Highly Automated Method for Facial Expression Synthesis

A Thesis
Presented to
The Academic Faculty

By
Nikolaos Ersotelos

A' Supervisor: Prof. Feng Dong
B' Supervisor: Prof. Marios Angelides

Submitted for the degree of
Doctor of Philosophy

Department of Information Systems and Computing (DISC)

Brunel University

March 2010

Dedication

To my parents Theologo and Kalliopi Ersotelou, for supporting me financially, psychologically in all my academic studies.

Acknowledgments

I would like to thank all those people who supported and encouraged me during my work on this project. Without their help I would not have achieved my goal.

Firstly, my supervisor, Professor Feng Dong, who I first met at Lancaster University in a project based on the synthesis of 3D objects using C++. I have to admit that before then I did not have any knowledge of programming and I had never thought I would become involved in computer science. However, at that first meeting, he inspired me and taught me, even privately, the first principles of programming. From his first lesson I selected his project as my final Master's Dissertation, for which I achieved a distinction and eventually, publication. Since then I have not given up in spite of all the difficulties I have faced because I did not want to disappoint him.

I never expected that he would offer me a PhD research position during his first months at Brunel University, and even when he moved to Bedfordshire University to be a Professor he has supported me by coming to Uxbridge and giving me tutorials. I shall always remember, with nostalgia, our lunch break talks about academic and research issues.

I also would like to thank him for his encouragement when I learned that my father was suffering from leukemia. He contacted me almost every day, asking about my father's health.

Secondly, I would like to thank the Institute 'Nikolaos Passadelis' for their scholarship and financial support.

Finally, I would like to dedicate this work to my family: Theologos, Kalliopi and Michael. Without their understanding support and love from childhood until now I would never have made it through this process, or through any of the tough times in my life.

Thank you.

Contents

Figure Contents

Table Contents

Abstract

The synthesis of realistic facial expressions has been an unexplored area for computer graphics scientists. Over the last three decades, several different construction methods have been formulated in order to obtain natural graphic results. Despite these advancements, though, current techniques still require costly resources, heavy user intervention and specific training and outcomes are still not completely realistic. This thesis, therefore, aims to achieve an automated synthesis that will produce realistic facial expressions at a low cost.

This thesis, proposes a highly automated approach for achieving a realistic facial expression synthesis, which allows for enhanced performance in speed (3 minutes processing time maximum) and quality with a minimum of user intervention. It will also demonstrate a highly technical and automated method of facial feature detection, by allowing users to obtain their desired facial expression synthesis with minimal physical input. Moreover, it will describe a novel approach to the normalization of the illumination settings values between source and target images, thereby allowing the algorithm to work accurately, even in different lighting conditions.

Finally, we will present the results obtained from the proposed techniques, together with our conclusions, at the end of the paper.

Chapter 1 - Introduction

1.1 Introduction

One of the most important characteristics of human beings is the ability to communicate in several ways for many different purposes; primarily, they communicate with each other to express their thoughts and beliefs and to give instructions or show their feelings. Apart from by auditory means, such as speaking and singing and tone of voice, humans communicate physically by using body language, sign language, paralanguage, touch, eye contact, or writing and drawing. In the context of the face, without verbal communication, a change of facial expression indicates a change of mood, much as a painting can capture a sentiment.

Because faces have only a limited range of movement, expressions rely on minuscule differences in the proportions, and the relative positions, of their features. According to portrait painting canons, the difficulty of creating a realistic expression lies in the illumination settings of the finer details of the features – such as creases and wrinkles. However, facial expression synthesis, based only on a geometrical deformation of the features, lacks fine details, because those wrinkles, creases and illumination settings are required to portray a realistic and recognisable expression.

The main aim of this thesis, therefore, is to present novel techniques for detecting automatically the appropriate features and synthesizing a realistic expression, based on both the geometrical deformation and on the synthesis of realistic illuminative settings.

Generally, the realistic synthesis of facial expression is one of the most fundamental problems in computer graphics. Since the early 1970s many research papers and books have been published describing either innovative methods or improvements to existing ones. Even if the results have been, in some cases, very close to naturalism, they still do not satisfy the four factors specified by Magnenat-Thalmann [32] in order to provide a fully functional and accurate graphic result: a) simple technical equipment, such as a personal computer, b) a single facial target image, c) an ‘automatic’ algorithm, which does not require the user to be trained, and d) the quickest possible result.

The reason why an appropriate research has not yet been implemented that satisfies the above requirements is because of the problem of re-synthesizing, with

sufficient accuracy, characteristics that differ from face to face, such as the muscles that are activated with every expression, the shape of the features and the appropriate illumination settings for each expression.

In the literature review chapter we present several approaches for synthesizing expressions by categorizing most of them into two sections: (a) synthesizing, and, (b) manipulating 3D facial models or 2D images. The techniques various researchers have used for establishing appropriate results vary, but no-one, to date, has been able to satisfy all the above requirements.

The main idea of this research, therefore, was to construct an innovative algorithm that would satisfy all four requirements. With that in mind, we firstly discussed the fundamental principles that lay behind each existing technique, together with their strengths and limitations, from which we developed a new algorithm that combines the benefits of the most effective methods and improvisations in order to produce more accurate, automatic and speedy results.

Consequently, this thesis will present an analytical approach for detecting automatically or manually, all features, whilst eliminating, as far as possible, user-interaction. It will also offer an improved illumination transfer process, which will work with all picture lightings and in all colors.

1.2 Aims and Objectives

This thesis has been motivated by the large amount of existing facial animation work and the recent advances in Image Based Modeling and Rendering (IBMR), which uses 2D and 3D models to achieve quick and realistic results, by utilising images, rather than polygons. This novel approach allows for the synthesis of a single picture expression, or video animation, by using one input image and a group of facial expression source images.

Our primary goal is to generate a lower cost, more automated algorithm to synthesize new expressions that will require only occasional and minimal intervention from the user. We consider that we have achieved this by creating the following key procedures, which, we believe, will be of great scientific interest:

- a human facial expression generator
- an automatic facial features detection procedure
- an image colour normalization procedure, and

- an image illumination settings transfer procedure

Consequently, the methods developed in this thesis emphasize speed and efficiency. All the algorithms were initially implemented in Microsoft Visual Studio 6 and all the implementations have been made using C++.

1.2.1 The human facial expression generator

The 'human facial expression generator' is a highly automated algorithm for synthesizing facial expressions. Firstly, the algorithm geometrically deforms specific features in order to give the new expression; it then transfers the appropriate illumination settings from the source images, which provide the predefined original expression, including natural wrinkles, creases and lighting conditions.

Our algorithm is an improved version of Liu's, [31] which was designed for transferring a facial expression from one person to another. Our approach, however, produces faster and more accurate results based on three factors, (a) automatic features detection, (b) extraction of specific facial areas for geometrical deformation and, (c) an illumination transfer process and colour-lighting settings threshold.

Instead of defining all the facial features – ears, hair, neck, eyes, eyebrows, nose, mouth, cheek, etc – our algorithm extracts the eyes, eyebrows and mouth areas from the images, then, after the deformation process is complete, places them back in their original positions. The advantages of this process is the reduction of the processing time and the simplification of the geometrical deformation because only small areas of the image are affected, thereby avoiding possible distortions.

1.2.2 Automatic facial features detection procedure

We will present two automatic algorithms for the detection and definition of specified facial features that affect the target image in order to execute the whole process, as far as is possible, automatically. This process is based on an edge detection algorithm by which the my code can calculate the size, the shape and the position of the features in relation to their edges, by surrounding them with dots, which will be used in the geometrical deformation and colour ratio process.

1.2.3 Image colour normalization procedure

We will outline a novel approach for normalizing the illumination settings values between the source and the target images in order for the algorithm to work accurately, even in different lighting conditions.

1.2.4 Image illumination settings transfer procedure

We will suggest a threshold, corresponding to ERI, for transferring a specific percentage of illumination settings data to the target image. This threshold is valuable when the lighting, or color conditions, of source images are different. In the final, synthesized result, the illumination threshold, combined with image color normalization, provides high graphic details.

1.3 Applications

One of the main reasons for researching the algorithm for facial graphics improvisation is to explore its several potential applications, such as in the movie industry, the computer games industry, video teleconferencing and personal image manipulation.

1.3.1 The Movie Industry

The movie industry has received huge benefit from the advance of facial modeling and animation techniques. The increasing number of computer-made movies, such as Toy Story, Shrek, Monsters Inc, Monster House and King Kong, has demonstrated that the current techniques are well suited to the creation of cartoon-styled films. To this end, traditional techniques, such as geometrical face modeling and simulation-based facial animation [27, 46, 50], have proved to be quite competent for the purpose.

However, major work still needs to be done in order to achieve highly realistic results. Due to ‘post-processing’ procedures, movie production usually allows for a considerably long time-lapse in order for the desired quality to be achieved. Hence, high cost resources are tolerated, such as the creation of large face models, which can contain minute details when captured from 3D scanners; however, facial motion capture uses costly equipment and considerable ‘man-hours’. Therefore, given the current state of the art, there is still a long way to go before artificial looks can be

completely removed from computer synthesized faces and for products to be created that are indistinguishable from real faces.

We believe that by using our algorithm the movie industry will be able to create realistic animated movies featuring famous actors or actresses who have stopped producing movies; the only additional requirement being the employment of an actor as the source model, together with a 'neutral' facial picture of the subject actor or actress.

1.3.2 Computer Games

The nature of computer games implies that rapid response speed and, in many instances, the need for real-time processing is essential. Given that a large number of computer games are designed for hand-held devices nowadays, many employ moderate quality face-models with artificial facial movement in order to facilitate the necessary speed.

Highly accurate computation for facial animation, such as anatomy-based simulation, or high degree of shape-blending (interpolation), is not usually necessary for computer games. To maintain a good balance between speed and image quality, pre-processed, or captured, facial movement data can be stored, thereby reducing the negative aspects of real-time computing. Although this algorithm only supports a limited number of facial expressions, it is considered to be suitable in the context of current computer game design; therefore, many games rely on limited facial expressions.

As a consequence, given the above limitations of technique and power, there is an immense barrier to overcome before totally free-form and realistic facial animation in computer games can be achieved.

1.3.3 Video Teleconferencing

The capability of displaying a talking face is always a desirable feature in telecommunication. Here the challenge is to reduce the time-lag whilst improving the display quality. Due to bandwidth limitations, transmitting high resolution face images in practice is usually prohibitive. A feasible solution, therefore, is to store, locally, high-resolution generic face model reproductions of those involved in the communication, which will necessitate transmission of the various movements only, thereby reducing bandwidth costs.

However, because a fully automatic model adaptation has not, as yet, been available, the manual identification of facial features has been necessary; also, completely traceless, fast and reliable facial performance tracking has proved to be a challenge. Our algorithm, therefore, is capable of enhancing video teleconferencing by using a database of facial models with several expressions that can be incorporated into the target image in order to improve facial animation in order to facilitate a 'tele-animation' conference.

1.3.4 Personal Image Manipulation

Already, several companies, such as Adobe, Paint Pro Shop, and even some cameras manufacturers, have invested in research in order to fix, improve or manipulate images and videos. Our algorithm can easily be used for such personal image manipulation, where, by using the relevant software, the user may easily select the chosen 'personal image manipulation option' in order to synthesize facial expressions, or even facilitate video animation.

1.4 Outline of thesis

The remainder of this thesis is organized as follows: in Chapter 2 we will survey some of the most important approaches in facial modeling and animation. In Chapter 3 we will analyze the general methodology, starting from the extraction of the facial features from the source and target images and the analysis of the expressions database. In Chapter 4 we will present the Sobel Edge Detection method, together with noise reduction process specifications. We will also present the automatic 1, the automatic 2, and the manual facial features detection processes. In Chapter 5 we will present the algorithm, based on geometrical deformation, which generates the new expression. In Chapter 6 we will present the techniques for transferring facial illumination details for expression synthesis. In Chapter 7 we will present a novel approach for synthesizing a facial video animation, which will be followed in Chapter 8 by an examination of several results, based on the above algorithms; we will also discuss the advantages and limitations of the process and offer some proposals for the direction of future research. The thesis will conclude in Chapter 9 with a summary of the whole process.

1.4.1 Literature Review

Chapter 2 provides a comprehensive survey on the techniques for human facial expressions modeling and animation synthesis. We divide all the techniques into two categories: facial modeling, which involves making 3D models, and facial animation, which involves the making of synthetic facial expressions. To generate an individual face model we either 'individualize' a generic model, or combine a number of models from an existing face collection. Regarding facial animation, we have further categorized the techniques into simulation based, performance driven and blend-shape based approaches. We shall discuss the strengths and weakness of these techniques and draw conclusions at the end of the chapter.

1.4.2 Theoretical analysis of the process; detection of specific facial features and presentation of the Facial Expression Database

In Chapter 3 we shall present a summary based on a diagram of the whole process for creating a realistic facial expression. We will identify all the target image requirements necessary for the algorithm to work accurately and we will also explain why specific features were selected in order to produce the expressions. We will follow this by an analysis of the facial expression database to demonstrate its importance in the process.

1.4.3 Facial features identification processes

In Chapter 4 we will present three ways of identifying the features of the target image. Firstly, we will describe the automatic 1 approach, where the algorithm automatically defines the features; this can, if necessary, be corrected manually. Secondly, we will give the innovative specifications for the automatic 2 process detection and the placement of dots around them that define the facial characteristics. Finally, we will explain the manual intervention process for defining the facial features.

In both automatic processes, the algorithm will firstly transform the colour image to a greyscale image; this simple differentiation between black and white pixels, rather than the thousands that constitute colours and their numerous different lighting conditions, is important, because it is much less complicated. After this, the edge detection process, which utilises rectangular masks, will produce a series of dots to identify and position the eyes, eyebrows and mouth shapes. The algorithm will also reduce noise. Finally, the correct position and coordinates of the dots will be copied

and placed on the original colour target image for the geometrical deformation and illumination transfer processes.

1.4.4 Facial expression generation

In Chapter 5 we will present an algorithm for the geometrical deformation of the facial features. In this process the target image will be morphed according to the expression selected from the facial expression database. Moreover, solutions will be provided should there be distorted areas and the user will be shown how to remove them. Results of geometrical deformations, based on different facial expressions, will be provided.

1.4.5 Facial details generation

In Chapter 6 we will demonstrate the 'illumination settings transfer' process. Firstly, by using geometrical deformation again, the algorithm will equalize the source images' facial features in shape and position with the new synthesized facial expression of the target image. Subsequently, each of the source images will be divided by 50% with the target image in order for their pixel values to be combined. The two new images, when synthesized, will be added to the deformed target picture. The amount of the illumination data to be added to the target image will depend on the threshold settings, which the user will select. Solutions for removing possible distorted areas will also be provided.

At the end of the process the user will have the option to add a Gaussian filter to the final result in order to normalize the colours to the lighting setting.

1.4.6 Video animation generator

Another function that the algorithm can provide is the synthesis of six facial expression images simultaneously for video editing animation (see Chapter 7). As with the individual facial expression synthesis process, the algorithm contains groups of source animated pictures together with their documentations, which are stored in the library. When an expression is selected by the user, the edge detection of the features of the target image begins. Even if the user intervenes in the process by correcting the detected dots, the algorithm stores their positions in a temporal folder for re-use. All the synthesized target images are produced by taking, as inputs, the

neutral expression of both the source and the target images, together with the subsequent source image expression.

After the sequence of the target facial animation images have been synthesized, video editing software may be used to render the images in a video format; thus, by using a 'fade in fade out' video effect, very realistic results of facial video animation can be achieved.

1.4.7 Results

In Chapter 8 we will show several results based on all the processes, together with an analysis of the deformation and illumination transfer settings; from these the reader will be able to see the algorithm's effectiveness and consistency in synthesizing realistic facial expressions. It must also be mentioned that our results will be included in each chapter in order to provide clearer information during each process.

1.4.8 Conclusion

In Chapter 9 we will present a summary of the capability of our algorithm and a discussion regarding its importance to the computer graphics industry, together with suggestions for future improvisations.

Chapter 2 Literature Review

2.1 Introduction

Human figure synthesis has been one of the most difficult problems faced by the computer graphics industry, with facial expression and animation being the most significant aspects. Two decades have passed since Parke's, Water's, and Koch's pioneering work in the animation of faces [35, 36, 50 and 25] and during that time a significant effort has been devoted to the development of computational facial models in applications over such diverse areas as entertainment, low bandwidth teleconferencing, surgical facial planning and virtual reality. However, the task of accurately modeling the expressive human face by computer still remains a major challenge.

Since the appearance of these pioneer works, significant progress has been made by computer graphics researchers, who have developed numerous techniques in order to generate good quality facial models with highly realistic expressions. However, despite these efforts, computer synthesized human facial animation still requires costly resources and sometimes involves considerable manual labor. Cost-effective solutions, therefore, are still sought in order to attain fully realistic facial animation.

Because of the complexities of facial anatomy, our natural sensitivity to facial appearance and the fact that no time algorithm exists that is capable of capturing on an avatar the subtleties of expression and emotion, the most ambitious attempts to attain realism through modeling and rendering in real time have proved unsuccessful. Therefore, the ultimate goal in facial modeling and animation is to develop a algorithm that 1) creates realistic animation, 2) operates in real time, 3) is automated as much as possible, and, 4) depts. easily to individual faces.

The recent interest in facial modeling and animation has been spurred by an increase in the appearance of virtual characters in film and video, inexpensive desktop processing power and the potential for a new 3d immersive communication metaphor for human-computer interaction.

Much facial modeling and animation research is published in specialist papers that are relatively unknown to the general graphics community; consequently there have been few surveys and detailed historical treatments of the subject. This chapter,

therefore, aims to be an accessible reference to the range of reported facial modeling and animation techniques.

Facial modeling and animation research falls into two major categories: geometric manipulations and image manipulations, with each category comprising several sub-categories. Geometric manipulations include key-framing and geometric interpolations, parameterizations, finite element methods; muscle based modeling, visual simulation using pseudo muscles, spline models and free-form deformations. Image manipulations include image morphing between photographic images, texture manipulations, image blending, and vascular expressions. At the pre-processing stage, a person-specific individual model may be constructed using anthropometry, scattered data interpolation, or by projecting target and source meshes onto spherical or cylindrical coordinates. 'Such individual models are often animated by feature tracking, or performance driven, animation' [15].

Facial massage and animation are strongly correlated. In fact, the production of realistic facial animation is often applied to modeling techniques – for example, in the building of multiple face poses, or in carrying out the deformation of face models in order to achieve the desired expressions. Consequently, the quality of facial animation is determined by the methods used in facial massage modeling and facial animation. Such a report/ratio is shown later in this chapter.

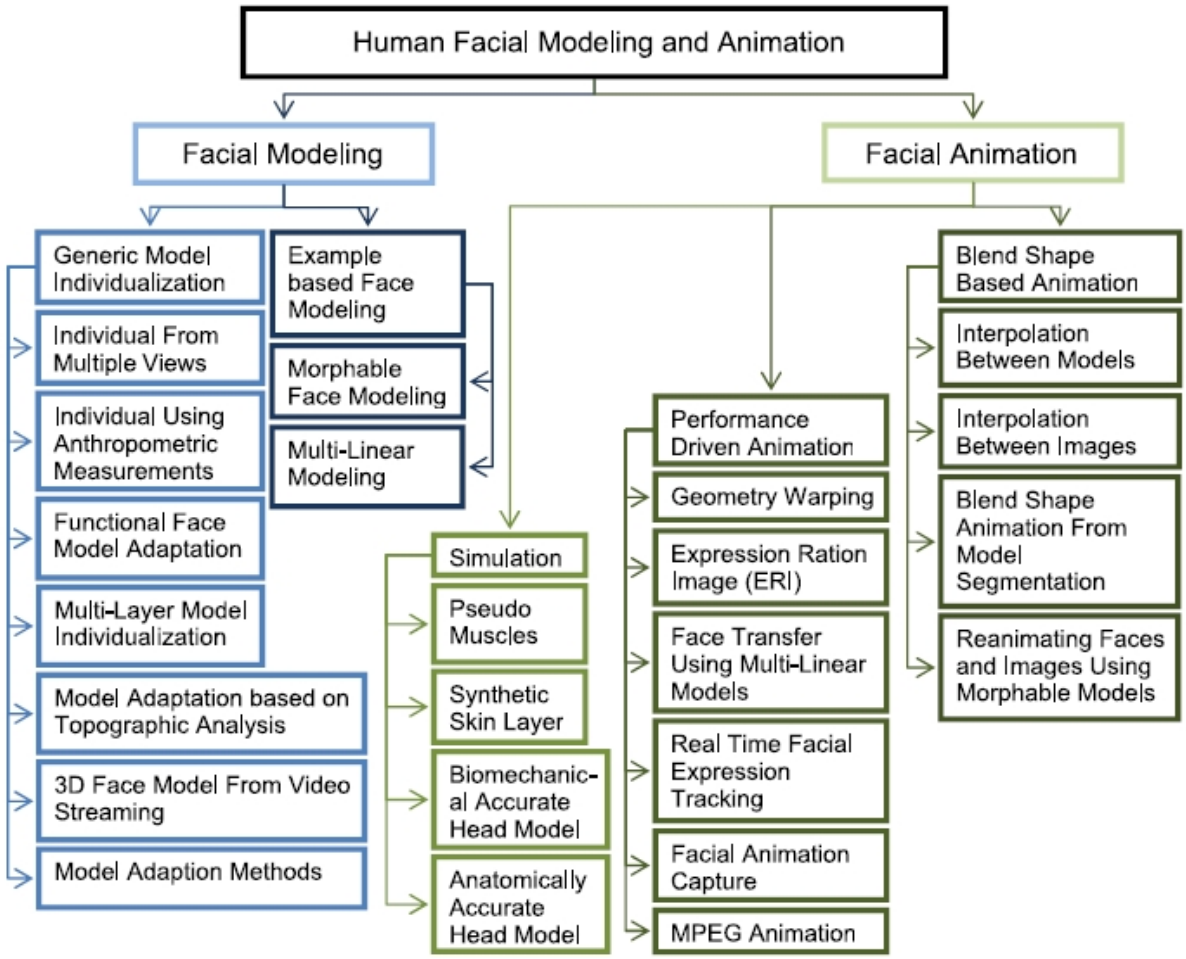


Figure 1: Categorization of facial modeling and animation methods *[13]*.

2.2 Brief history

Great interest has been generated in the computer simulation of the human face and its expressive movements during the last few decades. A facial expression is the result of a confluence of muscle contractions that deform the neutral face into an expressive face and this constitutes a primary form of human visual communication. Ekman has cataloged 55,000 distinguishable expressions, with about 30 semantic distinctions, by identifying six primary emotional expressions – anger, disgust, fear, happiness, sadness and surprise. Together with Friesen, Ekman [12] has proposed the Facial Action Coding Algorithm (FACS) – a quantified abstraction of the actions of facial muscles – as a means of recording facial expressions independent of cultural or personal interpretation. The FACS represents facial expressions in terms of 66 action

units (AUs), which involve one or more muscles and associated activation levels [28]. The AUs are grouped into those that affect the upper and the lower face, and include vertical actions, horizontal actions, oblique actions, orbital actions, and miscellaneous actions, such as nostril shape, jaw drop and head and eye position.

Early work on computer facial modeling and animation dates to the 1970s, when the first 3D facial animation was created by Parke [35]. This was followed by a few landmark works in the 1980s, such as the deformable face model [3] based on a 3D facial mesh – "pseudo-muscles" – and the classic work on facial animation using specified muscles areas [50].

In recent years, in order to better understand the mechanism of facial movements, high quality, anatomically-based simulations have been created, including the principal 'multi-layered model', which comprises the skin, the muscles, the skeleton, etc, and the 'volumetric model', requiring an understanding of the muscles, fabrics etc, which covers all the principal facial structures. However, in spite of recent technical advances, currently available techniques still cannot meet the requirements of many of the required applications. Therefore, in order to offset these limitations, the calculations required for the creation of anatomically accurate models had to be simplified. However, imaging and video analysis – the 'execution-control' approach – requires highly exacting equipment, therefore facial massage and animation modeling research will continue to be a crucial element in the field. There is still, though, a long way to go before a satisfactory technological solution is arrived at.

2.3 Facial modeling

This section includes the techniques regarding the synthesis of a high quality 3D head.

Two basic methods may be used for constructing a 3D head: firstly, a triangular mesh that consists of connecting dots along the edges of triangles, and, secondly, a laser cylindrical scanner, such as those produced by a Cyberware scanner [8] as used by Yuencheng Lee and Demetri Terzopoulos [27] who created highly realistic models, with facial expressions, based on pseudo muscles consisting of several layers of triangles simulating the skin, nerves, skull, etc; thus, by changing the settings of the triangular meshes (pseudo muscles), new expressions can be synthesized.

Over the last few years, researchers have proposed and developed various other techniques for the production of quality face models, and these can be divided into two categories:

- A generic model individualization, which is based on the idea of creating a face model for a specific subject by carrying out feature-based deformation to a generic model
- An example-based face model, which creates a model that has desired facial features through the linear combinations of existing facial models collection.

2.3.1 Generic model individualization

Generic model individualization, also called model adaptation, is the procedure of deforming a general model in order to create a new facial model for a specific person. In order to create the model, the facial characteristics and positions of the specific person, such as his/her mouth, nose and eyes, are used to align the corresponding characteristics of the general model. The various elements used as inputs for model adaptation are:

1. Facial characteristics, which may be extracted manually by using a number of multi-viewed photographs [39] *(further details follow in section 2.3.1.1)*.
2. Anthropometric measurements, which may be deployed to describe faces with particular features [10] *(more details follow in sections 2.3.1.2, 2.3.1.3 and 2.3.1.4)*.
3. Frontal view image and image analysis of the individual face, which may be used to detect the facial features [55] *(more details follow in section 2.3.1.5)*.
4. Video streams, which may be captured by multiple cameras [59] *(more details follow in section 2.3.1.6)*.

These model individualization methods will be compared in terms of their efficiency and effectiveness in section 2.3.1.7

2.3.1.1 Individualization from multiple views

Generating facial models constitutes an important step in the work of Pighin [39] in the creation of realistic 3D animation depicting several expressions. A generic model, adapted to synthesize an individual subject, is derived from several digitized facial

photographs taken from different viewpoints in order to define the position of the face on the images and the 3D model.

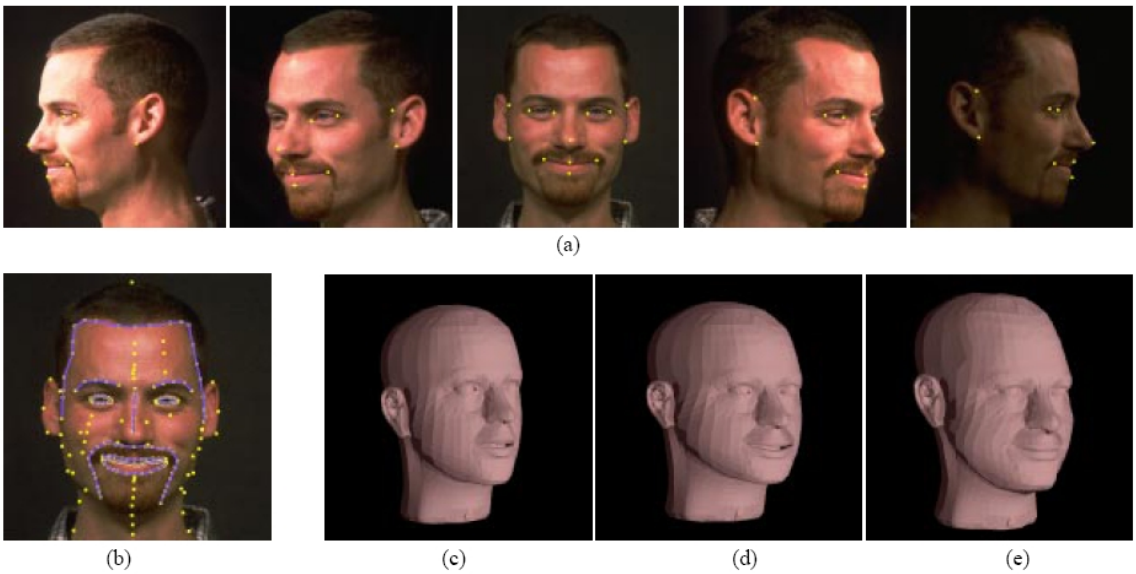


Figure 2: Model-fitting process: *a) a set of input images with marked feature points, b) facial features annotated using a set of curves, c) generic face geometry (shaded surface rendering), d) face adapted to initial 13 feature points (after pose estimation) and, e) the face after 99 additional correspondences have been given [39].*

To help in the adaptation process, the algorithm moves the corner of the eyes, bulge of nose and corners of mouth that have been manually identified in all the pictures. A scattered data interpolation technique is then used to deform the generic model to fit into the feature points identified from the images. This process is repeated for several facial expressions, creating a generic model for each expression, whilst, simultaneously, a 3D model shape morphing technique is applied in order to differentiate the various expressions.

However, creating a set of computations to produce a series of meshes with different facial expression is time consuming; also, the gaze direction of the eyes is fixed in all the models. This algorithm was mainly designed for offline authoring purposes and it requires a user to specify blending weights manually in order to obtain the desired expression.

2.3.1.2 Individualization using anthropometric measurements

Anthropometrics science data libraries have also been used as principal element in the accurate synthesis of 3D models.

Anthropometry is the biological science of human body measurement and it informs a range of enterprises that depend on an understanding of the distribution of measurements across human populations. For example, in human-factor analysis, a known range for human measurements can help guide the design of products to fit most people; in medicine, quantitative comparison of anthropometric data with patients' measurements before and after surgery furthers planning and assessment of plastic and reconstructive surgery and for constructing realistic 3D models. Anthropometry libraries, therefore, contain data characterizing human bodies by gender, color, age, size and shape.

DeCarlo et al [10] approach is based on facial anthropometric measurements, which are the fundamental elements used to generate a geometric 3D head model. The first stage – the characteristics of the human face (e.g. in Figure.3 below, "ex" left corner of the eye, "en" right corner) – is determined by measuring the distances between the points derived (an accurate 3D model might need a hundred and thirty such points). The second stage – variational modeling – is a framework for generating geometric surfaces according to anthropometric measurement optimization, the algorithm for which creates a static facial surface. The anthropometric data that DeCarlo et al [10] uses in his algorithm are based on the Farkas [14] algorithm.

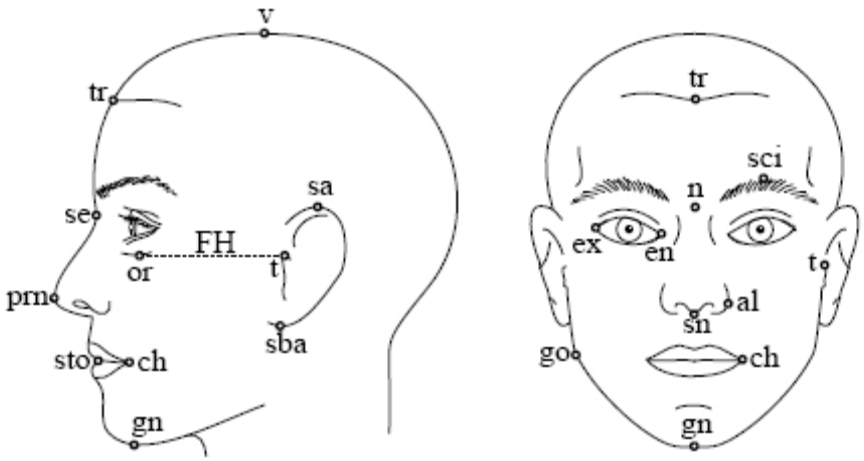


Figure 3: Anthropometric points on the face *[10]*

A major disadvantage of the DeCarlo approach is that it is based on statistical facial measurements, for instance, it cannot create a hooked nose or double chin. Moreover, the optimization process requires much calculation time in order to generate a model.

2.3.1.3 Functional face model adaptation

Zhang et al. [58] presented an algorithm to achieve functional face model adaptation, based on the principle that the generic model is equipped with a number of pseudo-muscles to support facial animation. More specifically, it is necessary for a small set of anthropometric landmarks to be defined on the 2D images of both the generic and the scanned models. Thereafter, the algorithm automatically recovers the 3D position of the landmarks by using a projection mapping approach. Later, a global adaptation procedure is carried out in order to adapt size, position and orientation of the generic model towards the scanned model, referring to a series of measurements based on the recovered 3D landmarks. A local adaptation procedure is then initiated to deform the generic model in order to fit all its vertices to the scanned model. Meanwhile, the underlying muscle structure of the generic model is automatically adapted so that the reconstructed model might resemble the facial animation based on the adjusted pseudo muscles. In this way, efficient face model adaptation is achieved, not only in terms of color and shape, but also in terms of animation.

