, & Tiecheng, 2006) and the object tracking is from metrics based on the performance evaluation criteria outlined in (Bashir & Porikli, 2006).

The metric on the accuracy of the object segmentation uses the figure ground assumption. The object segmentation can split the shot into only two defining regions, foreground and background. The foreground may contain one or more objects. The first frame of the shot is taken as the segmentation image to be evaluated. The same frame is then manually segmented using Photoshop CS6[11] and used as the groundtruth image. The performance of the segmentation is evaluated on the degree of overlap between the segmented image S and the groundtruth image G. The following formula measures the accuracy of the intersection between the two images:

Eq. (4.5)
$$P(S\backslash G) = \frac{|G \cap S|}{|G \cup S|} = \frac{|G \cap S|}{|G| + |S| - |G \cap S|'}$$

This measure has no bias to the segmentations that produces overly large or small number of segments. The numerator $|G \cap S|$, measures how much the ground-truth structure is detected. The denominator, $|G \cup S|$, is a normalisation factor of the accuracy measure to the range of [0, 1]. With this normalisation factor, the accuracy measure penalizes the error of detecting irrelevant regions as the foreground segments (false positives). It is easy to see that this region-based measure is insensitive to small variations in the ground-truth construction and incorporates the accuracy and recall measurement into one unified function.

The benchmark test for object tracking was originally meant for evaluating the performance of video surveillance systems. Whereas they used video of pseudo synthetic environments as the test data, MAC-REALM will be using the clips from “Alien vs. Predator”. This should not affect the experiment as the video of the pseudo synthetic environments used provided a controlled environment were the complexity could be controlled to mimic different scenarios. No such

[11] http://www.adobe.com/mena_en/products/photoshop.html

control is needed here as the generic nature and purpose of MAC-REALM means that the environment the test should be carried out in should not be controlled.

To evaluate the tracking of the object in MAC-REALM the Object Tracking error must be measured. The error in tracking is calculated by the average deviation from the centroid of the object segmented by MAC-REALM to the centroid of the groundtruth object. This is given by:

Eq. (4.6) $$Object\ Tracking\ Error\ (OTE) = \frac{1}{N_{rg}} \sum_{i \in g\ (t_i) \wedge r(t_i)} \sqrt{\left(x_i^g - x_i^s\right)} - \left(y_i^g - y_i^s\right)$$

where N_{rg} represents the total number of overlapping frames between ground truth and system results, x_i^g, y_i^g represents the coordinates of the centroid of object in the i^{th} frame of ground truth whilst x_i^s, y_i^s represents the coordinates of the centroid of object in the i^{th} frame of MAC-REALM segmentation.

4.2.2.4 Scene Detection

To evaluate the performance of the scene boundary detection (SD) algorithm of MAC-REALM, the metrics must measure the effectiveness of locating the boundaries of all the scenes compared to the groundtruth of scene boundaries. The MAC-REALM algorithm already implicitly formulates a metric for this purpose, the fitness function. It is used by the GP algorithm to select the rules that are the most accurate at identifying scene boundaries. This though is only theoretical and needs to be evaluated to see how accurate the prediction will be when the rule is actually used to segment the video stream.

To achieve this we use the same metrics as used for the SBD evaluation, recall, precision and F1 score. This is because structurally they are physically similar and the metrics only measure the physical attributes of the feature. This choice of metrics provides a mechanism for validating the accuracy of the selected rule, and therefore the fitness function.

4.2.2.5 Spatial relationships

There are no standard benchmark tests for evaluating spatial relationships between objects. Most spatial relationships are formulated as a post-process feature that is not part of the content modelling authoring tool, and where they have been explicitly stated they have been manually created and, therefore, an evaluation of captured relationships is redundant. The problem of spatial relationships is the ambiguity of the direction of the spatial relationship as it can be calculated with a different formula depending on where measurements are taken from. MAC-REALM uses the centroid method that gives the most natural orientation of the objects from a human perspective.

To evaluate the accuracy of the formulation of MAC-REALM's derived spatial relationships we use the precision metric. Precision gives us the metric of how correct the position of the derived spatial relationship is. This metric allows us to analyse the quality of the spatial relationships. Recall, which would give us the quantity of spatial relationships, is not required as we know this will depend on how many objects are identified and segmented by the OBD algorithm.

To produce a groundtruth for the experiment we manually define both absolute, and where applicable, relative spatial relationships. The spatial relationships are stated by the user as giving the most obvious spatial relationship from their perspective. To evaluate the precision of relative (SR_R), absolute (SR_A) and total number (SR_T) of spatial relationships we use the formulas:

Eq. (4.7)
$$SR_R = \frac{SR_r}{SG_R}$$

Eq. (4.8)
$$SR_A = \frac{SR_a}{SG_A}$$

Eq. (4.9)
$$SR_T = \frac{SR_t}{SG_T} = \frac{SR_r}{SG_R} + \frac{SR_a}{SG_A}$$

where SR_r, SR_a and SR_t are the derived relative, absolute and total number of spatial relationships spatial relationships from MAC-REALM, respectively, and SG_r, SG_a and SG_t is the groundtruth for the respective relative, absolute and total number of spatial relationships.

4.2.2.6 Temporal relationships

As is the case when benchmarking spatial relationships, there are no standard benchmark tests available for temporal relationships. Typically, such benchmarks are either manually created, or are calculated post process to the creation of the content features extraction. However, unlike spatial relationships there is no ambiguity in the interpretation of the temporal relationships, as content features all have a definitive start and end point. Thus, making precision reliable for all temporal relationships.

The evaluation is therefore not an empirical evaluation of the retrieval of the temporal relationships, but of the data collected. Examining the data collected we look at the types of feature and quantity of relationships for each feature. This evaluates the trends in the complexities of describing features temporally and the temporal richness of each feature. The number of each type of relationship is evaluated to examine trends and infer reasons for the proportions of the features.

4.2.2.7 Content modelling

The content model once assembled from all the constituent content features must be able to be validated as an MPEG-7 compliant document. As the purpose of the content model is to be universally accessible to all the MPEG-7 compliant applications the content model must be validated against the MPEG-7 Schema.

To validate the content model an MPEG-7 Validator tool called VAMP (Troncy, Bailer, Höffernig, & Hausenblas, 2010) is used to validate the content model. This tool can validate MPEG-7 files to many different profiles, so is ideal at checking compatibility for different versions of MPEG-7. The tool comes with a downloadable client tool for uploading and testing local files on the VAMP servers.

4.2.3 Results

All results for the tests were carried out using the testbed computer and using the benchmark tests explained in section 4.2.2. The results for each benchmark test were run three times and the median average taken for all three runs used and presented here. Where a benchmark test uses need input from another process that has been part of another benchmark test, the results that are presented are used for the input into the new benchmark test.

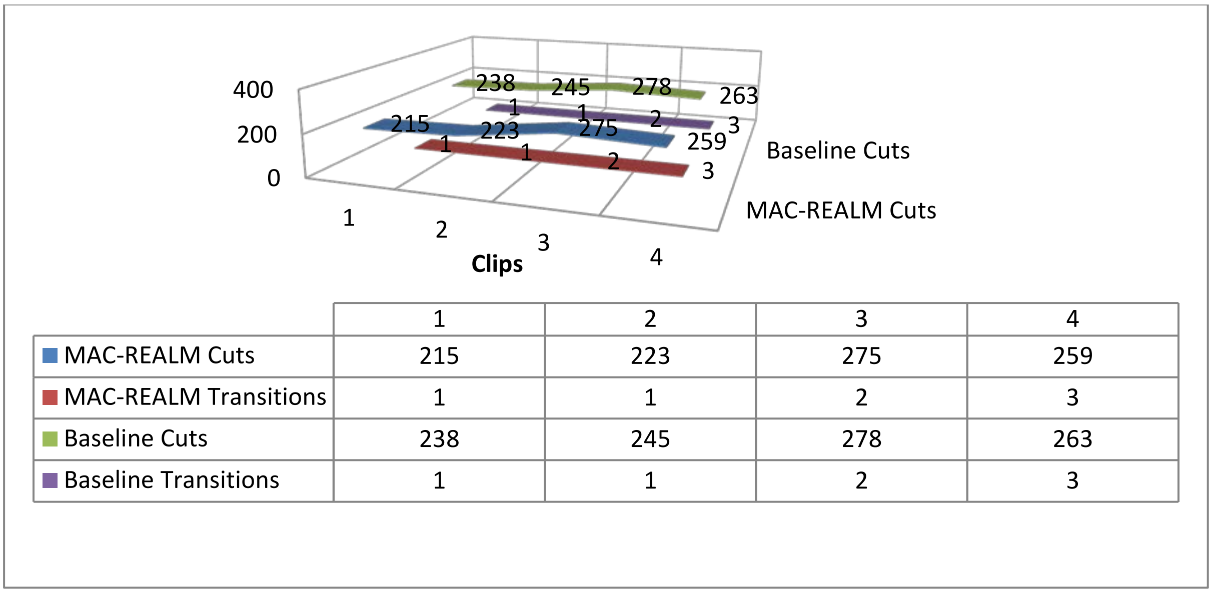
4.2.3.1 Shot Boundary detection

Before beginning the experiments, the segmentation algorithm was tuned on a number of small (< 10 minute) video segments extracted from the test set. These training runs enabled fine-tuning of the adaptive threshold levels for each clip.

The experiment was conducted with the four sample clips, and the results are depicted in Figure 4.17, alongside the number of groundtruth shots. Detection rates are provided separately for both cut and gradual transition shots.

All the gradual transition shots were detected with a 100% recall. The cut shots had variable rates of detection that were: clip 1 = 90.34%, clip 2 = 91.34%, clip 3 = 98.92% and clip 4 =

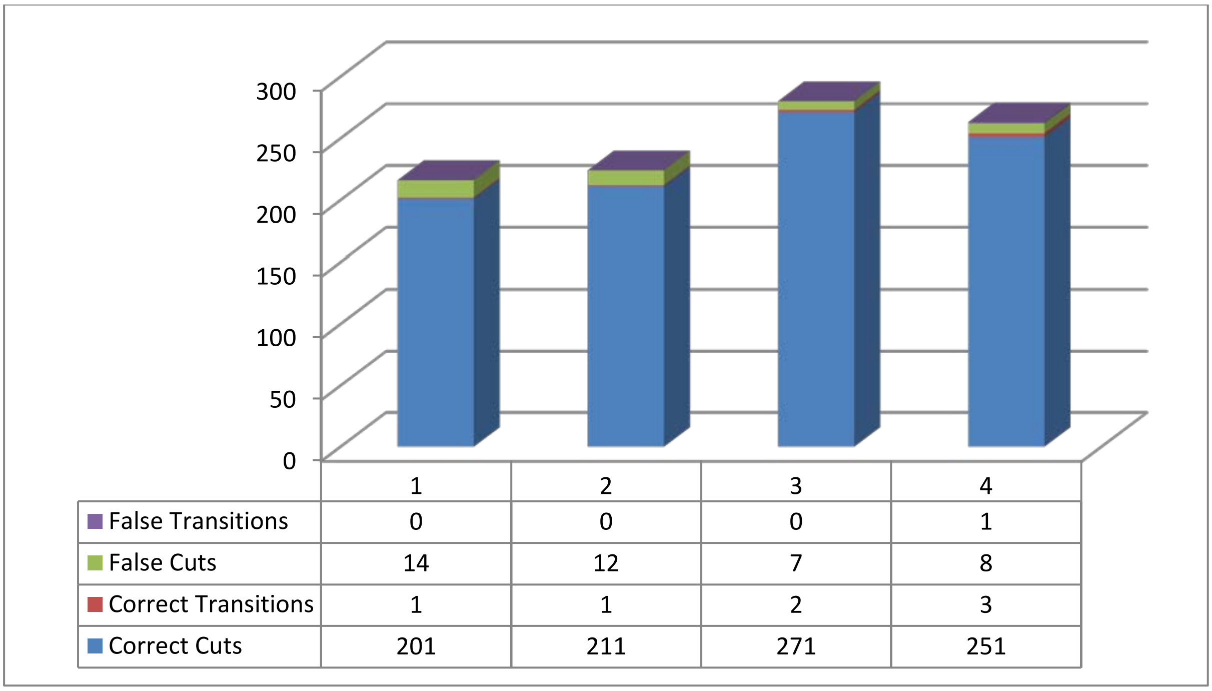
98.48%. Clips 1 and 2 where poorly-lit scenes and, therefore, the colours in them were not as vivid as in clips 3 and 4. If they had been then they would have had similar detection rates. The gradual transitions relied on edge information, which although it was diminished, was still able to accurately detect the gradual transitions.