2.3.1.4 Multi-layer model individualization

The DeCarlo et al [10] and Kahler et al [22] approach continued into animated facial 3D models and expressions based on 'landmark' data (see Figure 4 a-e below) where, because the landmarks are connected to the skin and the skull as layers, simply by moving them new expressions and elements depicting the model's appearance can be synthesized.

To construct the 3D model Kähler et al [22] used multiple layers to simulate the anatomical principal structures, including the skin, the muscle, the cranium, and other separate components – for example, eyes, teeth, tongue, etc – (see Figure 4 below). For each model twenty-four important muscles are used for facial expressions and elocutions, and the skin and muscles are fixed at the cranium via a mass spring algorithm. These principal models allow for animation in real-time, based on the simulation of the facial muscles and the elastic properties of the skin. These tagged, anthropometrically meaningful landmarks allow us to fit generic models to scanned 3D face models, thus allowing for the creation of a wide variety of animated face models.

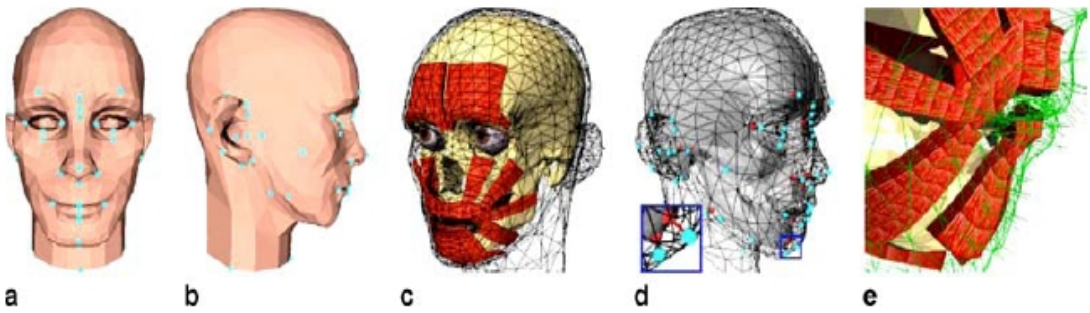


Figure 4: The reference head*: a head geometry with landmarks, front view; b side view; c skull and facial components; d skull landmarks related to subset of skin landmarks; e facial detail showing spring mesh connecting skin and muscles [22].*

A generic model is equipped for the construction of the 3D model consisting of five layers. With the use of anthropometric landmarks, which are small dots placed on the face, the facial features and the skull shape were defined.

Anthropometric measurements are also important in order to synthesize the 3D model growth. According to Kähler et al. the algorithm can automatically apply adult measurements on to a child's 3D head with a reliable output.

2.3.1.5 Model adaptation based on topographic analysis

Rather than adopt the anthropometric data of terminal limits, as presented in the preceding sections, Yin and Weiss [55] suggest using a topographical approach to represent facial characteristics for the individualization of a generic model. To produce a topographic representation, input face model is necessary, thereafter, an analysis treats the facial image as a section of land, marking each of its pixels with topographic labels, such as peaks, edges, saddles, hills, ravines and pits, with the face being divided into convex, concave saddles or slopes. After this topographic analysis, individualization is achieved by carrying out an optimization process using the criteria measured by these labels, thus adapting a generic model into an individualized model.

2.3.1.6 3D face model from video streaming

Zhang et al [59] presents an approach to produce models and expressions facial by using multiple video cameras. These video cameras record facial animation via stereo matching, after which a suitable process is used to adapt a gauge of face to the captured data, which is composed of six synchronized visual jets (four monochromic

and two colors), functioning at 60 fps (images a second). This provides a stereo algorithm that deploys three cameras to capture the left wing while the three other cameras cover the right wing. A stereo algorithm of space/time is used during the data capture, and this computes a time sequence of depth maps using stereo matching.

Thereafter, a fitting of precision of gauge and a process of advance are used to adapt a gauge of face to in-depth charts in the first reinforcement and then to carry out the advance by the whole order. The result of this fitting of precision gives meshes of quality with the correspondence of top during time. The data gives a high-resolution to the 3D meshes within the order of 20 fps, thus capturing the facial geometry, the color and the movements. Once acquired, these models can be operated interactively in order to create expressions through a technique called 'faceIK' by the authors.

The main disadvantages of this algorithm are the expensive equipment required and the absence of any indication as to where the cameras (or equipment) should be placed. Moreover, the work recorded only provides for the animation of a simple face without taking into account individual variations.

2.3.1.7 Model adaptation methods

All the generic model methods of individualization, as described above, involve model adaptation, a technique that deforms a generic model with an individual target model. A basic problem, however, is the interpolation of data of scattering, which drives the movement for each vertex on the generic model towards the target model, given only a sparse set of feature point positions as input.

In order to solve such a problem, a common approach is to build and to the minimum reduce a function of interpolation based on radial basic functions. To improve quality further from the model adaptation, Kahler et al [22] worked out an automatic process that enabled refinement by employing the model subdivision without requiring users to enter a large number of dense points of device.

Moreover, DeCarlo et al [10] use variational modeling for adaptation purposes. This works on models of face of B-groove. Anthropometric measurements are employed while linear and nonlinear external pressures, and later the external fairing within these constraints, are applied. This produces the soft external models, which match the anthropometric devices. Zhang et al [58] deform a generic model with an individual model by total, local adaptation.

The total adaptation comprises the adaptation of size and orientation based on some limiting terminals which are identified semi-automatically, whereas the local adaptation includes to enter the model tops locally to the adjustment the individual model. Primarily, the total adaptation is a process of re-establishing of installation of face and graduation of model, and adaptation provides small adjustments for the positions of tops according to the local geometry. In Yin et al [55], equations of the second order are employed to define the adaptation of the generic model.

In these equations, topographic devices are employed to define the internal and external forces, where the external force leads in the deformation of the model, and the internal force maintains the form of the model during the deformation process. The result of the deformation matches the generic model by way of a simple frontal sight of an individual face.

2.3.2 Example-based face modeling

The techniques presented in this section create the model of face by a combination of existing models, however such methods require a collection of models of face. When the desired facial devices are given, the method of optimization is employed in order to find the coefficients of the best combination. The linear combination of the gathered models of face, together with the optimized coefficients, provides a narrow correspondence between the synthetic model and the desired facial devices.

2.3.2.1 Morphable face modeling

Blanz & Vetter [5] created an algorithm that constructs a morphable 3D model derived from a 2D image. The advantage of a morphable model is the synthesis of variety face modeling with minimum user interference, also, the user can change the facial expression, shape, texture and the position of the 3D model by using data from the 2D photograph.

The first step of this process is to input the algorithm onto a 2D novel image. The algorithm will then automatically separate the facial characteristics of the 2D image, excluding the hair and the neck, and will transform it into a 3D facial model through an illumination scanning process.

In order for the user to create new facial expressions, or change the shape and texture of the 3D face, a library of 200 X 3D heads – 100 male and 100 female – must be used in a morphable process, where the user chooses several of the models

according to the changes he is planning to make, thus creating a 3D modeler interface menu that can be manually manipulated to change the shape of the nose, the chin, and the position of the eyebrows and lips. Afterwards the algorithm will automatically place the new facial characteristics on the 2D input image.

The output of the impressive morphable model of Blanz & Vetter [5] could be used to retrieve accurate depth. However, this technique, like the previous one, is not easy to implement, since it requires a large database of 3D face scans, it unfortunately, therefore, takes a long time to obtain the geometry of a face from a photograph.

A different approach from Blanz et al [6] for creating a morphable model was presented where an algorithm can extract the facial characteristics from the source image and continuously reconstruct a 3D face according to the facial image characteristics, i.e. illumination, position, and viewpoint. More specifically, by giving a set of seven feature points that are manually defined by the user in an interactive interface, this algorithm automatically fits a morphable model of 3D faces to the image and optimizes all model parameters, such as 3D orientation, position, focal length of the camera, and the direction and intensity of the illumination.

When the algorithm has collected all the necessary data from the source image facial surface, a library of 200 Cyberware scans, consisting of 3D faces with shapes and textures, is used in order to find common materials with the 2D facial surface. The algorithm for 3D face reconstruction uses two different actions between the face shape and texture, and the scene parameters. Both of the actions are estimated simultaneously in an analysis-by-synthesis loop.

Remarkably, this morphable face modeling has been utilized in the work of Hiwada [20], in which a 3D face model is tracked from a real-time video sequence. It has been found that 3D morphable modeling is extremely well suited to the task of fitting a 3D model to a target video in real time.

2.3.2.2 Multi-linear modeling

By using existing face model examples, multi-linear modeling is another approach that enables the production of a desired facial model [48]. As in Blanz's morphable modeling [5, 6], the approach demands careful pre-processing of the collected examples in order for the full vertex to vertex correspondence to be appropriately set-up between the examples. These examples are then organized in the form of a data

tensor, which encodes model variations in terms of different attributes, such as identity, expression and viseme. In this way, the attributes may vary independently. By using the organized data tensor, any face model with any desired facial expression can be modeled as a linear combination of examples. This multi-linear face modeling was presented by Valsic [48], where face models had been deployed for face transfer. By using this method, it is possible to adapt the facial animation of a specific face to the video recorder performance of another face.

More details regarding the face transfer will be presented in Section 2.4.2.3.

2.3.3 Discussion

The generic model individualization (GMI) and the example-based face modeling (EFM) approaches present advantages and disadvantages as follows:

- The GMI approach requires only a generic model, while the EFM is not effective unless a face model collection is available. Also, the face models in the collection need to be in vertex-to-vertex correspondence. The disadvantage of this is that such a collection may not easily be available, therefore the GMI approach is preferable to the EFM if no large model collection is available.
- The GMI approach involves the quite demanding process of defining facial features. This requires either a considerable amount of manual work, or the deployment of a demanding automated procedure, which requires both expensive equipment and sophisticated vision techniques. On the other hand, defining facial features in EFM is not a required process, providing a face model collection is available.
- The GMI approach can be used to develop multi-layered, anatomically-based models, while the EFM, to the best of our knowledge, cannot be used for face models with single layers. Extending the EFM to accommodate multi-layered models can be a difficult challenge since it involves having a multi-layer model collection available. In addition, the optimization method, which is required in the EFM approach, may be a prohibitive prospective if it has to be applied to multi-layered models.

Table 1 provides a summary of the two approaches:

Table 1 - Comparison between GMI and EFM

	Examples	Strength	Weakness
Generic Model Individualization (GMI)	[39] [10] [22] [58] [59] [55]	1. Only require a generic model 2. Works for models with multiple layers	1. Need to identify facial features 2. Need considerable user inputs
Face Model Combination (FMC)	[5] [6] [48]	1. Do not need to identify facial features 2. Only need a single face image input 3. Allow generate facial expressions	1. Need support of a registered face collection 2. Difficult to generate multi-layer models.

2.4 Facial expression and animation

Driven by the desire to improve visual realism of facial animation and to create natural facial expressions, great efforts have been made by the graphics community, which can be summarized as follows:

- The simulation based approach, which generates synthetic facial movements by mimicking the contraction of facial muscles. *More details are given in Section 2.4.1.*

- Performance driven animation, which tries to learn facial expressions from recorded videos, or captured face movements, and subsequently makes synthetic facial expressions by applying them to a face model. *More details are given in Section 2.4.2.*
- The blend shape based approach, which creates new facial expressions from the linear combination of collected examples of expressions. *More details are given in Section 2.4.3.*

2.4.1 Simulation-based approach

The aim of the simulation-based approach is to create synthetic facial expressions by simulating actions of facial muscles on a model. This requires a definition of the functions of a certain number of pseudo muscles, and the places where they appear on the model. The functionality of a pseudo muscle is defined in terms of its influence on the model, and this depends on the method of simulation used. Therefore, the total synthetic facial expression is determined by the combination of the pseudo contractions of muscles. Launched with the idea of employing the pseudo simulation of muscles, a certain multi-layer number of models was developed in order to simulate the anatomical structure of the human face, including the cranium, the muscles, soft fabric, skin etc.

2.4.1.1 Pseudo muscles

In order to synthesize an elastic facial skin several authors created an interactive mesh consisting of virtual vertices and springs attached to the bone structure. By morphing the vertices and the springs of the mesh, the algorithm synthesized different facial expressions. This approach has been identified as pseudo-muscles activation. To perform this technique only a small amount of data is required because the morphing effect on the pseudo-muscles takes place on small areas of the facial mesh.

According to Waters [50] the algorithm splits the mesh into muscle areas so that when the user wants to change the lip shape, the algorithm will only change the mouth area, thereby avoiding facial distortion. Individual muscles, or small group of muscles, are identified as Action Units (AUs), some remaining consistent between the source and the generated images and some being activated during manipulation. AUs activate specific muscles according to the expression that is to be synthesized.

The pseudo muscles can be identified according to their orientation on the face either in parallel or in linear – “oblique” or “spiraled” – relative to the direction of pull at the point of manipulation. These muscle types are divided into upper and lower face. The upper facial muscles are responsible for changing the appearance of the eyebrows and the upper and lower eye-lids, while the lower facial muscles delineate the appearance of the chin, the ears, the lips, the areas around the eyes, and the neck.

2.4.1.2 Synthetic skin layers

An important innovation of Terzopoulos & Waters [46] is the combination of an anatomically-based facial muscle process with a physically-based model of human facial tissue. This combination significantly improves the realism of the facial expression synthesis compared to earlier techniques [35, 50 and 27] its basic feature being the animation of facial expressions by contracting synthetic muscles embedded in an anatomically motivated model of three spring-mass layers of skin. This physical simulation propagates the muscle forces through the physics-based synthetic skin, thereby deforming the skin to produce facial expressions. The advantage of the physics-based approach is that it greatly enhances the degree of realism over purely geometric facial modeling approaches, whilst it reduces the amount of work the animator has to do. It is also computationally efficient and is amenable to improvement, with an increase in computational expense, since it requires more sophisticated biomechanical models and more accurate numerical simulation methods.

2.4.1.3 Biomechanical skin model

Lee et al [27] have extended the facial muscle procedure by creating an algorithm that uses a 3D Cyberware scanner, which automatically develops facial models with different expressions. The difference between this method of synthesizing a 3D head model and the methods described in Sections 3.1 is that the Cyberware scanner can collect more accurate data from the 3D object and is therefore capable of providing more details for the algorithm.

Following the data collection, the algorithm automatically produces a generic geometric 3D muscle model with a dynamic skin. More specifically the skull is covered by five individual layers of tissue: the epidermis, dermis, sub-cutaneous connective tissue, fascia, and the facial expression muscles. Each layer is described as

a triangular deformable tissue connected to all the other layers (see Figure 5). The benefit of the five layers helps the user to change not only the shape, the depth, and the light up of the muscles, but also the skull. This approach to skin tissue is identified as "physically-based model of human facial tissue" and the biomechanical face skin model is claimed to be more effective than methods used previously, such as that used by Terzopoulos [46].

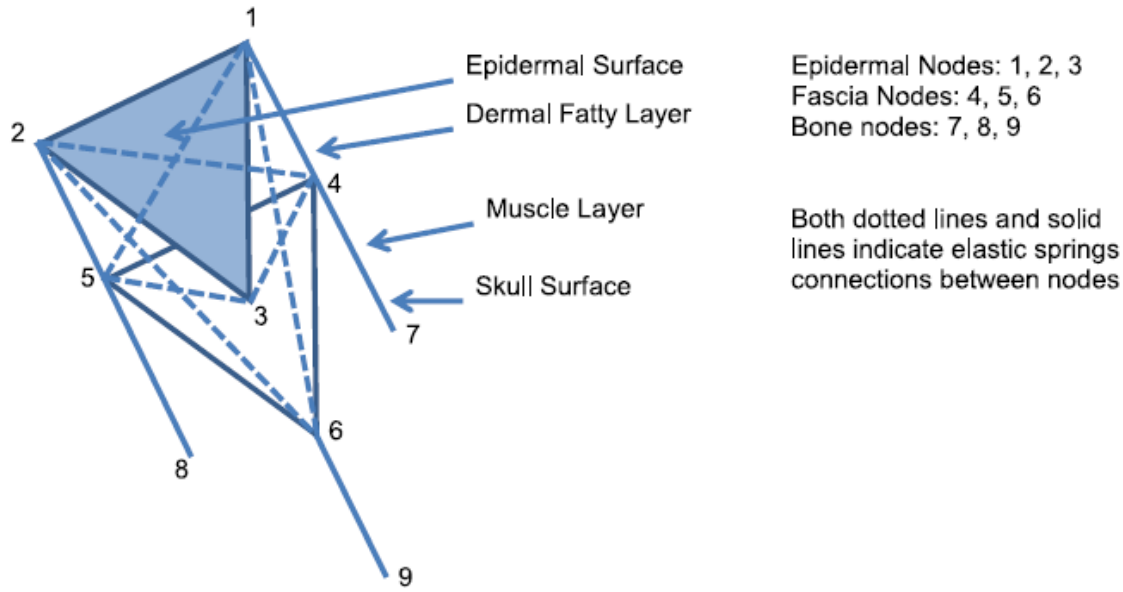


Figure 5: Triangular deformable tissue *[34]*.

In order to synthesize a new facial expression the user has to pull out, or pull in, or even move, the elastic angles of the triangles. The disadvantages of this are the complexities and difficulties of obtaining a reliable facial expression, because transforming the triangles requires great skill. Moreover, it is difficult to model the detailed skin deformations, such as expression wrinkles. Therefore problems are encountered in rendering a photorealistic face model by this method.

2.4.1.4 Anatomically accurate head model

Another approach creating facial expressions based on feature points was presented by Sifakis et al [45] who have created a 3D head model consisting of a rigid articulated cranium and jaw with about 30,000 surface triangles. The flesh mask of the model consists 850,000 tetrahedrals contained within 32 muscles. According to Sifakis, this model was constructed over a two month's period by five undergraduate

students, which makes it difficult, even for a professional 3D developer, to execute the algorithm quickly (see Figure 6).

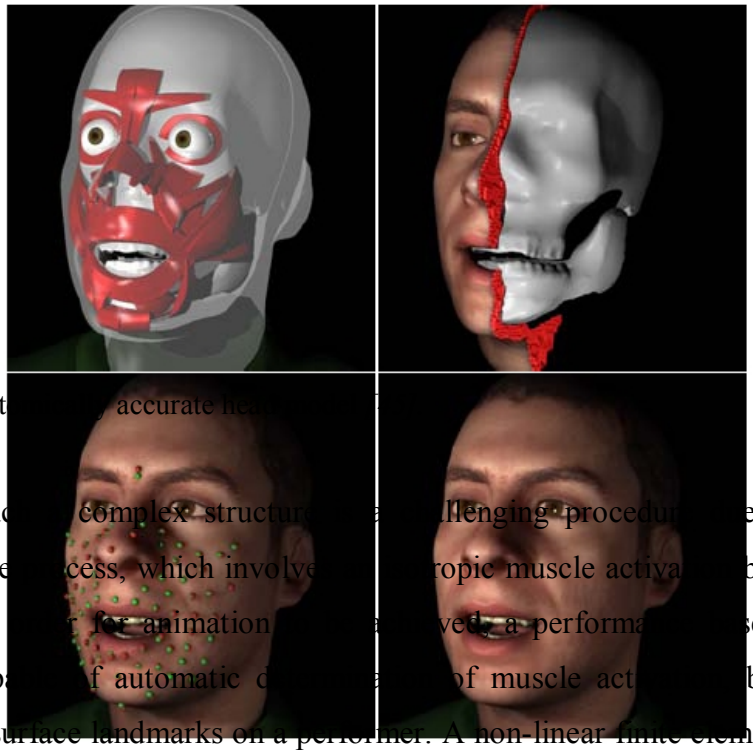


Figure 6: Anat accurate

Animating suc complex structur llenging procedu e to the non-linearity of the cess, which involve ropic muscle activa based on fiber directions. In for animation ved, a performa sed method is deployed, capa f automatic d of muscle act by tracking a sparse set of surface landmarks on a performer. A non-linear finite element method is then used to enable final animation. Once the controls are reconstructed, the model can be subjected to many procedures, such as interaction with external objects and dynamic simulation to capture ballistic motion; also the facial expression can be edited in the activation space. The results can be characterized as both realistic and anatomically accurate. This approach has also been applied to simulate body structures and movements.

Wilhelms & Gelder [51] designed individual muscles, bones and generalized tissues, all fully connected in order to create animated movements between skin, skeleton and muscles. This is considered to have been one of the first successful techniques for constructing an animated 3D body. However, according to Allen Van Gelder [16], by assigning the same stiffness to all springs, the algorithm failed to simulate a uniformly elastic membrane to enable it for maintaining equilibrium.

2.4.2 Performance-driven animation

A simple performance driven animation method is based on a geometric wrapping process where the facial features are tracked though a sequence of facial expression images that enables subsequent movements to be transferred onto a new face in order to synthesize a facial expression [30, 37, 52]. In addition, a more delicate algorithm

can be applied that deploys the Expression Ratio Image (ERI) in order to have fine facial details captured and transferred onto the new image. Other examples of performance driven animation are: live facial performance capture [19]; face transfer using multi-linear models [48]; vision based facial animation control [7].

2.4.2.1 Geometry warping-based method

The Geometry warping-based method consists of a simple form of animation where two photographs of the same person (the source images) are used – the first photo with a neutral face and the second with an expression – together with a photograph of another person with a neutral face (the target image). Either by manually or automatically locating the facial features, such as the eyes, mouth and nose in all three images, the algorithm deploys difference vectors in order to trace the movement of features when emotions are being expressed. Afterwards the difference vectors are applied to the target image in order to transfer the expression to the neutral face.

2.4.2.2 Expression ratio image (ERI)

Normally, facial animation generated from the geometrical deformation of facial models lacks the fine details that often appear in real human facial expressions, such as cresses and wrinkles. These are normally captured by illumination changes during facial movement. To meet the challenge of presenting realistic facial expression, Liu et al [31] invented the Expression Ratio Image (ERI) methodology, a technique by which the user can capture the illumination change of one person's expression and fit them to another person's face.

Two photographs of the same person are the materials necessary to run the algorithm – the first with a neutral face and the second with an expression. The main facial characteristics, such as eyes, mouth, nose, and eye-brows are located either manually or automatically. The last required item is a neutral picture of another person, onto which the desired expression – similar to the other person's second image – will be transferred.

This algorithm is based on lighting changes that occur when the face changes its expression. In the case of monochromatic images the process will be based on analyzing the features exposed by just one light. If the source images are colored then the features will be based in analyzing the most common colors, such as red, green and blue (RGB) for each separate color (weight map).

After the weight map is computed, an adaptive Gaussian filter runs on the ERI. For pixels with a large weight, Gaussian filter is used so that it will smooth out the details of the expression. For pixels with a small weight, a large window is used in order to smooth out the noise in the ERI (see Figure. 7).

With geometric warping the user can continuously map an ERI to any other person's face image of 3D model and generate several facial expressions. This approach requires the ERI process to be executed manually by the performer, which is difficult to achieve. However, this method works when it is assumed that the illumination change only happens when the facial expression changes, and when the source and target images are subjected to similar lighting conditions. If just one of the above conditions cannot be applied, the ERI based approach is unable to create a high quality output.

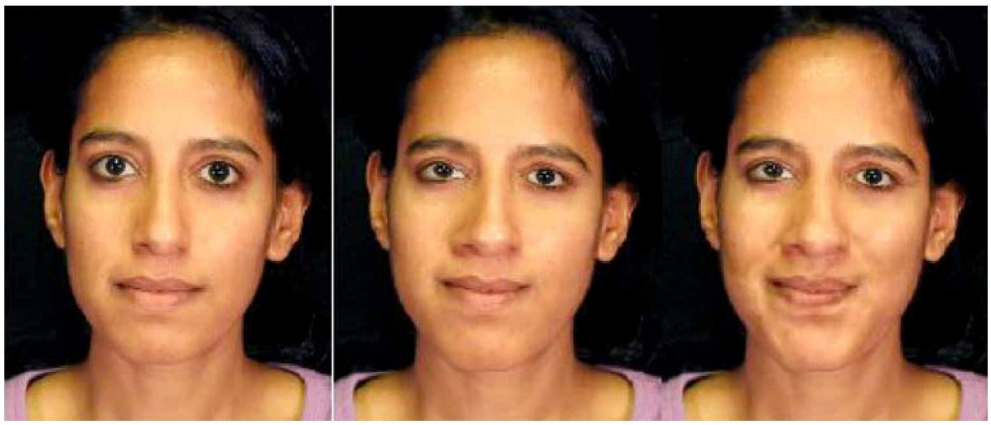


Figure 7: Expression Ratio Image approach. *The first image has a neutral expression (input image). The second image follows the geometric warping, and the third is the one with ERI. The wrinkles in the third image are the result of using the ERI approach [31].*

2.4.2.3 Face transfer using multi-linear models

Another methodology for transferring a face was presented by Vlasic et al [48]. The main contribution of this algorithm [48] is a face transfer method achieved from a multi-linear model onto which the user can transfer a recorded face video on to a 3D model.

As introduced in Section 3.2.2, face transfer is based on a multi-linear modeling of 3D face meshes, therefore, a multi-linear model is a 3D model consisting of meshes that can parameterize the space of geometric variation that possess different attributes, such as facial identity, expression and viseme (speech-related mouth

articulations). By using a multi-linear model these attributes can be varied independently.

The multi-linear model is consisted of several modes. The first mode contains vertices, while the second and the third modes correspond to expression and identity respectively, in the fourth mode the algorithm scans in each slice share the same viseme (see Figure 8).

Figure 8: In the multi-linear model*, the expression and the viseme from the second and third left, respectively, can be transferred to a neutral subject face (far left). The far right gives the result [48].*

In order to construct the multi-linear face model, it is firstly necessary to acquire a range of 3D scans and to use the N-mode SVD in order to compute a model that captures the geometry of the face and its variations caused by identity and expression.

By linking the multi-linear model to optical flow, a tracker continuously estimates the facial expression parameters and detailed 3D geometry from video recordings. The model defines a mapping from performance parameters back to a 3D shape, thus it can arbitrarily mix pose, identity, expressions, and visemes from two or more videos and render the result back into a target video. As a result, the algorithm provides an intuitive interface for both animators (via separable controllable attributes) and performers (via acting). And because it does not require performers to wear visible facial markers, or to be recorded by special face-scanning equipment, it is an inexpensive and easy-to-use facial animation algorithm.

2.4.2.4 Real time facial expression tracking

Video based facial modeling is a method in which the algorithm automatically detects facial regions, extracts features from the video and recognizes facial identity if a face is present. Such a method is the Chai et al [7] algorithm – a real time facial tracking algorithm that extracts a small set of animation control parameters from a video where animation can be achieved by using the recorded actions of the face. The input of the algorithm is a single video stream recording the user's facial movement, a

preprocessed motion capture database, and a 3D head model captured by a laser scanner.

The algorithm has four components: firstly, the visual analysis, which detects the principal position in the video, identifies a restricted number of devices and gives later parameters of the head pose and the facial expression. Secondly, the pre-treated data of capture of movement contains a head motion and expression data. The principal movement and the facial deformations are automatically uncoupled in the capture of movement data during the pre-treatment. Thirdly, expression control and animation transform noisy and low resolution control parameters into high quality motions. Fourthly, expression re-targeting applies the movement synthesized to animate high 3D resolution models.

The expression tracking step involves tracking nineteen 2D features on the face, one for each point of the upper and lower lip, one for each corner of the mouth, two for each eyebrow, four for each eye and three for the nose. The parameters of ordering of expression produced in the first stage include a description of the movement of the devices, such as the mouth, the nose, the eye and the eyebrows of the actor in the video. These noisy and low resolution parameters are converted into high resolutions by using an example-based motion synthesis method, which compares the low resolution parameters with the motion data captured in the second step, and subsequently synthesizes a proper high resolution motion.

Finally, to map the synthesized motion to a target 3D model, an efficient expression cloning technique is used, which pre-computes a number of bases for the facial deformation of the target model, and then blends them to create run-time facial expressions according to the synthesized motion.

2.4.2.5 Facial animation capture

Guenter et al [19], by using a different approach, have presented a algorithm for capturing both the 3D geometry and color and shading information for human facial expressions.

An actress was used, together with six studio-quality synchronized video cameras, arranged in the pattern shown, each one individually calibrated to determine its intrinsic, and extrinsic, parameters to correct for lens distortion. 182 fluorescent pigmented dots of six different colors, arranged to follow the contours of the actress's face and positioned so that dots of the same color were as far apart as possible from

each other, were glued onto her face; a foam-padded box was used to reduce body motions and ensure her face remained centered.

The actress was then illuminated with a combination of visible and near UV light, which increased the brightness of the dots and moved them further away in color space than they would ordinarily be from the colors of her face, making them easier to track. Before the shoot the actress' face was digitized by a Cyberware scanner in order to create a 3D face mesh base, which was then distorted by using the positions of the tracked dots. The fiducials were used to generate a set of 3D points, acting as control points, to warp the Cyberware scanned mesh of the head. They were also used to establish a stable mapping for the textures generated from each camera. This required that each dot had a unique label, consistent over time, so that it was associated with a consistent set of mesh vertices.

For each camera view the 2D coordinates of the centroid of each colored fiducial was computed using three steps: color classification, connected color component generation, and centroid computation.

The data was used to reconstruct photorealistic 3D animations of captured expressions by utilising a large set of sampling points on the face in order to accurately track its three dimensional deformations, whilst simultaneously capturing multiple high resolution registered video images, which were used to create a texture map sequence for a three dimensional polygonal face model, which was rendered on standard 3D graphics hardware. The texture sequence was compressed by utilising an MPEG4 video codec. Animations reconstructed from 512x512 pixel textures produce good data rates as low as 240 Kbits per second.

This method was considered to be novel and the results were very life-like, however analysing video streams to determine deformation is a time-consuming process requiring specialized and expensive hardware.

2.4.3 Blend shape-based approach

The blend shape based approach obtains desired facial expressions by combining a set of existing examples. The approach is similar to the face modeling approach presented in section 2.3.2, which also deploys linear combination from a number of existing face models. To achieve the required combination, linear interpolation [56, 57] or morphable modeling [6] can be deployed either on images or face models.

2.4.3.1 Interpolation between models

Given the availability of a number of face models with different facial expressions, a simple idea is to generate facial expressions in-between by using linear interpolation, and this has been used frequently in many applications of facial animation. For example, Pighin [39] and Zhang [58] generate facial expression by following the reconstruction of face models with various expressions.

Noticeably, Pighin [39] used lineally combined 3D face models to recover face position and expression by using an input video sequence. This recovery process takes place in each frame of the video and it employs a continuous optimization method in order to find the best matched model at each frame. The 3D model used for the fitting is based on the linear combination of a set of face models bearing different expressions, generated by using the technique described in Pighin [39].

However, an obvious disadvantage of this approach is that only facial expressions in- between existing examples can be created. Therefore, the technique requires a very large number of facial expression examples. Also, linear interpolation is not highly accurate, hence it is not a perfect solution for generating in-between expressions. Remarkably, Zhang [58] has overcome the weakness of using linear interpolation by presenting a technique called "faceIK", which is, essentially, an inverse kinematics technique blending the models to generate different facial expressions under user-specified controls. Moreover, Zhang also presents a new representation name – "face graph" – that encodes the dynamics of the face sequence and can be traversed to create desired facial animations.

2.4.3.2 Interpolation between images

Given a set of examples of different facial expressions, the technique presented in Zhang et al [56, 59] allows for quality facial expression to be generated with significant details, such as wrinkles, showing. This technique can also be applied to 3D models.

To use these example images, geometry positions of the feature points in the example images need to be identified. Then, a photorealistic facial expression can be obtained starting from a convex combination of the expressions of example based on these positions. Since this technique makes use of high quality expression examples, it can generate photorealistic and natural looking expressions with fine facial details.

Further, to overcome the challenges of automatically recovering feature points from the images, the authors develop a technique to infer missing feature points from the tracked face by using an example-based approach. This enables the re-establishment of point of device and technique of advance to be less demanding. In other words, the algorithm should still be well-functioned even if the number of tracked feature points is fewer than what the algorithm requires.

2.4.3.3 Blend shape animation from model segmentation

To generate high quality facial expression for 3D face models by using blend shape methods, Joshi et al. [21] present a method that segments the face models into small regions. The shape blend animation based on this segmentation allows for the manipulation of specific parts of a face without effecting other non-relevant parts, hence the preservation of a significant number of complex human expressions.

2.4.3.4 Reanimating faces and images using morphable models

Continuing from their previous work [5], Blanz et al [6] present a technique that allows for changes of facial expressions in existing images and videos. The morphable modeling method, which was used previously for face modeling (see Section 2.3.2.1), is extended here to cover facial expressions. To achieve this, instead of only collecting neutral faces as in Blanz [5], 35 laser scans of facial expressions are also captured and stored in the face model collection. Morphable modeling using such a face model collection allows 3D reconstruction of non-neutral face models, i.e. it generates face models with expressions. The reconstructed model can be adjusted to change its facial expression and rendered back to its original image or video.

The advantage of using this morphable modeling based approach is that it can work on any face shown in a single image/video, without requiring example expression data from that particular person.

2.4.4 Discussion

By comparing the three major facial animation approaches mentioned above, it can be concluded that their strengths and weaknesses are as follows:

- The simulation-based approach only needs the use of computing resources, since it basically employs and simplifies a number of biomechanical muscle equations to produce visually correct facial movements. This performance-

driven approach, which requires the capture of live facial expressions, involves a considerable outlay on high cost computer vision data capturing equipment. Given the state of the art of computer vision, the performance of such a data capture process may not be susceptible to rapid improvement in the near future. For the blend shape approach, a collection of models with different facial expressions is required. Again, obtaining or accessing such a collection is not feasible for an individual user.

- The performance driven approach has the most potential for achieving realism, however, the simulation-based approach will always be limited by the underlying biomechanical simulation method. Unfortunately, most of the biomechanical simulations, including those claimed to have high accuracy – as in [Sifakis 05] – are greatly simplified in comparison to the real world models. In contrast, the performance driven approach, which adopts facial expression from real performance, offers a simulation that is as close as possible to real life.
- The shape blend approach is only capable of creating animation in-between the existing examples, which is a great limitation. Also, its performance relies heavily on the quality of the model examples and this is similar to the employed interpolation methods.
- In principle, the simulation based approach has great potential for medical applications. It can work in conjunction with anatomical and medical data and medical simulation, which provides high quality medical models and has the potential for providing guidelines for medical practice.

Table 2 summarises the comparisons of the three major approaches.

Table 2 Comparison between the Simulation, Performance Driven and Shape Blend Approaches

	Examples	Strength	Weakness
Simulation Based Approach	[50] [46] [27] [45]	1. Only requires computing resources 2. Great potential in medical applications	Artificial-looking facial expression
Performance Drive Approach	[52] [31] [48] [7] [19]	Great potential to achieve visual realism	Need to involve motion capture equipment.
Shape Blend Approach	[39] [59] [56, 57] [21] [6]	Easy implementation	Need high quality facial expression examples

2.5 MPEG-4 facial animation

A superb feature of MPEG-4 is the support of integration of natural and synthetic scenes through an object-based audiovisual representation. This allows MPEG-4 to generate natural, as well as synthetic, visual data without executing conversion to pixel based representations of synthetic models. Synthetic and Natural Hybrid Coding (SNHC) is a subgroup of MPEG that addresses issues related to synthetic data representation and synchronization. The SNHC technology in MPEG-4 version 1 supports applications such as multimedia broadcasting and presentation, virtual talking humans, speech rehabilitation etc, with its 3D graphic capabilities. MPEG-4 Version 2, a later edition, also supports functionalities for body animation.

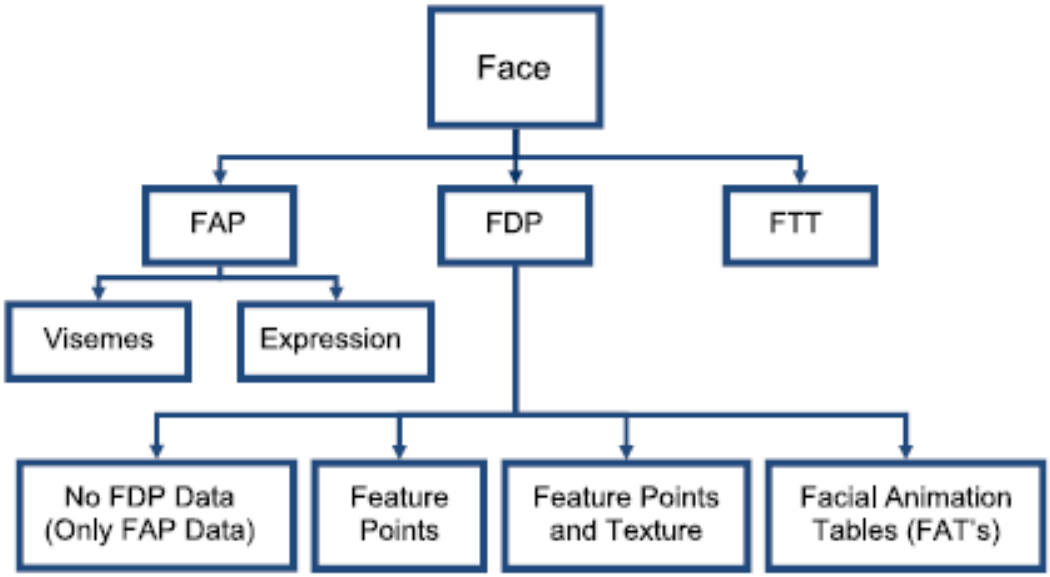


Figure 9: MPEG-4 facial animation data

In order to specify a 3D head, and more precisely facial expression, by using MPEG-4 three types of data of facial animation tools are used (see Figure 9). These are necessary for the construction of facial expressions because they allow the speaker's mood to be, as far as possible, faithfully reproduced.

FAPs were designed to allow the animation of a 3D facial model available at the receiver. More specifically FAPs were invented to perform the animation of faces, reproducing movements, expressions, emotions and speech pronunciation. With the FAP the user can deform the facial features by changing feature points independently or in groups. These facial actions are closely related to muscle actions. The FAP interpretation model strongly enhances the realism of the animation.

1. The FAP set includes 68 FAPs, 66 low-level parameters related to the lips, jaw, eyes, mouth, cheek, nose, etc, and two high-level parameters (FAPs 1 and 2) related with expressions and visemes. Low-level FAPs are associated with movements of key facial zones, typically referenced by a feature point, such as a rotation of head and eyeballs, whereas expressions and visemes represent more complex actions, typically associated with a set of FAPs.

   It is possible to select six different expressions – joy, sadness, anger, fear, disgust, and surprise – by using the expression FAP. Visemes are the visual analog to phonemes and allow the efficient rendering of speech

pronunciation, as an alternative to having them represented by using a set of low-level FAPs. Zero-valued FAPs correspond to a neutral face.

The reason for using FAP compression algorithms is to reduce the bit rate necessary to represent a certain degree of animation data with a certain predefined quality, or to achieve the best outcome for that data with the resources available (bit rate).

2. Facial Definition Parameters (FDPs). FDPs allow us to configure a 3D facial model, which is to be animated by means of FAPs as described above. This can be done by either adapting a previously available model or by sending a completely new model alongside with the information about how to perform its animation. A MPEG file can potentially carry A) no FDP data, or B) feature points, which constitute a set of 3D features represented by their coordinates, or C) feature points plus textures, which include additional texture information for the model, or D) facial animation tables (FATs), which are designed to define full animation control of a complete new model.
3. FAP Interpolation Table (FIT). FIT allows interpolation for the FAPs. This facilitates the definition of facial animation as only a small set of FAPs needs to be included in the MPEG file to describe a dynamic facial animation. This small set of FAP is used to determine the values of other FAPs during the animation using the interpolation based on FIT.

2.6 Limitations and future trends

Good progress has been made in the modeling and animation of facial manipulation during the last several years. Comparing recent results with those of earlier years, faces of glance can be synthesized much more sharply and realistically given the advances in computerisation. The techniques covered in this chapter have been designed in order to create facial shapes and movements. In order to achieve a complete simulation it is necessary to use the true face in combination with the modeling and animation of associated components, such as hair, eyes, and tongue. In particular, the modeling of hair is uniformly forbidden in computer graphics because of its complicated and dynamic nature. Although the modeling of hair has already been adopted in many commercial software packages, it is still only in an infantile

stage in that computer-synthesized results appear artificial and only bear a vague similarity to real hair.

2.7 Conclusion

This paper presents a complete outline of the techniques adopted for the modeling and animation of the human face. It covers facial modeling, facial animation and facial massage and it considers the techniques involved in making 3D facial models. The generic approach of individualization is less demanding of resources, needing only one generic model, whereas face example-based modeling requires a large collection of models.

Face modeling and facial massage are in great demand for a variety of applications, including the cinema, computer games and telecommunication industries, and also in medicine Given the current state of technology, a great effort is required before artificiality can be eradicated and results that are indistinguishable from real faces are obtained. This is particularly important in the manufacture of computer games, where the need for real-time calculations is significant.

Chapter 3 Facial Expression Synthesis Process Overview & Preparation

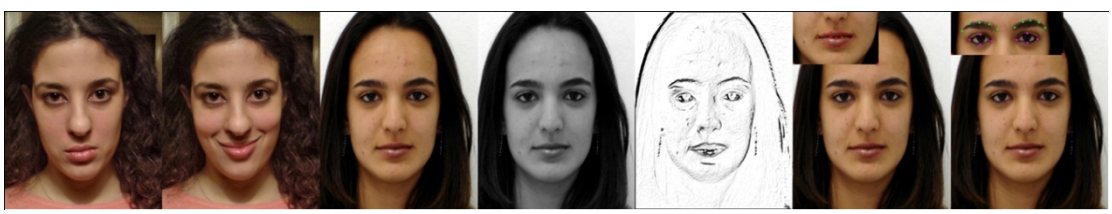
3.1 Introduction – overview process

According to facial drawing canons, the difficulty of creating a realistic expression lies in the illumination settings of the features. By referring to [54] drawing tutorials for constructing a realistic facial expression, artists must concentrate firstly on the shape and the correct values of the features and, secondly, they must pay a similar level of attention to the illumination facial settings that give weight and realistic values to a natural expression. Facial expression synthesis, based only on geometrical values, lacks fine detail, by which it is very difficult to obtain a recognisable expression. The illumination settings, therefore, give the fine details of facial expression by way of creases and wrinkles.

The synthesis of a facial expression, as with the painting rules for attaining realism, is based on two parameters: firstly, obtaining the correct geometrical settings of the expression, which can be performed with a proper geometrical deformation process, and, secondly, transferring the illumination settings from the original source image onto the target image.

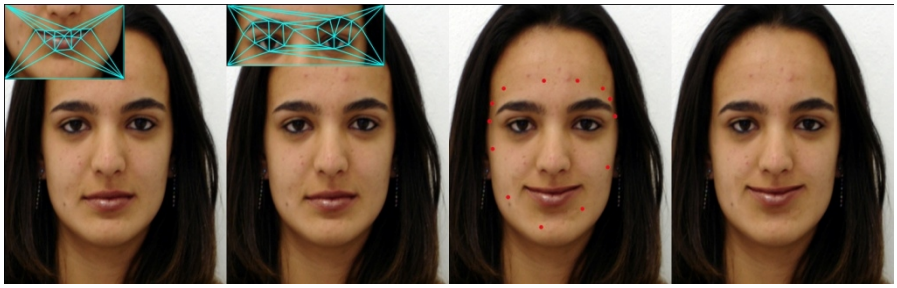
As in portrait painting tutorials, I have divided the facial muscles into two categories: those that must be geometrically deformed and those that must be affected by illumination.

The muscles that must be geometrically deformed are the mouth, the eyes and the eyebrows. The muscles that must be affected by illumination are those around the geometrically deformed muscles, which can be categorized as wrinkles, creases etc (see Figure 10 below).

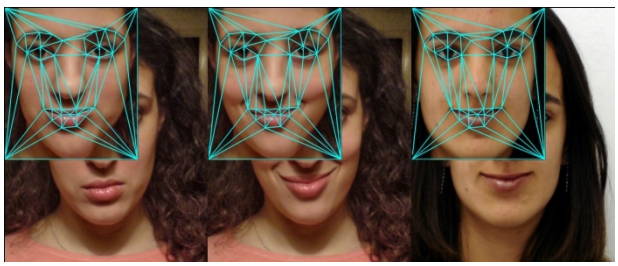


Source im. 1 Source im. 2 Target im. Greyscale Edge map Mouth detect. Eyes detect.

After the expression selection has been made, the source images are loaded from the library, the algorithm transforms the target image to a greyscale format and starts the automatic detection process for the mouth and eyes-eyebrows features.



The geometrical deformation process starts (chapter 5) with the mouth and continues with the eyes and eyebrows. Afterwards the distortion elimination process begins – if it is needed. The algorithm extracts the well deformed area avoiding the distorted parts and pastes it on top of the original target image.



Source images equalization with the deformed target image process (chapter 6.2)

The "source images' colour and lighting normalization process" with the target image (chapter 6.3). After the normalization process, a division of the source images will extract the illumination data map (blue picture)

The user selects the percentage of the illumination data that will be transferred onto the target image by using the 'Ratio Image Threshold' (chapter 6.4). Afterwards the 'distorted illumination elimination' process starts by the undistorted area being extracted and pasted onto the geometrically deformed target image. (chapter 6.5)

Figure 10: Overall process example for synthesising a smiling expression

More analytically, the first stage starts with the deformation of the mouth the eyes and the eyebrows by extracting them from the target image. The sizes of the areas are then defined by rectangles, which already appear in the documentation of the source images. After the automatic detection of the specified characteristics is complete, dots will be placed around them to define their size and positions. Next, the coordinates of the dots for the same facial characteristics are loaded from the source image documentation, and, finally, a geometrical deformation is used to calculate the difference in the dot coordinates of the source images, which is then added to the coordinates of the dots in the target image. This will result in a new expression without, however, proper lighting conditions.

The second stage is the illumination settings transfer process, which means the effectiveness of the defined illuminated muscles. By using the geometrical deformation process again, the algorithm will equalize the source images facial features with the new synthesized (facial expression) features of the target image. Subsequently each of the source images will be divided by 50% pixel by pixel with the target image in order for their pixel values to be combined and normalize huge differences (picture quality, skin colour etc). The two new images are then divided to extract the final lighting settings, which will be added to the geometrically deformed target image at the end of the process.

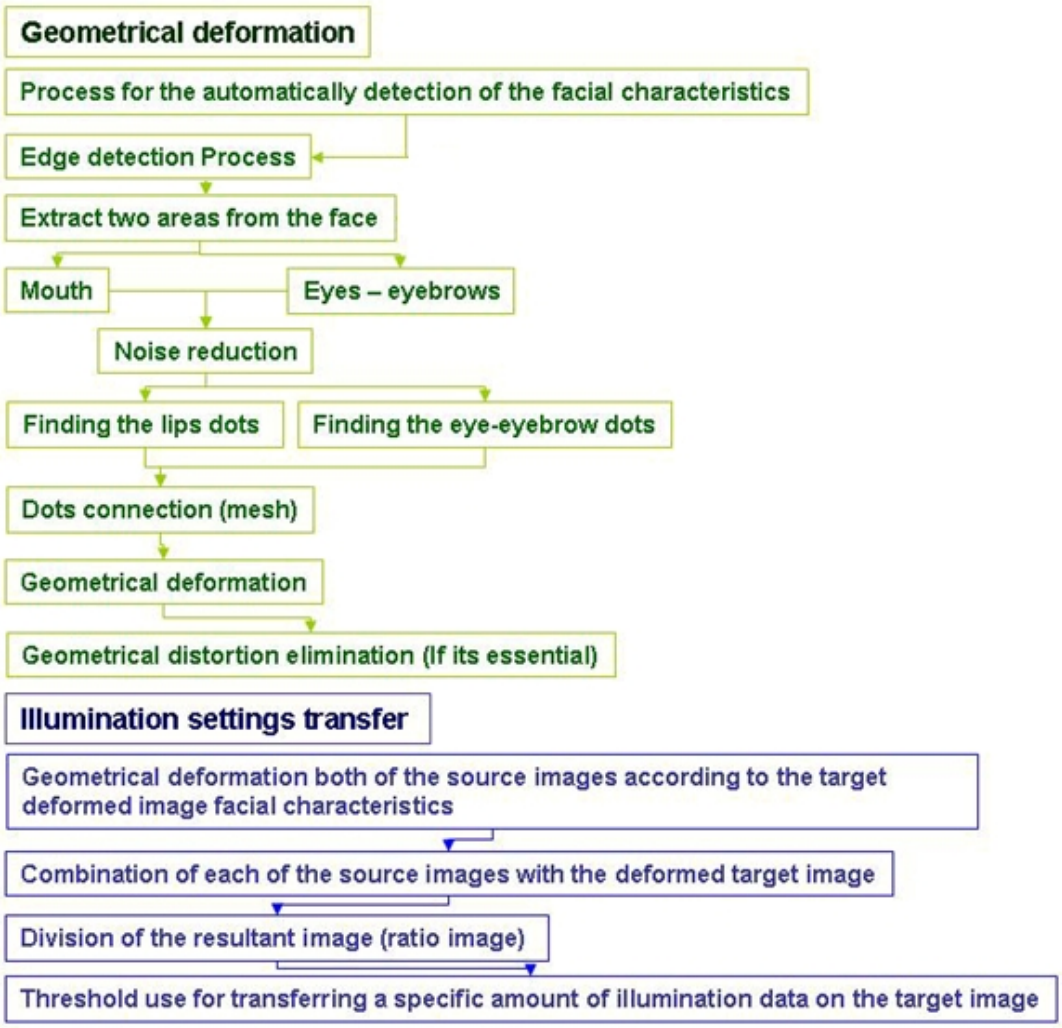


Figure 11: Process diagram

3.2 Facial expression synthesis process preparation

The first step in implementing the new facial expression synthesis is to import a neutral facial picture onto the algorithm. It is important that a complete picture, such as a passport photo, be fitted into the frame, since this will help the algorithm to accurately define the mouth, eyes and eyebrows. The edge detection process is now ready to begin.

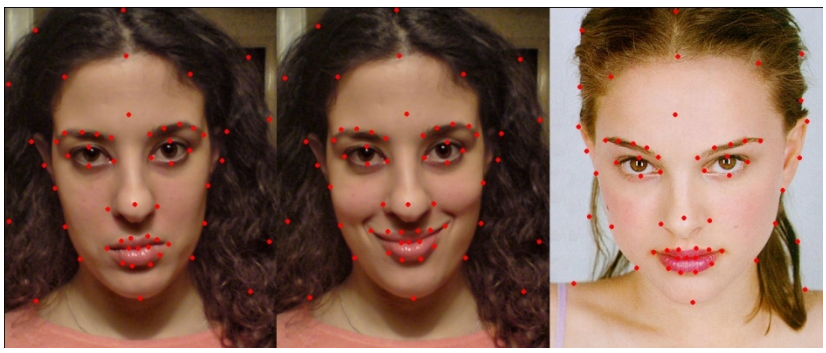
Continuously, the algorithm will automatically scale the target image to the size of the source image for further processing in order to ensure that the features are correctly fitted onto the layers. At the end of the facial expression synthesis process the altered target image will automatically revert to its original size.

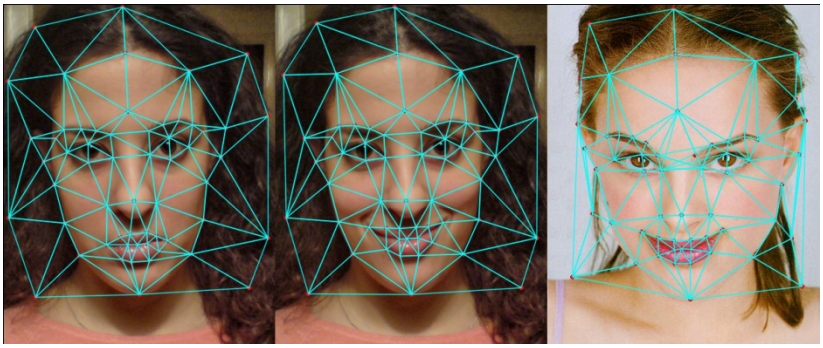
Another important requirement which refers to the automatic facial features detection is the colour transformation of the target image into a greyscale format,

which will provide better edge detection results, since the algorithm will only have to decide between the different values of black and white pixels instead of between thousand of different colours and lighting conditions. Therefore, after the algorithm has completed the target image size process, it will copy the coloured target image into a new window, where it will be transformed into a greyscale format to enable the edge detection processing to take place.

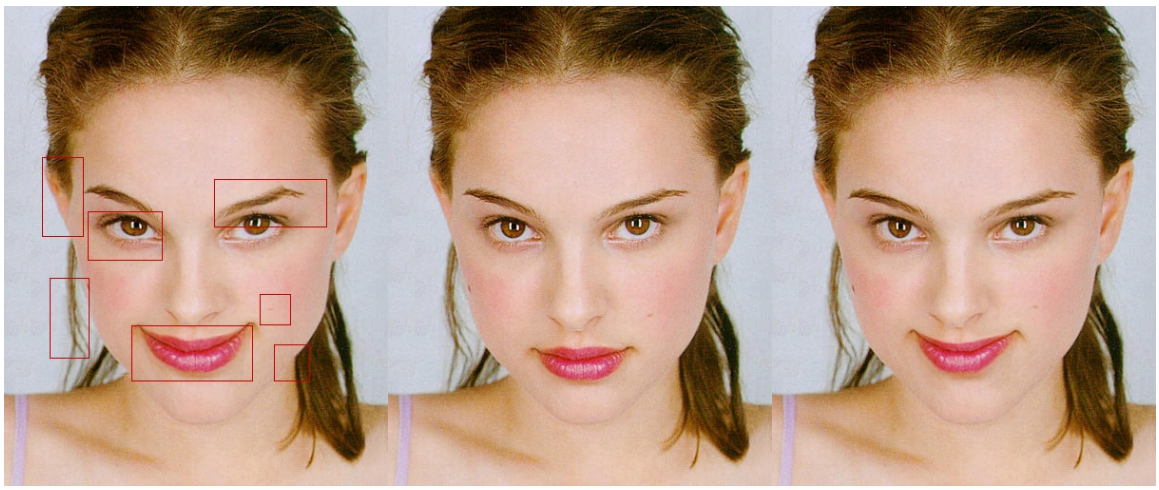
3.3 Splitting the face into areas

The whole geometrical deformation process for synthesizing a new expression is based on the facial features movement of the source image. As it has been mentioned previously, two expressions are used in the independent facial expression synthesis process: (a) the neutral expression, and (b) the desired expression. In the geometrical deformation process the algorithm calculates the movement difference between the source image's facial features and then adds that difference to the target image. Therefore, if there is a small positional change of the facial features, such as hair, neck, ears, shoulders, etc. that difference will distort the corresponding target image's features. Because the identification of all these features is not actually necessary for the synthesis of a new expression, time is wasted by positioning dots and calculating the difference between the correspondent source images features and geometrical deformation process of the target image; distortion also occurs in the areas covered by the triangles. The following examples (Figure 12) show some cases of distortion and demonstrate the need to identify specific areas for facial expression synthesis.





(A) (B) (C)



(D) (E) (F)

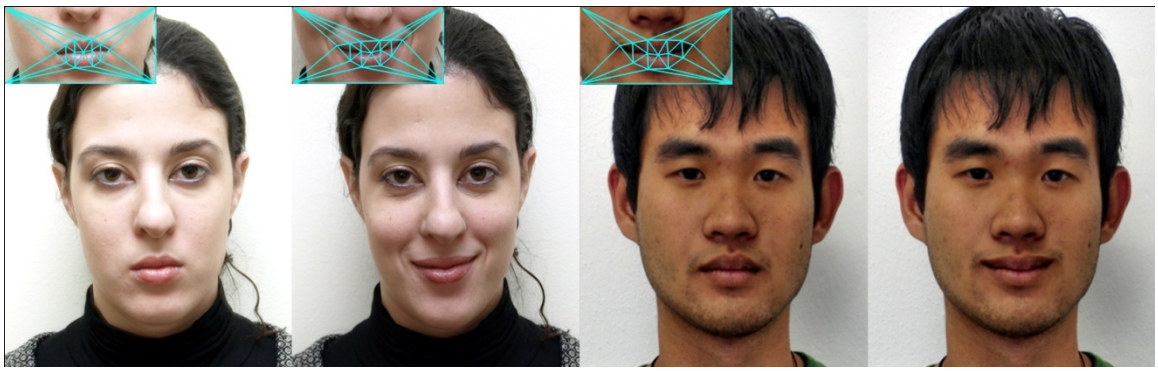
Figure 12: cases of geometrical distortions. *(A) and (B) are the source images, and (C) the target images with dots and triangles. It took fifteen minutes to place the dots around all the facial features in the three images. (D) is the target image after the geometrical deformation process has been completed. Based on the (C) dots deformation the result (D) produced several distorted parts. (E) is the original target image. (F) is the target image after the deformation process, based on my algorithm without distortion.*

Regarding the geometrical deformation process, a facial expression indicating an emotional state derives from the activation of one or more facial muscles; therefore the whole process should focus on those muscles. The muscles, therefore, have been divided into two areas: (a) those containing the muscles that must be geometrically deformed in order to produce a new expression and, (b) those muscles that under proper lighting conditions – i.e. wrinkles, creases, etc – can upgrade the geometrically deformed expression into a realistic one.

The first area of muscles consists of the mouth and the eyes-eyebrows. It is not necessary to identify the other features with dots, because, even if they can be deformed geometrically, they will not provide more realistic details – i.e. hair, neck,

ears, etc. Therefore, the algorithm requires fewer dots and triangles. The process also eliminates distortion of other characteristics; moreover, it is much faster (see Chapter 5) since only thirty dots are required for the mouth and eyes-eyebrow muscles instead of one hundred and fifty in the Liu et al. [31] method.

These muscle areas are extracted from all the images and pasted in layers (as shown in figure 13 in the top left corner of the pictures), each forming a rectangle, which, for the mouth, covers the area vertically defined by the nose and the chin, and horizontally by the edge of the face, (Figure 13) and, for the eyes-eyebrows covers the area vertically defined by the forehead and the middle nose, and horizontally by the edge of the face – and sometimes the hair, depending on the target image.



First Step Second Step

Figure 13: Target area. *First Step: the mouth area has been extracted from the source images and the target image, and placed on the rectangle. The target image mouth is geometrically deformed in accordance with the positions of the dots. Second Step: the target image deformed result has been copied and pasted back on top of the original target image. The size of the rectangle adjusts to fit every facial size in accordance with the picture framing specifications. The rectangles, used to deform the specific areas geometrically, avoid other facial parts. After the geometrical deformation, both layers are copied and pasted back on top of the original target image.*

Further benefits of this approach are the simplification of the automatic facial features detection, since the algorithm will not need to spend much time identifying the indicated features. Also, the edge detection process is activated inside the rectangles, thus reducing the searching time to less than a minute for both areas.

This division of the face into regions was invented by Waters [50], who created a 3D model from a 3D mesh. He found that when he moved specific parts of the mesh in order to create a new expression, such as to the mouth or eyes, because the whole face was built on the mesh, all the other features, which were connected by triangles,

became geometrically deformed. However, the part that he manipulated manually produced proper results. Therefore, he divided the area into specific regions, so that, when he applied a geometrical deformation, the changes were affected only on the corresponding area.

Several other researchers have based their algorithms on Waters' facial sub-regions. Zhang et al [57] used fourteen manually defined sub-regions, all of which were necessary in order for expressions to be changed with the least possible distortion. Also, in order to avoid image discontinuities along the sub-region boundaries, the Zhang et al [57] algorithm performs a 'fade in fade out' blending process by utilizing a weight map, which affects the distorted areas only and does not change the other naturally colored sub-areas.

3.4 Facial expression database

For a faster process, and user convenience, a database of several expressions has been created. This library, which contains all the source images required to synthesize new expressions, has been divided into two sections. The first section, is for the 'one picture synthesis' process, where the images have been categorized in pairs, with one showing a neutral expression and the other the selected expression to be transferred to the target image.

The second section is the expressions database for the video editing synthesis. This contains groups of several images of the same person taken for animation purposes. The first image has a neutral expression, then, from the second to the last, the expressions gradually change, until the final one is selected (Figure 14).

Figure 14: Group of source images for video animation purposes

Both 'neutral' sections also contain two sub-categories containing images of old and young persons; this is important because of the numbers of wrinkles required for transfer to the target image. If, for instance, the target image is a young person and the

source image corresponds to an old person, then the distortion on the illumination settings will be observable in the final result.

Each of the source images for both main sections contain three types of data sets in documentation format, which correspond to all the data point coordinates for the mouth area, eyes and eyebrows area. Also, a set contains a combination of both these areas (mouth, eyes-eyebrows), in order to activate the 'wrinkles' process. All expressions in the database contain the necessary documentation for positioning the corner coordinate dots that define the rectangles of the mouth, the eyes/eyebrows, and the other main facial area, which also are affected by the 'wrinkles process'.

The benefit of this library is that it eliminates both the need to place the dots on the source images manually, and the need to spend time on the automatic detection definition process. Therefore, by using the predefined dots position of the source images' features, this algorithm focuses only on the detection and definition of the target image characteristics.

My algorithm stores all Ekman et al's [12] six primary emotional expressions – anger, disgust, fear, happiness, sadness, and surprise – in the database library.

Chapter 4 Facial features identification processes

4.1 Introduction

In the area of facial expression synthesis, over the last few years several researchers have concentrated on synthesizing realistic expressions and have produced algorithms that automatically detect those features requiring limited user interaction. In this chapter we will present two approaches for the automatic detection of the target's features and then describe the advantages and disadvantages we discovered during several tests on the various images.

We based both automatic approaches on the numbers of facial feature edges on the target image that had been specified and identified, together with their shapes and positions. Firstly, it was essential that we used for both of the approaches a Sobel edge detection technique to transform the image from greyscale (see Chapter 3.1 above) to a map of edges based on the contrast pixel different values of the features with the skin color. Secondly, the algorithm identified the edges around the features by utilising rectangles, masks and a noise reduction process. Thirdly, the facial feature edges were automatically surrounded by dots, which afterwards were copied and placed over the original colour target to enable the geometrical deformation and lighting transfer process to be activated.

In case the algorithm is unable to automatically detect the required facial features, a simple explanation of the manual process is available at the end of this chapter.

4.1.1 Edge detection approaches review

Because the detection of sharp changes in image brightness enables the recording of important global events and settings changes, edge detection algorithms are very useful in many areas, such as architecture, image processing, the mathematics of facial features detection, writing detection, buildings surface detection, etc.

The following is a description of some of the more important edge detectors.

4.1.1.1 Canny's Edge Detector

This consists of two parameters: a) the Gaussian filter, and b) masks that reveal edges. The Gaussian filter 'blurs' the noise pixels in order to clean the picture, while the

masks, which are small rectangles commonly 3x3 pixels, scan the image thereby revealing facial features edges. The Sobel, Canny-Derishe and Differential Detectors are all based on Canny's technique.

4.1.1.2 Interest Point Detector

This recent 'computer vision' terminology is used in processing for the detection of 'interest points' – i.e. points on the image characterized as:

- clear, preferably mathematically well-founded definitions
- well-defined positions in image space
- surrounded by an image structure that is rich in local information, thereby simplifying further processing within the vision algorithm
- stable, when the image domain is either locally, or globally, perturbed; this includes deformations arising from perspective transformations (sometimes reduced to affine transformations, scale changes, rotations and/or translations)
- being capable of computing illumination/brightness variations and can be reliably computed to a high degree of reproducibility
- including an attribute of scale, thereby making it possible to compute them from real-life images

The notion of interest points relates to 'corner detection', where corners were detected in order to obtain reliable, stable and clearly-defined features for object tracking and for the recognition of three-dimensional CAD-like objects obtained from two-dimensional images. In practice, though, corner detectors were sensitive not only to corners, but also to local regions that possessed high variations in direction. Interest points also relate to the concept of regions of interest, which were used in order to recognize objects; this was often referred to as 'blob detection' (see 4.1.1.3 below).

Although such detection methods were not necessarily considered to be of interest to point operators, there is no good reason to exclude them, since, in most common detectors, each blob has a clearly-defined point that often corresponds to (a) a local maximum, or (b) a local maximum in the operator response, or (c) a centre of gravity of a non-infinitesimal region. Blob descriptors, in all other respects, satisfy the criteria of an interest point, as defined above. Although a number of blob descriptors contain complementary information, this should not prevent them from falling into the scope of 'interest points'.

4.1.1.3 Blob Detectors

'Blob detection' in computer vision refers to visual modules for detecting points and/or regions in an image that is either brighter or darker than their surroundings. The two main classes are ‘differential’ methods based on derivative expressions, and methods based on local extremes of intensity. More recently, blob detector operators have become ‘interest point’ operators, or, alternatively, ‘interest region’ operators (see 4.1.1.2 above).

There are several reasons for the development of blob detectors. One is that they provide complementary regional information that cannot be obtained from edge, or corner, detectors. Another, as described in 4.1.1.2, is that they indicate regions of interest for processing purposes, and they can recognize, or track, objects, or parts of objects, in the image domain. In other domains, such as histogram analysis, they are used for peak detection, as applied to segmentation. More recently, blob descriptors have been used as interest points for wide baseline stereo matching and to indicate informative image features for appearance-based object recognition relating to local image statistics and for ridge detection and the plotting of elongated objects.