	1	2	3	4
🟦 MAC-REALM Cuts	215	223	275	259
🟥 MAC-REALM Transitions	1	1	2	3
🟩 Baseline Cuts	238	245	278	263
🟪 Baseline Transitions	1	1	2	3

Figure 4.18: SHOTS DISCOVERED PER CLIP

In Figure 4.18 we can see how many shots detected were correctly identified and how many were incorrectly labelled as shot boundaries.



	1	2	3	4
🟪 False Transitions	0	0	0	1
🟩 False Cuts	14	12	7	8
🟥 Correct Transitions	1	1	2	3
🟦 Correct Cuts	201	211	271	251

Figure 4.19 NUMBER OF SHOTS CORRECTLY IDENTIFIED

From these results we see that $TP = 941, FP = 42, FN = 52$ and $P = 1031$. From this we calculate that the recall, precision and $f1$ score of the shot detection algorithm as:

$$Precsion = \frac{941}{941 + 42} = 95.73\%$$

Eq. (4.10)
$$\text{Recall} = \frac{941}{1031} = 91.27\%$$

$$f1 = \frac{2 \times 95.73 \times 91.27}{95.73 + 91.27} = 93.44$$

From this result we conclude that the shot detection algorithm is close to the optimal score of 1 which makes it a very good shot detector. The transition detection of the ECR is good but the cut detection missed a small percentage of shots and mislabelled a relatively few shot boundaries incorrectly. This could be due to lighting which is dark and the colours are not distinct enough.

4.2.3.2 Object Detection

To detect an object, the object detection method, requires an object to be segmented accurately so that the contour of the object is the boundary of the object, and that the object boundary is tracked accurately once segmented.

The methodology employed to measure the object segmentation, uses 4 randomly selected objects. These objects, the intersection between them and the groundtruth samples were recorded. Using Eq. (4.5) we calculated the accuracy for all four objects. They are presented in Table 4.1:

Object	Accuracy
1	0.87
2	0.80
3	0.73
4	0.83

Table 4.2: SEGMENTED OBJECT ACCURACY

The results for sample selections of the object extraction are shown in Figure 4.19, with each row showing the results for a key frame of a shot. The original colour images are shown on the far left of each row. The user-defined label traces overlaid on the image are shown in the left middle column of each row. Each colour represents a different label, object 1 is red and the background is yellow. Green is used as the colour for second objects. The object segmentation is shown in the third from left of each row. On the far right of each row we have a colour map of the objects, clearly showing the boundaries of each object and the backgrounds through colour.

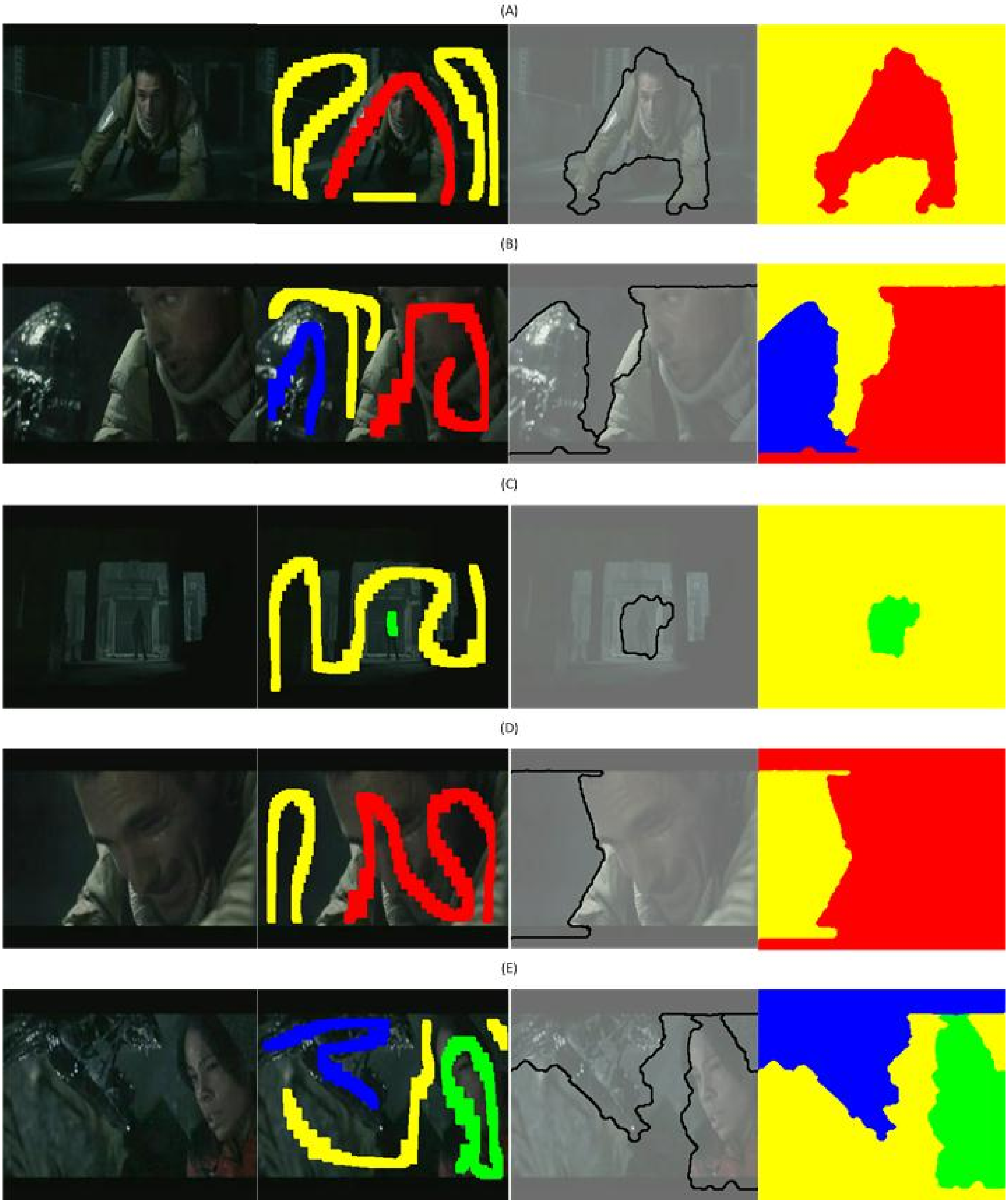


Figure 4.20 EXAMPLES OF OBJECTS EXTRACTED FROM IMAGES

The image regions may present similar grey-levels due to dark scenes and belong to different model classes defined by the user labels. Also, there are some image regions with substantial grey-level variation because of belonging to non-homogeneous textured regions, which are traditionally very difficult to segment. The structural information leads to a robust segmentation performance even in such cases. For brighter regions with well contrasted boundaries the segmentation has accuracies of between 0.97 – 0.998.

We can see that an object has bled into the "letterbox" lines of the image in frames B, D and E. These lines were never traced as background and so were not eliminated. If done so they would have been removed too. The rough tracing of objects has led to some objects edges not being defined, as in C and E.

For the object tracking we have tracked the same objects. These did not handle scenarios of occlusion and partial occlusion. Table 4.2 shows the OTE calculated from Eq. (4.6) for the four objects

Object	OTE
1	0.98
2	0.75
3	0.55
4	0.93

Table 4.3: OTE FOR SEGMENTED OBJECTS

The tracking was good for 1 and 4 as the object contours were well defined and the motion smooth. Object 2 had problems with tracking as it was a fast moving scene and there was some motion blur that affected the integrity of the object boundary. For object 3 the problem was that the object had not been segmented well and therefore the tracking became erroneous.

4.2.3.3 Scene detection

Scene detection was tested by using the first clip of AVP as the training data for the GP algorithm. The resulting rule was then used on the remaining three clips to ascertain how well the clips were segmented into scenes. The four features used in the GP algorithm (shot duration, number of objects, colour histogram and shot transition) was provided by the results of the shot detection and objects detection on the four AVP clips as java data structures that had been serialised to data text files.

Each test run was set up with parameters p (population size) = 500, k (maximum generation) = 300 and f (maximum fitness) set at 98%. The experiment was run three times on the same dataset. What was found was that an optimal rule was found with 98% fitness around the 118 – 120 generations mark. The best machine-generated rule from each run is shown in Table 4.3 in Reversed Polish Notation (RPN).

Best Rule 1	(((dC03<cA75<dA39<&)((cC04>cA85<cA09<^)(dE03<cA78<cA39<&)&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dB01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)((cC04>cA72<cA08<&)(dC03<cA78<cA39<&)&)^)(cC04>cA72<cA08<&)&)((cC04>cA72<cC09<&)(dC03<cA78<cA39<&)&)&)&)&)&)&)&)&)&)&)&)&)&)	118
Best Rule 2	(((dC03<cA75<dA39<&)((cC04>cA85<cA09<^)(dE03<cA78<cA39<&)&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(cD04<cA72<cC09<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dE01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)(cC04>cA72<cA08<&)^)(cC04>cA72<cA08<&)&)((cC04>cA72<cC09<&)(dC03<cA78<cA39<&)&)&)&)&)&)&)&)&)&)&)&)&)	120
Best Rule 3	(((dC03<cA75<dA39<&)(dC03<cA78<dA19<&)&)((dE01<cA79<cA33<&)((cC04>cA42<cA09<^)((dC03<cA71<cA39<&)(((cE03<cA78<cA39<&)((((cC04>cA82<cA09<^)(cD04<cA72<cC09<&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)((cC06>cA72<cC09<&)(dE01<cA79<cA39<&)&)&)^)((bC04>cA52<cA09<&)((cC04>cA42<cA09<^)((cC04>cA42<cA09<^)((dC03<cA75<cA39<&)((dE01<cA79<cA39<&)(((dC03<cA78<cA39<&)(((cC04>cA72<cA08<&)(((cC04>dA72<cA04<&)(dC03<cA78<cA39<&)^)(dC03<cA78<cA39<&)&)&)((dB03<dA78<cA39<&)(dE01<cA79<cA39<&)^)&)&)((((cC04>dA72<cA04<&)(cC04>cA72<cA08<&)^)(cC04>cA72<cA08<&)&)(cC04>cA42<cA09<^)&)&)&)&)&)&)&)&)&)&)&)&)	119

Table 4.4: Best Scene boundary change Rules generated by the GP algorithm for detecting scene changes in AVP film

The rules are applied to testing data for measuring its accuracy. We use the same performance measures used for shot detection, precision and recall, to evaluate the accuracy of the rule. The methods have been used extensively to compare the performance of shot boundary detection techniques. Since the nature of scene boundary detection is similar to shot boundary detection, it is plausible to use the method as well without any modification.

There are 786 shots in total in the three AVP clips to be segmented. Among them, there are 23 scene boundaries (manually counted) and the rule has discovered 21. But only 13 of them are correct, so there are 8 false alarms. Of the 21 scenes found only 13 have correct boundaries, which means it missed the other 8. Hence, with equation 4.1, the recall value is computed as:

Eq. (4.11)
$$Recall = \frac{\begin{array}{c}Number\ of\ scenes\ correctly\\ identfied\ by\ MAC - REALM\end{array}}{\begin{array}{c}Total\ number\ of\\ baseline\ scenes\end{array}} = \frac{13}{23} = 56.5\%$$

With Eq. (4.12), the precision value is calculated as:

Eq. (4.12)
$$Precision = \frac{\begin{array}{c}Number\ of\ scenes\ correctly\\ identfied\ by\ MAC - REALM\end{array}}{\begin{array}{c}Total\ number\ of\ scenes\\ identfied\ by\ MAC - REALM\end{array}} = \frac{13}{21} = 61.9\%$$

From this we can calculate the $f1$ score for the scene boundary detection algorithm as:

Eq. (4.13)
$$f1 = \frac{2 \times 61.9 \times 56.5}{61.9 + 56.5} = 59.08$$

As discussed in section 3.2.2.3 the performance for the GP algorithm is better using the four video features compared to two video and two audio features for the sample clip. When tested using the audio/video feature combination 23 clips were identified and only 10 were correct with a rule that had 96% fitness. The results for this test were 47.6% for precision and 43.5% for recall, giving an f1 = 45.45. One of the possible reasons for this is because AVP does not have much dialogue and long pauses of silence for suspense, making audio breaks are rare.

After close examination of the result we discovered that there is a substantial number of "near-misses", where the correct boundary is just one or two shots away from the boundary detected by the rule. By observation, we discovered that some of the missed boundaries are very close to the correct ones. However, the measurement we use can only indicates that the results are correct or not but cannot specify the margin of error. Most researchers agree that since scene boundary is a subjective concept, people may have different perception on where the scene boundaries are located (Hua & Zhang, 2009). The solution is to treat false alarms unequally. The distance between the false alarms and the correct ones is taken into account during the evaluation.

Furthermore, insufficient terminal and function sets are a potential problem, which is a common difficulty in genetic programming. We selected the terminal and function sets based on the attributes that have been proved to be valuable clues in determining scene boundaries. It is impractical to include as many functions and terminals as possible because the presence of extraneous functions and terminals would adversely affect the algorithm's performance.

4.2.3.4 Spatial relationships

Two things are of main concern when evaluating the spatial relationships: are all the spatial relationships captured and are they captured accurately. The results of the object extraction process are used as the input dataset for the spatial relationship analysis. To be able to accurately calculate the spatial relationships the centroid of the objects are calculated to provide the reference point of the objects.

A groundtruth of spatial relationships is produced for all objects by visually examining all shots and manually determining where the centre of the main body of an object lies and stating its absolute position. If two objects are within the shot frame then a relative position is calculated for them.

As we can see in Figure 4.20 recall was 100% and precision was 100%. All absolute and relative spatial relationships were correctly identified and all spatial relationships were correct according to the groundtruth observations of the spatial relationships.

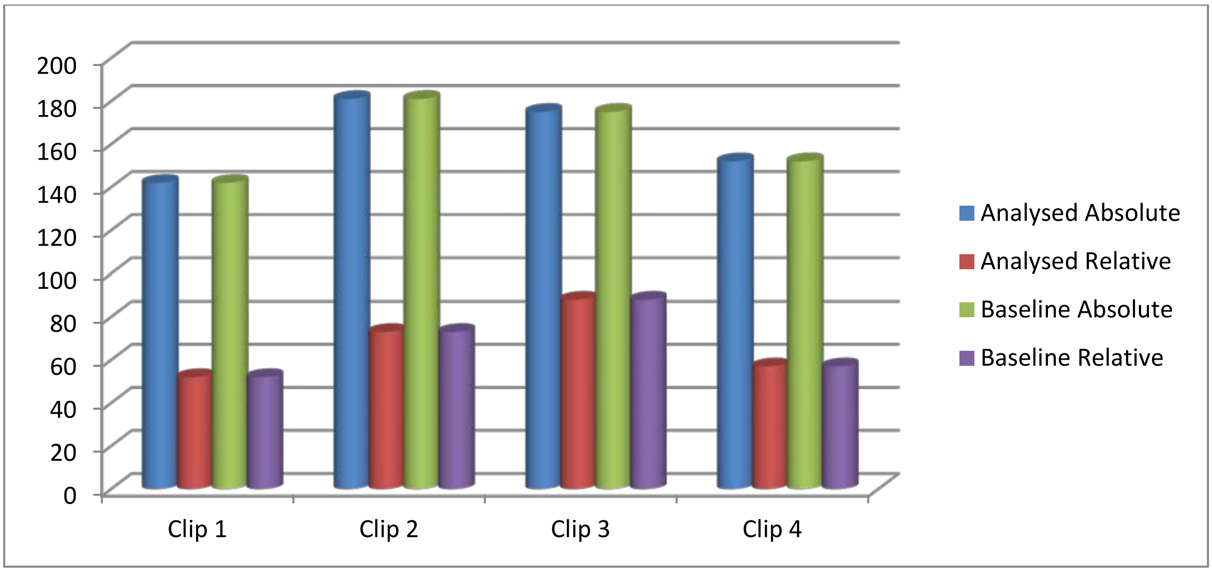


Figure 4.21: DERIVED SPATIAL RELATIONSHIPS VS. TOTAL AMOUNT OF SPATIAL RELATIONS

Figure 4.21 contains a list of absolute spatial relationships derived by the spatial relationship analyser. Using the centroid function as the point of calculation for spatial relationships holds true for the dataset used.

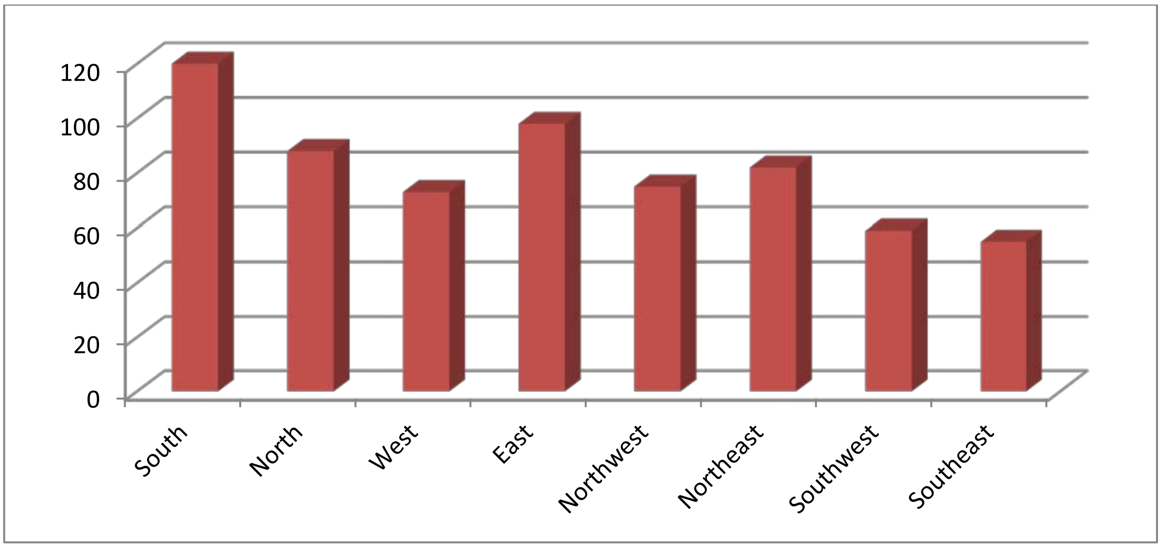


Figure 4.22: NUMBER OF ABSOLUTE SPATIAL RELATIONSHIPS FOUND FOR EACH POSITION

When the absolute spatial relationships are analysed a lot of the objects were only marginally within the boundary of the particular position they were allocated. A lot of objects central mass, and not their central point were in the centre of the screen. A more accurate derivation for a number of spatial relationships would be more accurately described as “central”. Seeing that “central” or “centre” are not in the MPEG-7 spatial relationship CS, this could not be used.

In Table 4.4 we have a grid of relative spatial relationships. The table does not show inverse relationships.

	Above	Below	Left	Right
Above	40	▨	25	22
Below	▨	43	24	28
Left	25	24	40	▨
Right	22	28	▨	38

Table 4.5: NUMBER OF RELATIVE SPATIAL RELATIONSHIPS FOUND FOR COMBINATION OF POSITIONS

The relative spatial relationships have a fuzzier categorisation as they can have both a horizontal and vertical element for their relative positioning. Of the 270 relative spatial relationship's 60% were single positions whilst 40% had two positions both horizontally and vertically. With 40% of spatial relationships having 2 positions, compared with those with just 1 position, it was correct for MAC-REALM to state two positions as it gives more clarity to the positioning of the objects.

4.2.3.5 Temporal relationships

Temporal relationships form the basis of semantic querying by allowing the user to investigate both semantic and syntactic features through their chronological relationship to each other and the meaning of those relationships.

All features have a temporal component and can therefore have a temporal relationship with any other feature. This intra/inter temporal relationship dependency allows for more intuitive search queries from the user that allows them to link abstract concepts to physical elements. For example a query can be formulated that states “When does an object A and object B reverse positions”. This query involves all the content features that have been extracted and sets a context for a user query.

In Table 4.5 we have types of content feature along both axes. The intersection where they meet shows the amount of temporal relationships between them. The table shows both binary and inverse binary relationships. These are shown together as to eliminate pointless duplication.

Feature vs. Feature	Shots	Scenes	Objects	Spatial Relationships
Shots	685584	12420	117576	761760
Scenes	12420	225	2130	13800
Objects	117576	2130	20164	130640
Spatial Relationships	761760	13800	130640	846400

Table 4.6: Number of Temporal relationships found for combination of features

From the table we can see that the amount of temporal relationships increases by multiple factors depending on the number of the instances of the content feature within the video stream. As there are only a few scenes the amount of temporal relationships is small. As there are a large amount of spatial relationships there are thousands more temporal relationships. The table shows temporal relationships add descriptive meaning exponentially depending on the increased presence of a feature, thus, giving more querying advantages.

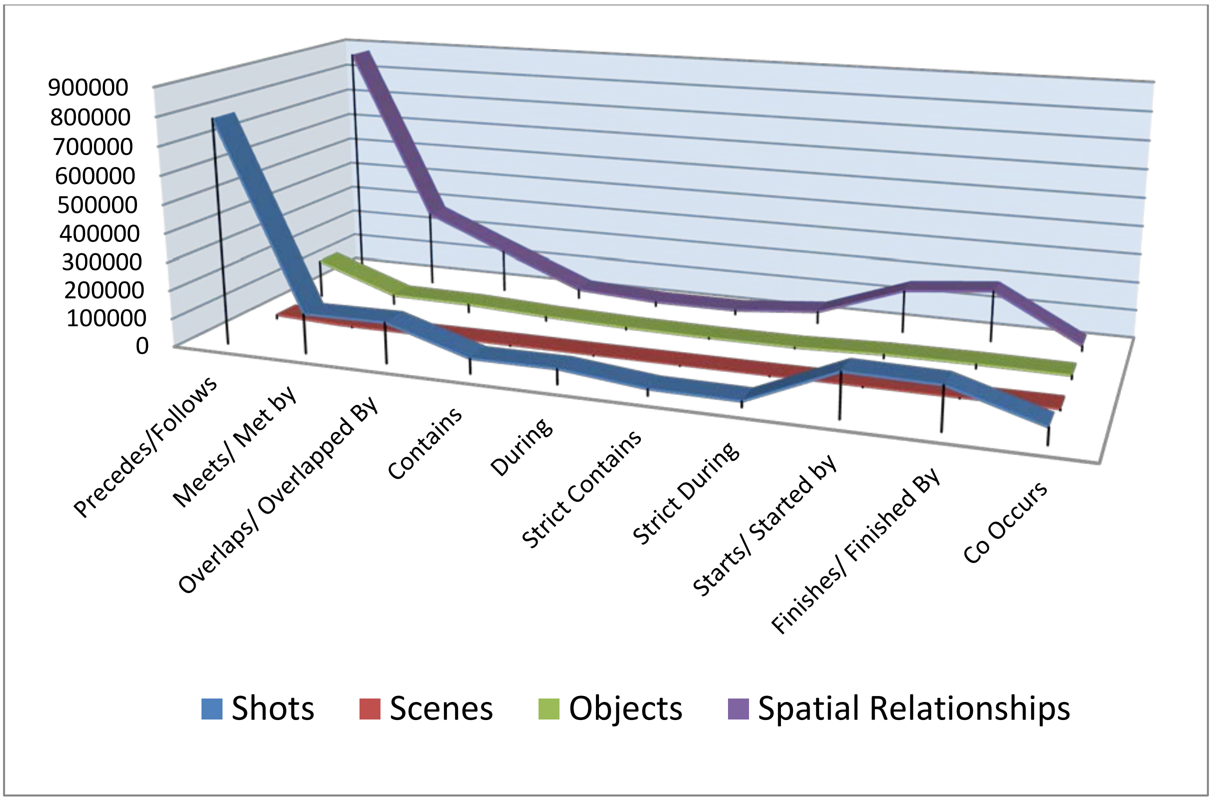


Figure 4.23: Number of Temporal relationships found for each Feature

From the graph in Figure 4.22 we can see that the majority of temporal relationships are precedes/follows. This makes sense when we consider the generalised case of a feature's time point $[i_x, j_x]$. If it is but one of many features $[i_N, j_N]$ then $\forall[i_{N-x}, j_{N-x}] < [i_x, j_x] < \forall[i_{N+x}, j_{N+x}]$ which therefore means that for every feature there will be an exponential increase for every other feature that it is compared too. We can also see that temporal relationships only require one time point of a feature to meet another time point of another feature with the most popular being meets, overlaps, starts, finishes, co-occurs and their inverses. Finally, those that needed both time points of both features to be satisfied i.e. contains, during, strict contains and strict during were the least used.

4.2.3.6 Content Modelling

The MAC-REALM content model was validated against different versions, profiles and constraints of MPEG-7 standard. Both versions 1(1999) and 2(2004) of MPEG-7, with three different profiles,, namely DAVP, TRECVID and AVDP, along with temporal validation. The MAC-REALM content model successfully completed all possible permutations. The MPEG-7 valid MAC-REALM content model means that the model is accessible to all MPEG-7 compliant content based video search applications.

To fully test the content model for its multi-content type ability and granular search capabilities, the MAC-REALM content model needs to be tested against a number of MPEG-7 compliant content based video search applications. At the time of testing there were no MPEG-7 compliant content based video search applications available. The MAC-REALM content model validates for all known profiles of MPEG-7, and is compliant to all parts of the standard. From the compliance results it can also be extrapolated that all other MPEG-7 compliant content based video search applications will also be interoperable. Even though MAC-RELAM is compliant with the MPEG-7 standards, there could be some integration work necessary (i.e. libraries, or implementation of API's) to ensure compatibility across other MPEG-7 applications and devices.