4.2 The Sobel Edge Detection Process

Detecting sharp changes in image brightness is to capture important events; therefore an edge detection algorithm is needed to eliminate skin color detail in order to reveal only the required characteristics. Because the edges in an image are pixel areas with strongly contrasting values, the intensities need to be measured according to both the previous and the subsequent pixel areas. If, for instance, the first target area consists of bright value pixels and its neighbor has dark value pixels, the algorithm will define the second area as ‘edges’. Edge detection is crucial since it reduces the amount of data required by filtering out useless information, whilst still preserving important structural properties. [6]

Green [6] has presented an algorithm for an automatic edge detection process – the Sobel Edge Detector – whereby the operator moves masks over the image, manipulating a square of pixels at a time; the pixels values are then scanned onto an image by way of a 2-D spatial gradient measurement. Two 3x3 convolution masks are

used, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

The magnitude of the gradient is then calculated using the formula:

$$|G| = \sqrt{Gx^2 + Gy^2}$$

This approach is fast and accurate for non-facial pictures; therefore we considered it to be ideal to modify it for facial scanning. We have also considered increasing the sizes of the masks in order to eliminate noises. For example, in the 3x3 mask shown as 'A' in Figure 15, below, the fourth pixel is black and the second one white, therefore the algorithm will identify the observed pixel as an edge. When the mask size is increased to 5x5 pixels (see example 'B') the algorithm will recognize the smaller black pixel as noise and it will keep the originally observed pixel's color value.

The size of the mask depends on the size of the picture; for instance, if the picture is 150x150px it would be appropriate for the mask to be 3x3px. If the mask were to be bigger, then it would avoid important pixels that describe the facial characteristics, and if it were smaller, it would define the noise pixels as edges. For a larger picture, different sized masks are required. For our purposes, the most appropriate picture size is 500x500px, which requires a 5x5 mask

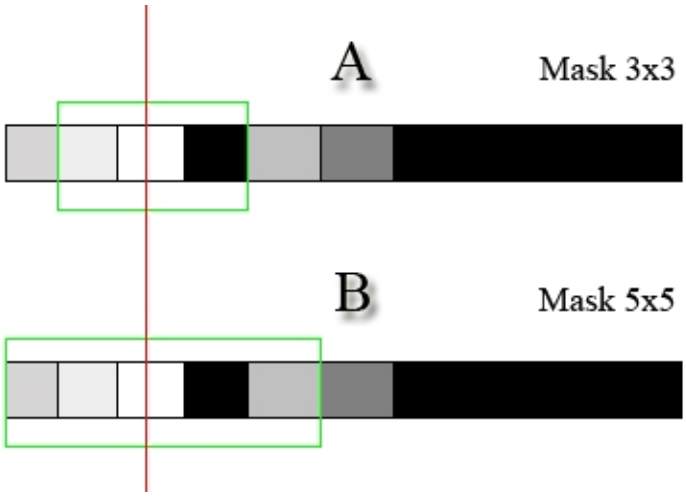


Figure 15: Showing the differences in accuracy between mask A and mask B

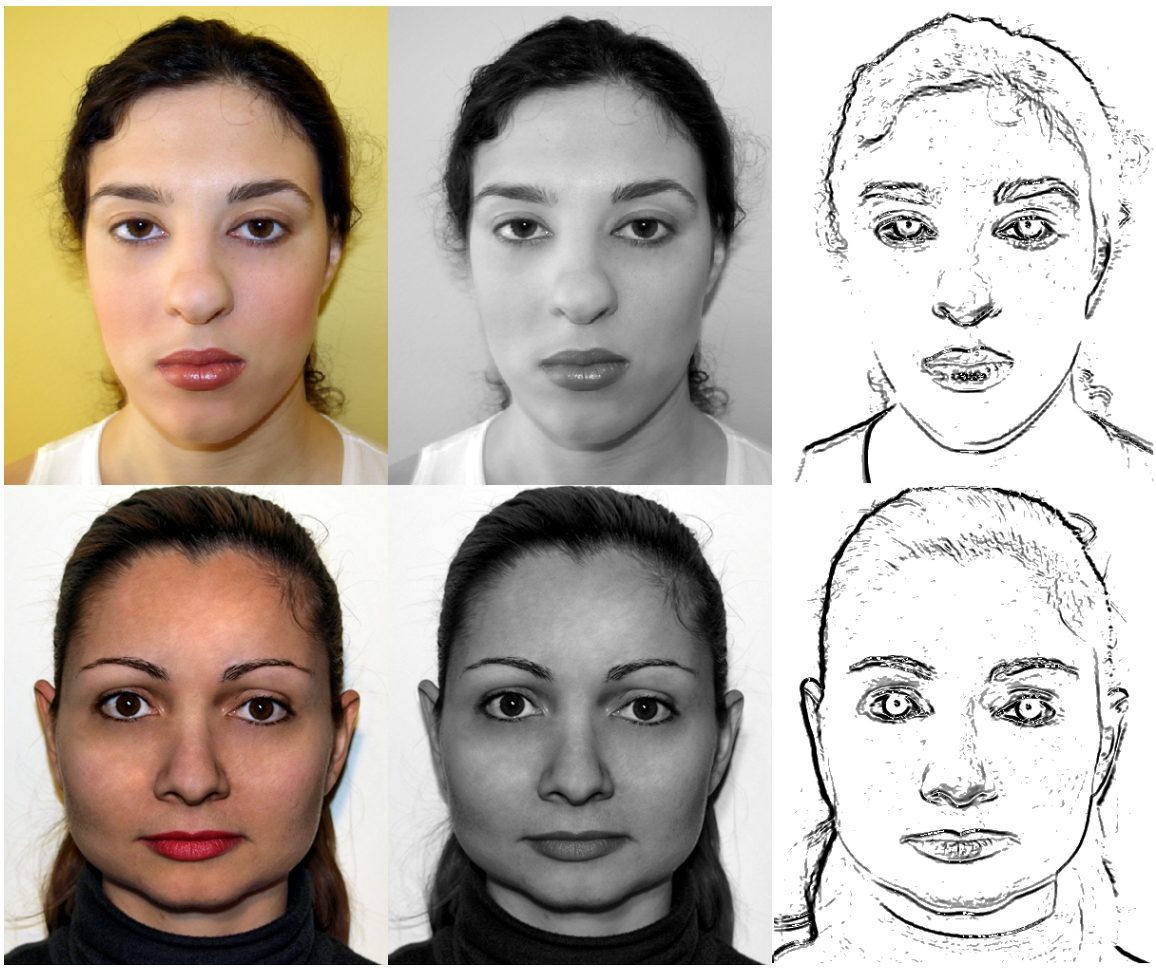


Figure 16: The original colored images after the edge detection process. *The shape of the mouth, eyes and eyebrows have been identified clearly by using the Sobel Edge Detector approach.*

We selected the Sobel detector for its simplicity, accuracy and speed; it is also incredibly sensitive to noise, which it highlights effectively as edges. It is therefore superior to Canny's highly complex optimal edge detector, which is based on a multi-stage algorithm that takes significantly longer to compute. We also tested other edge detection algorithms, such as the interest point detector and the blob detector, but we found they needed further refinement. The Sobel Edge Detector, therefore, is recommended for its capacity to reveal massive data from facial pictures.

4.3 Noise reduction

After the rectangular shape is defined, the process of noise reduction begins. A new mask will be used to identify the 'almost white' pixels. A colour threshold defines how much non-white colour is permitted to be inside the mask rectangle for an area to be considered 'almost white'. If the area covered by the mask is characterised as 'almost white', then the pixel in the centre of the mask will be changed to 255 (i.e. 100% white) (Figure 17). This process is important for removing those pixels that do not coincide with the shape and position of the facial characteristics and are, therefore, just noise.

A square the same size as the mask rectangle will then be defined around the observation pixel, so that the observation pixel is an equal distance from each of the angles of the square. In order to decide whether the observation pixel is noise, or whether it belongs to the contours of the facial characteristics, the algorithm will calculate the whiteness value ratio, which is defined by computing the average normalized colour value of the mask pixels, each of which are described as MPCol/255.

$$\sum_{x \in M} \frac{X}{255*n}$$

M = group that contains all the mask pixels (their colour values)

X =the pixel of the mask

n = the number of pixels in the mask

The equation calculates the sum (Σ) of pixels color values of the rectangle

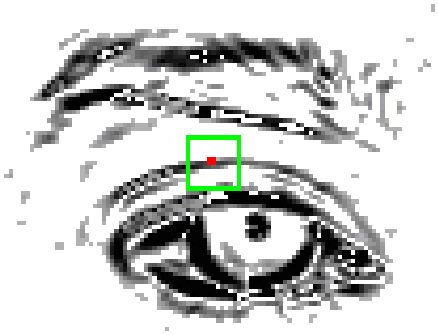


Figure 17: Noise reduction process. *The green colored square is the noise reduction mask. The red dot is the observation pixel.*

If the average normalized colour value of the mask pixels is 1, then all the mask pixels are totally white. If it is 0, then the pixels are totally black. The threshold has been defined as 0.7, therefore the algorithm considers the observation point to be part of the facial characteristics when the whiteness color value is less than 0.7.

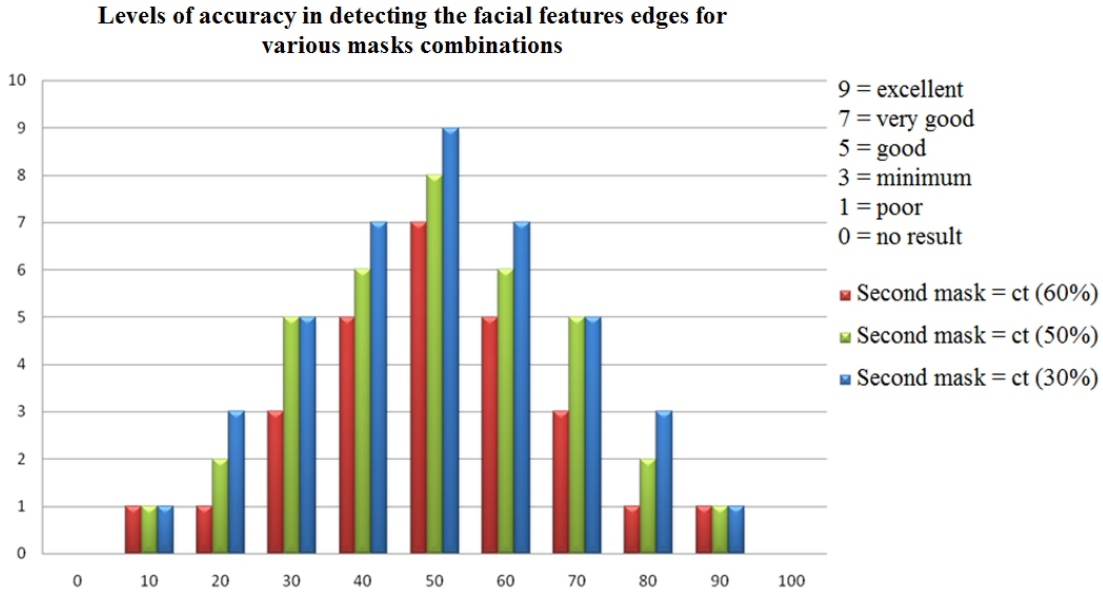
4.4 Automatic process 1

4.4.1 The search for the eyes and eyebrows

As mentioned in chapter 3.2, the algorithm uses two areas for the geometrical deformation process. These are the mouth and the eyes-eyebrows areas, the size and position of which are already defined in the library expression database. When the user loads an expression, a document will also load the points' position and rectangle sizes of the source images. The rectangles will also be used for the target image, since they are big enough to include the specific facial features of any possible target image. The algorithm will then copy the area from the target image, in proportion to the size of the rectangle, in order to separate it from the original image and to start the detection of that area's facial characteristic – such as the mouth – subsequently achieving geometrical deformation.

The process is more complex for the eyes and eyebrows, because the algorithm has to detect four different facial characteristics in the same area. For that reason the rectangle is divided into two sub-areas, each containing an eye and an eyebrow. The algorithm starts first with the left area (left eye-eyebrow) and, after the detection process is complete, continues with the right. It then searches inside the left half of the rectangle to identify the shape of the eyebrow and continues by shaping the eye. For

this purpose, two search masks – one 5x5 pixels and the other 30x30 pixels – will be used in sequence to scan the entire rectangle. The first mask will search for 50% non-white pixel areas, and the second mask, which will start from the same position following the first mask, will search for 30% non-white pixel areas. The referred percentages for the first and the second masks have been defined following multiple experiments with different values using different images. Those values have produced successful results for more than fifty pictures; we therefore accept the values as being correct in most cases. The following diagram provides statistical measurements of using different mask percentages values for searching non white areas and up to which grade those values succeeded in detecting the shape of the eyes-eyebrows features accurately.



(A)

Figure 18: This diagram presents the grade of success in the detection of eyes-eyebrows derived from more than fifty experiments, under different levels of *percentage of non-white area of the first mask increases, while the second mask has constant values*

The starting point for both masks will be the top middle pixel of the rectangle; they will move vertically downwards until they find the first big non-white pixel area, which will define the eyebrow. The masks will then move horizontally until the size and shape of the whole area is defined. If the conditions for both masks are not satisfied, then the algorithm will assume that particular area to be noise – hair, scars, etc – and the masks will move forward until the correct area – the eyebrow – is detected.

4.4.2 Dot placing

After correctly detecting the eyebrow, the algorithm will place five evenly spaced dots on its upper border. It will first place two dots on the left and right corners of the eyebrow, the left and the right dots being the furthest left and right border coordinates. The next dot will be placed in the middle, equidistant from the left and right corner dots. The algorithm will then identify the last two dots by the same process; the first being the dot between the left border dot and the middle one; then, by using the same logic, it will identify the right middle dot.

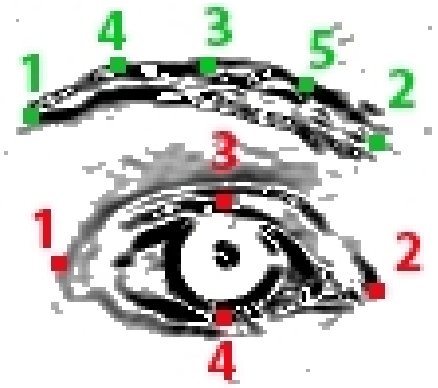


Figure 19: Eye dots placing. *The numbers next to the dots indicate the order in which the algorithm places them. The eyebrow dots are green to contrast with the red eye dots; this helps the user to adjust the position of the dots more accurately in case the algorithm fails to identify the features correctly.*

After the eyebrow has been defined by dots, the algorithm will continue with the eye. Again, there will be two masks, each with different starting points and settings, also using a smaller detection percentage for the non-white area. The starting point of the masks will be defined as ten pixels below the left corner dot of the eyebrow. In order to help the algorithm detect the proper area of the eye, and not be confused with edges that describe irrelevant characteristics, the edge detection was placed 10 pixels under the left eyebrow corner, because that distance makes it certain that the algorithm will find the eye, it will also define the eye shape faster and it won't confuse it with the eyebrow; and under the eyebrow because the eye shape always starts there. The masks will then scan horizontally to the right, looking again for the first non-white area. If the non-white area satisfies the conditions for both masks, it will be accepted as the eye area and the mask will proceed to identify the lower side of the eye border.

The placement of the eye dots is as for the eyebrow, with the first two dots being placed at the corners of the border shape – left and right. The algorithm will then continue to calculate the distance between the points, and, depending on the result, will define the middle pixel. It will then divide the distance – width – between the left 1 and right 2 corner dots by 20% of that distance, in order to calculate how much higher, or lower, dots 3 and 5 should be placed above the central dot (Figure 19). It must be mentioned again that the specific percentage has been selected following several experiments with target images, which produced acceptable results. The following diagram (Figure 20) shows different percentages of the distance between the left and right dots along with the corrected defined percentage results.

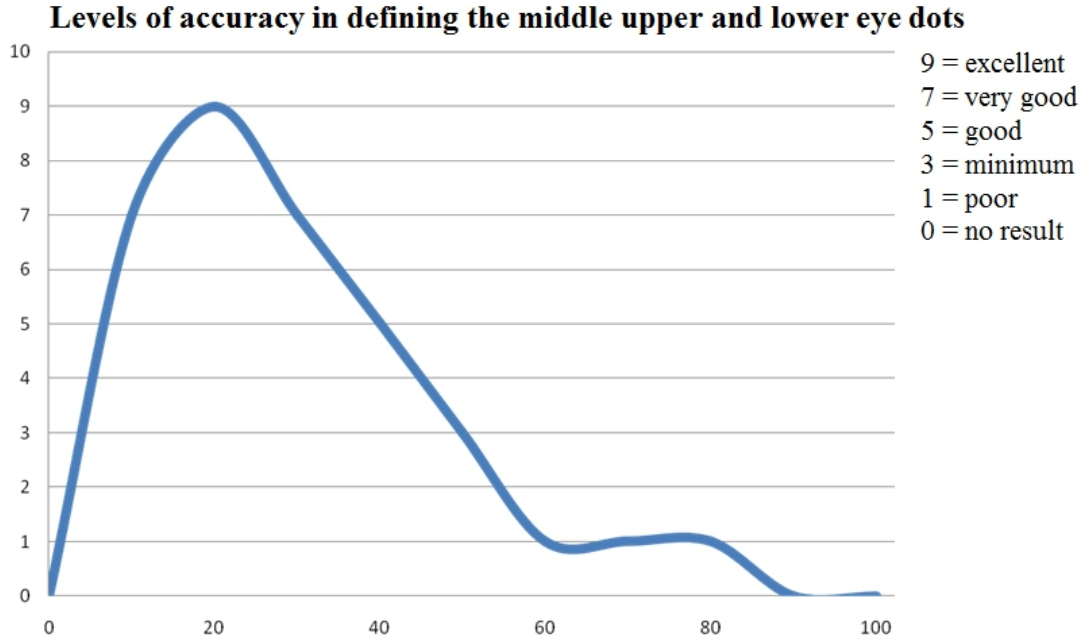


Figure 20: Levels of subjective rating of accuracy *as the distance percentage changes*

Depending on the eye size, this procedure is faster and more accurate than trying to separate the middle and upper edges of the eye from the edges of the eyebrow.

.

Figure 21: The dot placement order on the mouth

The method for detecting the mouth is as for the eyebrow; six evenly spaced dots are selected for the upper lip from the area border pixels, the first being the furthest left corner pixel and the last the furthest right border pixel. The lower lip dots define the pixels that are vertically adjacent to the upper lip dots and the order is as for the upper lip dots, reading from left to right. Three dots are placed in the middle line of the mouth by dividing the middle, upper and lower points in order to produce accurate geometrical deformation results and to eliminate distortion (Figure 21).

4.5 The Automatic Process 2

4.5.1 The search for the eyes and eyebrows

Because this process is fully automatic, the user is not able to correct dots that might be misplaced from the algorithm. It is important to point out, though, that this approach can only work accurately where there is a high contrast of lips. This is also true for the automatic 1 process (Chapter 4.4) where the most difficult features for detection are the eyes and the eyebrows. After the division of the main eye rectangle into two semi-rectangles for the left and right eyes, the algorithm will start, as described above, by searching inside the left half of the rectangle for the eye and eyebrow.

Both automatic approaches differ from the eyes-eyebrows detection processes. In the automatic 1 first approach (Chapter 4.1) the detection process relies on mathematical equitation (calculating the distance between the left and the right corner dots on the eye in order to define the position in which the middle upper and lower dots will be placed) and is based fully on the edges target image diagram, rather than on the following approach, which uses more dots for the eye-eyebrows features.

In order to define the eyes and the eyebrows properly, it is necessary to identify the separate area between the eyebrow and the eye. If the algorithm defines this correctly, then it will be able to identify the eyebrow borders separately from the eye borders. In the automatic 2 process the algorithm does not place dots around the eye based on mathematic equations, as in the automatic 1 process, since it is based on the shape of the eye. In order to define the eye border, however, it is important to be clear which area is the eye and which is the eyebrow, therefore, the rectangle must change its size in order to fit them (Figure 22); this will avoid those edges that refer to hair or nose.

As in chapter 4.4 above, two search masks will be used in sequence using the same settings and percentage specifications. The difference, however, is the new starting point and the direction that it will follow. The starting point for both masks is the top left corner of the already defined rectangle; they will then move first vertically down and then back to the top and from left to right.

The first area to satisfy the conditions of both masks will define the left side border of the eye and eyebrow and a dot will be produced. Then, in order to find the top, right and bottom sides of the new rectangle, the algorithm will rotate the predefined rectangle clockwise, each time by ninety degrees; after producing dots at the left, top, right and bottom, the rectangle then changes its shape in order to fit the border.

(a) (b)

Figure 22: The search of the eye-eyebrow area. *The purple rectangle in (a) is the predefined rectangle. Picture (b) shows the new rectangle size that fits over the target area*

4.5.2 Finding the separation border between the eye and the eyebrow

After discarding the noise pixels in order to define the border, the algorithm will select the first white pixel near to the bottom left side, it will then search for, and

separate into two, the large area between the eye and the eyebrow, which it will define by the placement of a series of dots.

(a) (b) (c)

Figure 23: Separation area detection process. *(a) the point inside the large area (b) the border of the area (c) the definition of the separation area*

The algorithm will then scan upwards from the left bottom until it identifies the first black pixel area; it will then identify the contour of the pixels by underlining the border line. If this border line is not continuous it will mean that the algorithm has identified a gap, it will, therefore, return to the initial black pixel and continue upwards to find the first white pixel, which it will select as a new starting point (red dot, Figure 23c). The algorithm will then follow a similar process until it identifies a continuous line.

If the width of a border area is greater than 70% of the rectangle – i.e. the eye-eyebrow – the algorithm will consider it to be the separation area, and, when this is defined, it will create two layers – the first containing only the eyebrow, and the second only the eye.

4.5.3 Dot placing

In the fully automatic process, the algorithm finds the borders that specify the shapes of the eyes and eyebrows separately. Next, the algorithm searches for five evenly spaced dots on the upper border of the eyebrow; it will find the left and the right side dots, from where it will identify the middle dot, equidistant from them. Finally, it will identify the dot between the left border dot and the middle (5) and then the dot between the right border dot and the middle (4) (see Fig 24 below and automatic 1

process at 4.5.2 above).

The algorithm will find the dots on the upper and the lower eye in the same way as for the eyebrow; the only difference being that it will align the dots below the eye with the upper side dots, vertically. The algorithm will then move the lower dots to satisfy this condition, depending on the shape of the border.

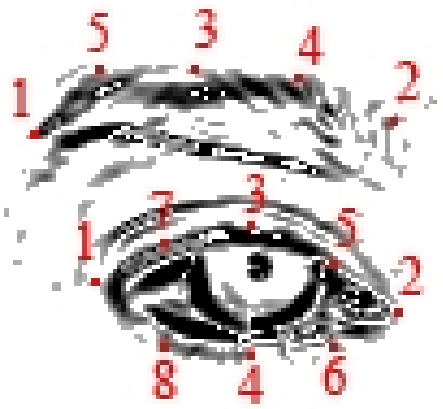


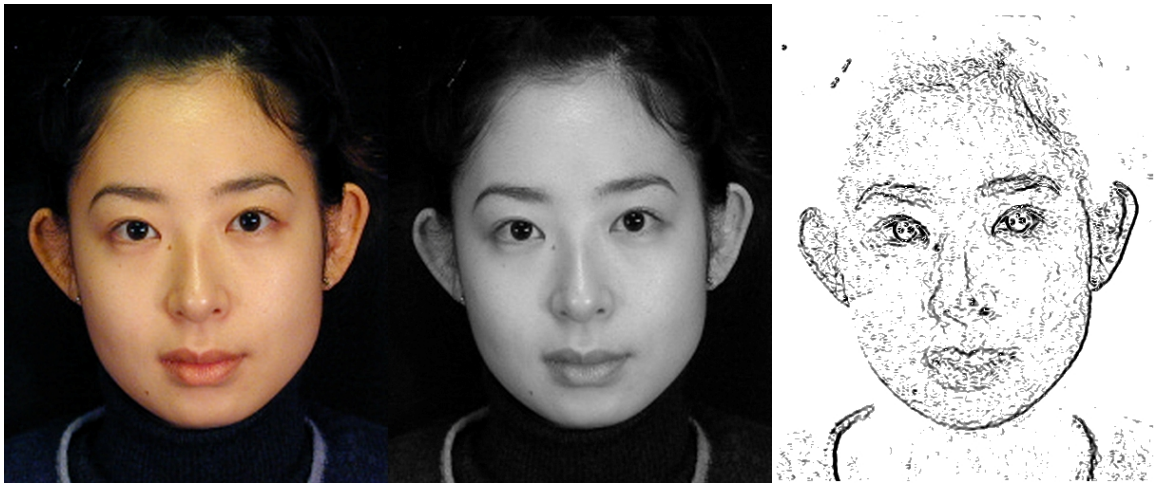
Figure 24: Eye dots placement. *The numbers next to the dots indicate the order in which the algorithm places them*

Finally, the algorithm will find new right and left corner dots, so that the left dot (1) is, as near as possible, equidistant from dots (7) and (8). The dot that best satisfies this condition will be the left dot. The same process will be followed for the right corner dot according (5) and (6) dots distance (Figure 24). The automatic process will place more dots around the eye, because the algorithm has identified its border. Although this process produces better result, they are not a great deal different from the automatic 1 process.

The algorithm detects the lips area by using the method described in the automatic 1 process (Chapter 4.4.2, Figure 21).

4.6 Manual Process

In a few cases of target imaging, the algorithm cannot detect the features. This occurs when characteristics, such as the mouth, don't differ much from the colour of the skin, resulting in the algorithm being unable to indicate the shape properly, rendering it undetectable and difficult to define by dots.



.

Figure 25: Manual Process. *The lighting settings of the source image here do not satisfy the edge detection masks specifications on the mouth area; therefore, the algorithm could not specify its shape. Moreover the noise reduction process avoided a lot of noise edges.*

Also, in some cases the 'edge picture transformation' procedure produces more edges than are necessary, which usually results in poor quality pictures, because the algorithm, assuming the noise edges to be those of facial feature, defines them mistakenly, as can be seen in Figure 25. In such cases, the noise reduction filter cannot delete them, because it, also, defines them as parts of the facial features. This error can be resolved by the user manually changing the size of the masks that detect the picture edges (see Chapter 4.2) or by changing the precedence of the non-white pixel areas during the noise reduction process (see Chapter 4.3). However this procedure is complicated, requiring time and several attempts until the user achieves the most suitable settings.

In both of these 'edge definition distortion' instances, the algorithm recognises the difficulty and changes the relevant automatic process into a manual process. As previously described, the masks have specific settings enabling the search for specific areas of non-white pixels – i.e. the facial features. However, if the process cannot be completed successfully, a message appears on the screen prompting the user to continue by placing the dots manually.

If the user prefers to use only the manual process from the beginning, in order to avoid wasting time on the edge detection procedure, a button allowing for this appears on the main menu.

The only disadvantage to the manual process is that the user has to place the dots in the same order in which they appear on the source images. It is important to note that the deformation process is based on the difference in the positioning of the dots; therefore the position of a dot indicating, say, the left mouth corner of Source Images 1 and 2, must be the same as it appears on the target image – if it isn't, the algorithm will produce an unwanted distortion. To avoid such errors and for the user's convenience, when the manual process is activated, a picture containing the correct position and numerical value of the dots automatically appears on the screen.

4.7 Results of the automatic processes

In this chapter we will present several results of facial features detection with the use of both automatic detection algorithms. The pictures have been chosen especially for their different lighting and colour settings in order to explore the limits of each process. We have also chosen male and female pictures, with, and without, lipstick and without eye make-up. The features of each picture will be detected by both automatic methods in order to define the limits of each.

(A) (B)

Figure 26: Detection process results 1. *(A) Original target image and (B) Transformed target image using the Sobel Edge Detector approach (C) facial features detection with the first automatic process 1 (D) facial features detection with the second automatic process*

In the above example both detection approaches produced similar results. The mouth detection, as has been described, is the same for both approaches. Please note that the top right corner dot is slightly outside the mouth area, because of the mouth wrinkle edges, which continue up to that point. This is the reason why the algorithm identified that area as part of the mouth shape, because it is connected with the mouth. The difference between the automatic detection approaches can be seen on the eyes. More dots are needed in the second approach, which is based entirely on edge detection; this is why the dots are not placed fully corrected around the eyes. This model has strongly coloured eyelashes (Figure 26 (A) purple square) with hard edges, which makes the detection approach to identify the correct eye limits shape impossible. Moreover, there is a beauty spot on the middle top of the left eyebrow (Figure 26 (A) green circle) which has been identified as part of the eyebrow because it is close to the eyebrow edge.

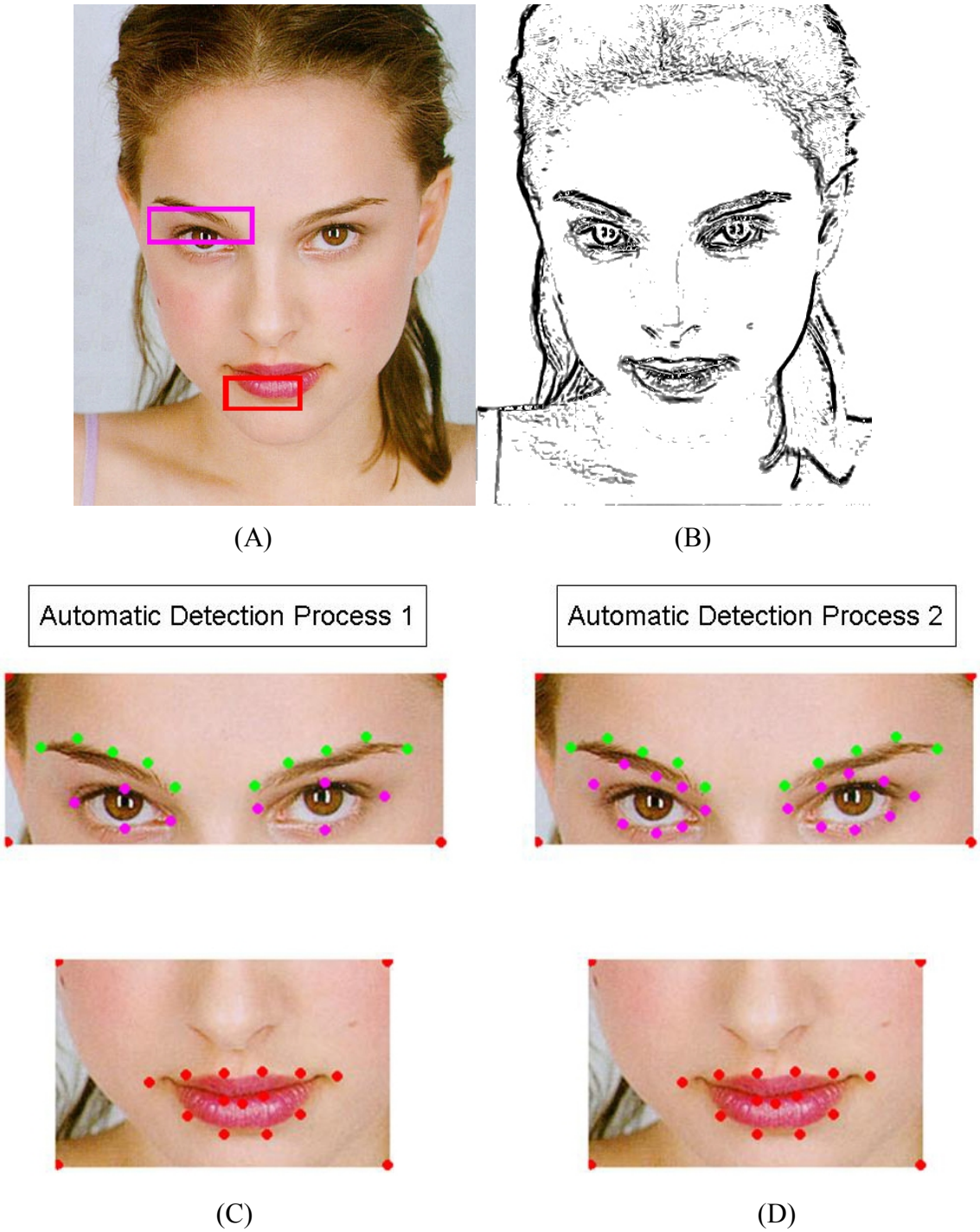


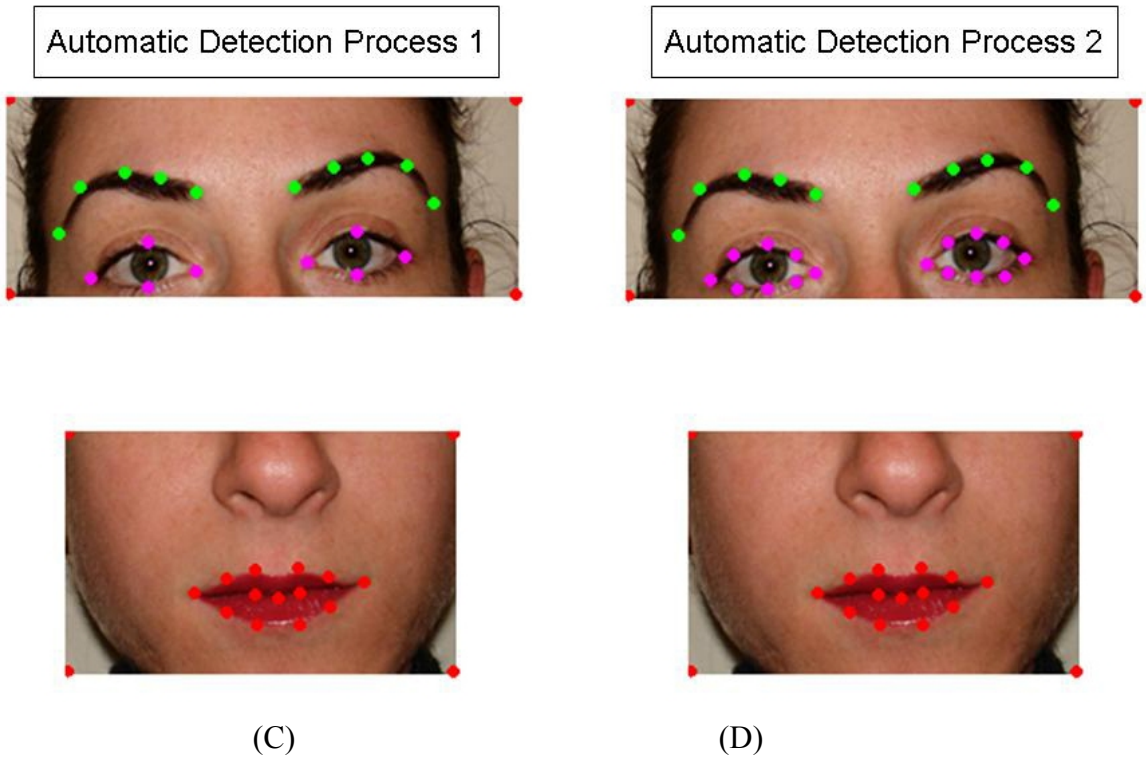
Figure 27: Detection process results 2. *(A) Original target image, and (B) Transformed target image, using the Sobel Edge Detector approach (C) facial features detection with automatic process 1 (D) facial features detection with automatic process 2.*

In Figure 27 please note the strong values of edges in this picture (A) and the shadows under the mouth, which, in the edge map, has been transformed into a large area of edges. Also note the eyes position, which is very close to the eyebrows. Because the image has been produced under studio conditions, the area under the mouth, which

contrasts strongly with the skin, produces a shadow. Unfortunately, the shadow area is very close to the mouth; therefore it is identified automatically as part of the mouth shape.