During the development of the MAC-REALM prototype, there was no standardised formal query syntax defined, to create search queries with. Other related works used solved this problem using XQuery (Bas̡tan et al., 2010; Döller, Stegmaier, Stockinger, & Kosch, 2011; Kannan et al., 2010). Often extensions were added to XQuery for multimedia (Xue, Li, Wu, & Xiong, 2009a) and SQL/MM, MMDOC-QL (Kang, Kim, & Ko, 2003). The proprietary nature of some metadata descriptions and the lack of formal semantics, are two main issues that do not allow using XQuery based applications, as a query testbed for MAC-REALM. The first issue is that XQuery search tools where created application by application to fit the needs of the content model, and act as a proof-of-concept for the particular application. Such XQuery search tools were not guaranteed to be MPEG-7 compliant, as combining these varied query approaches with alternate metadata description formats and retrieval interfaces prevents effective interoperability between MPEG-7 multimedia retrieval systems. The non-standardised process to designing MPEG-7 search functionality means, although many content based video search systems claim to be "MPEG-7 compliant", they were in practice limited in their compatibility of the MPEG-7 standard, especially with regards to the semantic querying of multimedia content. The second issue is that XQuery lacked any formal syntax for semantics. This along with the ability to not be able to handle "fuzzy" query types (e.g. "query-by-example") and having no formal semantics for processing multimedia objects, meant that it was unsuitable for testing of multi-content type queries.

MPEG-7 query format (MPQF) was created to solve the problem of search interoperability and was ratified officially into the MPEG-7 standard (MPEG, 2012b). MPQF provides a query syntax that makes access to distributed multimedia resources unified. The standardisation of MPQF into MPEG-7 leads to two main benefits; interoperability between parties in a distributed environments and platform independence. The key feature of MPQF is that it addresses the weaknesses of XQuery such as fuzzy request handling and formal semantics for syntax and processing multimedia objects. MPQF allows for queries specifically targeted by the MAC-REALM content model such as query-by-example media, query-by-example-description, query-by-keywords, query-by-feature-range, query-by-spatial-relationships and query-by-temporal-relationships. The MAC-REALM content model is better suited to MPQF queries, as it allows the search and retrieval of complete, or partial multimedia content data, metadata by specification of a filter condition tree and desired processing granularity. This would allow querying to be performed in a "coarse to fine" manner, with the capability of searching only relevant content features within the MAC-REALM content model.

Applications that are fully MPEG-7 compliant would provide a better test platform for the MAC-REALM content model, but there are currently no applications available for testing. The MPQF reference software was only available after the development and testing of MAC-REALM had been completed. As MAC-REALM was validated against a broad range of specifications for the MPEG-7 standard and that MPQF has been ratified into the standard, it can be concluded that interoperability between MAC-REALM and MPQF could facilitate multi-content type and granular searches.

4.3 Discussion of MAC-REALM Framework

The GUI front end screens are user friendly and allow the user to navigate through the process of creating a content model from a video stream. The interaction between the user and the MAC-REALM front end is intuitive and guides the user step-by-step through each process. The user is shown the results of each process after completion and can analyse the MPEG-7 content descriptors at the end of every stage. The design of the GUI puts the interaction between the user and MAC-REALM at the forefront of creating the mid-level syntactic content descriptions. The user input and feedback is taken to create mid-level content descriptions that are more accurate semantically. The GUI design is very user concentric and hides the underlying processes well, but in doing so the user does feel disconnected with the functionality of MAC-REALM as a content creation tool in the unsupervised machine driven parts of the processing. Extensive user testing needs to be carried out on the GUI to make it more HCI friendly, but this was outside the scope of this thesis as it is only a proof of concept for the MAC-REALM framework.

The pre-processing method is proven to reduce the computational expense of processing by a multiple factors, depending on the video frame rate, by eliminating redundant frames from the processing chain. The filtering is improved, whilst reducing processing time, by choosing the RGB colour space that is shown to be an all-round good choice as a trade-off for performance vs. improved results. The colour profile increases the feature extraction potential of the media and is less computationally expensive than other comparable colour profiles. Noise removal and flattening of the images improves the feature extraction potential of the RGB colour space by removing the susceptibility it has to noise and luminance changes.

The syntactic feature extraction processes are shown to segment the low-level and mid-level features accurately and with high detection rate for each feature. The shot boundary technique has high precision, recall and F1 score of 95.73%, 91.27% and 93.44% respectively. Detection rates and accuracy could have been higher if footage was used that had more definition through improved lighting. The object detection was a semi-supervised process, where the objects were manually identified through brush strokes. This allowed for all objects to be identified. The object boundaries were then segmented from the background and then tracked by the MAC-REALM OBD algorithm. The accuracy of the contours for varied between clip from 0.73 – 0.87 as the quality of the image affected the segmentation process due to the dark scenes and lack of definition of the objects. The object tracking accuracy was 0.55 – 0.98 for the same set of sample objects. The varying rate of the tracking was down to the initial problem of poor segmentation leading to incorrect pixels being tracked and the integrity of the tracking process being impeded because of this. The scene detection algorithm recorded results of precision, recall and F1 score for the scene segmentation as 56.5%, 61.9% and 59.09% respectively. The score was much lower than the predicted 98% of scene detection predicted by the fitness function of the GP algorithm that tested the evolved rule used. The discrepancy comes from the fact that a large number of scene boundaries were a less than three frames away from the correct boundary. This is within a margin of error that If there were taken into account the scene detection rate would have been very close to the hypothetical figure predicted by the fitness function.

The semantic relationships are analysed and discussed to see how they describe the spatial and temporal relationships of the content features, and the implications of this. All the spatial relationships are captured and the accuracy of their positions are shown to be 100% when compared to the user defined groundtruth positions. This shows that using the centroid function, as the reference point for measuring the spatial relationships positions is the best method for defining relationships that are intuitive to that of the human perspective. The temporal relationships between features are shown to increase exponentially as the number of instances of the features increase. This implies the amount of temporal information associated with the each feature increases and the querying potential increases of the content model. This increase in temporal information between different feature sets facilitates inter/intra temporal querying that provides tighter integration of the content model. The types and amounts of temporal relationships were also analysed and it was shown how they exponential increase for the features.

The final MAC-REALM content model is shown to integrate the extracted content features into a standardised content description document, which has been validated as MPEG-7 compliant. MAC-REALM's content model was validated against different types of profiles from the MPEG-7 standard, along with different configurations, and was found to be compliant with all of the specifications. This means that the MAC-REALM content model will be compatible with all correctly MPEG-7 compliant applications, regardless of what version of MPEG-7 is being used. To fully test the MAC-REALM content model, MPEG-7 compliant video search applications were required, but no application at the time were found suitable, nor would they allow the satisfactory querying of the MAC-REALM's content model. Achieving the research objective to facilitate granular and multi-content type searches, employs that the MPQF specification is used as a basis of comparison to the MAC-REALM content model. From this comparison and MPEG-7 content model validations, it can be inferred that the content model fulfils the research objective and steps closer to 'bridging' the semantic gap.

4.4 Summary

Chapter 4 provides a walkthrough of MAC-REALM including GUI front end screens that enable the users to extract low level features and from those features derive high level features, which are then integrated to together to produce a MPEG-7 compliant content model.

A performance evaluation is presented, based on how well the objectives have been fulfilled by MAC-REALM. The first section of the performance evaluation presents the testbed and the benchmark tests that are required to test MAC-REALM. The second section presents the results, firstly of the syntactic extraction techniques using empirical evaluation, followed by an analysis of the semantic features and their relation to the lower level features. The section finishes by validating the MAC-REALM content model against different MPEG-7 profiles. Finally, there is a discussion about MAC-REALM's content model and how it can have the facility of multi-content type and granular searches capabilities using MPQF.

Finally, this chapter presents an overall discussion about the MAC-REALM framework. The main points of the interaction and functionality of the GUI is examined and how it helps users navigate through MAC-REALM. The main points of the results are summarised and their implications of MAC-REALM's for objectives discussed. The summary concludes the chapter.

The next chapter concludes by summarising the thesis, presenting and measuring the research contributions against the research objectives and considering further research and development.

CHAPTER 5: CONCLUSION

In chapter 4 we provide a walkthrough and evaluation of MAC-REALM prototype. The walkthrough provides a step-by-step examination of the MAC-REALM GUI and how a user is guided through the process of converting a video stream into a standardised content model that describes the content in both syntactic and semantic terms. An evaluation of each component of MAC-REALM

The chapter is organised as follows. Section 5.1 reviews the thesis chapter by chapter discussing the main points of each chapter. Section 5.2 we examine the research contributions against the research objectives and section 5.3 considers future research and development.

5.1 Thesis Overview

Chapter 1 aims to establish the thesis by introducing the overarching themes and by placing the inspiration for the research undertaken into context. Subsequently, the motivation and goals defined for the investigation of the thesis are discussed, followed by a summary of the thesis project. Finally, an overview of the dissertation is given on a chapter-by-chapter basis.

Chapter 2 proposes a design of a content feature extraction and modelling framework called MAC-REALM. The framework is introduced and the motivations behind the requirements of MAC-REALM are examined. The following two sections examine automatic content feature extraction and content modelling design requirements in further detail. These are then stated as formal design requirements that elaborate on the requirements from the objectives in chapter 1.

The MAC-REALM Framework is presented as an architecture that incorporates the design requirements into function components that are linked by a custom video processing pipeline. Content passed through the pipeline and is converted from content media to content descriptions in layers of different content feature levels as the video stream is translated into a content model.

The design of the content, application and MPEG-7 layers is then looked at. For the content layer we describe the media to content description conversion for each plane. The content layer stores the media for each plane that will be processed. The application layer converts the content for each plane into content descriptions that are relevant for that planes function. The MPEG-7 layer is where the content description are modelled into MPEG-7 content descriptions. An in depth view is given of the planes and how they are to perform their function. The choices of the processing strategy for each component are discussed in reference to the function it performs in the MAC-REALM framework. Where applicable the sub-processed are discussed and the techniques employed are focused on in their own sections.

Chapter 3 presents MAC-REALM and the implementation of its four planes, three layers architecture. The implementation requirements are presented based on the research methods and design requirements stated in earlier chapters. An overview of the MAC-REALM prototype is shown, and how the custom video processing pipeline passes through the planes and the interaction of the components of each plane play a part in transforming the content. Next each plane is presented individually and the function of each component within the plane detailed.

The raw media plane removes redundant data by removing frames incrementally, converts the colour space to be more beneficial to extraction and finally the noise is removed from the frames using morphological filtering to stop impurities affect the performance of the extraction process. The filtered frames and the colour histograms are sent to the syntactic media component in the extraction plane to await processing.

The extraction plane integrates two shot detection algorithms to detect two different types of shot transition, abrupt cut and gradual transition. The shots are then used for object extraction, where a two-phase approach is used. Graph cuts segmentation is used to extract the object/s from the background, and then covariance matrix tracking is used to track the pixels across the shot. Both shots and objects, along with the colour histograms are used by the scene segmentation algorithm as the input of features that will be used by the GP Algorithm to evolve rules that will identify a scene boundary. The resulting shots, objects and scenes extracted in this plane are then used as input for the next plane. They are also modelled and serialised into syntactic MPEG-7 content descriptions.

The analysis and linkage plane analysis and links the content features together to form spatial and temporal relationships between them. Before the spatial relationships are defined the centroid of each object is defined to provide the reference point of measurement. Then the absolute, relative and inverse relative spatial relationships are calculated. Then all the temporal relationships between all the content features are mapped and modelled. The spatial and temporal relationship are then serialised into MPEG-7 semantic content descriptions.

The MPEG-7 syntactic and semantic features are then integrated together in the modelling plane. They are combined in a hierarchical structure that uses a MPEG-7 content model wrapper to interlink the syntactic and semantic content features into a tightly coupled integrated content model that is capable of granular search and facilitates multi-content type search.

Chapter 4 provides a walkthrough of MAC-REALM including GUI front end screens that enable the users to extract low level features and from those features derive high level features, which are then integrated to together to produce a MPEG-7 compliant content model.

A performance evaluation is presented, based on how well the objectives have been fulfilled by MAC-REALM. The first section of the performance evaluation presents the testbed and the benchmark tests that are required to test MAC-REALM. The second section presents the results, firstly of the syntactic extraction techniques using empirical evaluation, followed by an analysis of the semantic features and their relation to the lower level features. The section finishes by validating the MAC-REALM content model against different MPEG-7 profiles. Finally, there is a discussion about MAC-REALM's content model and how it can have the facility of multi-content type and granular searches capabilities using MPQF.

Finally, this chapter presents an overall discussion about the MAC-REALM framework. The main points of the interaction and functionality of the GUI is examined and how it helps users navigate through MAC-REALM. The main points of the results are summarised and their implications of MAC-REALM's for objectives discussed. The summary concludes the chapter.

The next chapter concludes by summarising the thesis, presenting and measuring the research contributions against the research objectives and considering further research and development.

5.2 Research contributions

This section will provide an overview to the objectives and research contributions of this thesis. Our objectives were presented in Chapter 1 and are summarised in Table 5.1, along with each corresponding research contribution. Objectives 1 and 3, as listed in Table 5.1 contain a number of sub-objectives, each of which are detailed later in this section.

Objective	Research Contribution
O1. To design an abstract framework that transcodes video stream into content descriptions. The framework must extract both syntactic content and semantic relationship descriptions and interlink them; Helping to bridge the semantic gap. • A method to pre-process video suited for extraction • Extract syntactic features • Create semantic relationships and content descriptions	RC1. The abstract MAC-RELAM Framework design • the raw media plane - improves extraction and reduces computational expense • The extraction plane - extracts low-level syntactic features and mid-level syntactic features with semantic attributes into syntactic content descriptions • The analysis and linkage plane - links the spatial and temporal relationships of all the features into semantic content descriptions
O2. To integrate the syntactic and semantic descriptions into a content model that is accessible to a wide range of applications and that supports granular search and facilitates multi-content type search. • Interoperable content model • A content description that supports querying • Combine syntactic and semantic features for querying	RC2. The output produced from MAC-RELAM is a standards based content model • MAC-REALM's content model validates against all profiles and versions of MPEG-7, thus is usable by MPEG-7 compliant applications. • The content details are in a hierarchical structure that is mapped on the structure of the content model. This structure allows for "coarse to fine" searches to be performed • interlinks content on a syntactic and semantic level, facilitating syntactic and semantic search queries on all content features
O3. To develop a prototype of the framework as a proof of concept. • Extensible and modular • Allow for custom video processing pipelines to be created	RC3. The MAC-RELAM prototype that implements a four plane and three layer system architecture. • An object oriented and portable framework that allows for modules to be developed, extended, reused, shared and modified independently • The framework allows for sequences of modules to create custom video processing pipelines

Table 5.1: Research objectives vs. Research contributions

The main contribution of this thesis is the abstract framework that was developed to achieve objective 1 and its sub-objectives 1.1 to 1.4, which aims to "To design an abstract framework that transcodes video stream content features into content descriptions. The framework must extract both syntactic content and semantic relationship descriptions and interlink them, in order to take a step closer to 'bridging' the semantic gap between syntactic and semantic features" MAC-REALM conceptualises the entire content feature extraction and modelling process. It uses a mixture of existing algorithms that have been extended or adapted in order to enable extract content features into content description from raw media, then amalgamates and structures the content descriptions into a content model. The abstract design of MAC-REALM provides enough flexibility in order to customise both its architecture and functionality and therefore, its implementation. In turn, this yields several advantages, firstly, separation of its functionality into four distinct planes with three layers, allows customisation at each layer rather than the entire framework and helps to achieve low level of coupling, for example modularity. Secondly, use of a novel pre-processing technique improves the feature extraction from the media and reduces unnecessary processing by removing redundant data. Thirdly, the syntactic feature extraction plane uses a hierarchical architecture, which extracts three syntactic features using a mixture of unsupervised and semi-supervised algorithms, reducing the need for user interaction. Where human interaction is required it is to provide input that defines the semantic characteristics of the syntactic features. Fourthly, the semantic relationships of the content features are derived from analysis, along with the subsequent linking of the syntactic features, provides semantic links between all features. Spatial relationships provide a spatial context to the content model, facilitating spatial search parameters on the content. The temporal relationships allow queries of the syntactic and semantic features in the same temporal context. The semantic relationships provide a semantic foundation to the content model that can enable the addition of high level concepts and facilitates event based querying. The second research contribution relating to objective 2 and its sub-objectives of this thesis is the modelling of the content into a widely accessible standard compliant content model. The choice of MPEG-7 as the content model feature description language led to problems of its own, as the standard has been revised numerous times. To make sure that the MAC-REALM content model was acceptable to the widest range of applications the descriptions were made backward compatible by using unrevised elements; a hierarchical structure with the main four category elements connected to a top level element and a general profile. The hierarchical structure also makes possible "coarse to grain" search by of any feature or combination of features. Through the temporal relationships, a semantic temporal search is possible on all of the features. The interlinking of the syntactic and semantic features through the syntactic features elements, physical attributes and the semantic modelling nodes, provides tight coupling between the syntactic and semantic features. The tight integration of these features on a structural level also allows the content to be queried using logic based queries. Modelling all the temporal relationships between all content features provides an abstract temporal relationship, which allows the content to be queried using event based queries. These two types of querying provide multi-content type search, through the integration of logic and semantic search capabilities.

The third and last contribution is the implementation of the MAC-REALM prototype that aims to develop "a proof-of-concept application which implements objectives 1 and 2". It must be extensible and modular to allow for customisation and updates. The proof of concept should provide a framework that allows for modules to be added, re-used, extended and modified. The re-use of modules allows for modules to be shared and further developed and can potentially reduce processing time, while allowing for custom video processing pipelines. MAC-REALM is a functional prototype and proves that MAC-REALM can convert "raw media into a content model through a process of content feature extraction and modelling". The framework is a novel approach to content conversion, where there are three layers to the content extraction and four modelling planes. Each layer provides modularity, by defining the function of each component at the intersection between planes. These components can be updated to provide better extraction of existing features, or extended to extract new features that better fulfil the desired functionality. The sequential arrangement of custom modules at layers, allows for custom video processing pipelines to be created. To the best of the author's knowledge, there is no other tool that attempts to bridge the 'semantic gap', by combining automated syntactic and semantic content extraction, into a MPEG-7 standards based and searchable content model, while providing an extensible and modular development framework, which allows for custom pipelines to be created.

5.3 Further Research

This section discusses possible future research and development work that may be undertaken to improve or extend the development of MAC-REALM.

5.3.1 Concept detection and classification of semantic events and objects

Automatic detection of complex events in unconstrained videos has great potential for many applications, such as web video indexing, consumer content management, and open-source intelligence analysis. Semantic concept detection is a research topic of current interest, as it provides semantic filters to help analysis and search of multimedia data. It is essentially a classification task, which determines whether an image or a video shot is relevant to a given semantic concept. The ability to detect events and label objects through concept detection is a possible extension to the MAC-REALM framework. This facility allows concept detectors to use spatial and/or temporal relationships to find syntactic features matches to concepts through the relationships between those features. In the study by (Jiang, Zeng, & Ye, 2010) it is stated that spatial - temporal features are very effective at multimedia event detection, when combined with other content descriptors, such as SIFT descriptors and audio features.

Using spatial and temporal relationships between the features, could potentially improve the detection rates of events further then just spatial-temporal interest points (STIP), which capture space-time volumes where the image values have significant local variations in both space and time. STIP has a problem with variations in length and complexity of the content as it is a direct measure of the spatial and temporal properties of certain features. Using spatial and temporal relationships between the features removes length and complexity of the feature from the equation, while normalising the feature description of these properties between all features. This would build a set of semantic feature descriptions that are built around a vocabulary that is more suited to learning methods (e.g. classifiers) that need fixed dimensions of input.

Once the concept are detected they can be classified using standardised semantic web ontologies, such as Dublin Core ("Dublin Core Website," 2012), TV-Anytime (Rey-López et al., 2010) or the suite of IPTC G2 News Exchange Format Standards ("IPTC News Exchange Format Standards," 2014). This means that the content model produced by MAC-REALM must be translated from its MPEG-7 XML-Schema base and converted into MPEG-7 RDF Schema, as suggested by (Jane Hunter, 2005). This would allow the metadata terms to be accessible, re-usable and interoperable with other ontological domains.

5.3.2 Crowd sourcing to extract semantic features

To improve the extraction capabilities of MAC-REALM for semantic content features, it could be extended to include crowd sourcing as a semantic feature extraction technique. Crowd sourcing has become a powerful tool in collaborative classification schemes that can build a structured knowledge base through user feedback via the world wide web (Doan, Ramakrishnan, & Halevy, 2011). The crowd sourcing could be used to extract semantic content features more efficiently and effectively as the semantics of the content would be directly perceived and could be mapped onto syntactic features through an interface. This would bridge the semantic gap as the knowledge base grows and could be used to train concept detectors to more accurately match concepts through a larger corpus of semantic material. Similar work has already been done where semantically annotated sport video clips from users are crowd sourced to provide fan-centric video summaries based on team supported (Tang & Boring, 2012).

REFERENCES

Aburjanidze, N., & Boucher, J. (2010). The Dot-Com Boom… and Burst: Identifying the Causes and Characteristics of the 21st Century's First Speculative Bubble. *The University of Utah Financial Research Report.*

Agius, H., & Angelides, M. C. (2005). COSMOS-7: Video-Oriented MPEG-7 Scheme for Modelling and Filtering of Semantic Content. *The Computer Journal, 48*(5), 545-562. doi: 10.1093/comjnl/bxh115

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832-843.

Aly, R., Doherty, A., Hiemstra, D., & Smeaton, A. (2010). Beyond Shot Retrieval: Searching for Broadcast News Items Using Language Models of Concepts. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger & K. van Rijsbergen (Eds.), *Advances in Information Retrieval* (Vol. 5993, pp. 241-252): Springer Berlin / Heidelberg.

Amel, A. M., Abdessalem, B. A., & Abdellatif, M. (2010). Video shot boundary detection using motion activity descriptor. *arXiv preprint arXiv:1004.4605.*

Amiri, A., & Fathy, M. (2011). Video shot boundary detection using generalized eigenvalue decomposition and Gaussian transition detection. *Computing and Informatics, 30*(3), 595-619.

Amri, A., & Fathy, M. (2010). Video shot boundary detection using QR-decomposition and gaussian transition detection. *EURASIP Journal on Advances in Signal Processing, 2009*, 1-12. doi: 10.1155/2009/509438

Angelides, M., & Sofokleous, A. (2013). A game approach to optimization of bandwidth allocation using MPEG-7 and MPEG-21. *Multimedia Tools and Applications, 62*(1), 287-309. doi: 10.1007/s11042-011-0981-0

Angelides, M., Sofokleous, A., & Parmar, M. (2006). Classified ranking of semantic content filtered output using self-organizing neural networks *Artificial Neural Networks–ICANN 2006* (pp. 55-64): Springer Berlin Heidelberg.

Angelides, M. C. (2003). Guest Editor's Introduction: Multimedia Content Modeling and Personalization*, 10,* 12-15.

Angelides, M. C., & Agius, H. (2006). An MPEG-7 scheme for semantic content modelling and filtering of digital video. *Multimedia Systems, 11*(4), 320-339.

Angelides, M. C., & Kevin Lo, T. S. (2005). A video content independent mining algorithm for evolved rule-based detection of scene boundaries. *Ingénierie des systèmes d'information, 10*(1), 81-99.

Angelides, M. C., & Lo, K. T. S. (2005). *A video content independent mining algorithm for evolved rule-based detection of scene boundaries* (Vol. 10). Paris, FRANCE: Lavoisier.

Apache. (2013, February 20th 2013). Hadoop Retrieved February 20th, 2013, from http://hadoop.apache.org/

Appiah, K., Hunter, A., Dickinson, P., & Meng, H. (2010). Accelerated hardware video object segmentation: From foreground detection to connected components labelling. *Computer Vision and Image Understanding, 114*(11), 1282-1291. doi: 10.1016/j.cviu.2010.03.021

Ayadi, T., Ellouze, M., Hamdani, T., & Alimi, A. (2012). Movie scenes detection with MIGSOM based on shots semi-supervised clustering. *Neural Computing and Applications*, 1-10. doi: 10.1007/s00521-012-0930-5

Ayvaci, A., & Soatto, S. (2012). Detachable Object Detection: Segmentation and Depth Ordering from Short-Baseline Video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 34*(10), 1942-1951. doi: 10.1109/tpami.2011.271

Babenko, B., Ming-Hsuan, Y., & Belongie, S. (2011). Robust Object Tracking with Online Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33*(8), 1619-1632. doi: 10.1109/tpami.2010.226

Baber, J., Afzulpurkar, N., & Bakhtyar, M. (2011, 5-6 Sept. 2011). *Video segmentation into scenes using entropy and SURF.* Paper presented at the Emerging Technologies (ICET), 2011 7th International Conference on.

Bai, X., Wang, J., & Sapiro, G. (2010). Dynamic Color Flow: A Motion-Adaptive Color Model for Object Segmentation in Video. In K. Daniilidis, P. Maragos & N. Paragios (Eds.), *Computer Vision – ECCV 2010* (Vol. 6315, pp. 617-630): Springer Berlin / Heidelberg.