The Automatic Detection Process 2 produced incorrect results in the upper eye detection because it could not find the separation area between the eyes and the eyebrows; therefore it placed the dots under the eyes, and then, given those positions, corrected the dots in the upper area, not on the eye, but, in some cases, on the lower eyebrow. This is a disadvantage of the second process because it is only based on the edges, whereas the first process calculates mathematically the middle, upper and lower dots. Unfortunately, even if the eye detection in the second process only used four dots, the misplacement of some dots would be unavoidable.

(A) (B)



(C) (D)

Figure 28: Detection process results 3. *(A) Original target image, and (B) Transformed target image using the Sobel Edge Detector approach (C) facial features detection with automatic process, 1 (D) facial features detection with automatic process 2.*

Figure 28 presents another example showing correct detection results. The main reason why both processes produce satisfactory results is because of the target image colour and the illumination conditions. Although it is not a professional picture, it is clear enough for the Sobel algorithm to produce a proper edge map. According to the 'edge detection process' transformation result the algorithm will detect the facial features correctly, or incorrectly. It is very important to mention, however, how much lipstick helped in detecting the mouth shape.

In the following examples we will present several target images with poorer colour and illumination qualities in order to explore the limits of each automatic detection process.

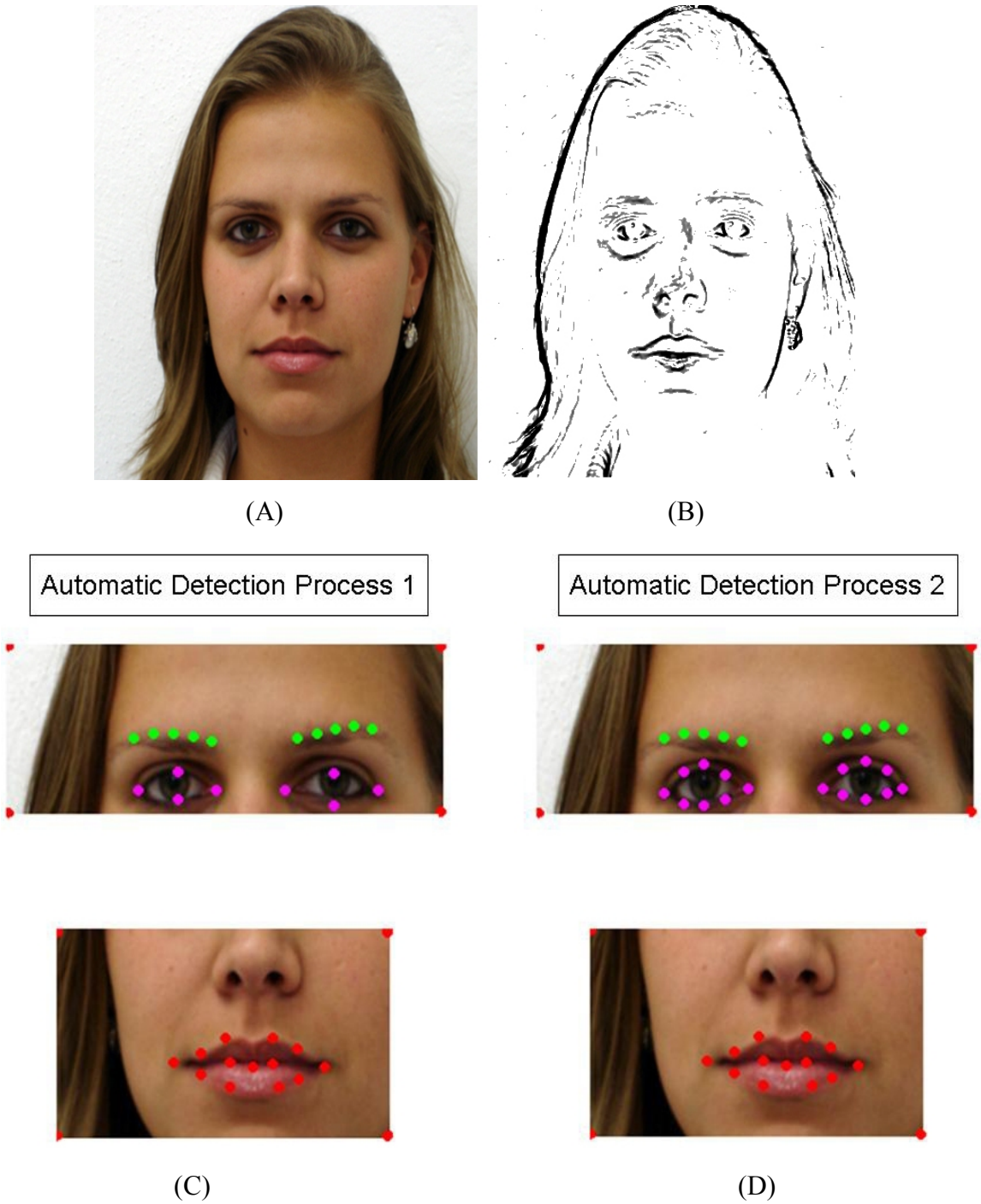


Figure 29: Detection process results 4. *(A) Original target image, and (B) Transformed target image using the Sobel Edge Detector approach (C) facial features detection with Automatic Process 1 (D) facial features detection with Automatic Process 2.*

The specific target image (Figure 29) is a low quality picture with dark illumination values. This creates difficulties for facial features detection, especially with the mouth and eyebrows. As can be seen from the edge map (Figure 29 B) the lower part of the lip is not identified clearly, therefore all the dots on the lower lip were placed one step

inside the original shape of the mouth. Moreover the eyebrows are very close to the eyes, which makes eye identification difficult when using the second process. In the first process (Figure 29 C) the distance between the corners of the eyes is very small compared with the eye opening space, consequently the algorithm correctly calculated the middle point of the eyes, but placed the upper and the lower middle dots too close together to accommodate the area the eyes cover when fully open.

(A) (B)

Figure 30: Detection process results 5. *(A) Original target image (B) Transformed target image using the Sobel Edge Detector approach*

In Figure 30, above, none of the automatic detection algorithms achieved results because the edge map, which is actually very bright, contained only small areas of edges as the dark color of the skin, together with the dark colored mouth, produced insufficient contrast to enable identification. In such cases, the manual process, where the user places the dots around the facial features, is activated automatically.

Chapter 5 Geometrical Deformation

5.1 Introduction

In order to geometrically deform a 2D picture it is necessary to construct a triangular, or polygonal mesh that will contain all the elements of a picture. Thereafter, by manipulating the mesh, the picture's characteristics will also be manipulated. The mesh comprises of a set of triangles connected by their common edges or corners, which forms either a triangular or polygonal grid consisting of a collection of vertices, edges and faces that define its shape (Figure 31).

Different representations of polygonal meshes are used for different purposes. Volumetric meshes are distinct from polygonal meshes in that they represent explicitly both the surface and volume of a structure, while polygonal meshes only represent the surface – with the volume remaining implicit. Because they are used extensively in computer graphics, algorithms exist for ray tracing, collision detection and rigid-body dynamics of polygonal meshes.

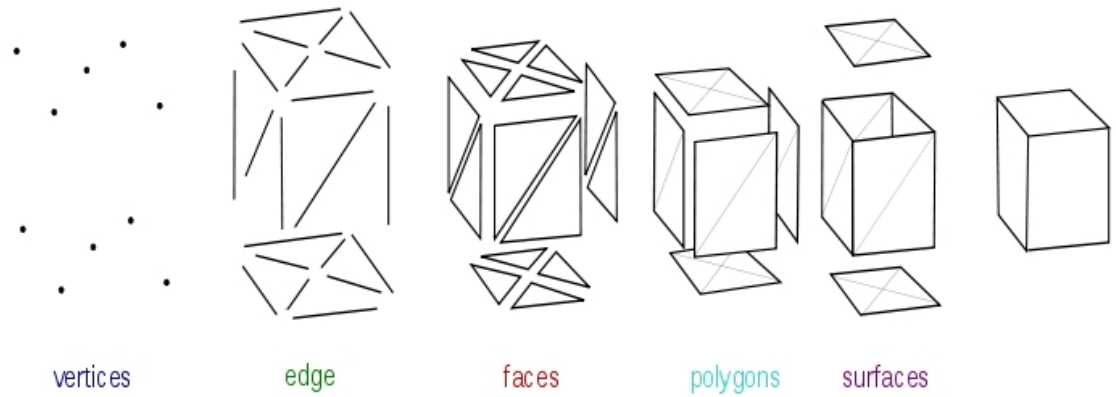


Figure 31: Grid Anatomy *[63]*

Objects created from polygonal meshes must store different types of elements, including vertices, edges, faces, polygons and surfaces (figure 31).

By using a triangle mesh, or grid, an object's shape may be changed, or a facial expression manipulated, either in 3D or 2D. As mentioned in Chapter 2.4.2 a facial expression can be produced, either by synthesizing a simple 2D facial surface or (Chapter 2.3.1.4. - 2.4.1.4) a 3D head by creating several interconnected triangular

meshes, each depicting a specific muscle, bone, skin, etc, when the user manipulates the grid edges and/or corner dots. This technique, which is called 'geometrical deformation', is often used to deform facial features in the computer vision field. In the case of 2D images, the triangular mesh is similar to that found in 3D objects, with vertices, edges and faces, with the edges defining the borders, or shape, of each feature and the faces defining the surface. It is then connected magnetically to the surface of an image, allowing the user to move an edge manually, thereby deforming the corresponding face.

In the early years [50, 5, 9, 10, 16, 19] synthesising a facial expression was a very difficult process, since, being fully manual, it required much time and a well-trained user to achieve accurate results.

One of the main disadvantages of the geometrical deformation technique is that, in 2D mode, it captures feature changes correctly, but completely ignores illumination changes; it therefore produces unrealistic results.

In this chapter we will show how a geometrical deformation process is used to deform the target image facial features in order to obtain realistic results. We will also provide solutions for possible distortions.

5.2 The Geometrical Deformation Process

As mentioned in Chapter 3.1, the geometrical deformation process does not require any user interaction, because there is no need to synthesize a facial expression from scratch; all that needs to be done is to transfer an original expression from source images onto the target image. In a similar way to Liu's [31] process, the algorithm captures the facial features movement from source image 1 to source image 2 and the movement is continuously added onto the target image. This approach explains the necessity for both source image 1 and the target image to have neutral expressions.

After detecting the features of the target image, and placing dots around them, the algorithm loads the corresponding dots on to both source images. A triangle mesh is then synthesized, based for every picture on the dots that define the shape of the features (Figure 32).

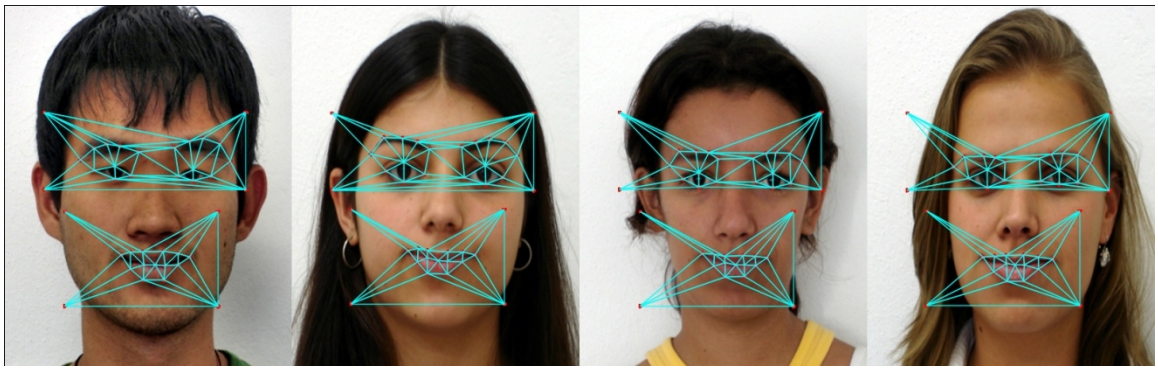
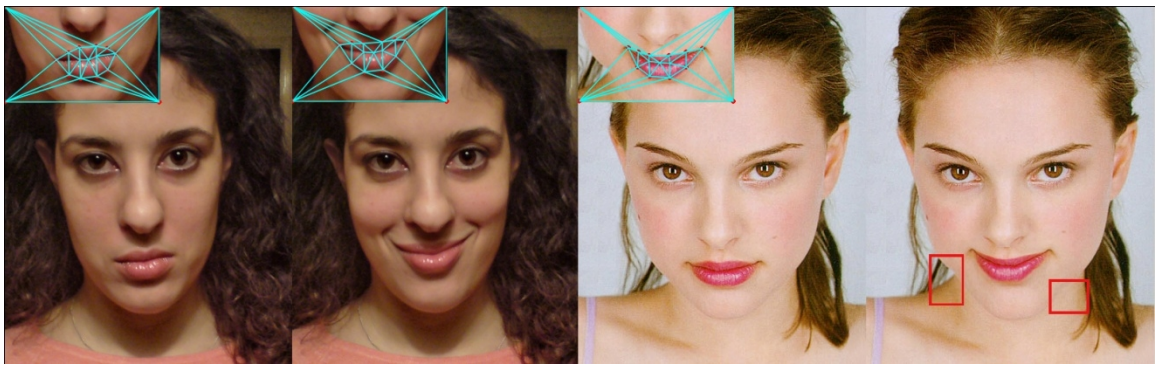


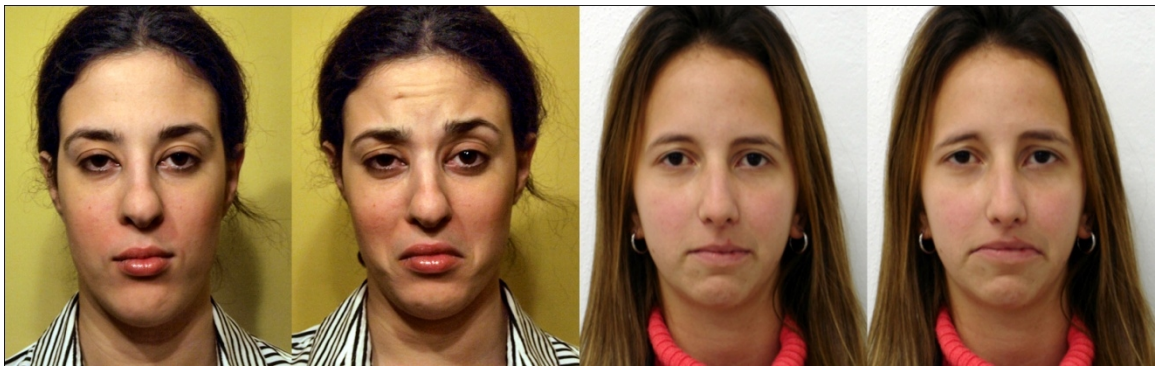
Figure 32: examples of the triangle meshes. *Two rectangles define specific facial features, such as mouth, eyes and eyebrows. The facial features have been defined with the automatic detection process 1 (Chapter 4.4)*

Based on Liu's process [31] the algorithm calculates for every dot on source image 1 the position difference of the correspondent dot on source image 2. That difference is added afterwards on the corresponding dot on the target image; the image will be warped accordingly:

$$D(u,v) = A_1(u,v) - A_2(u,v)$$
$$B_{gd} = D(u,v) + B(u,v)$$

Therefore, if source image 1 (A_1) has a neutral expression, and source image 2 (A_2) a smiling expression, the algorithm will calculate the vector difference (D) for every dot, and it will add that difference to target image (B). Because the target image has a neutral expression, as in source image 1 (A_1) that difference will produce, geometrically, the same smiling expression as in source image 2 (A_2) (Figure 33 B_{gd}).





A_1 A_2 B B_{gd}

Figure 33: Geometrical deformation of the target image 1. *A1 and A2 are the source images, B is the original target image, and Bgd the deformed target image, which accords with the facial features movement of the source images*

In the following example (Figure 34) the deformation process affects the target image more in the eye and eyebrow areas. The 'surprised' expression is an interesting example, which shows the importance of performing geometrical deformation with the proper illumination settings. It can be seen from the final picture that, after geometrical deformation processing, the model does not have the proper wrinkles; he therefore lacks a natural expression.

A_1 A_2 B B_{gd}

Figure 34: Geometrical deformation of the target image 2. *A1 and A2 are the source images, B is the original target image, and Bgd the deformed target image, according with the facial features movement of the source images*

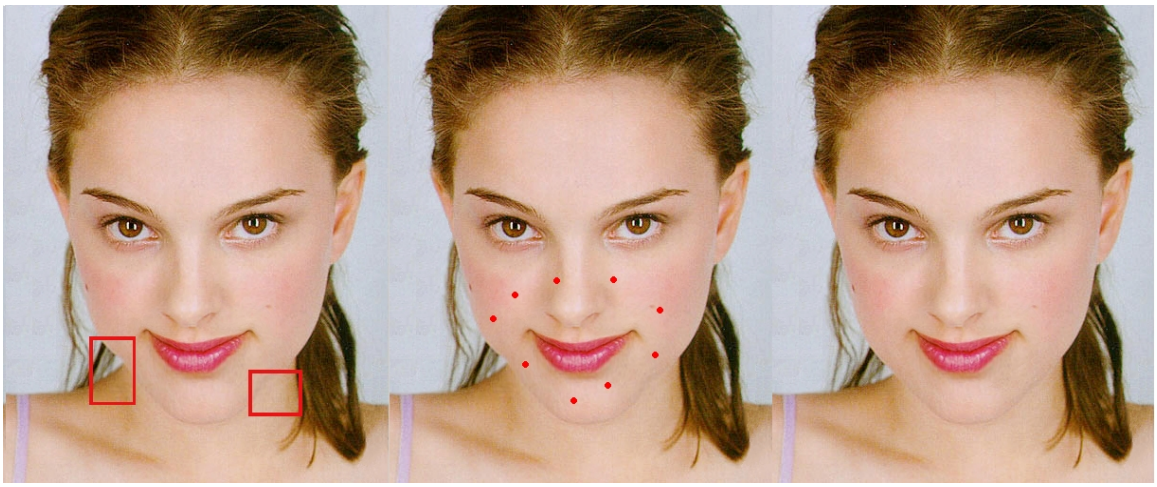
5.3 Geometrical Distortion Elimination

It is important to mention that the geometrical deformation process by itself, in some cases, lacks highly graphic results. It has been noted that, after the geometrical deformation process has been completed, the selected target's features have also been

changed, together with other characteristics. This unwanted deformation takes place because of the grid size and its cover area. If, for instance, the rectangle that includes the mouth area is much bigger than the actual mouth shape, the geometrical deformation will affect the other facial parts included in the rectangle; that distortion occurs because the triangular mesh covers the whole rectangle. As has been noted in Chapter 5.1, the triangle mesh has been magnetized with the target image's surface, therefore, when the grid changes shape during the deformation process, it not only changes the mouth shape, but it also affects the other characteristics in the rectangle.

If the rectangle includes only the mouth and some parts of the skin around the mouth, then the distortion is not visible and the result is correct. In the opposite case, though, small distortions occur, and these must be removed; to avoid this, the user may copy the correctly deformed area and place it on top of the neutral expression target image.

More specifically, if the person in the source image has a large mouth, then the rectangle for that area must be big enough to cover it. A larger rectangle may also cover the target image's chin, or other parts of the face. Unfortunately, after the geometrical deformation process is complete, this might produce distortion of the target image (Figure 35 A). To eliminate this, the user may place as many dots as he/she wishes on the target image, avoiding the distorted parts, in order to delineate the correctly deformed area (Figure 35 B); afterwards, a new window will appear showing the target image without any distortion. The copied area will then be placed on top of the corresponding place on the new window, which does not necessarily have to be a rectangle, since the user defines its shape by placing the dots him/herself (Figure 35 C). The geometrical distortion elimination process can be performed following the mouth geometric deformation, or following the mouth and the eyes/eyebrows deformation process.



(A) (B) (C)

Figure 35: The geometrical distortion elimination process 1. *(A) The user has indicated the distorted part in the corresponding automated algorithm and has questioned the quality of the result. (B) A new window appears so that the user can manually specify the corrected area by avoiding the distortion. (C) The extracted, correctly deformed, area is pasted on top of the original target image*

In the above example the geometrical deformation process has produced several small distorted areas. Those around the mouth area are similar to those on the above example, which is based on the target image's chin, which is quite different in size to that of the source image.

However, distorted parts can also be found in the eyes and eyebrows areas. This happens primarily because the hair of the model is very close to the area to be deformed (figure 36). Similarly, in the above example, in order to avoid all the distorted parts around the geometrically deformed area, the user must manually define the correct area and avoid the distortions.

(A) (B) (C)

Figure 36: The geometrical distortion elimination process 2. *(A) The user has indicated the distorted part in the corresponding automated algorithm and has questioned the quality of the result. (B) A new window appears so that the user can manually specify the corrected area by avoiding the distortion. (C) The extracted, correctly deformed, area is pasted on top of the original target image*

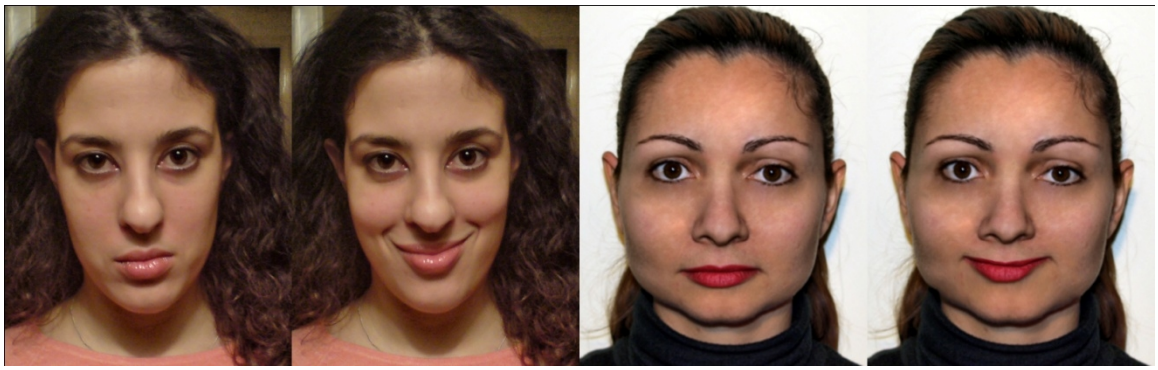
5.4 Difficulties with Geometrical Deformation

5.4.1 Introduction

The main purpose of this algorithm is to transfer realistically new expressions onto a 2D neutral expression image. However, even if the functionality of the process has been proved mathematically, sometimes the algorithm cannot produce realistic results. The failure occurs in those target images, either where the characteristics differ greatly from the source image, or where those images, when they are geometrically deformed, produce definite distortions.

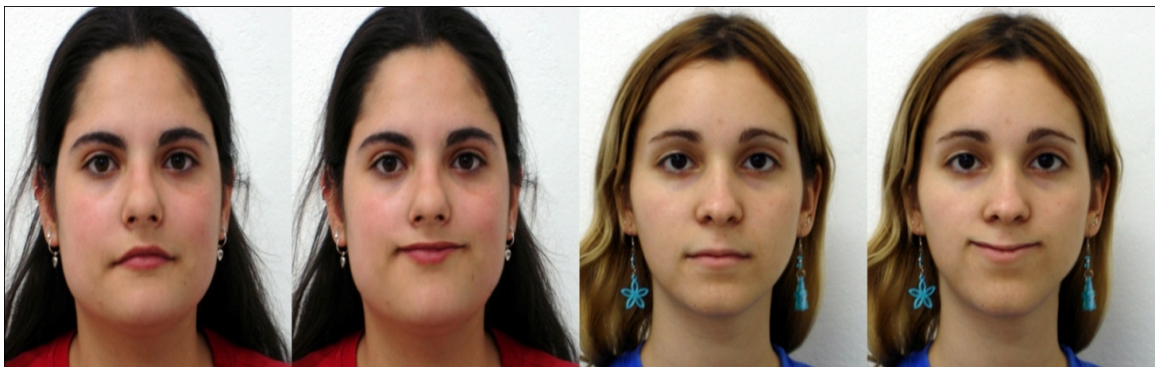
5.4.2 Facial Features Differentiation

Similar geometrical distortions can be found also between models whose facial characteristics differ a great deal, even if they are from the same race. For instance, if the source image model has a much bigger upper lip and the target image a smaller one, distortion during the geometrical deformation process is possible (Figure 37). Inaccurate results can occur with specific expressions, and accurate results with other expressions, even when using the same source image model (Figure 54 Results Chapter).



(A) (B) (C) (D)

Figure 37: Facial Features Differentiation 1. *(A) and (B) are the source images. (C) is the neutral expression target image and (D) the target image after the deformation. Note: the distortion on the mouth derives from the thin upper lip which seems to disappear after deformation, thereby providing an artificial result.*



(A) (B)

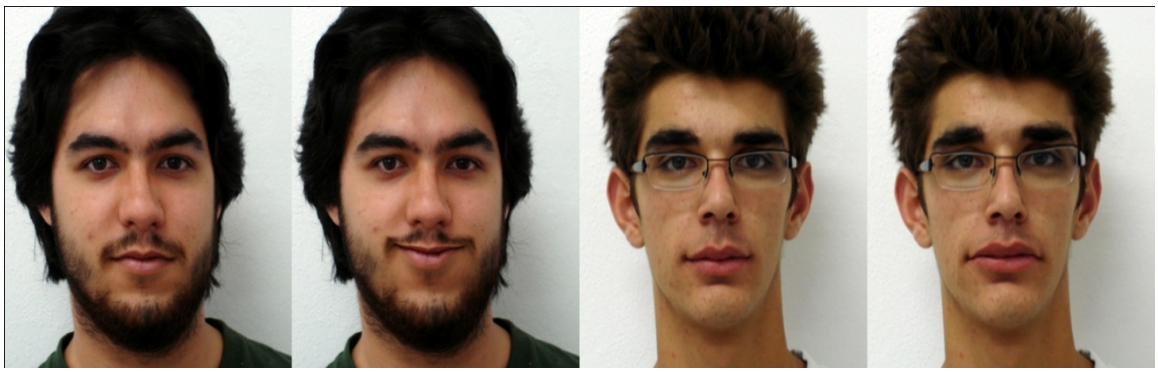
Figure 38: Facial Features Differentiation 2. *First pair: after the geometrical deformation the process produced a better neutral expression than a smiling one. The second distorted example has occurred because of the unclear mouth shape and the thin upper lip.*

Another issue that can affect a synthesized new expression is the original shape of the mouth, or other features. The geometrical deformation might give correct results, but because of the difficult original shape of the target image feature, the result can appear either distorted or artificial, or it can produce little change. For instance, the first target image from the pair on the left of Figure 38 has a neutral expression where both corners of the mouth turn down. After geometrical deformation according to the upper (Figure 37 (A) and (B) source images, both corners of the lips have been correctly moved upwards correctly according to the vectors difference from the source images, however this provides a better neutral expression than a smiling one. In the second distorted example in Figure 38 (B), the target image facial model with a

thin upper lip has not got a clearly defined shape. After the geometrical process takes effect, the upper lip almost disappears and the lower lip becomes shorter, producing a distorted final result.

5.4.3 Facial Characteristics that Produce Distortion

Another difficulty for accurate results appears on the male target images and more specifically with those that have beards. When a geometrical deformation takes effect on the mouth area, not only the lips but also the area around the lips is also deformed. If the model has a beard or a moustache those features will also be deformed, giving artificial results (Figure 39 A). Unfortunately those distortions cannot be avoided, even with the Geometrical Distortion Elimination process, because it cannot avoid the area around the lips.



(A) (B)

Figure 39: Examples of geometrical distortions. *First pair: after the geometrical deformation the process produced a smiling expression, mainly because of the target image's moustache. The second example is distorted because of the target image's glasses. Following the geometrical process, the glasses take on an unclear shape – the right glass is bigger than the left, and the shape of the left has been distorted at the right top corner.*

In (B), above, the inevitable distortion in the facial characteristics includes the effect of the shape of the glasses (Figure 39 B).

Chapter 6 Illumination Settings

6.1 Introduction

A few decades ago, when the first algorithms for facial expressions synthesis were presented, illumination settings of 2D images were avoided because of the difficulty in generating realistic illumination parameters. As previously noted, the synthesis of realistic facial expressions is the most complex process in the computer graphics area, since illumination settings differ with every expression. For instance, when a person smiles, besides the facial features, which can be transformed geometrically in shape and position, wrinkles appear around the eyes and mouth and different lighting conditions occur on the skin; these not only produce the expression, but also indicate the person's age, since, obviously, older people have more wrinkles.

Chapter 5 showed that, in the geometrical deformation process, a facial expression synthesis without proper lighting lacks naturalism. Liu's process turned out to have been one of the most innovative approaches in the area of synthesizing realistic expressions, because, while he did not focus on producing new lighting conditions for synthesising an expression, which would be almost impossible, he transferred natural illumination settings onto the target image.

The illumination settings transfer approach is divided into five steps:

1. ***Source images equalization with the target image****:* is the process where the source image's facial features, using the geometrical deformation process, are equalized in shape and position with the correspondent facial features of the target image. In order to extract the illumination settings of the second source image, which has the preferred expression, it is necessary to divide it with the first source image. However, in order to extract the illumination settings only it is important to equalize the features of the fist source image with those of the second source image, by using the geometrical deformation process. If we divide both equal images the only differences extracted will be the illumination settings from the first source image which is contained in the second source image's expression. Unfortunately, if that map has been directly transferred to the target image it is quite possible that the illumination settings will not be properly placed. Therefore, instead of equalizing the facial

features of the first source image with the second source image, the algorithm will equalize both of them by using the target image's facial features, shapes and positions.

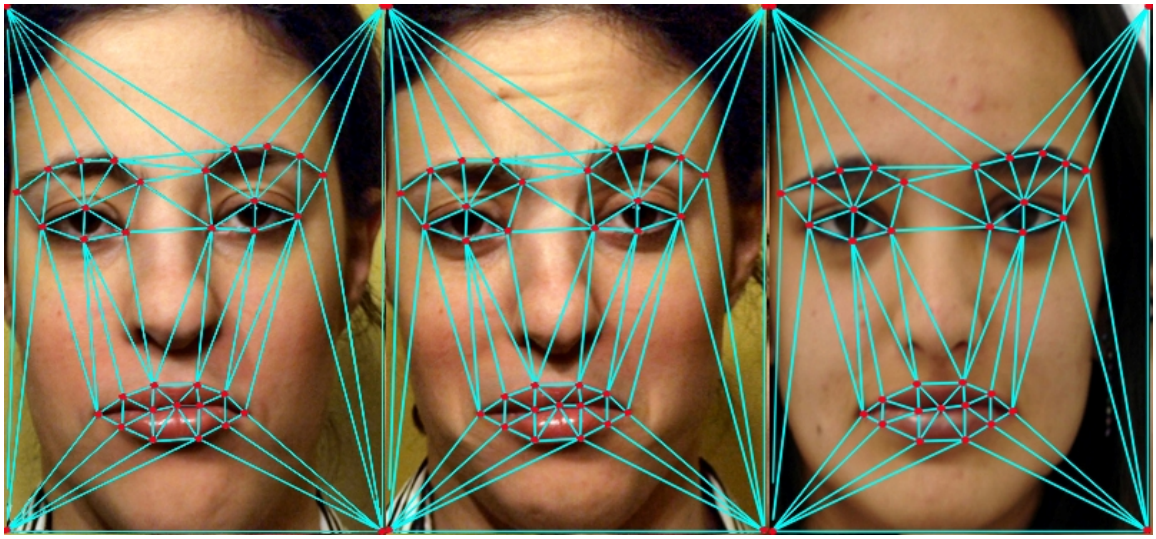
2. ***The colour normalization process:*** is the process which combines the colour value of each pixel of the source images with the correspondent pixel's colour value of the target image. This affects the illumination settings of both source images: the purpose of this process is to minimise the colour and lighting differences of the source and target images, such as image quality, skin colour etc, which will be normalized during with the division of each source image with the target image. This process takes place just before the division of the source image, pixel by pixel.
3. ***The division of the source images, pixel by pixel:*** during this step the algorithm will automatically extract the illumination settings, resulting in a map containing all the proper lighting settings for a high graphic quality expression synthesis, which will be transferred onto the target image.
4. ***The ratio image threshold:*** this has been invented in order for the user to transfer a specific amount of illumination data onto the target image for a realistic final result.
5. ***The distortion elimination on the illumination transfer approach:*** as can be seen, the geometrical deformation process focuses on specific features; therefore, if other features are included in the rectangle, distortion will result because the previous steps (geometrical deformation – equalization process) will not have been applied to them. By using the distortion elimination process the user will be able to transfer a specific area containing the preferred illumination settings, thereby avoiding distortions.

6.2 Source images equalization with the target image

The first stage of the wrinkles transfer approach requires the source image facial characteristics to be equalized with the corresponding characteristics of the target image. The process of enabling equalization will again be geometrical deformation. The characteristics of both source images must match perfectly with those of the target image in position and shape. If this process is avoided, or is unsuccessful,

distortion will appear on the lighting settings in the result. Figure 42 shows examples of the distortion that appear when it fails:

As stated in Chapter 3.3 it is not necessary to deform features that are not involved in the chosen expression; consequently, in the geometrical deformation process, the hair, neck, ears and shoulders are avoided. In Figure 40, a rectangle is used to contain the eyes, eyebrows and mouth areas for all of the images.



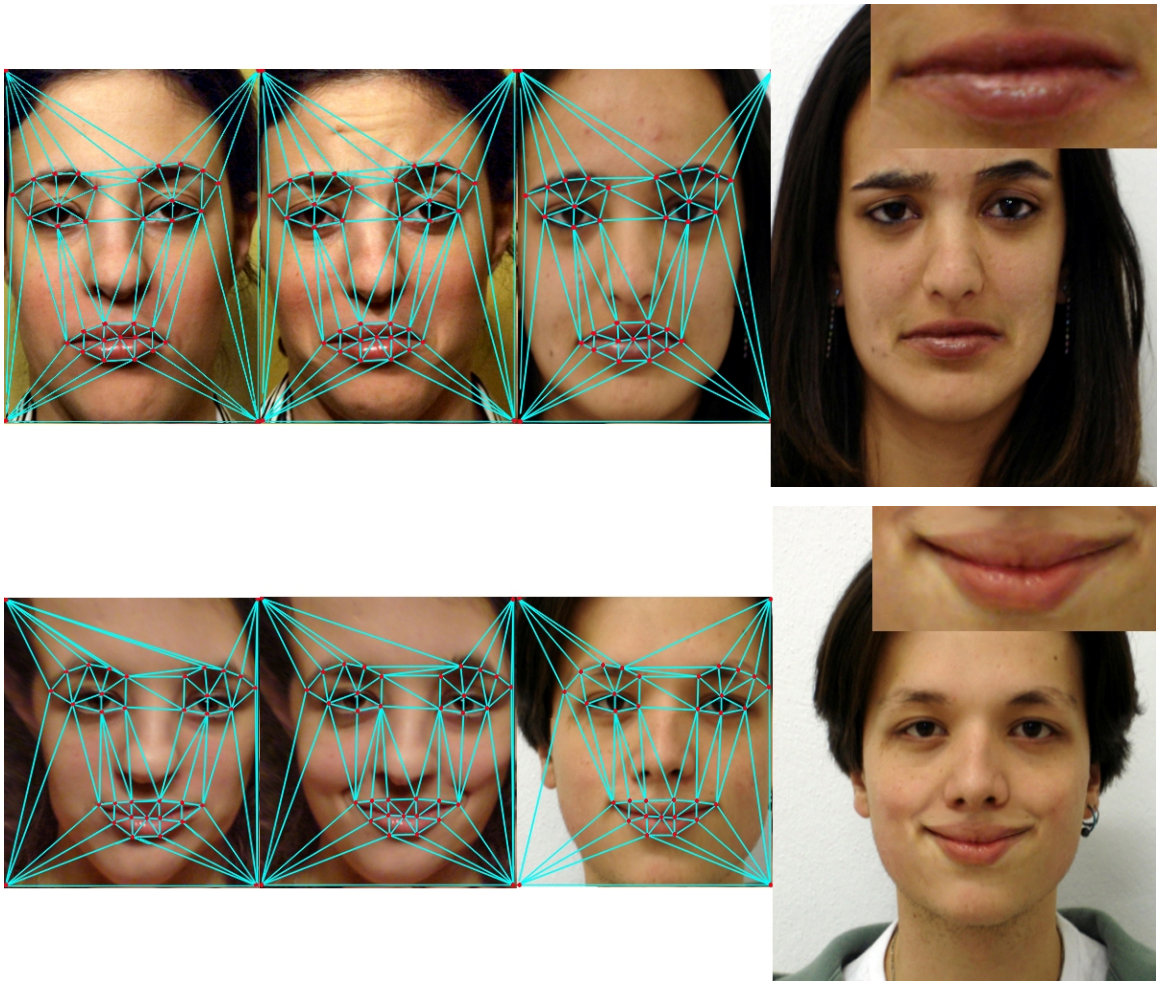
(A) **(B)** **(C)**

Figure 40: Source images equalization process. *The first step of the illumination settings transfer process is to equalize the characteristics of the source image with those of the target image. The dots indicating the features of target image (C) are temporarily stored following the geometric process. The dots specifying the features of the source images are loaded from the facial expressions library.*

Specific features were used because the synthesized expression will be based on them. When the source images in the following chapters are divided, one into the other, those characteristics, and only those, must be perfectly equalized, the remainder, even if they appear in the final result, will be removed.

The same number of dots were used to define the eyes, eyebrows and mouth as were used on the target image during the first process, described in Chapters 4.4 - 4.5 five for each eyebrow, four for automatic 1, or eight for automatic 2, for each eye and twelve for the mouth. The dots defining the features of the source images are stored and loaded automatically at the beginning of the process. The dots defining the features of the target image are copied from the previous process and pasted in the new rectangle.

The geometrical process is repeated automatically several times until the dots on the source images are aligned with the corresponding dots on the target image.

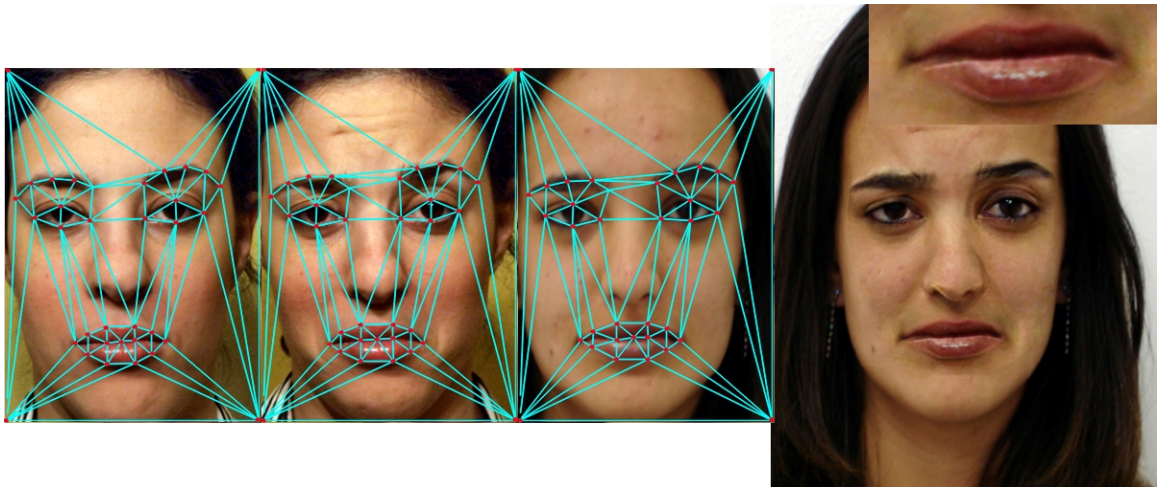


(A) (A') (C) (F)

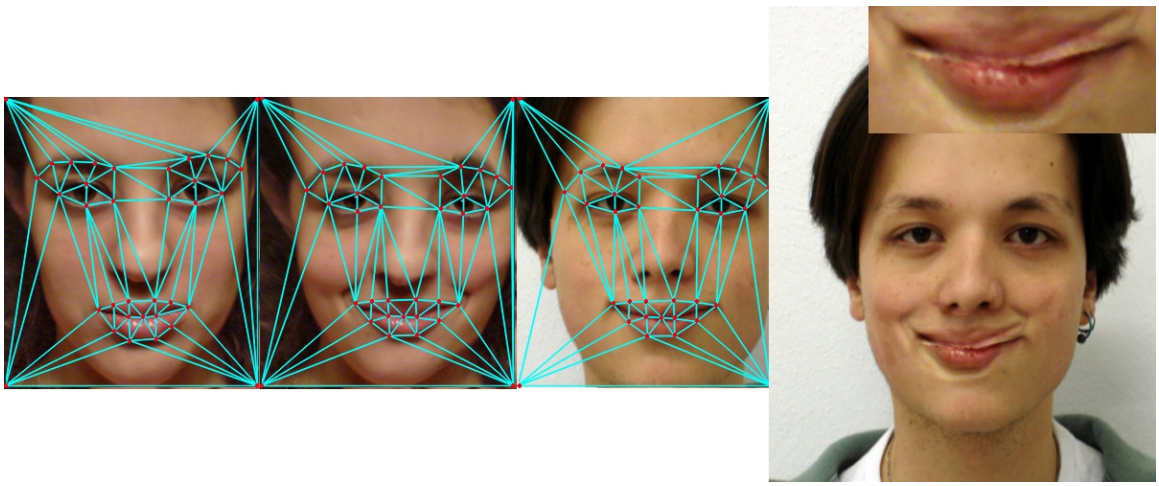
Figure 41: Correct repetition process. *After ten repeats of the geometrical deformation process, the features of source images (A) and (A') are now identical to those of the target image (C). All the dots have equal coordinates with the corresponding dots on the target image. It can be seen that even the triangle meshes are almost identical. In the final result, the lighting conditions have been transferred correctly without any distortion, especially to the mouth.*

We have discovered that the optimum number of repetitions the geometrical deformation process requires is ten, since this gives better result on the source images; any more than ten repeats is a waste of processing time, because the dots are equalized statistically against the position of the target image dots. Figure 42 shows examples of distortion with fewer than ten timed geometrical deformation procedures

and Figure 41 highly in graphics results under the predefined amount of the repetition the geometrical deformation process.



(A) (A') (C) (F)



(D) (D') (E) (F2)

Figure 42: Distorted repetition process. *After repeating the geometrical deformation process the source images (A) and (A') seven times, are not completely identical with the features of target image (C). The distorted result has been transferred onto the final result on the mouth and nose areas. Distortion occurs when source images (A) and (A') are not completely identical in all their features; these small differences appear as very bright lines. A bright line appears on the left corner of the mouth and on the nose. In the second example (D), (D') (E) and (F2) the geometrical deformation process was repeated five times. The mouths of source images (D) and (D') show enormous differences in shape and position. The distortion is great and, in the final result, it can appear as if a second mouth has been imposed on top of the synthesized expression.*

The mathematical method for calculating the position of the dots differences between the source and the target image is:

$$T_g(u,v) - S(u,v) = D(u,v)$$

$T_g(u,v)$ is the position of the dots on the target image, $S(u,v)$ is the position of the source image dots, and $D(u,v)$ is the difference between the positions of the dots on the target and source images. The distance becomes shorter with each repetition until it disappears completely.

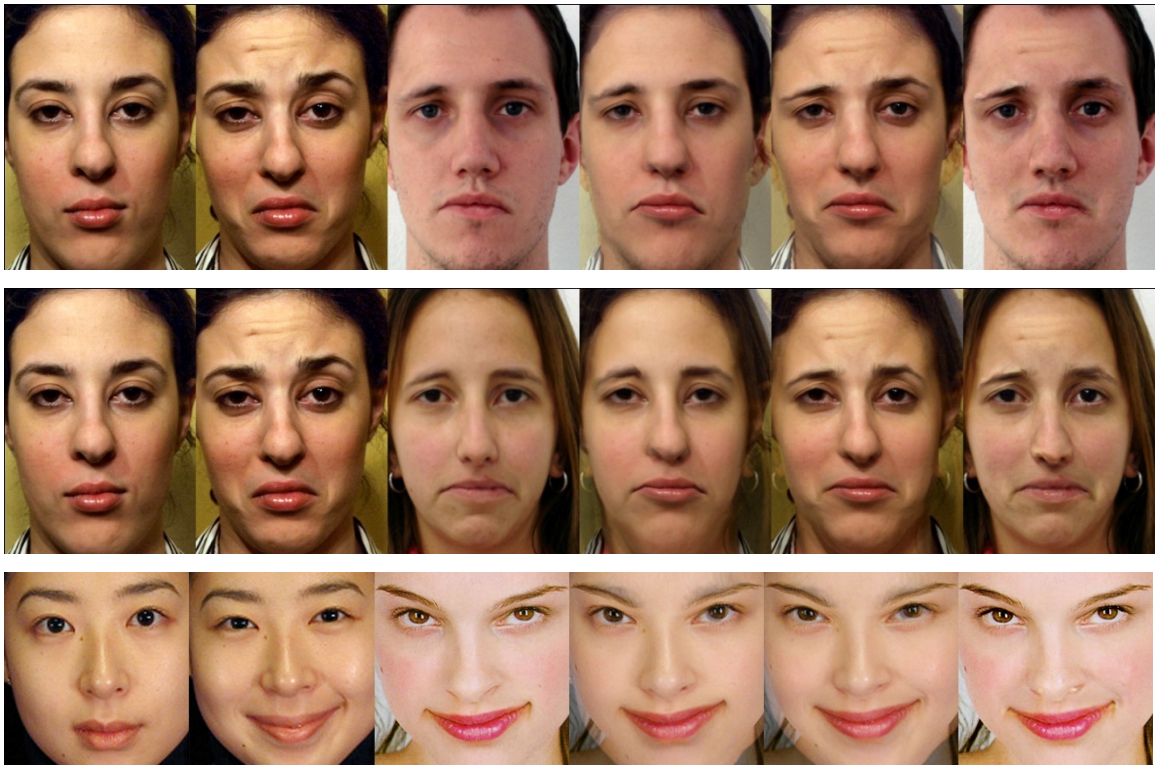
According to the source images equalization results of the target image, it can be seen that the distortion appears on the source images, especially on the neutral expression on the first source image. To avoid any misunderstanding, this result can be characterized as 'normal' and accords with the procession specifications.

From the beginning, we aimed to equalize the facial expressions of the source images with the already geometrically deformed target image. We also aimed not to transfer an expression, but to equalize it. Therefore, even if the source images look geometrically distorted, when they are divided one by the other, this distortion is eliminated, revealing only the illumination settings, which may then be superimposed on the target image.

6.3 Colour Normalization

Colour normalization is a process that prepares the source images for the extraction of the proper lighting settings. If we follow Liu's approach, we only have to divide the source images in order to extract the illumination settings. However, this can only work properly if the colour texture, or skin, has similar specifications to the target image, since, if they are different, the final result will be distorted.

In order for the synthesized result to be of high graphic quality, it is essential that the skin texture of the source images be normalized with the target image skin colour. If there is much difference, then the transferred illumination settings will be distorted and the result will appear artificial. Also, if the source images are in black and white, the distortion will be greater, since, because the illumination settings are part of the skin, the lighting settings will contain a grey shadow. In this section we will present an effective solution for eliminating such problems.



(A) (A') (Bd) (ABd) (A'Bd) Bd'

Figure 43: The colour normalization process. *(ABd) and (A'Bd) are the source images combined with the target image. (Bd') is the final synthesised expression.*

It is important to note that the picture quality, and the colour of the skin, are crucial factors in the illumination transfer approach, therefore, instead of dividing the source images, one by the other, each is combined, pixel by pixel, with the target image, not only to retain the wrinkle values of the source images, but also to combine, or normalize, them with the illumination and skin colour settings of the target images.

The results are two new source images – *ABd* and *A'Bd,* where *Bd* is the target deformed image. However, in order not to mislay some percentage of the source images illumination settings, the first pair, *ABd* uses 50% of its images and the second pair, which has the desired expression, uses 70% from the source image and 30% from the target image – *A'Bd*. The different percentage will collect all the source image illumination details, together with the skin colour settings of the target image.

6.4 Ratio Image Threshold

As has already been mentioned, a specific facial expression can be obtained from a combination of a highly graphical deformation process and a synthesis of realistic

illumination settings. Except for the colour normalization of the source images with the target image it is essential to transfer a specific amount of illumination data onto the final result.

My research has found that, for every target image used, a specific amount of data produces a realistic result, and that the application of more, or less, illumination data produces artificial results, and in some cases distortion. The following examples show the huge effect the illumination data has on the final result.

0% 30% 50% 70% 100%

Figure 44: Examples of the 'Ratio Image Threshold' *usage in different percentages*

To eliminate artificial effects we have established an illumination data threshold, which allows specific amounts of data to be transferred onto the target image. More specifically, following the synthesis of the two new images, the algorithm divides them, one with the other (*ABd*/*A'Bd*), in order to extract their illumination settings; this we describe as a 'ratio image'. A threshold has been invented in order for the user

to transfer a specific amount of data from the ratio image to achieve a realistic final result on the target image.

By using this threshold, the user has the option of transferring a specific percentage of data. The ratio image acts rather like a 'transparent file', which is added on the target image through a multiplication process – *Ratio Image x Bd = Bd'*. This threshold helps to avoid details that produce lighting distortion on the target image, such as beauty spots, scars, etc, which derive from the source images.

6.5 Distortion elimination on the illumination transfer approach

If the rectangle containing the facial features for the 'illumination transfer approach' is bigger than the area of the eyebrows, eyes, nose mouth area, and it contains hair, or ears, or scars, it will produce a distortion, because all the head characteristics of the source images have not been equalized with the corresponding characteristics of the target image. This will, therefore, transfer the distorted illumination settings to the final result. To correct this, simple 'user interference' is provided.

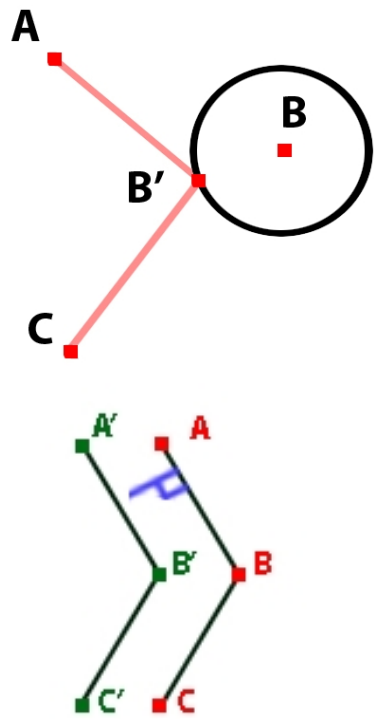


Figure 45: *Illumination elimination approach. (a) B is the original dot placed by the user, B' is the internal dot on the perimeter of the circle that describes the shortest distance between A*

and C, b) The blue color diagram describes the orthogonal vectors used to gradually change the color of the pixels area.

The user can identify the non-distorted area by simply placing any number of dots around the area containing the proper lighting settings (Figure 46). Afterwards the algorithm will automatically extract that area and paste it back to the original target image, thereby avoiding the distorted parts.

When the user places the dots on the face, a small circle of 20 pixels diameter is drawn with the specified dot at its center. According to the neighbors – dots A and C – the algorithm will calculate a new dot, B', which describes the shortest path from A to C via the circle. B' will be defined as the inner dot (Figure 45 a). This process will be repeated for all the dots placed by the user. The purpose of the internal dots is to achieve a gradual change of color, which takes place in an approximately 10 pixels area between the outer and the inner dots, from the color of the original target image to the color of the newly synthesized area. Therefore, two borders, ten pixels apart, are created around the extracted facial area. The outer border consists of the line achieved by connecting the original dots, placed by the user, and the inner border, by connecting the new internal dots created for the above procedure.

Orthogonal vectors are used to move from one pixel to another, inwardly, from the outer border lines, in order to gradually change the pixels in the area (Figure 45 b).

Figure 46: Two final results from the "Distortion elimination on the illumination transfer" process. *The images on the left are the final rectangles that have distorted areas around the eyes eyebrows and mouth areas. The red dots have been placed by the user in the preferred places. The right hand images are the final results after the process is complete. The extracted illuminated area has been pasted smoothly on top of the geometrically deformed target images producing high graphic and detailed results.*

Chapter 7 Video Animation Synthesis

7.1 Introduction

Video facial animation is a major scientific area in computer graphics that encapsulates models and techniques for generating and animating images of the human head and face. The quest for the accurate representation of human faces during verbal and non-verbal communication, together with advances in computer graphics hardware and software, have generated considerable scientific, technological and artistic interest in the subject.

Computer video facial animation is not a new endeavor. The earliest work with video animation was presented in 1973 by Gilleson, who developed an interactive algorithm to assemble and edit line-drawn facial images. The early 1980s saw the development of the first physically-based muscle-controlled face model by Platt and also the development of techniques for facial caricatures by Brennan. A new muscle-based model was developed by Waters in the late 1980s and in 1985 [50], the short animated film "Tony de Peltrie" became a landmark, when, for the first time, computer facial expression and speech animation were fundamental to the story.

There followed the development of an abstract muscle action model by Magnenat-Thalmann and colleagues [32] together with advances in automatic speech synchronization by both Lewis and Hill. The 1990s, therefore, saw a steady development of these techniques as the key storytelling components in animated films such as "Toy Story", "Antz", "Shrek" and "Monsters", and computer games like "Sims". In 1995 "Casper" became an important milestone in that it was the first movie to have a digital facially animated lead actor produced exclusively for it; this was followed later in the same year by "Toy Story"

Two-dimensional facial animation is commonly based on a morphing technique that allows in-between transitional images to be generated between a pair of target still images, or between frames from sequences of video. The technique usually consists of a combination of geometric deformations that align the target images and a cross-fade that creates a smooth transition into the image texture. An early example of image morphing can be seen in Michael Jackson's video for "Black or White".

Another form of animation obtained from images consists of concatenating sequences captured from video. In 1997 Bregler et al. described a technique called 'video-rewrite', where existing footage of an actor is cut into segments corresponding to phonetic units that are blended together to create new animations of a speaker. Video-rewrite uses computer vision techniques to track lip movements automatically in video and these features are used in the alignment and blending of the extracted phonetic units. However, this animation technique only generates animations of the lower part of the face, so they need to be combined with video of the original actor in order to produce the final animation.

After 2000 films became more sophisticated. In "The Matrix Reloaded" and "Matrix Revolutions" the use of dense optical flow from several high-definition cameras captured realistic facial movement at every point on the face and "Polar Express" film utilised a large Vicon algorithm to capture upward of 150 facial points. However, although these algorithms are automated, a considerable amount of manual clean-up is needed to make the data usable. Shortly thereafter another milestone in facial animation was reached with the character-specific shape-base algorithm applied to "The Lord of the Rings". Pioneered by Mark Sagar, this FACS (Facial Action Coding Algorithm, Chapter 2.2) based algorithm was also used on "Monster House", "King Kong", and other films.

7.2 Overview process

Another operation that our algorithm can perform is the synthesis of six facial expressions simultaneously for purposes of video editing animation. As with the individual facial expression synthesis process, our algorithm contains a set of source animated pictures with the relevant documentations stored in a library. When the process is selected by the user the automatic edge detection of the target image's features begins. Where the user intervenes manually by correcting the dots, the algorithm stores the position of the dots in a temporary folder so they may be re-used in the same positions in order to obtain the remainder of the image. All the synthesized target images are capable of being produced by taking the neutral expression of both the source and the target images as inputs, along with the subsequent source image expression. Figure 47 B presents several examples of synthesized target images based on the same sad facial expression Figure 47 A.

After having synthesized the sequence of the target's animation images, a video editing software package may be used to render them in video format. By using a "fade in, fade out" effect, very realistic video animation can be achieved.

(A)

(B)

Figure 47: Video animation synthesized examples. *The first group, (A), contains all the source images for the sad young person video animation. The second group, (B), contains examples of target images and how they were animated according to group (A).*

In this chapter we will present the video synthesis function created by the algorithm, together with the predefined 'individual facial expression synthesis' (Chapters 3-6). The video animation process is divided into the following steps:

1. **Preparation process:** after the expression has been selected, the target image is automatically transformed; if required, the size may be adjusted according to the size of the source image. Afterwards, it loads the source images, together with their documented corresponding features
2. **Automatic detection, or manual definition, of the target image's facial features:** when the position of the dots is approved by the user, they will be stored temporarily for further use until the video synthesis process is completed
3. **Geometrical deformation of the target image:** the target image will be deformed four times in sequence for every facial movement of the source images until the final expression has been achieved
4. **Elimination of possible distortion:** after every geometrical deformation the user will have the option to remove any possible distortion
5. **Source images equalization:** the equalization process for each of the five source images with the corresponding geometrically deformed target image
6. **Colour normalization:** the source image's colour normalization process with the target image skin colour settings for the extraction of smooth illumination settings
7. **Ration image threshold:** utilising the ratio image threshold for the proper amount of illumination data, which will be added to the final target images' results
8. **Distortion elimination on the illumination transfer approach:** for each of the four final results the user will be asked, if necessary, to eliminate any illumination distortion

7.3 Practical considerations

The purpose of the algorithm is to create a video animation synthesis. The difference between this process and the one picture synthesis is the repetition of the main process several times until the desired expression has been achieved on the target image. In order to synthesize a proper video animation expression, it is important that more than four images, where each image represents a slight change of expression, are synthesized until a satisfactory result is achieved. Therefore pairs of source images will be used – the first pair being neutral and the others depicting the slight changes.

For practical reasons the whole process does not allow for the simultaneous execution of all the images. Unfortunately, even the one picture synthesis method requires a great deal of algorithm memory and power processing; moreover, several windows are required, which is true even for the one picture synthesis. These windows correspond to the automatic facial features detection process, the geometrical deformation process and the illumination settings transfer approach. Therefore, in order to run the video animation process, more than forty-four windows would be needed, which would be difficult for the user to manage.

Based on the above practical considerations, therefore, we have decided that the algorithm should operate with one pair of images only.

7.4 Preparation process

After the user-preferred facial expression is selected, the algorithm starts the video animation synthesis process. Firstly it identifies the original target image size and compares it with the size of the source image. Note that all the source images have a specific size of 404x505px. If the target image size differs, then the algorithm will rescale it accordingly and it will store the original image size in the algorithm memory. After the video animation synthesis of the target image is complete, the algorithm will restore the original size image to all the results.

After the target image size rescale process is completed, the algorithm will continuously ask the user to specify the preferred automatic process. As described in Chapter 4, the algorithm will transform the target image into a greyscale format, then, by utilising the Sobel detector, into an edge map, which will be used to define the facial features. After the facial features automatic detection process, as described in Chapter 4, is completed, the user may make manual corrections. The great advantage

to using the video animation process is that the dot coordinates for the mouth, eyes and eyebrows of the neutral expression target image are stored in the algorithm library (Figure 48).

(A) (B) (C) (D)

Figure 48: Preparation process. *(A) to (D) presents all the preparatory steps before the geometrical deformation of the target image. (A) is the original neutral target image, rescaled if needed. (B) is the target image transformed into greyscale format suitable for the automatic detection process. (C) is the greyscale target image that has been transformed into an edge map by the Sobel process (Chapter 4) (D) is the result after the automatic detection process 1 (Chapter 4.4) and the manual corrections, if they are needed*

The reason for storing the target image dot coordinates in the algorithm's memory is because we wished to reduce the procedure time. As mentioned earlier, the video animation synthesis process is based on pairs. It would be annoying were the algorithm to have to identify, from scratch, the target image facial features for every change of expression; therefore, with the first pair, the algorithm is the same as for the one picture synthesis process; after identifying the facial features automatically it moves to the second pair and continues until it reaches the last pair. The algorithm will then load the temporary target image neutral expression documentation and continue to the geometrical deformation process.

7.5 Geometrical deformation process

The geometrical deformation process is as described in Chapter 5; the algorithm proceeds with one pair of images at a time, then, after the facial features detection process is completed, all the dots are connected by triangles. For every pair of source images, the differences between the coordinates of the respective dots will be

calculated and added to the corresponding dot coordinates of the target image, first on the mouth, then on the eyes and the eyebrows (Figure 49).

Figure 49: geometrical results of the video animation *synthesis from the neutral to the final sad expression*

7.6 Geometrical Distortion Elimination

After geometrical deformation the user will be able to remove any distorted areas by activating the geometrical elimination feature. As described in Chapter 5.3, there is an option whereby several dots may be placed around the correctly deformed area in order to avoid the distorted parts. The example in Figure 50 shows the distorted areas around the mouth and the eyes-eyebrows that have been avoided.

(A) (B)

Figure 50: geometrical distortions *appear on image (A) inside the red squares. By using the geometrical distortion elimination process the distorted parts have been avoided. Image (B) presents the corrected geometrical deformation result*

7.7 Source images equalization with the target image

The next step of the geometrical elimination process is to equalise the source images with the deformed target image (Chapter 6.2). As with the geometrical process, the current function is performed each time on one pair of source images. Therefore, the algorithm will take the current pair of source images and, by using the geometrical deformation process, will equalize them with the corresponding deformed facial features of the target images.

7.8 Colour normalization

The colour normalization process then takes place, whereby the algorithm will divide each of the deformed source images with the target image. The source images' pixels will be mixed – i.e. normalized with the illumination and skin colour settings of the target image pixels (Chapter 6.3).

7.9 Ratio Image Threshold – Distortion elimination on the illumination transfer approach

After the colour normalization process is complete, the next step is the illumination settings transfer, where each deformed target image will be changed, realistically, according to the illumination settings of the expressions on the source images. Then, the newly created images, which derive from the division of each source image with the target image, will be divided, one with the other, in order to extract the illumination setting of the new expression; they will then be added to the deformed target image by utilising the 'ratio threshold tool' (Chapters 6.3 and 6.4). That process will be repeated for every pose of the source image until the video animation sequence is completed.

(A)

(C) (D)

Figure 51: Distortion removal process. *the two images in group (A) present the source images 4 and 5 according to the source images sequence on figure 47 A; more specifically they indicate the facial wrinkles and beauty spot, indicated by coloured squares, which will be transferred in the final target image results as distortions. Group (C) presents the target images 4 and 5 with the specified characteristics of group A as distorted areas. Group D shows the same target images after the 'distortion elimination on the illumination transfer' process, with no distortion and realistic expressions*

After the synthesized result of the target image has been established, the user will be given the option to use the 'distortion elimination on the illumination transfer' process in order to avoid the distorted parts, by placing dots manually around the relevant areas and copy-paste them on top of the original neutral target image (Chapter 6.5).

It is important to mention why this process must be performed separately for each synthesized target image: the first synthesized target image's dots coordinates, which specify the selected illuminated area, is stored temporarily on the algorithm, to be loaded later, in order to obtain the remaining synthesized results. This will reduce processing time and also facilitate a more automated process. Unfortunately, the automatic process, although it reduces the processing time, in some cases it may also miss important illumination settings, or create facial expression inaccuracies, possibly resulting in distortions. Therefore, each synthesized target image must contain all the correspondent illumination settings provided by the source images, which, of course, can differ from one pose to another. Therefore, if only the dot position coordinates from the first target image results are stored, important data might be lost from the remaining synthesized images. Consequently, the process *must* be run separately for each target image.