Bailer, W., & Schallauer, P. (2006). *Detailed audiovisual profile: enabling interoperability between MPEG-7 based systems.* Paper presented at the Multi-Media Modelling Conference Proceedings, 2006 12th International.

Ballan, L., Bertini, M., Bimbo, A., Seidenari, L., & Serra, G. (2011). Event detection and recognition for semantic annotation of video. *Multimedia Tools and Applications, 51*(1), 279-302. doi: 10.1007/s11042-010-0643-7

Bartolini, I., Patella, M., & Romani, C. (2011). SHIATSU: tagging and retrieving videos without worries. *Multimedia Tools and Applications*, 1-29. doi: 10.1007/s11042-011-0948-1

Bashir, F., & Porikli, F. (2006). Performance evaluation of object detection and tracking systems. *In PETS, 6*.

Baştan, M., Çam, H., Güdükbay, U., & Ulusoy, O. (2010). Bilvideo-7: an MPEG-7- compatible video indexing and retrieval system. *MultiMedia, IEEE, 17*(3), 62-73. doi: 10.1109/mmul.2010.5692184

BSkyB. (2012). BSkyB Corporate Timeline Retrieved 22.08.12, 2012, from http://corporate.sky.com/about_sky/timeline

Bursuc, A., Zaharia, T., & Prêteux, F. (2012). OVIDIUS: A Web Platform for Video Browsing and Search. In K. Schoeffmann, B. Merialdo, A. Hauptmann, C.-W. Ngo, Y. Andreopoulos & C. Breiteneder (Eds.), *Advances in Multimedia Modeling* (Vol. 7131, pp. 649-651): Springer Berlin Heidelberg.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*(6), 679-698.

Carmona, E. J., Martínez-Cantos, J., & Mira, J. (2008). A new video segmentation method of moving objects based on blob-level knowledge. *Pattern Recognition Letters, 29*(3), 272-285. doi: http://dx.doi.org/10.1016/j.patrec.2007.10.007

Chan, C., & Wong, A. (2011, 5-7 Dec. 2011). *Shot Boundary Detection Using Genetic Algorithm Optimization.* Paper presented at the Multimedia (ISM), 2011 IEEE International Symposium on.

Chao, L., Changsheng, X., Jian, C., & Hanqing, L. (2011, 20-25 June 2011). *TVParser: An automatic TV video parsing method.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.

Chen, J., Ren, J., & Jiang, J. (2011). Modelling of content-aware indicators for effective determination of shot boundaries in compressed MPEG videos. *Multimedia Tools and Applications, 54*(2), 219-239. doi: 10.1007/s11042-010-0518-y

Chen, X., & Liu, W. (2010, 13-14 Oct. 2010). *Study on Shot Boundary Detection Based on Fuzzy Subset-Hood Theory.* Paper presented at the Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on.

Chen, Y., Deng, Y., Guo, Y., Wang, W., Zou, Y., & Wang, K. (2010, 26-28 Feb. 2010). *A Temporal Video Segmentation and Summary Generation Method Based on Shots' Abrupt and Gradual Transition Boundary Detecting.* Paper presented at the Communication Software and Networks, 2010. ICCSN '10. Second International Conference on.

Chiarcos, C., Nordhoff, S., & Hellmann, S. (2012). *Linked Data in Linguistics: Representing and Connecting Language Data and Language Metadata*: Springer.

Choroś, K., & Pawlaczyk, P. (2010). Content-Based Scene Detection and Analysis Method for Automatic Classification of TV Sports News. In M. Szczuka, M. Kryszkiewicz, S.

Ramanna, R. Jensen & Q. Hu (Eds.), *Rough Sets and Current Trends in Computing* (Vol. 6086, pp. 120-129): Springer Berlin / Heidelberg.

Christodoulou, L., Kasparis, T., & Marques, O. (2011, 6-8 July 2011). *Advanced statistical and adaptive threshold techniques for moving object detection and segmentation.* Paper presented at the Digital Signal Processing (DSP), 2011 17th International Conference on.

Dal Mutto, C., Dominio, F., Zanuttigh, P., & Mattoccia, S. (2012). Stereo Vision and Scene Segmentation.

Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., & Kompatsiaris, Y. (2011). A survey of semantic image and video annotation tools. In P. Georgios, D. S. Constantine & T. George (Eds.), *Knowledge-driven multimedia information extraction and ontology evolution* (pp. 196-239): Springer-Verlag.

Dasiopoulou, S., Tzouvaras, V., Kompatsiaris, I., & Strintzis, M. G. (2010). Enquiring MPEG-7 based multimedia ontologies. *Multimedia Tools and Applications, 46*(2), 331-370.

Daylamani Zad, D., & Agius, H. (2010). An MPEG-7 Profile for Collaborative Multimedia Annotation *The Handbook of MPEG Applications* (pp. 263-291): John Wiley & Sons, Ltd.

del Fabro, M., & Boszormenyi, L. (2010, 13-19 June 2010). *Video Scene Detection Based on Recurring Motion Patterns.* Paper presented at the Advances in Multimedia (MMEDIA), 2010 Second International Conferences on.

Doan, A., Ramakrishnan, R., & Halevy, A. Y. (2011). Crowdsourcing systems on the World-Wide Web. *Commun. ACM, 54*(4), 86-96. doi: 10.1145/1924421.1924442

Döller, M., Stegmaier, F., Stockinger, A., & Kosch, H. (2011). *XQuery Framework for Interoperable Multimedia Retrieval.* Paper presented at the Grundlagen von Datenbanken.

Dropbox. (2013). Dropbox Retrieved February 20th, 2013, from https://www.dropbox.com/

. Dublin Core Website. (2012), from http://dublincore.org/specifications/

Dumont, É., & Quénot, G. (2012). Automatic Story Segmentation for TV News Video Using Multiple Modalities. *International Journal of Digital Multimedia Broadcasting, 2012*, 11. doi: 10.1155/2012/732514

Ellouze, M., Boujemaa, N., & Alimi, A. (2010). Scene pathfinder: unsupervised clustering techniques for movie scenes extraction. *Multimedia Tools and Applications, 47*(2), 325-346. doi: 10.1007/s11042-009-0325-5

Ercolessi, P., Bredin, H., Sénac, C., & Joly, P. (2011). *Segmenting TV Series into Scenes Using Speaker Diarization.* Paper presented at the WIAMIS 2011:, 12th International Workshop on Image Analysis for Multimedia Interactive Services.

Fei, W., & Zhu, S. (2010). Mean shift clustering-based moving object segmentation in the H.264 compressed domain. *Image Processing, IET, 4*(1), 11-18. doi: 10.1049/iet-ipr.2009.0038

Feng, G., Song, W., & Tiecheng, L. (2006, 17-22 June 2006). *Image-Segmentation Evaluation From the Perspective of Salient Object Extraction.* Paper presented at the Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on.

Fromme, J., & Unger, A. (2012). *Computer Games and New Media Cultures: A Handbook of Digital Games Studies*: Springer.

Gargi, U., Kasturi, R., & Strayer, S. H. (2000). Performance characterization of video-shot-change detection methods. *Circuits and Systems for Video Technology, IEEE Transactions on, 10*(1), 1-13. doi: 10.1109/76.825852

Ghuffar, S., Brosch, N., Pfeifer, N., & Gelautz, M. (2012, 11-13 April 2012). *Motion segmentation in videos from time of flight cameras.* Paper presented at the Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on.

Gibbon, D., Liu, Z., Basso, A., & Shahraray, B. (2011). Using MPEG Standards for Content Based Indexing of Broadcast Television, Web, and Enterprise Content. *The Handbook of MPEG Applications*, 343-361.

Goss, P. (2011). YouView: We welcome Google TV competition Retrieved 30.08.11, 2011, from http://www.techradar.com/news/television/youview-we-welcome-google-tv-competition-1005658

Goss, P. (2012). Virgin TV Anywhere officially outed, arriving autumn 2012 Retrieved 07.09.12, 2012, from http://www.techradar.com/news/television/virgin-tv-anywhere-officially-outed-arriving-autumn-2012-1095475

Grana, C., & Cucchiara, R. (2007). Linear Transition Detection as a Unified Shot Detection Approach. *Circuits and Systems for Video Technology, IEEE Transactions on, 17*(4), 483-489. doi: 10.1109/tcsvt.2006.888818

Group, D. T. (2006a). BT Vision reveals December launch date Retrieved 28.11.2012, 2012, from http://www.dtg.org.uk/news/news.php?id=2092

Group, D. T. (2006b, 18.05.2006). Industry unites to promote Freeview Playback Retrieved 21.08.12, 2012, from http://www.dtg.org.uk/news/news.php?id=1674

Grundmann, M., Kwatra, V., Mei, H., & Essa, I. (2010, 13-18 June 2010). *Efficient hierarchical graph-based video segmentation.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.

Guo, W., Xu, C., Ma, S., & Huang, S. (2010). *Hausdorff matching based SVD-covariance descriptor for object tracking.* Paper presented at the Proceedings of the Second International Conference on Internet Multimedia Computing and Service, Harbin, China.

Güsgen, H. W. (1989). *Spatial reasoning based on Allen's temporal logic*: International Computer Science Institute.

Haller, M., Krutz, A., & Sikora, T. (2009, 6-8 May 2009). *Evaluation of pixel- and motion vector-based global motion estimation for camera motion characterization.* Paper presented at the Image Analysis for Multimedia Interactive Services, 2009. WIAMIS '09. 10th Workshop on.

Hameed, A. (2009, 19-20 Oct. 2009). *A novel framework of shot boundary detection for uncompressed videos.* Paper presented at the Emerging Technologies, 2009. ICET 2009. International Conference on.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques*: Morgan Kaufmann Pub.

Harikrishna, N., Satheesh, S., Sriram, S. D., & Easwarakumar, K. S. (2011, 28-30 Jan. 2011). *Temporal classification of events in cricket videos.* Paper presented at the Communications (NCC), 2011 National Conference on.

Haskell, B. G., Puri, A., Netravali, A. N., & Langdon, G. G. (1998). Digital video: an introduction to MPEG-2. *Journal of Electronic Imaging, 7*(1), 265-266.

Heejun, H., & Jaesoo, K. (2011, 19-21 Oct. 2011). *An useful method for scene categorization from new video using visual features.* Paper presented at the Nature and Biologically Inspired Computing (NaBIC), 2011 Third World Congress on.

Heuer, J., Hutter, A., & Niedermeier, U. (2010). Method for improving the functionality of the binary representation of MPEG-7 and other XML based content descriptions: Google Patents.

Höffernig, M., Hausenblas, M., Bailer, W., & Troncy, R. (2010). VAMP: Semantic Validation of MPEG-7 Profiles.

Hu, W. C., & Hsu, J. F. (2011). Foreground extraction-based video object segmentation using motion information and gradient compensation. *International Journal of Innovative Computing, Information and Control, 7*(8), 4849-4859.

Hua, X. S., & Zhang, H. J. (2009). Automatic Home Video Editing. *Multimedia Content Analysis*, 1-35.

Huang, Q., Ostermann, J., Puri, A., & Rajendran, R. K. (2009). Synthetic Audiovisual Description Scheme, Method and System for MPEG-7: US Patent App. 20,100/106,722.

Huang, S. N., & Zhang, Z. Y. (2010). Scene detection in videos using mutual information. *Applied Mechanics and Materials, 34*, 920-926.

Huang, Y.-F., & Tung, L.-H. (2010). *Semantic scene detection system for baseball videos based on the MPEG-7 specification*. Paper presented at the Proceedings of the 2010 ACM Symposium on Applied Computing, Sierre, Switzerland.

Hui, C., & Cuihua, L. (2010, 9-11 July 2010). *A practical method for video scene segmentation*. Paper presented at the Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on.

Humax. (2008). Freesat+ launches with humax foxsat-hdr in November 2008 Retrieved 28.09.12, 2012, from http://www.humaxdigital.com/freesat/press_081023.asp

Hunter, J. (2005). *Adding multimedia to the Semantic Web-Building and applying an MPEG-7 ontology*: Wiley.

Hunter, J., & Iannella, R. (2009). The application of metadata standards to video indexing. *Research and Advanced Technology for Digital Libraries*, 514-514.

Inigo, S. A., & Suresh, P. (2012). General Study on Moving Object Segmentation Methods for Video. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1*(8), pp: 265-270.

. IPTC News Exchange Format Standards. (2014), 2014, from http://www.iptc.org/site/News_Exchange_Formats/

Jacobs, A., Miene, A., Ioannidis, G., & Herzog, O. (2004). *Automatic shot boundary detection combining color, edge, and motion features of adjacent frames*. Paper presented at the TRECVID 2004 Workshop Notebook Papers.