Chapter 8 Results

8.1 Introduction

According to the Liu et al [31] approach, the executable time for a facial expression synthesis is approximately thirty minutes. The user has to place dots manually to cover all the features on both the source and target images. The user must be very accurate, for, if a dot is placed incorrectly, distortion will certainly occur during geometrical deformation. However, because our process is primarily automatic, only when geometrical, or illumination, distortion occurs does the user intervene with a limited manual interaction.

Geometrical deformation is based on two specific areas, therefore, there is no need to place dots around all features, such as hair, ears, nose, etc; also, by using an expression library database, the algorithm can automatically load all the dot coordinates of the source images.

The number of dots required, therefore, is reduced from three hundred, as in Liu's approach, to sixty-nine, and the time required for execution is reduced from thirty minutes to less than two minutes – depending on the computer's processing capabilities.

In this chapter we will present several examples of synthesized facial expression, describe the various advantages and disadvantages of processing and their effect on the final result.

This section also highlights the difficulties of synthesizing one picture, or a facial video animation, firstly from a 'geometrical deformation' angle and then from an 'illumination settings transfer' point of view.

Each individual expression is unique, since, not only are the features settings, such as mouth, eyes and eyebrows, changed, but so, too, are the lighting settings; therefore, all our selected source images are of different persons who have had their images taken under dissimilar color and lighting conditions in order to prove the accuracy of our algorithm.

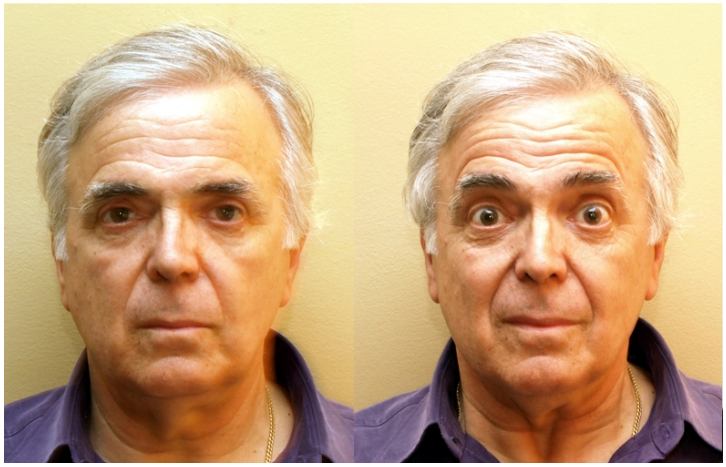
8.2 One picture synthesis process

A new approach has been applied in order to detect automatically the characteristics of the target image, to separate them into two areas, to deform them and then to transfer the appropriate illumination data from the source images to the final image.

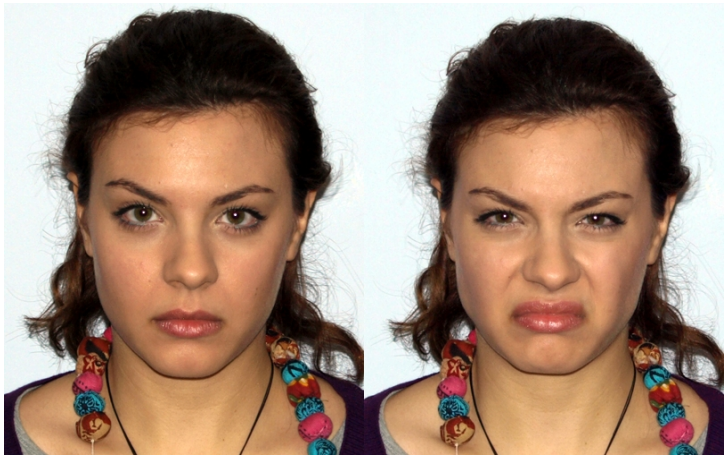
The source images have been carefully chosen because their facial expressions and illumination settings vary; these are presented in Figure 51. The images are grouped in pairs of neutral and non-neutral expressions.

Example 1 – neutral and smiling source images

Example 2 – neutral and sad source images



Example 3 – neutral and surprised source images



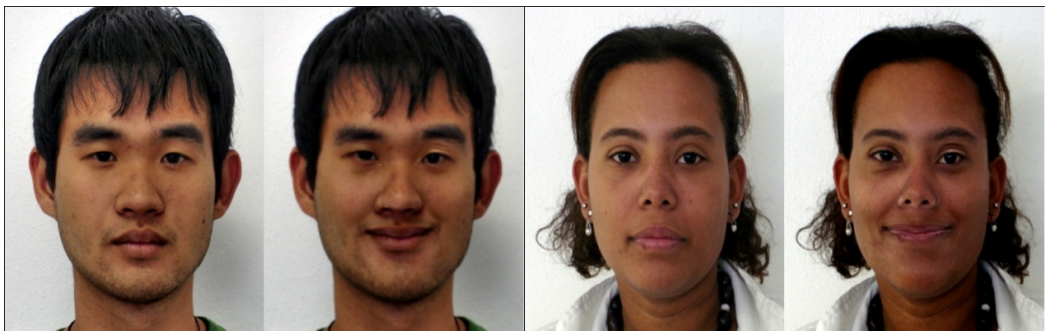
Example 4 – neutral and disgust source images

Figure 52: source images from the library *database used for synthesizing new expressions*

Figures 53 and 54 show the target images and the resultant images following completion of the deformation and the illumination setting transfer processes, after using, as source images, the pair at Figure 52, Example 1. Please note that the wrinkles around the mouth produce a realistic result by providing a high level of physical detail and deformation has only been applied to the mouth and eyes. Note also that splitting a facial image into areas does not detract from the naturalness of the expression. Even though no deformation was applied to the nose by the user, it is deformed according to the geometrical deformation of the mouth. Moreover, the illumination settings threshold enables the transfer of a suitable proportion of illumination data without creating distortion, resulting only in natural changes.

(a) (b) (c)

Figure 53: results based on source images: Figure 52, Example 1*, (a) target image with a neutral expression, (b) after the geometrical deformation, (c) the deformed image with a smiling expression and the corresponding wrinkles.*



(a) (b) (a) (b)

Figure 54: more results based on Figure 52, Example 1. *(a) The target image with a neutral expression, (b) the deformed image with a smiling expression and with corresponding wrinkles*

Figures 55 and 56, below, shown in Figure 52, Example 2, as source images the target neutral and deformed expressions, in that the wrinkles in the source image's expression have been transferred to the deformed image by capturing the illumination settings. The problem demonstrated in this example is the wrinkling in the forehead. However the result demonstrates that highly detailed graphics can be achieved, even though the face has been split into areas.

During the illumination distortion elimination process, in order to create a different expression and a better result, the area above the eyes and the eyebrows is avoided. In this specific case, in order to obtain the 'surprise' wrinkles in the final result, the user had to place the dots very precisely so as to avoid distortion caused by the different shaped hair of the source and target images.

(a) (b) (c) (d)

(a) (c) (d)

Figure 55: results based on source images: Figure 52, Example 2 *(a) target image with a neutral expression, (b) automatic edge detection, (c) after the geometrical deformation, (d) the deformed image with a sad expression and the corresponding wrinkles on the forehead.*

(a) (b) (a) (b)

Figure 56: more results based on Figure 52, Example 2*, (a) the target images with neutral expressions, (b) the deformed images with sad expressions and corresponding wrinkles.*

In Figure 57, the target image has been deformed according to the source image's surprised expression (Figure 52, Example 3). An interesting feature in this figure is the raised eyebrow, or surprised expression, and the resultant wrinkles in the forehead. It is very important to note that the source image should be of a similar age to the target image. If the target image is a young person and the source images an old one, the algorithm will transfer an appropriate number of wrinkles onto the younger face, thereby creating unwanted distortion on the final image.

(a) (c) (d)

Figure 57: results based on Figure 52, Example 3, *(a) the target image with a neutral expression, (b) after geometrical deformation (c) the deformed image with a surprised expression showing the corresponding wrinkles*

In Figures 58 and 59, the target images have been deformed according to the disgust expressions of the source images (Figure 52, Example 4). The difficulty with the syntheses is the geometrical deformation of the mouths in that their shapes change in such a way that geometrical distortion is likely. The following examples show precisely the importance of the illumination transfer approach. The second pictures, (c) in Figure 58, are distorted and give completely different results from the third, where they have been changed by the illumination settings into realistic expressions of disgust.

(a) (c) (d)

Figure 58: results based on Figure 52, Example 4, *(a) the target image with a neutral expression, (b) after the geometrical deformation, (c) the deformed image with a sad expression and corresponding wrinkles*

(a) (b) (a) (b)

Figure 59: more results based on Figure 52, Example 4, *(a) the target image with a neutral expression, (b) the deformed image with the disgust expression and the corresponding wrinkles*

8.3 Video animation synthesis results

In this section we will present several groups of synthesized target images for video animation purposes. The process, which has been analyzed in Chapter 7, follows the same procedure as in the 'one picture synthesis' process with differentiation on features detection for faster processing. The reader must focus on the features of each source image and how accurately those individual characteristics have been transferred onto the target images.

As in the previous section, we have divided the resulting expressions into two categories – old and young persons. It must be noted that the synthesis of the old person's expression was more complicated because of the number of wrinkles that had to be transferred onto the target image.

Starting from Figure 60 (a) we have produced a group of six target images with smiling expressions. From the first to the sixth picture the reader can perceive the difference at every step, not only in the geometrical deformation, but also in the lighting conditions that change in each picture – 60 (b).

The difficulty we found in producing these examples was, because the source image's head is held at the opposite angle to that of the target image, extreme accuracy was needed in order to transfer the illumination settings. As a consequence, it is important to stress the importance of the 'source images equalization' process

(Chapter 6.2). All the deformed source images, therefore, have been equalized specifically with the features of the target images in position and shape.

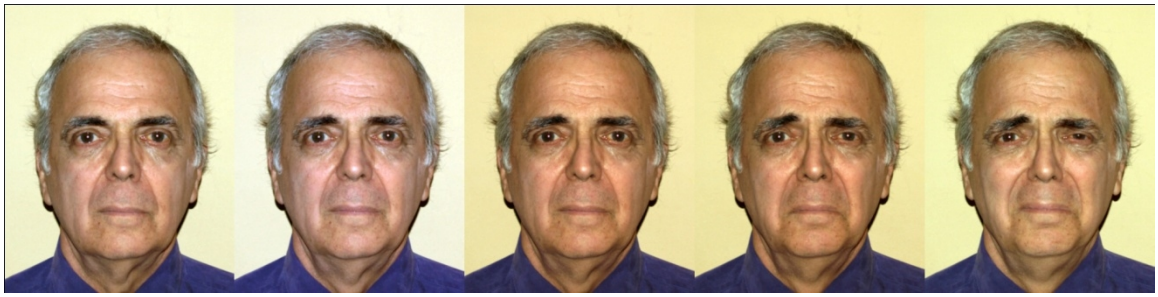
(a)

(b)

Figure 60: Video animation result smile *(a) source images, (b) the geometrically deformed target images with the correspondent facial expression and illumination settings.*

The Figure 61 (below) proved the most difficult of all the synthesized target images because of the geometrical deformation and the illumination settings transfer. In particular, the geometrical deformation of the eyebrows on the final two synthesized images produced a great deal of distortion, because both eyebrows on the corresponding source image had completely different shapes and positions. While these showed realistically on the source images, they proved difficult to reproduce accurately on the target. Therefore, the dots that defined the target image's eyebrows were placed manually, thereby avoiding the distorted parts in the geometrical distortion elimination process (Chapter 5.3).

The second problem involved the realistic illumination data that had to be transferred, since, because the source image is of an older person, it has more wrinkles than the target image. Therefore the 'ratio image threshold' (Chapter 6.4) had to be very accurate in order to transfer the specific illumination data onto the target image. The more illumination transferred data would change the target model age to a much older person and the less transferred data would not produced realistic results.. The last difficulty that should be mentioned is the model's soft smiling expression, which is one more factor emphasizing the importance of controlling the illumination setting transfer threshold.

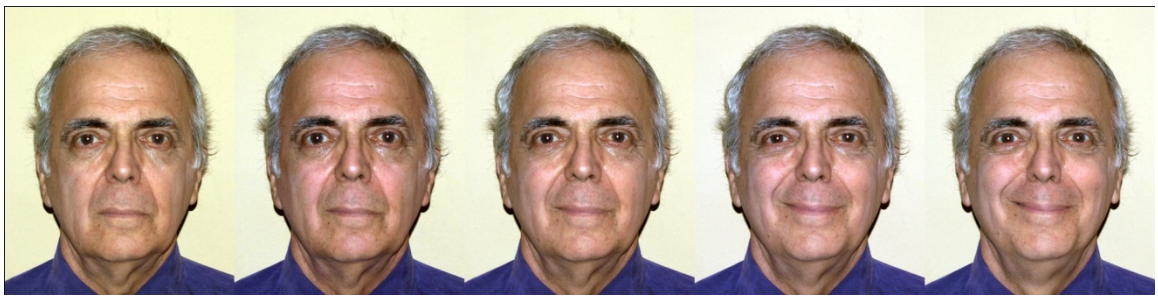


(a)

(b)

Figure 61: Video animation result old sad *(a) source images, (b) the geometrically deformed target images with the corresponding facial expression and illumination settings*

Figure 62 shows a smiling expression synthesis. The target image was deformed accurately, especially on the mouth, and follows the corresponding shapes of the source images. The correct amount of data was transferred in the illumination settings process, therefore all the results are highly graphical and realistic.



(a)

(b)

Figure 62: Video animation result old smile *(a) source images, (b) the geometrically deformed target images with corresponding expression and illumination settings*

Please Note: the target neutral expression images in Figures 10, 13, 30, 32, 33, 34 (second example), 36, 38-44, 46-51, 52 - 54 (second example), 53-60, belong to FEI Face Database [20].

Chapter 9 Discussion – Conclusion

9.1 Introduction

The scope of this chapter is to present the limitations of this research and refer to the future works which can improve the algorithm and enable it to expand its synthesis capabilities.

The limitations which will be presented are mainly issues which could not be resolved in this research since they were out of scope given that the main idea was to create an automatic detection process for the imported image's facial features under specific circumstances and to synthesize realistic expressions for specific purposes.

The "future works" chapter will provide solutions on those limitations which will also improve the final project to a more complete algorithm, capable to synthesize facial expressions even with open mouth characteristics such tongue and teeth and therefore to produce speech video animation.

The last section provides a summary of the presented algorithm along with algorithm's specifications which were established in order to synthesise accurate still pictures' facial expressions and video animations.

9.2 Limitations

9.2.1 Difficulties on synthesizing an open mouth facial expression or speech animation

One of the main limitations of this research is the algorithm's weakness to handle expressions based on an open moth. Examples of those expressions can be the feared, derisive, smiling expressions or even a speech animation.

The reason of that difficulty is the distortion which is produced after the geometrical deformation process. If for instance the target image facial model has a neutral expression and the user selects to synthesize an open mouth expression, the algorithm, according to the geometrical deformation process presented in chapter 5, will only deform the outer shape of the mouth, ignoring details such as tongue or the teeth. That difficulty is based on the fact that the algorithm cannot predict how the

target image's mouth would appear if it was open. Therefore the result will be a distorted stretched mouth.

That limitation prevents the algorithm from producing only video animations where the target image model has open mouth expressions.

9.2.2 Imported image with an opened mouth facial model

The previous limitation poses another one if applied in a reverse manner. If the imported target image has an open mouth the algorithm will not be able to recognize it and to synthesize accordingly a proper facial expression. As it is already mentioned, the algorithm during the geometrical deformation process will deform the mouth by adding to the target image's dots coordinates the difference between the coordinates of the correspondent dots between the source images and thus will deform the outer shape of the feature as well as the included area such as tongue and teeth.

9.2.3 Limitations on the automatic detection processes

Another limitation of the facial expression synthesis process derives from the automatic detection process of the facial features. More specifically, the algorithm cannot define the features of the imported facial image correctly or even totally fails if the Sobel detector does not manage to detect the shapes correctly. In case that the Sobel detector reveal part of the edges which describe the mouth or the eyes – eyebrows shapes the algorithm will not be able to suggest the original shape and will place the dots according to the specified edges.

That limitation in some cases (especially if the user do not interfere to correct the dots position) can produce distortion either on the geometrical deformation or on the illumination settings transfer process.

9.3 Future Work

9.3.1 Transformation of the synthesized 2D image into a 3D head model

Future work could involve the creation of 3D models generated from the synthesized 2D facial expressions. For this purpose, a library of 200 3D heads based on different anthropometric measurements could be used [60, 61] categorized by race, age, gender and sizes and shapes of facial characteristics.

The final deformed image from such an algorithm would contain information about the position and shape of the facial characteristics, as defined by landmarks and

triangles, together with data regarding the desired illumination settings. This would allow the algorithm to search the library by utilizing an efficient algorithm in order to identify the 3D head that most accurately matches the target face; the image could then be adjusted on the 3D model accordingly. The same geometrical deformation that was used on the 2D images would also have to be incorporated onto the 3D model in order for the new expression to be transposed without distortion. The advantage to such a process would be that the user could take images of the face from different angles.

9.3.2 Open mouth expression synthesis – speech animation

In order to synthesize a facial expression with an open mouth or to create a video speech animation, a generic 3D model is needed.

The dots which will describe the facial features' shape and position of the neutral facial picture will geometrically deform the correspondent dots of the generic model. Thus the 2D image will be able to be copied and pasted on top of the generic model. However more dots will be needed to describe the nose's position and shape so as to correctly fit to the 3D generic model nose.

When the user would like to synthesize a speech animation or an opened mouth expression the generic model will take colour samples from the mouth area of the 2D image in order its mouth to have similar characteristics with the target image mouth. Then the mouth from the 2D image will be removed and the animated mouth of the generic model will take the correspondent place.

9.3.3 Improvement of the automatic detection processes

In order to improve the already referred limitation of the automatic detection processes the algorithm must not be completely based on the Sobel's detector results. A 2D generic model must be synthesized as a definition of the facial features shape and positions instead. According to the mouth and eyes – eyebrows corner points of the target image the generic model's correspondent features will be deformed in order to fit to the target image features. If a specific part (of edges) from the facial features' edge map is not detected from the Sobel detector, it could be fulfilled from the generic model correspondent part.

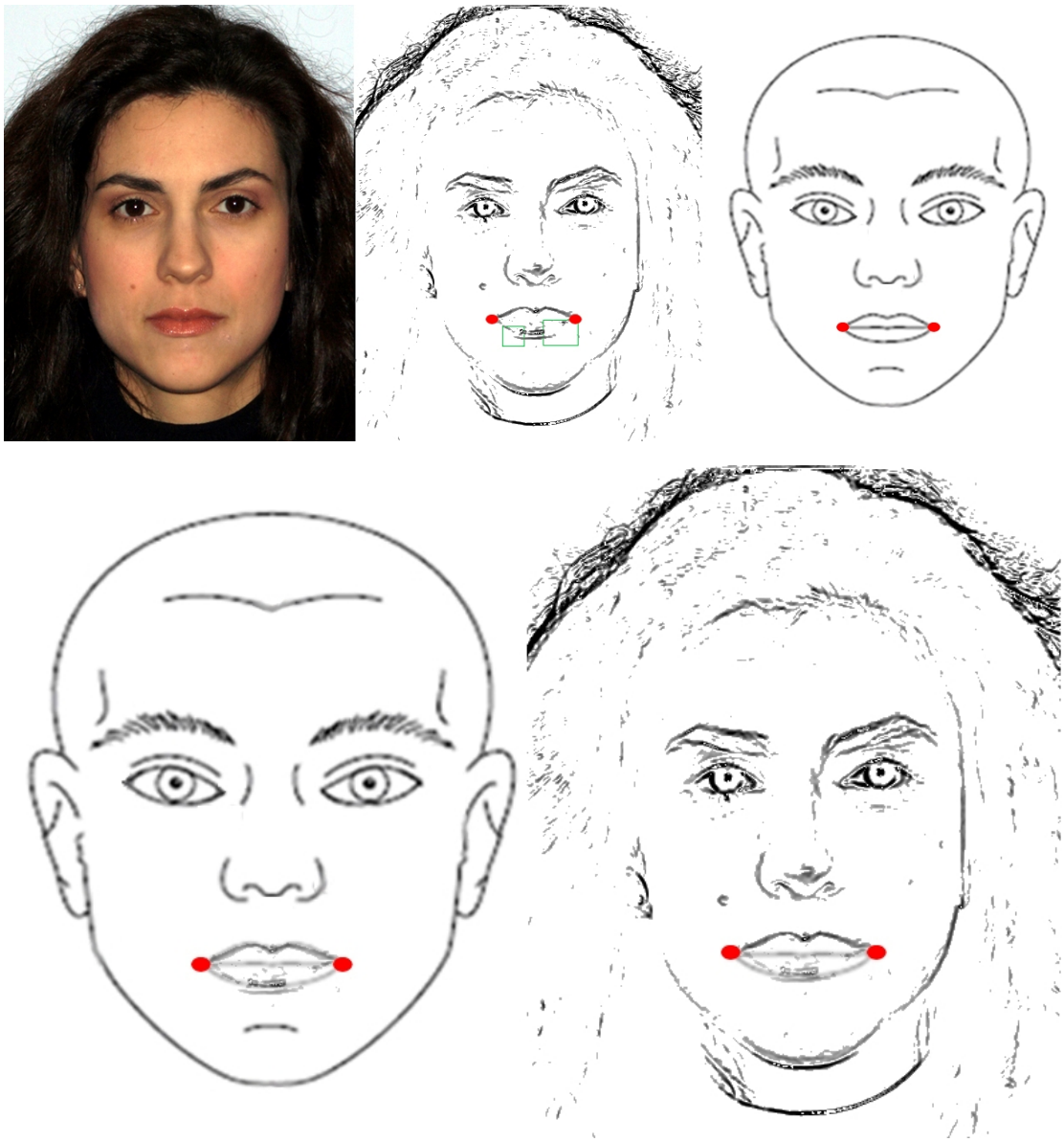


Figure 63: Approximation of the process which can be followed for fixing Sobel detector failures.

9.4 Conclusion

This thesis has focused on providing an accurate algorithm for synthesizing facial expression, simply, accurately and with minimum user interaction. This highly automated method provides for facial feature detection, geometrical deformation, and wrinkle transfer, whilst using different lighting and color settings between the source and the target images.

The process has been divided into four steps:

1. The preparation process, whereby the target image is analyzed and transformed according to its size; i.e. it is rescaled according to the specific

size of the source image and its color settings – this is known as the grayscale format. In this step, two facial areas will be extracted for facial feature automatic detection and for the geometrical deformation process. The process also includes the facial features documentation of the source image, which requires minimum processing time, since several actions, which in the past had to be manually operated, such as in Liu et al's algorithm [31], can be automated.

2. The automatic facial features detection process, which contains three accurate algorithms for detecting the mouth, eyes and eyes-eyebrows, two automatic processes requiring manual intervention, if needed, and one fully manual process for those target images that the algorithm cannot detect. Far fewer dots are required for the features than previous researchers have required, because we have concentrated on the principle features and ignored those that don't affect expression.

3. The geometrical deformation process, which includes the algorithm for morphing the features of the target image and transferring the desired expression from the source images onto the target by applying the dots coordinates difference. We have also provided a code to eliminate, or avoid, all the distorted areas in the geometrically deformed result.

4. The development of an algorithm for synthesizing a realistic facial expression; to this end we have provided an illumination settings transfer from the source images to the deformed target. In order to eliminate any huge differences we have also taken into consideration the color settings normalization between the source images and the target and, also, by utilizing a ratio image threshold. This is a useful algorithmic option whereby users may decide the amount of data they need to transfer in order to obtain a satisfactory end result. We have also made available an elimination code for possible distorted areas after illumination transfer has been completed.

In this thesis we have presented, in highly graphical detail, an accurate account of a process for establishing a one picture synthesis and video animation algorithm that

can be performed on a personal computer. This algorithm has the potential for application by both computer game designers and movie animators for the quick and realistic generation of various facial animations for their characters.

References

1. Abrantes, G.A., Pereira, F.: MPEG-4 facial animation technology: survey, implementation, and results. IEEE Trans. Circuits Syst. Video Technol. **9**(2), 290–305 (1999)

2. Allen, B., Curless, B., Popovi´c, Z.: The space of human body shapes: reconstruction and parameterization from range scans. ACM Trans. Graph. **22**(3), 587–594 (2003)

3. Badler, N., Platt, S.: Animating facial expressions. ACM SIGGRAPH Comput. Graph. **15**(3), 245–252 (1981)

4. Bickel, B., Lang, M., Botsch, M., Otaduy, M. A., Gross M.: Pose-Space Animation and Transfer of Facial Details. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 57-66. ACM, New York (2008)

5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM SIGGRAPH, New York (1999) (URL:http://www.kyb.tuebingen.mpg.de/bu/people/volker/)

6. Blanz, V., Scherbaum, K., Vetter, T., Seidel, H.-P.: Exchanging faces in images. Comput. Graph. Forum **23**(3), 669–676 (2004)

7. Chai, J.-X., Xiao, J., Hodgins, J.: Vision-based control of 3D facial animation. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 193–206. EUROGRAPHICS, San Diego(2003) (URL: http://faculty.cs.tamu.edu/jchai/projects/face-animation/)

8. Cyberware Laboratory, 3D Scanner with Color Digitizer", Inc, Monterey, California. 4020/RGB. (1990).

9. Debevec, P.E.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 189–198. ACM SIGGRAPH, New York (1998)

10. DeCarlo, D., Metaxas, D., Stone, M.: An anthropometric face model using variational techniques. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–74. ACM SIGGRAPH, New York (1998)

11. Du, Y., Lin, X.: Emotional facial expression model building. Pattern Recognit. Lett. **24**, 2923–2934 (2003)

12. Ekman, P., Friesen, W.: Facial Action Coding Algorithm: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto, (1978)

13. Ersotelos, N., Dong, F.: **Building highly realistic facial modeling and animation: a survey. In:** The Visual Computer: International Journal of Computer Graphics archive, pp. 13-30. Springer-Verlag New York (2007)

14. Farkas, L.: Anthropometry of the Head and Face, 2nd edn. Raven Press, New York (1994)

15. Fua, P.: **Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. In:** International Journal of Computer Vision, pp. 153-171. Kluwer Academic Publishers Hingham, MA, USA (2000)

16. Gelder, A.V.: Approximate simulation of elastic membranes by triangulated spring meshes. J. Graph. Tools **3**(2), 21–42 (1998)

17. Golovinskiy, A., Matusik, A., Pfister, H., Rusinkiewicz, S., Funkhouser, T.: A statistical model for synthesis of detailed facial geometry. In ACM Transactions on Graphics (TOG), pp. 1025 – 1034. ACM, New York (2006)

18. Green, B., Edge Detection Tutorial (2002) (http://www.pages.drexel.edu/~weg22/edge.html)

19. Guenter, B., Grimm, C., Wood, D., Malvar, H., Pighin, F.: Making faces. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 55–66. ACM SIGGRAPH, Boston, MA (1998)

20. Hiwada, K., Maki, A., Nakashima, A.: Mimicking video: real-time morphable 3D model fitting. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 132–139. ACM SIGGRAPH, Osaka (2003)

21. Joshi, P., Tien, W.C., Desbrun, M., Pighin, F.: Learning controls for blend shape based realistic facial animation. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 187–192. EUROGRAPHICS, San Diego (2003)

22. Kähler, K., Haber, J., Yamauchi, H., Seidel, H.-P.: Head shop: generating animated head models with anatomical structure. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 55–63. EUROGRAPHICS, San Antonio, TX (2002) (URL http://www.mpi-inf.mpg.de/-kaehler/slides/sca02-headshop_files/v3_document.htm)

23. Kähler, K., Haber, J., Seidel, H.-P.: Reanimating the dead: reconstruction of expressive faces from skull data. ACM Trans. Graph. **22**(3), 554–561 (2003)

24. Kalra, P., Garchery, S., Kshirsagar, S.: Facial deformation models. In: Magnenat-Thalmann, N., Thalmann, D (eds.) Handbook of Virtual Humans, chap. 6. John Wiley & Sons, West Sussex, England (2004)

25. Koch, A.: Structured design implementation – a strategy for implementing regular data paths on FPGAs. In: Proceedings of the 1996 ACM 4th International Symposium on Field Programmable Gate Arrays, pp. 151–157. ACM SIGDA, Monterey, CA (1996)

26. Kshirsagar, S., Egges, A., Garchery, S.: Expressive speech animation and facial communication. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) Handbook of Virtual Humans, chap. 10. John Wiley & Sons, West Sussex, England (2004)

27. Lee, Y., Terzopoulos, D., Waters, K.: Realistic modeling for facial animation. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 55–62. ACM SIGGRAPH, New York (1995)

28. Lee, W., Goto, T., Kshirsagar, S., Molet, T.: Face cloning and face motion capture. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) Handbook of Virtual Humans, chap. 2. John Wiley & Sons, West Sussex, England (2004)

29. **Leyvand, T., Cohen-Or, D., Dror, D., Lischinski, D.: Data-driven enhancement of facial attractiveness.** In: Proceedings of the 2008 ACM SIGGRAPH. Article No. 38. ACM, New York (2008)

30. Litwinowicz, P., Williams, L.: Animating images with drawings. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 409–412. ACM SIGGRAPH, New York (1994)

31. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 271–276. ACM SIGGRAPH, New York (2001)

32. Magnenat-Thalmann, N., Thalmann, D.: Handbook of Virtual Humans. John Wiley & Sons, West Sussex, England (2004)

33. Marschner, S.R., Greenberg, D.P.: Inverse lighting for photography. In: IST/SID 5t Colort Imaging Conference, pp. 262–265. IS&T, Scottsdale (1997) (URL:http://www.graphics.cornell.edu/pubs/1997/MG97.html)

34. Noh, J., Neumann, U.: A Survey of Facial Modeling and Animation Techniques. USC Technical Report 99-705, Integrated Media Algorithms Center, University of Southern California (1998)

35. Parke, F.: Computer generated animation of faces. In: Proceedings of the ACM Annual Conference, pp. 451–457. ACM, Boston, MA (1972)

36. Parke, F.I.: A Parametric Model for Human Faces. PhD Thesis, University of Utah, Salt Lake City, UTEC-CSc-75-047, USA (1974)

37. Parke, F.I., Waters, K.: Computer Facial Animation. AK Peters, Wellesley, MA (1996)

38. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM Trans. Graph. **23**(3), 664–672 (2004)

39. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.H.: Synthesizing realistic facial expressions from photographs. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 75–84. ACM SIGGRAPH, New York (1998)

40. Pighin, F., Szeliski, R., Salesin, D.: Resynthesizing facial animation through 3D model-based tracking. In: Proceedings of the 7th IEEE International Conference on Computer Vision, vol. 1, pp. 143–150. IEEE Computer Society, Los Alamitos, CA, USA (1999)

41. Pratscher, M., Coleman, P., Laszlo, J., Singh, K.: Outside-in anatomy based character rigging. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 329–338. Eurographics, Los Angeles (2005)

42. Seo, H., Magnenat-Thalmann, N.: An automatic modeling of human bodies from sizing parameters. In: Proceedings of the 2003 Symposium on Interactive 3D Graphics, pp. 19–26. ACM SIGGRAPH, Monterey, CA (2003)

43. Seo, H., Cordier, F., Magnenat-Thalmann, N.: Synthesizing animatable body models with parameterized shape modifications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 120–125. Eurographics, San Diego (2003)

44. Shashua, A., Riklin-Raviv, T.: The quotient image: class-based re-rendering and recognition with varying illuminations. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 129–139 (2001)

45. Sifakis, E., Neverov, I., Fedkiw, R.: Automatic determination of facial muscle activations from sparse motion capture marker data. ACM Trans. Graph. **24**(3), 417–425 (2005)

46. Terzopoulos, D., Waters, K.: Physically-based facial modeling, analysis, and animation. Vis. Comput. Animation **1**, 73–80 (1990)

47. Tu, P.-H., Lin, I.-C., Yeh, J.-S., Liang, R.-H., Ouhyoung, M.: Surface detail capturing for realistic facial animation. J. Comput. Sci. Technol. **19**(5), 618–625 (2004)

48. Vlasic, D., Brand, M., Pfister, H., Popovi´c, J.: Face transfer with multilinear models. ACM Trans. Graph. **24**(3), 426–433 (2005)

49. Ward, K., Bertails, F., Kim, T.Y., Marschner, S.R., Cani, M.P., Lin, M.C.: A survey on hair modeling: styling, simulation, and rendering. IEEE Trans. Vis. Comput. Graph. **13**(2), 213–234 (2007)
(URL: http://www.cs.unc.edu/-ardk/research.html)

50. Waters, K.: A muscle model for animating three-dimensional facial expression. Comput. Graph. **22**(4), 17–24 (1987)

51. Wilhelms, J., Gelder, A.V.: Anatomically based modeling. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 173–180. ACM Press/Addison-Wesley Publishing Co., New York (1997)

52. Williams, L.: Performance driven facial animation. Comput. Graph. **24**(4), 235–242 (1990)

53. Yang, C. K., Chiang, W.T.: **An interactive facial expression generation algorithm. In** Multimedia Tools and Applications, pp. 41 – 60. Kluwer Academic, Hingham (2008)

54. Yaun, D.: Drawing: Faces & Features (How to Draw & Paint/Art Instruction Program). Walter Foster. California, (2006)

55. Yin, L., Weiss, K.: Generating 3D views of facial expressions from frontal face video based on topographic analysis. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 360–363. ACM SIGMULTIMEDIA, New York (2004)

56. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-driven photorealistic facial expression synthesis. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation pp. 177–186. Eurographics, San Diego (2003)

57. Zhang, Q., Liu, Z., Guo, B., Terzopoulos, D., Shum, H.: Geometry-driven photorealistic facial expression synthesis. IEEE Trans. Vis. Comput. Graph. **12**(1), 48–60 (2006)

58. Zhang, Y., Sim, T., Tan, C.L.: Rapid modeling of 3D faces for animation using an efficient adaptation algorithm. In: Proceedings of the 2nd International, pp. 173-181. ACM SIGGRAPH, New York (2004)

59. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high resolution capture for modeling and animation. ACM Trans. Graph **23**(3), 548–558 (2004)

60. (URL: http://www.sic.rma.ac.be/~beumier/DB/3d_rma.html)

61. (URL: http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/)

62. (URL: http://www.fei.edu.br/~cet/facedatabase.html)

63. (URL: http://en.wikipedia.org/wiki/Polygon_mesh)

Appendix

Paper 1

A Highly Automated Method for Facial Expression Synthesis

Nikolaos Ersotelos, Feng Dong
Department of Information Systems & Computing, Brunel University, UB8 3PH

Abstract
This paper proposes a highly automatic approach for a realistic facial expression synthesis that allows for enhanced performance in speed and quality, while minimizing user interferences. It will present a highly technical and automated method for facial feature detection, by allowing users to perform their desired facial expression synthesis with very limited labour input. Moreover, it will present a novel approach for normalizing the illumination settings values between the source and the target images, thereby allowing the algorithm to work accurately, even in different lighting conditions. We will present the results obtained from the proposed techniques, together with our conclusions, at the end of the paper.