Jacobs, A., Miene, A., Ioannidis, G., & Herzog, O. (2004). *Automatic shot boundary detection combining color, edge, and motion features of adjacent frames*.

Jiang, Y.-G., Zeng, X., & Ye, G. (2010). *Columbia-UCF TRECVID2010 Multimedia Event Detection: Combining Multiple Modalities, Contextual Concepts, and Temporal Matching*. Paper presented at the NIST TRECVID Workshop.

Kaleka, J. S., Singh, J., & Sharma, R. (2012). Different Approaches of CBIR Techniques. *INTERNATIONAL JOURNAL OF COMPUTERS & DISTRIBUTED SYSTEMS, 1*(2), 76-78.

Kang, J.-H., Kim, C.-S., & Ko, E.-J. (2003). *An XQuery engine for digital library systems*. Paper presented at the Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, Houston, Texas.

Kannan, R., Andres, F., & Guetl, C. (2010). DanVideo: an MPEG-7 authoring and retrieval system for dance videos. *Multimedia Tools and Applications, 46*(2-3), 545-572. doi: 10.1007/s11042-009-0388-3

Khatoonabadi, S. H., & Bajic, I. V. (2013). Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields. *Image Processing, IEEE Transactions on, 22*(1), 300-313. doi: 10.1109/tip.2012.2214049

Koprinska, I., & Carrato, S. (2001). Temporal video segmentation: A survey. *Signal Processing: Image Communication, 16*(5), 477-500.

Kristensen, F., Nilsson, P., & Öwall, V. (2006). Background segmentation beyond RGB. *Computer Vision–ACCV 2006*, 602-612.

Krulikovska, L., Pavlovic, J., Polec, J., & Cernekova, Z. (2010, 15-17 Sept. 2010). *Abrupt cut detection based on mutual information and motion prediction*. Paper presented at the ELMAR, 2010 PROCEEDINGS.

Küçüktunç, O., Güdükbay, U., & Ulusoy, Ö. (2010). Fuzzy color histogram-based video segmentation. *Computer Vision and Image Understanding, 114*(1), 125-134. doi: 10.1016/j.cviu.2009.09.008

Ladický, Ľ., Sturgess, P., Alahari, K., Russell, C., & Torr, P. (2010). *What, Where and How Many? Combining Object Detectors and CRFs*. Paper presented at the Computer Vision – ECCV 2010. http://dx.doi.org/10.1007/978-3-642-15561-1_31

Lavee, G., Rivlin, E., & Rudzsky, M. (2009). Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurrences in Video. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 39*(5), 489-504. doi: 10.1109/tsmcc.2009.2023380

Lawrence, E., Newton, S., Corbitt, B., Lawrence, J., Dann, S., & Thanasankit, T. (2012). *Internet commerce: digital models for business*: John Wiley & Sons.

LawTo, J., Gauvain, J. L., Lamel, L., Grefenstete, G., Gravier, G., Despres, J., . . . Sebillot, P. (2011). A Scalable Video Search Engine Based on Audio Content Indexing and Topic Segmentation. *arXiv preprint arXiv:1111.6265*.

Lezama, J., Alahari, K., Sivic, J., & Laptev, I. (2011, 20-25 June 2011). *Track to the future: Spatio-temporal video segmentation with long-range motion cues.* Paper presented at the Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.

Li, H., & Ngan, K. N. (2011). Image/Video Segmentation: Current Status, Trends, and Challenges Video Segmentation and Its Applications. In K. N. Ngan & H. Li (Eds.), (pp. 1-23): Springer New York.

Li, J., Ding, Y., Shi, Y., & Li, W. (2010). A divide-and-rule scheme for shot boundary detection based on sift. *International Journal of Digital Content Technology and its Applications, 4*(3), 202-214.

Li, S. B., Wang, L. F., & Wang, J. L. (2010). Video Segmentation Method Based on Film Script and Subtitle Information. *Computer Engineering, 15*, 077.

Li, W., Chen, T., Zhang, W., Shi, Y., & Li, J. (2012, May 1, 2012). *Music video shot segmentation using independent component analysis and keyframe extraction based on image complexity.* Paper presented at the Proc. SPIE 8334, Fourth International Conference on Digital Image Processing (ICDIP 2012), Kuala Lumpur, Malaysia.

Lienhart, R. (2001). Reliable transition detection in videos: A survey and practitioner's guide. *International Journal of Image and Graphics, 1*(03), 469-486.

Lin, G., Zhu, H., Fan, C., & Zhang, E. (2011). Object segmentation based on guided layering from video image. *Optical Engineering, 50*(9), 097006-097006. doi: 10.1117/1.3625415

Liu, L., Chen, R., Wolf, L., & Cohen-Or, D. (2010). *Optimizing Photo Composition.* Paper presented at the Computer Graphics Forum, 2010.

Liu, Z., Shen, H., Feng, G., & Hu, D. (2012). Tracking objects using shape context matching. *Neurocomputing, 83*(0), 47-55. doi: http://dx.doi.org/10.1016/j.neucom.2011.11.012

Luan, H., Zheng, Y.-T., Wang, M., & Chua, T.-S. (2011). VisionGo: Towards video retrieval with joint exploration of human and computer. *Information Sciences, 181*(19), 4197-4213. doi: http://dx.doi.org/10.1016/j.ins.2011.05.018

Ma, C., Yu, J., & Huang, B. (2012). A Rapid and Robust Method for Shot Boundary Detection and Classification in Uncompressed MPEG Video Sequences. *Computer Science Issues, International Journal of (IJCSI), 5*(2), 368-374.

Ma, Y., & Chen, Q. (2010). Stereo-Based Object Segmentation Combining Spatio-Temporal Information. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, R. Chung, R. Hammound, M. Hussain, T. Kar-Han, R. Crawfis, D. Thalmann, D. Kao & L. Avila (Eds.), *Advances in Visual Computing* (Vol. 6455, pp. 229-238): Springer Berlin / Heidelberg.

Mahesh, K., & Kuppusamy, K. (2012). Video Segmentation using Hybrid Segmentation Method. *European Journal of Scientific Research, ISSN*, 312-326.

Manjunath, B., Salembier, P., & Sikora, T. (2002). *Introduction to MPEG-7: multimedia content description interface* (Vol. 1): John Wiley & Sons Inc.

Marghitu, D. B. (2012, 9th October 2012). Centroids and centre of mass Retrieved 9th October, 2012, from http://www.eng.auburn.edu/users/marghdb/MECH2110/C_3.pdf

Mezaris, V., Papadopoulos, G. T., Briassouli, A., Kompatsiaris, I., & Strintzis, M. G. (2009). Semantic Video Analysis and Understanding. *chapter in "Encyclopedia of Information Science and Technology", Second Edition, Mehdi Khosrow-Pour.*

Mezaris, V., Sidiropoulos, P., Dimou, A., & Kompatsiaris, I. (2010). *On the use of visual soft semantics for video temporal decomposition to scenes.* Paper presented at the Proc. Forth IEEE Int. Conf. on Semantic Computing (ICSC 2010).

Microsoft. (2013, 2013). SkyDrive Retrieved February 20th, 2013, from https://skydrive.live.com/

Mika, P., & Greaves, M. (2012). Editorial: Semantic Web & Web 2.0. *Web Semantics: Science, Services and Agents on the World Wide Web, 6*(1).

Minter, R. (1999, 15.11.1999). QVC launches interactive shopping channel Retrieved 02.08.2012, 2012, from http://www.campaignlive.co.uk/news/31797/

Mitrović, D., Hartlieb, S., Zeppelzauer, M., & Zaharieva, M. (2010). Scene Segmentation in Artistic Archive Documentaries. In G. Leitner, M. Hitz & A. Holzinger (Eds.), *HCI in Work and Learning, Life and Leisure* (Vol. 6389, pp. 400-410): Springer Berlin / Heidelberg.

Moens, M. F., Poulisse, G. J., & VRT, M. M. (2012). State of the art on semantic retrieval of AV content beyond text resources.

Mohanta, P. P., Saha, S. K., & Chanda, B. (2010). *A heuristic algorithm for video scene detection using shot cluster sequence analysis.* Paper presented at the Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing, Chennai, India.

Mohanta, P. P., Saha, S. K., & Chanda, B. (2012). A Model-Based Shot Boundary Detection Technique Using Frame Transition Parameters. *Multimedia, IEEE Transactions on, 14*(1), 223-233. doi: 10.1109/tmm.2011.2170963

Money, A. G., & Agius, H. (2008). Video summarisation: A conceptual framework and survey of the state of the art. *J. Vis. Comun. Image Represent., 19*(2), 121-143. doi: 10.1016/j.jvcir.2007.04.002

MPEG. (2010, 2010). Information technology -- Multimedia content description interface -- Part 3: Visual ISO/IEC 15938-3. 2013, from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=34230

MPEG. (2012a, 2012). Information technology -- Multimedia content description interface -- Part 5: Multimedia Description Schemes ISO/IEC 15938-5. 2013, from http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=34232

MPEG. (2012b). Information technology -- Multimedia content description interface -- Part 12: Query format. *ISO/IEC 15938* ISO/IEC 15938-12:2012. from http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=61195

Noma, A., Graciano, A. B. V., Cesar Jr, R. M., Consularo, L. A., & Bloch, I. (2012). Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition, 45*(3), 1159-1179. doi: http://dx.doi.org/10.1016/j.patcog.2011.08.017

Ochs, P., & Brox, T. (2011, 6-13 Nov. 2011). *Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions.* Paper presented at the Computer Vision (ICCV), 2011 IEEE International Conference on.

Ohm, J.-R., Cieplinski, L., Kim, H. J., Krishnamachari, S., Manjunath, B., Messing, D. S., & Yamada, A. (2003). The MPEG-7 Color Descriptors.

Parmar, M., & Angelides, M. (2010). Automatic Feature Extraction to an MPEG-7 Content Model. *Advances in Semantic Media Adaptation and Personalization, 2*, 399.

Parmar, M. J. (2007). *Automatic feature extraction to COSMOS-7 content models.* Paper presented at the Semantic Media Adaptation and Personalization, Second International Workshop on.

Parmar, M. J., & Angelides, M. C. (2005). Multimedia Information Filtering.

Parmar, M. J., & Angelides, M. C. (2007). XML-based Genetic Rules for Scene Boundary Detection in a parallel processing environment. Retrieved from doi:http://bura.brunel.ac.uk/handle/2438/601

Porikli, F., Bashir, F., & Huifang, S. (2010). Compressed Domain Video Object Segmentation. *Circuits and Systems for Video Technology, IEEE Transactions on, 20*(1), 2-14. doi: 10.1109/tcsvt.2009.2020253

Poulisse, G.-J., Patsis, Y., & Moens, M.-F. (2012). Unsupervised scene detection and commentator building using multi-modal chains. *Multimedia Tools and Applications*, 1-17. doi: 10.1007/s11042-012-1086-0

Quan, Z., & Zhiwei, Z. (2011, 16-18 April 2011). *An MPEG-7 compatible video retrieval system with support for semantic queries.* Paper presented at the Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on.

Ren, W., Singh, S., Singh, M., & Zhu, Y. S. (2009). State-of-the-art on spatio-temporal information-based video retrieval. *Pattern Recognition, 42*(2), 267-282. doi: 10.1016/j.patcog.2008.08.033

Rey-López, M., Fernández-Vilas, A., Díaz-Redondo, R. P., López-Nores, M., Pazos-Arias, J. J., Gil-Solla, A., . . . García-Duque, J. (2010). Enhancing TV programmes with additional contents using MPEG-7 segmentation information. *Expert Systems with Applications, 37*(2), 1124-1133.

Richardson, I. (2010). *The H. 264 advanced video compression standard*: Wiley.

Rosman, B., & Ramamoorthy, S. (2011). Learning spatial relationships between objects. *The International Journal of Robotics Research, 30*(11), 1328-1342.

Ryoo, M. S., & Aggarwal, J. K. (2009, Sept. 29 2009-Oct. 2 2009). *Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities.* Paper presented at the Computer Vision, 2009 IEEE 12th International Conference on.

Ryoo, M. S., Lee, J. T., & Aggarwal, J. K. (2010). *Video scene analysis of interactions between humans and vehicles using event context.* Paper presented at the Proceedings of the ACM International Conference on Image and Video Retrieval, Xi'an, China.

Sakarya, U., & Telatar, Z. (2010). Video scene detection using graph-based representations. *Signal Processing: Image Communication, 25*(10), 774-783. doi: http://dx.doi.org/10.1016/j.image.2010.10.001

Sakarya, U., Telatar, Z., & Alatan, A. A. (2012). Dominant sets based movie scene detection. *Signal Processing, 92*(1), 107-119. doi: http://dx.doi.org/10.1016/j.sigpro.2011.06.010

Sang, J., & Xu, C. (2010). *Character-based movie summarization.* Paper presented at the Proceedings of the international conference on Multimedia, Firenze, Italy.

Sarmiento, A. S., & Lopez, E. M. (2012). *Multimedia Services and Streaming for Mobile Devices: Challenges and Innovation*: Information Science Reference.

Scott, K. (2012). BSkyB to launch pay-as-you-go IPTV service called Now TV Retrieved 21.08.12, 2012, from http://www.wired.co.uk/news/archive/2012-03/21/bskyb-launching-now-tv

Seeling, P. (2010). Scene Change Detection for Uncompressed Video. In M. Iskander, V. Kapila & M. A. Karim (Eds.), *Technological Developments in Education and Automation* (pp. 11-14): Springer Netherlands.

Seidl, M., Zeppelzauer, M., & Breiteneder, C. (2010). *A study of gradual transition detection in historic film material.* Paper presented at the Proceedings of the second workshop on eHeritage and digital art preservation, Firenze, Italy.

Seung-Bo, P., Heung-Nam, K., Hyunsik, K., & Geun-Sik, J. (2010, 13-15 Dec. 2010). *Exploiting Script-Subtitles Alignment to Scene Boundary Dectection in Movie.* Paper presented at the Multimedia (ISM), 2010 IEEE International Symposium on.

Shao, L., Ji, L., Liu, Y., & Zhang, J. (2012). Human action segmentation and recognition via motion and shape analysis. *Pattern Recognition Letters, 33*(4), 438-445. doi: http://dx.doi.org/10.1016/j.patrec.2011.05.015

Sharir, G., & Tuytelaars, T. (2012, 16-21 June 2012). *Video object proposals.* Paper presented at the Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on.

Sharmila Kumari, M., & Shekar, B. H. (2010, February 2010). *Color-SIFT model: a robust and an accurate shot boundary detection algorithm.* Paper presented at the Second International Conference on Digital Image Processing, Singapore, Singapore.

Sidiropoulos, P., Mezaris, V., Kompatsiaris, I., Meinedo, H., Bugalho, M., & Trancoso, I. (2011). Temporal Video Segmentation to Scenes Using High-Level Audiovisual Features. *Circuits and Systems for Video Technology, IEEE Transactions on, 21*(8), 1163-1177. doi: 10.1109/tcsvt.2011.2138830

Singhai, N., & Shandilya, S. K. (2010). A Survey On:"Content Based Image Retrieval Systems". *International Journal of Computer Applications IJCA, 4*(2), 22-26.

Smeaton, A. F., Over, P., & Doherty, A. R. (2010). Video shot boundary detection: Seven years of TRECVid activity. *Computer Vision and Image Understanding, 114*(4), 411-418.

Snoek, C. G. M., & Worring, M. (2009). Concept-Based Video Retrieval. *Found. Trends Inf. Retr., 2*(4), 215-322. doi: 10.1561/1500000014

Sofokleous, A. A., & Angelides, M. C. (2008). DCAF: an MPEG-21 dynamic content adaptation framework. *Multimedia Tools and Applications, 40*(2), 151-182.

Su, X., Bailan, F., Peng, D., & Bo, X. (2012, 25-30 March 2012). *Graph-based multi-modal scene detection for movie and teleplay.* Paper presented at the Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on.

Subudhi, B. N., Nanda, P. K., & Ghosh, A. (2011). *Moving objects detection from video sequences using fuzzy edge incorporated Markov random field modeling and local histogram matching.* Paper presented at the Proceedings of the 4th international conference on Pattern recognition and machine intelligence, Moscow, Russia.

SugarSync. (2013). SugarSync Retrieved February 20th, 2013, from https://www.sugarsync.com/

Tang, A., & Boring, S. (2012). *#EpicPlay: crowd-sourcing sports video highlights.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Austin, Texas, USA.

Tapu, R., & Zaharia, T. (2011). High Level Video Temporal Segmentation. *Advances in Visual Computing, 6938*, 224-235. doi: 10.1007/978-3-642-24028-7_21

Tapu, R., & Zaharia, T. (2011). Video Segmentation and Structuring for Indexing Applications. *International Journal of Multimedia Data Engineering and Management (IJMDEM), 2*(4), 38-58.

Tian, Z., Xue, J., Lan, X., Li, C., & Zheng, N. (2011). *Key object-based static video summarization.* Paper presented at the Proceedings of the 19th ACM international conference on Multimedia, Scottsdale, Arizona, USA.

Tjondronegoro, D. W., & Chen, Y. P. P. (2010). Knowledge-Discounted Event Detection in Sports Video. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 40*(5), 1009-1024. doi: 10.1109/tsmca.2010.2046729

Torres, R. S., Falcão, A. X., Gonçalves, M. A., Papa, J. P., Zhang, B., Fan, W., & Fox, E. A. (2009). A genetic programming framework for content-based image retrieval. *Pattern Recognition, 42*(2), 283-292.

Troncy, R., Bailer, W., Höffernig, M., & Hausenblas, M. (2010). VAMP: a service for validating MPEG-7 descriptions w.r.t. to formal profile definitions. *Multimedia Tools and Applications, 46*(2-3), 307-329. doi: 10.1007/s11042-009-0397-2

Tryon, C. (2012). 'Make any room your TV room': digital delivery and media mobility. *Screen, 53*(3), 287-300.

Tsao, H. H. (2011). *DCT Based Fast Object Detection and Segmentation Design for Compressed Video and Implementation on Embedded System.* Master, National Yunlin University of Science and Technology, Douliu City, Yunlin County, Taiwan. Retrieved from http://ethesys.yuntech.edu.tw/ETD-db/ETD-search/getfile?URN=etd-0819111-142117&filename=etd-0819111-142117.pdf

Tsinaraki, C., & Christodoulakis, S. (2011). Domain Knowledge Representation in Semantic MPEG 7 Descriptions. *The Handbook of MPEG Applications*, 293-316.

Tsinaraki, C., Polydoros, P., & Christodoulakis, S. (2004). *Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant Semantic Indexing*. Paper presented at the Advanced Information Systems Engineering.

Tsingalis, I., Vretos, N., Nikolaidis, N., & Pitas, I. (2012, 25-28 March 2012). *Anthropocentric descriptors and description schemes for multi-view video content*. Paper presented at the Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean.

Tuzel, O., Porikli, F., & Meer, P. (2006). Region Covariance: A Fast Descriptor for Detection and Classification. In A. Leonardis, H. Bischof & A. Pinz (Eds.), *Computer Vision – ECCV 2006* (Vol. 3952, pp. 589-600): Springer Berlin Heidelberg.

Van den Bergh, M., & Van Gool, L. (2012, 9-11 Jan. 2012). *Real-time stereo and flow-based video segmentation with superpixels*. Paper presented at the Applications of Computer Vision (WACV), 2012 IEEE Workshop on.

Vazquez-Reina, A., Avidan, S., Pfister, H., & Miller, E. (2010). Multiple Hypothesis Video Segmentation from Superpixel Flows. *Computer Vision – ECCV 2010, 6315*, 268-281. doi: 10.1007/978-3-642-15555-0_20

Ventura, C., Martos, M., Giró-i-Nieto, X., Vilaplana, V., & Marqués, F. (2012). Hierarchical Navigation and Visual Search for Video Keyframe Retrieval. In K. Schoeffmann, B. Merialdo, A. Hauptmann, C.-W. Ngo, Y. Andreopoulos & C. Breiteneder (Eds.), *Advances in Multimedia Modeling* (Vol. 7131, pp. 652-654): Springer Berlin Heidelberg.

Vijayakumar, V., & Nedunchezhian, R. (2012). A study on video data mining. *International Journal of Multimedia Information Retrieval, 1*(3), 153-172. doi: 10.1007/s13735-012-0016-2

Visser, A. (2011). On the ambiguation of Polish notation. *Theoretical Computer Science.*

Vrochidis, S., Moumtzidou, A., King, P., Dimou, A., Mezaris, V., & Kompatsiaris, I. (2010, 23-25 June 2010). *VERGE: A video interactive retrieval engine*. Paper presented at the Content-Based Multimedia Indexing (CBMI), 2010 International Workshop on.

W3C. (2007, 14/08/2007). MPEG-7 and the Semantic Web Retrieved 24/05/13, 2013, from http://www.w3.org/2005/Incubator/mmsem/XGR-mpeg7/#conclusions

W.S. Anderson, P. (2004, 22/10/2004). AVP: Alien vs. Predator, from http://www.imdb.com/title/tt0370263/

Wang, H. H., Mohamad, D., & Ismail, N. (2010). Semantic Gap in CBIR: Automatic Objects Spatial Relationships Semantic Extraction and Representation. *International Journal Of Image Processing (IJIP), 4*(3), 192.

Weiming, H., Nianhua, X., Li, L., Xianglin, Z., & Maybank, S. (2011). A Survey on Visual Content-Based Video Indexing and Retrieval. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 41*(6), 797-819. doi: 10.1109/tsmcc.2011.2109710

Westland, S., Laycock, K., Cheung, V., Henry, P., & Mahyar, F. (2012). Colour harmony. *JAIC-Journal of the International Colour Association, 1.*

Which? (2009). Virgin Media V+ HD review Retrieved 04.10.12, 2012, from http://www.which.co.uk/technology/tv-and-dvd/reviews/pvrs/virgin-media-v--hd/review/

Williams, C. (2006). BT Vision is go Retrieved 04.07.2012, 2012, from http://www.theregister.co.uk/2006/12/04/bt_vision_launch/

Wilson, K. W., Divakaran, A., Niu, F., Goela, N., & Otsuka, I. (2010). Method for detecting scene boundaries in genre independent videos: Google Patents.

Wolf, M., & Wicksteed, C. (1998). Date and time formats. *W3C NOTE NOTE-datetime-19980827, August.*

Wollborn, M. (2010). USA Patent No. 7697613. Google Patents: U. S. Patents.

Wu, J., Liu, Y., Wang, J., & Cai, X. (2012, 16-19 July 2012). *A geographic information based video segmentation method*. Paper presented at the System of Systems Engineering (SoSE), 2012 7th International Conference on.

Xu, W., & Xu, L. (2010, 16-18 April 2010). *A novel shot detection algorithm based on graph theory.* Paper presented at the Computer Engineering and Technology (ICCET), 2010 2nd International Conference on.

Xue, L., Li, C., Wu, Y., & Xiong, Z. (2009a). VeXQuery: an XQuery extension for MPEG-7 vector-based feature query *Advanced Internet Based Systems and Applications* (pp. 34-43): Springer.

Xue, L., Li, C., Wu, Y., & Xiong, Z. (2009b). VeXQuery: An XQuery Extension for MPEG-7 Vector-Based Feature Query. In E. Damiani, K. Yetongnon, R. Chbeir & A. Dipanda (Eds.), *Advanced Internet Based Systems and Applications* (Vol. 4879, pp. 34-43): Springer Berlin / Heidelberg.

Yongquan, X., Weili, L., & Shaohui, N. (2009, 7-8 Nov. 2009). *A Simple and Fast Segmentation Approach for Sport Scene Images.* Paper presented at the Artificial Intelligence and Computational Intelligence, 2009. AICI '09. International Conference on.

Zajić, G. J., Reljin, I. S., & Reljin, B. D. (2011). Video Shot Boundary Detection based on Multifractal Analisys. *Telfor Journal, 3*(2), 105-110.

Zavřel, V., Batko, M., & Zezula, P. (2010). *Visual video retrieval system using MPEG-7 descriptors.* Paper presented at the Proceedings of the Third International Conference on SImilarity Search and APplications, Istanbul, Turkey.

Zeng, X., Zhang, X., Hu, W., & Li, W. (2010). Video Scene Segmentation Using Time Constraint Dominant-Set Clustering. In S. Boll, Q. Tian, L. Zhang, Z. Zhang & Y.-P. Chen (Eds.), *Advances in Multimedia Modeling* (Vol. 5916, pp. 637-643): Springer Berlin / Heidelberg.

Zhenyu, Y., & Zhiping, L. (2012, 18-20 July 2012). *Scene change detection using motion vectors and dc components of prediction residual in H.264 compressed videos.* Paper presented at the Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on.

Zhu, Q., Xie, Y., Gu, J., & Wang, L. (2012). A New Video Object Segmentation Algorithm by Fusion of Spatio-temporal Information Based on GMM Learning. In G. Lee (Ed.), *Advances in Automation and Robotics, Vol. 2* (Vol. 123, pp. 641-650): Springer Berlin Heidelberg.

Zhu, S., & Guo, Z. (2012, 23-25 Aug. 2012). *An Overview of Video Object Segmentation.* Paper presented at the Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on.

Zhu, S., & Liang, Z. (2011). Semantic scene segmentation for advanced story retrieval. *Information Technology Journal, 10*(1), 98-105.