Keywords: Computer Graphics, Image Processing, Facial Expression Synthesis

1. Introduction
The synthesis of realistic facial expressions has been an unexplored area for computer graphics scientists. In the last three decades several approaches have offered different construction methods in order to obtain natural graphic results. However, despite this progress, existing techniques require costly resources and heavy user intervention and training. Also, outcomes are not yet completely realistic. This paper, therefore, aims to achieve an automated synthesis of realistic facial expressions at low cost.

According to facial painting canons, the difficulty of creating a realistic expression lies in the illumination settings that give fine details, such as creases and wrinkles; but facial expression synthesis, based only on a geometric deformation process, lacks such fine details. By using an Expression Ratio Image (ERI) to capture the difference in the illumination settings during the expression, and then transferring it to a target image, Liu et al, [9] presented an algorithm for capturing and transferring a real facial expression into a geometrically deformed facial expression. However, in order to obtain the required accuracy, significant user training and processing time is required. Also, the algorithm only works by assuming that the illumination change only

happens when the facial expression changes and the source and target images are subject to similar lighting conditions. If one of the above conditions cannot be held, the ERI approach is unable to achieve high quality.

This paper will present a novel technique of highly automated facial expression synthesis that requires very limited user interaction and that has provided accurate results under different lighting conditions along the following lines:

- Firstly, we have established a facial expressions database library from which the user may choose an expression that can be automatically transferred onto the imported target picture. The importance of having such a library is to eliminate the processing time needed to manually, or even automatically, place coordinate dots around the facial characteristics, since each picture in the library will be stored with dots already in place.
- Secondly, instead of defining all the facial features – ears, hair, neck, eyes, eyebrows, nose, mouth, cheeks, etc – the system will extract from the images the eyes, eyebrows and mouth areas and, after the deformation process has taken place, will replace them in their original positions. This has the advantage of reducing the processing time by simplifying the geometrical deformation, because it affects small areas of the image, thereby avoiding possible distortions.
- Thirdly, we will present a highly automated algorithm for the detection and definition of specified facial features that will affect the target image minimally should it be interact with, if it's essential by a user. Our system is able to calculate the size, shape and position of the facial features in relation to those edges, by surrounding them with dots used for the geometrical deformation and colour ratio processes.
- Fifthly, for the algorithm to work accurately, even in different lighting conditions, we provide a novel approach for normalizing the illumination settings values between the source and the target images. A threshold corresponding to ERI is suggested for transferring a specific amount of illumination setting data to the target image.
- Finally, in cases of geometrical, or illumination, distortion, we will offer solutions that will require simple user interaction.

The remainder of this paper is organized as follows: Section Two will consist of a survey of some of the most important approaches in facial modelling and animation. Section Three will be an analysis of the general methodology. Section Four will contain the novel algorithm for an automatic and accurate depiction of a 2D facial picture. Section Five will present the techniques of transferring facial details for expression synthesis, and Section Six will show some experimental results. We will conclude our paper by discussing the advantages and limitations of the technique, and we will offer some proposals for the direction of future research.

2. Previous Work

Early work on computer facial modelling and animation dates back to the 1970s, when the first 3D facial animation was created by Parke [10]. This was followed by Badler et al, [1] who introduced a 3D face based on a mesh of triangles, which, when morphed, could carry new expressions. The disadvantage, however, was that the user had to be very accurate, since large deformations could completely change the shape

of the face. Later, Waters [12] divided the mesh into facial areas, which could be changed individually, thereby producing new expressions without interfering with other parts of the face.

Since the 1980s several other approaches have produced accurate 3D and 2D facial models and expressions based on facial images, anthropometric libraries and even video data. As mentioned above, the basic method used for constructing a 3D head is a triangular mesh containing dots connected by common edges, while another innovative method – a laser scanner called Cyberware [3] used by Lee et al [9] to accurately scan a 3D cylindrical object – produces a 3D model with dynamic skin consisting six layers of triangular meshes, each forming a layer of pseudo muscle, which allows for the synthesis of new facial expressions by pushing, or pulling, or by moving the elastic angles of the triangles. The disadvantage of this process, however, is that the user has to be specially trained because of the complex manner in which the meshes are inter-connected.

DeCarlo et al [4] generated a static geometrical 3D facial surface variational model by synthesising the fundamental elements of anthropometric measurements held in statistical data libraries and ordered according to race, gender, age and specific characteristics.

Yet another approach for modelling a 3D head, also based on anthropometric measurements, has been presented by Kahler et al [7] who defined its features, position and shape by placing several landmark dots capable of being moved by the application of anthropometric data, which they used to calculate and synthesize the growth of the face.

Blanz et al [2] also used manually placed landmarks to describe the facial features of a 2D facial image. This system, which requires a set of 3D models, automatically separates the 2D face from the image – excluding the hair and neck – and fits it to a 3D morphable model. By optimizing all the parameters – such as 3D orientation, position, camera focal length and the direction and intensity of illumination – new facial expressions can be produced, which are then pasted back onto the 2D image.

Zhang et al [13] automatically synthesised a new expression manually by introducing system geometrical points that delineate features on a 2D image by dividing the face into fourteen sub-regions, which are necessary for the synthesis of new expressions. This system infers the feature points of the expression, derived from a subset of the tracked points, by using an example facial expression based approach whereby new expressions are generated by geometrical deformation.

Another approach for creating realistic facial expressions was presented by Sifakis et al, [11] who used a 3D head consisting of thirty thousand surface triangles. This analytical model of a head, which consists of eight-hundred and fifty thousand thresh-holds and thirty-two muscles, is controlled by muscle activations and the degrees of freedom allowed by the kinematic bones. The model is marked with coloured landmarks, each identifying different muscles, which may be activated to generate new expressions, or, even, animations.

Expression mapping [9] is a technique for transferring facial expression from one image to another. Generally this approach is based on geometric deformation and the transfer of illumination settings. The materials necessary are two pictures of the same person – the first, neutral, and the second with an expression – and one picture from another person with a neutral expression. The characteristics, such as eyes, mouth, nose, eyebrows, hair, ears and shape of head, are identified manually by dots and the landmarks around the characteristics are connected by triangles. By calculating the points position difference from the source images, the target image is geometrically

deformed. An image warping process then equalizes the source image according to the deformed target image. The two source images are then divided to give the Expression Ratio Image (ERI), which is the capture process of the illumination changes – creases and wrinkles – between the two images. The ERI is then multiplied pixel by pixel with the target deformed image in order to give the appropriate illumination settings for the new expression.

This method works on the assumption that the illumination change happens when the facial expression changes and the source-target images has similar lighting conditions. The landmarks have to be placed manually on all images simultaneously in order to describe all facial features; therefore, the user has to be very accurate so as to avoid distortion. Approximately three hundred dots are needed for every facial expression and several filters are necessary in order to reduce the hard colour values and the noise from the resulting target picture.

An algorithm synthesising facial expression based on music interaction was presented by Chuan-Kai Yang et al [14]. As in this paper the system requires one facial picture to be imported as a target image; it then necessitates user interaction to identify the facial features of the image. This can synthesize newly animated facial expressions when it is used continuously with a morphing process and the introduction of Midi (*Musical Instrument Digital Interface*) music files.

Tommer Leyvand et al [15] created an algorithm that was intended to enhance the aesthetic appeal of a human faces in frontal photographs. The key component of this beautification engine was the use of datasets of male and female faces selected following a process of attractiveness ratings, collected from human ratters. This semi-automatic process requires a frontal image as an input whereby the user identifies the facial features by using landmarks. Secondly, with the use of a planar graph and landmarks coordinates, the most proper position and shape of the facial features are redefined. Then, by using a warping process, the system deforms the existing facial features as closely as possible to the proposed coordinates obtained from the beautification data engine.

Bernd Bickel et al [16] presented a new methodology for establishing real-time animation of facial expressions based on a multi-scale decomposition of the facial geometry into large scale motion and fine-scale details. This algorithm requires a linear deformation model whereby the facial features are inscribed within a mesh, based on triangles, connected to approximately forty landmarks, which are manually placed on the target model and on the source image.

Another approach designed for the analysis and synthesis of 3D wrinkles and poses was introduced by Golovinskiy et al [17]. Using a 3D scanner and a custom-built face scanning dome, they produced highly detailed 3D face geometry; thereafter they subdivided the surface to separate the effected area from the rest of the facial mesh in order to add highly detailed elements, such as wrinkles and poses. The facial lighting details in this system derive from a database containing a cross-section of facial characteristics categorized by age and gender.

3. The Automatic Facial Expression Synthesis Process

This approach is based on two main elements: “geometrical deformation” and “illumination settings transfer” (Figure 1). In geometrical deformation, the system synthesizes the new expression by warping specific facial features, and, in illumination transfer, it gives realistic lighting settings based on the real lighting settings of the source image.

The main requirement for facial expression synthesis is the importation of a neutral face onto the system. It is important that a complete facial picture is fitted in the frame, in order provide for an accurate definition of the mouth, eyes and eyebrows. After the facial expression has been selected from the library, the geometrical deformation process starts with the target and source images being equalized in size; then the target image is re-scaled back to its original size. When the geometrical deformation process begins, the newly scaled target image is transformed into a greyscale format (each pixel carries only intensity information). This is necessary to enable the later edge-detection process, when the system will only have to decide between the different values of black and white pixels and not between thousand of different colours and lighting conditions. The system then extracts two specific areas from the target image – the mouth and the eyes-eyebrows. The sizes of the rectangles to be used are already defined in the documentation of the source images. After the automatic detection of the specified facial characteristics, dots are placed around them in order to define their size and positions; the coordinates of the dots for the same facial characteristics are then loaded from the source image.

Finally, a geometrical deformation is used to calculate the difference between the dots coordinates of the source image and add that difference to the dots coordinates of the target image. This results in the creation of a new expression, without, however, the application of proper lighting conditions.

At this point the “illumination settings transfer” process starts; by using the geometrical deformation process again, the system will equalize the source images with the new synthesized facial expression of the target image. After the equalization process of the source images with the target image is complete, the division of the source images will extract only the illumination settings, which afterwards will be added by a multiplication process, pixel by pixel, to the target image.

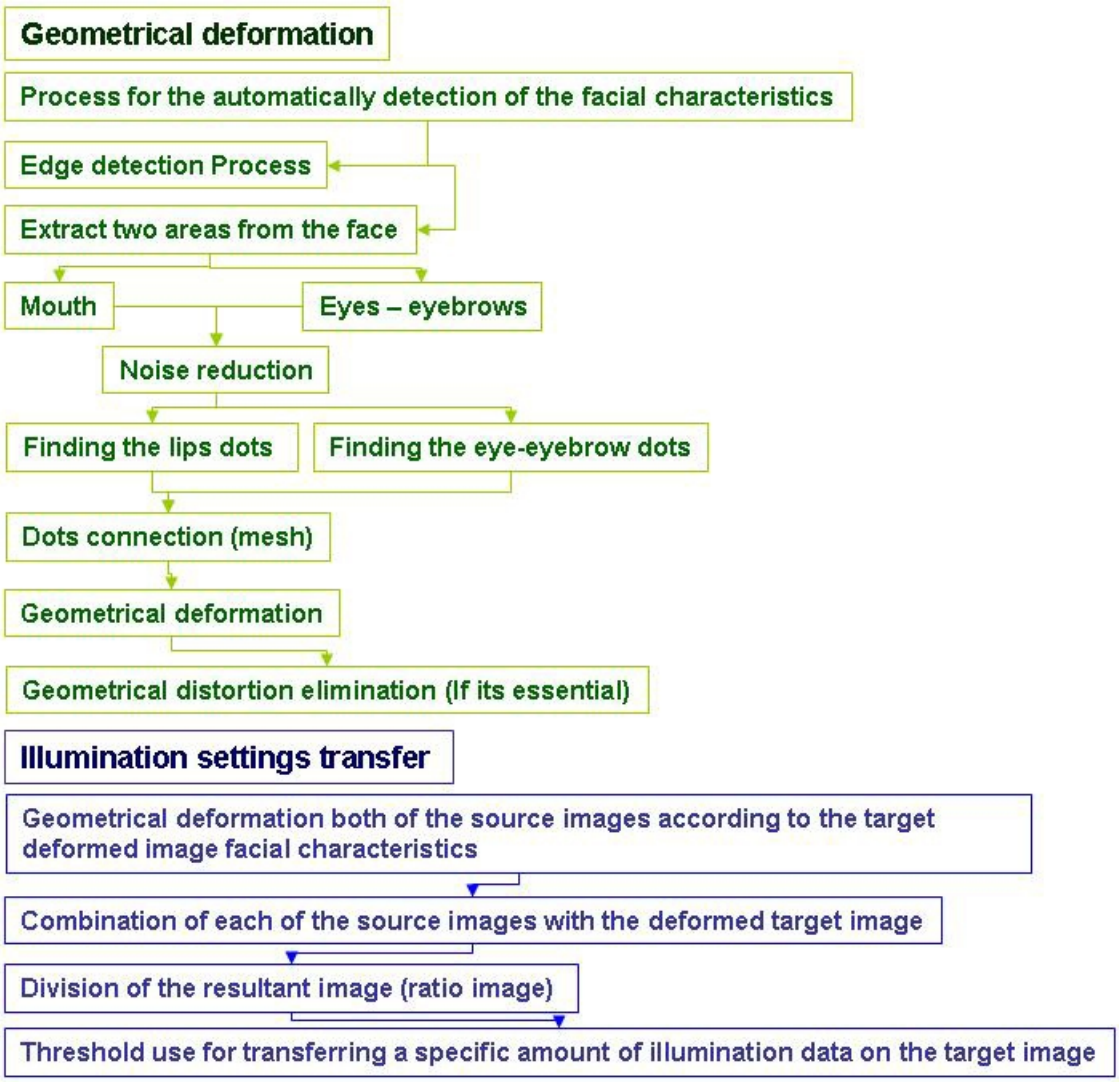


Figure 1: Process Diagram

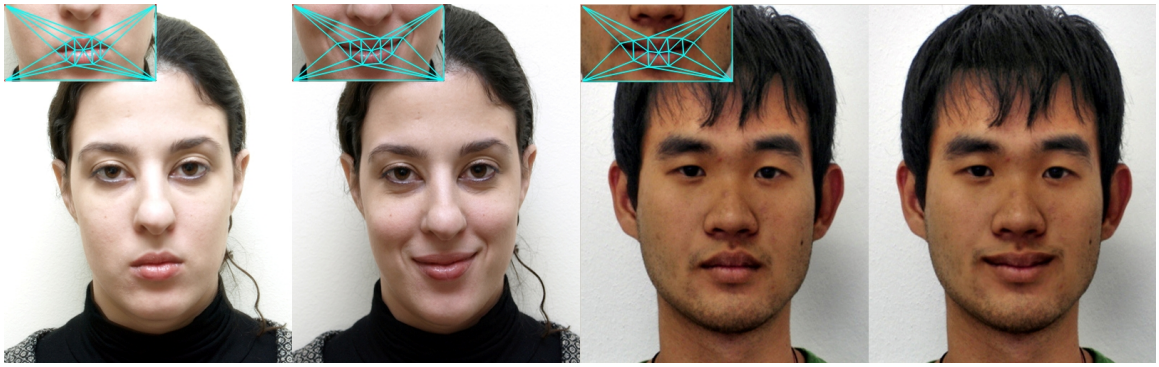
3.1 Splitting the face into areas

A facial expression derives from the activation of one or more facial muscles, which are movements that indicate the emotional state of the individual to observer.

Those muscles have been divided into two areas for the purposes of this paper. The first area contains those elements that must be geometrically deformed in order to produce a new expression. The second area contains those elements – wrinkles and creases – that, under proper lighting conditions, can upgrade a geometrically deformed expression into a realistic one. The only two areas of muscles that require delineating with dots and triangles are the mouth and the eyes-eyebrows areas; it is not necessary to identify any other facial features in this way. This system, therefore, requires fewer dots and triangles, and this also implies the elimination of distortion during the geometrical deformation of the target image. These areas are now extracted from all the images and pasted on layers, which are rectangles covering, in the case of the mouth, the facial area that is vertically defined by the nose and the chin, and horizontally defined by the edge of the face.

The size of the rectangle has been originally defined and stored in the library and it loads when an expression is being selected. The rectangle is big enough to include

any size of the target image's mouth (Figure 2). The same approach applies to the eye and eyebrows layer.



First Step Second Step

Figure 2: First Step: the mouth area has been extracted from the source images and the target image and placed on the rectangle. According to the source images dots positions difference, the target image mouth is geometrically deformed. Second Step: the target image deformed result has been copied and pasted back on top of the original target image.

3.2 The facial expression database

For a faster process, and for user convenience, a database library has been created that contains several pairs of source images with their points position coordinates documented. Each pair corresponds to a specific facial expression and consist one picture with a neutral expression and one with a specific expression. More specifically, three types of data sets have been created for each facial expression, which correspond to all the data point coordinates for the mouth and eyes-eyebrows areas, and also a set that contains both of the above-mentioned areas, together for the wrinkles process. This library also contains sets of seven source images for video animation projects, also together with their documentations. More analytically, it contains groups of seven images that have been taken for animation purposes. The first image has a neutral expression; from the second until the last image the group contains pictures of the same person with gradually changing expressions

The library is created to eliminate the need to place dots on the source images manually, or even to use the automatic detection process of the facial characteristics to determine them.

All six primary expressions to communicate emotions that Ekman et al [5] identified – anger, disgust, fear, happiness, sadness, and surprise – can be found in the library facial expression database, together with an option of more expressions that may be added by the user.

4. Automatic Detection Process

Instead of placing the dots manually around the mouth, eyes and eyebrows in the target image, this paper will present an innovative approach for the detection of the facial characteristics and the automatic placement of dots around them.

This system transforms the colour image to a greyscale image, then, by using an edge detection process, it will identify, by using dots, the eyes, eyebrows, mouth

shapes. Finally, the position and coordinates of the dots will be copied and placed on the original colour target image to activate the geometrical deformation and lighting transfer processes.

4.1 Edge detection

An edge detection process is used in order to eliminate the skin-color details and to reveal the required facial characteristics. The edges in an image contain pixel areas with strong values of contrasts and an edge detection process reduces the amount of data by filtering out useless information, while preserving its important structural properties.

Green [6] has presented an algorithm for an automatic edge detection process. His Sobel Edge Detector uses masks to move over the image, manipulating a square of pixels at a time and, for non-facial pictures, his approach is fast and accurate. We have also considered increasing the sizes of the masks in order to get rid of noise. In the 3x3 mask in example A (Figure 3), the fourth pixel is black, whereas the second one is white, therefore the system will identify the observed pixel as an edge. Consequently we have increased the size of the mask to 5x5 pixels. In example B, the mask size is bigger, so the system will recognize that the black pixel is noise and it will therefore retain the originally observed pixel color value.

The size of the mask, we found, is dependant on the size of the picture. For instance, if the picture in use is 150x150px it would be appropriate for the mask to be 3x3px. If the mask is bigger it avoids important pixels that describe the facial characteristics, and, if it is smaller, it defines the noise pixels as edges; therefore, we found that pictures measuring 500x500px, which required a 5x5 mask size, were the most suitable.

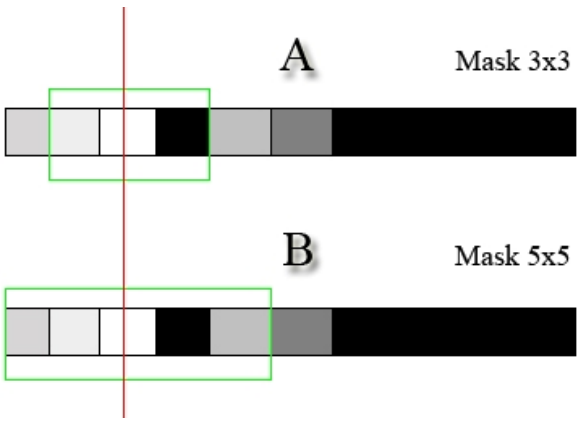


Figure 3: Accuracy between mask A and mask B

4.2 Noise reduction

After the rectangle shape is defined, the process of noise reduction begins. A new mask is used to identify the “almost white” pixels. A colour threshold defines how much none- white colour is allowed to be inside the mask rectangle for an area to be considered as “almost white”. If the area covered by the mask is characterised as “almost white”, then the pixel in the centre of the mask will be changed to 255 (255 is

100% white) (Figure 4). This process is important for removing those pixels that do not identify the shape and position of the facial characteristics and are thus just noise.

More specifically, a square of the same size with the mask rectangle is defined around the observation pixel, so that the observation pixel is an equal distance from each of the square's angles. In order to decide whether the observation pixel is noise, or whether it belongs to the contours of the facial characteristics, the system calculates the whiteness value ratio, which is defined by calculating the average normalized colour value of the mask pixels. Each pixel's normalized colour value is described as MPCol/255.

$$\sum_{MPCol \in M} \frac{MPCol}{255*n}$$

M = mask pixels (their color values)
n=the number of pixels in the mask

Figure 4: The green colored square is the noise reduction mask. The red dot is the observation pixel.

Therefore, if the average normalized colour value of the mask pixels is 1, then all the mask pixels are totally white. If it is 0, then the pixels are totally black. The threshold has been defined as 0.7, therefore the system considers the observation point as part of the facial characteristics when the whiteness color value is less than 0.7.

4.3 Search for the eyes and eyebrows and dot placings

As mentioned in section 3.1, the system uses two areas for the geometrical deformation process. These are the mouth and the eyes-eyebrows areas. The size and position of those areas are already defined in the library expression database. When the user loads an expression, a document will also load the points' position and rectangle sizes of the source images. The rectangles will also be used for the target image, since they are big enough to include the specific facial features of any possible target image.

The system will then copy the area from the target image according to the size of the rectangle in order to separate it from the original image and to start the detection of that area's facial characteristic – such as the mouth – and subsequently the geometrical deformation.

In the case of the eyes and eyebrows the process is more complex, because the system has to detect four different facial characteristics in the same area. For that reason we will divide the rectangle into two sub-areas, where each will contain an eye and an eyebrow. The system will start first with the left area and, after the detection process, will continue with the right. It will then search inside the left half of the rectangle to identify the shape of the eyebrow and, continuously, the shape of the eye. For this purpose two search masks will be used in sequence. Using those masks, the

system will scan the entire rectangle trying to find 50% non-white pixel areas by using a mask – a small square – of 5x5 pixels. The second mask, which will be 30x30 pixels, which searches for 30% of none-white pixel areas, will start from the same position after the first mask's process.

The starting point of the mask will be the top middle pixel of the rectangle and will firstly move vertically down, until it will find the first big non-white pixels area, which will possibly describe the eyebrow. After the detection of the first non-white pixel area, the mask will start moving horizontally until the size and shape of the whole detected area is defined. If the conditions for both masks are not satisfied, then the system will assume that area to be noise – hair, scars, etc – and the masks will move forward until the proper area – the eyebrow – is detected.

After correctly detecting the eyebrow, the system will place five evenly spaced dots on the upper border of the eyebrow. It will first place two dots on the left and right corners of the eyebrow; the left and the right dots being the furthest left and right border coordinates. The next to be identified is the middle dot, which is equidistant from the left and right corner dots; the system will then identify the last two dots by the same process. The first is the middle dot between the left border dot and the middle one, and then, by the same logic, it will identify the right middle dot (Figure 5).

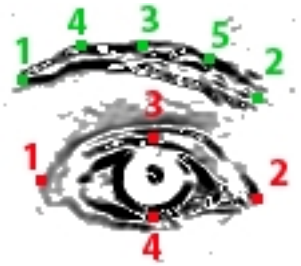


Figure 5: The numbers next to the dots indicate the order in which the system places them. The eyebrow dots are green to contrast with the red eye dots; this helps the user to correct the position of the dots more accurately in case the system fails to identify the facial features correctly.

After the eyebrow has been defined by dots, the system will continue by defining the eye. For that purpose, again, two masks will be used with different starting points and settings from the eyebrow detection, by using a smaller detection percentage for a non white area. More specifically, the starting point of the masks is defined as ten pixels below the left corner dot of the eyebrow. Afterwards the masks will scan horizontally to the right, looking again for the first non-white area. Once more, if the non-white area satisfies the conditions of both masks, it will be accepted as the eye area, and the mask will proceed to identify the lower side of the eye border.

The placement of the eye dots starts in the same way as for the eyebrow, with the first two dots being placed at the corners of the border shape – left and right. The system will continuously calculate the distance between the points, and, depending on the result, will define the middle central pixel. It will then divide the distance – width – between the left 1 and right 2 corner dots by 20% of that distance, in order to calculate how much upper or lower should dots 3 and 5 be placed above the central dot (figure 5). This procedure is faster and more accurate, depending on the eye size, than trying to separate the middle, upper edges of the eye from the edges of the eyebrow.

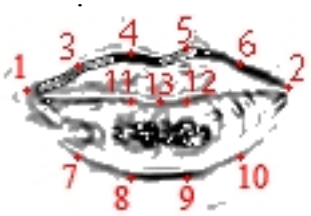


Figure 6: The dot placement order on the mouth

The methodology for the mouth is as for the eyebrow; for the upper lip six evenly spaced dots are selected from the area border pixels, the first being the furthest left mouth corner pixel and the last, the furthest right border pixel.

The lower lip dots define the pixels that are vertically adjacent to the upper lip dots (Figure 6). The order is the same as for the upper dots reading from left to right. Three dots are placed in the middle line of the mouth by dividing the middle upper and lower points in order to produce accurate geometrically deformation results and to eliminate distortion.

4.6. Geometrical distortion elimination

If geometrical distortion of the target image occurs, it generally affects the face shape; for instance, in the case of mouth deformation, the chin. In order to avoid this, the user can copy the correctly deformed area and place it on top of the neutral expression target image. More specifically, if the person in the source image has a large mouth, then the rectangle for that area (chapter 3.1) must be big enough to cover it. A larger rectangle may also cover the chin, or parts of the facial limits, of the target image. Unfortunately, this might produce a distortion on the target image following the geometrical deformation process. To eliminate this particular distortion, the user may place as many dots as are preferred on the target image, avoiding the distorted parts, in order to specify the correctly deformed area. Afterwards, a new window will appear showing the target image without any deformation. The copied area will then be placed on top of the corresponding place on the new window; this does not necessarily have to be a rectangle – since the user defines its shape by placing the dots.

5. Wrinkles Transfer Approach

In order to calculate the wrinkles of a facial expression and transfer them onto the target image, Liu et al [9] presented a method that has been analyzed and improved on in this paper in order to obtain better results under numerous illumination settings. In Liu's algorithm, the system aligns source images with the target image through a warping process. It then divides the warped source images in order to calculate the ratio image and multiplies the result with the target deformed image to transfer all the illumination setting from the source images to the target geometrically deformed image.

The disadvantages of such a process are the amount of wrinkles data, or the quality of those illumination settings that will be transferred on to the target image. If the source images are of poor quality, they will directly inflect the target image, produce

lighting distortions, create hard colours on the wrinkles areas, or simply look artificial.

5.1 Source images equalization with the target image

To eliminate the above disadvantages, Liu's process has been changed and split into four steps. The first step is the equalization of the source images with the target image. Both of the source images must be changed according to the new facial expression of the target image through a geometrical deformation process in order for both of the images' facial characteristics to match perfectly with the facial characteristics of the target image. Where there is a difference, a distortion appears on the lighting settings in the final result. As stated above, it is not important to deform facial characteristics that are not involved in the facial expression. Therefore the hair, neck, ears or shoulders are avoided in the geometrical deformation process. However, even if the rectangle is bigger than the face of the target image, the opposite solution will be used to eliminate any distortion.

The size and position of the rectangle is stored with the source images. The area it covers are the mouth, eyes and eyebrows, as in Figure 9. The dots that define the facial characteristics of the source images are stored and they are loaded automatically when the process begins. The dots that describe the facial characteristics on the target image are copied from the previous process (section 3) and are pasted in the new rectangle. Afterwards, the geometrical process begins and is repeated automatically several times until the dot positions on the source images are aligned with the dot positions on the target image.

5.2 Colour normalization

According to Liu's process [9] it is important that the source images are of a similar quality to the target image. Otherwise the differences will cause distortion on the final result. If, for instance, the faces of the source and target images have different skin colours, then that difference will also be transferred through the illumination settings. Moreover, if the source images are black and white, the distortion will be greater. All the lighting settings will be mixed with a grey shadow because the illumination settings are part of the facial skin. It must be assumed, therefore, that the picture quality, or the colour of the facial skin, matters.

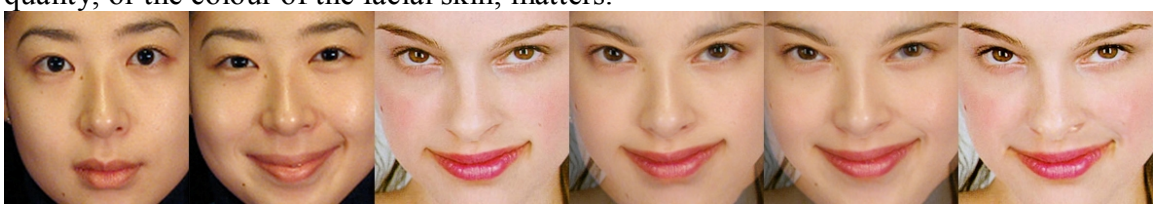


Figure 7: Skin colours and illumination setting are combined before the final result is achieved (The source images *A, A'* are from the Liu et al [9]).

For that purpose, instead of dividing the source images, one by the other, each of them is first combined, pixel by pixel, with the target image, not only to keep the wrinkle values of the source images, but also to combine, or normalize, them with the illumination and skin colour settings of the target image. The results are two new source images – *ABd* and *A'Bd,* where *Bd* is the target deformed image (Figure 7). However in order not to mislay some percentage of the source images illumination settings, the first pair, *ABd* uses 50% of its images and the second pair that has the

desired expression uses 70% from the source image and 30% from the target image – *A'Bd*. This different percentage will collect all the source image illumination details together with the skin colour settings of the target image.

5.3 Ratio Image threshold

Following the synthesis of the two new images, the system divides them one with the other (*ABd*/*A'Bd*) in order to extract their illumination settings, which are described as a 'ratio image'. A threshold has been invented in order for the user to transfer a specific amount of data from the ratio image for a realistic final result on the target image. More analytically, with the use of that threshold, the user has the option to transfer a specific percentage of data. The ratio image is like a transparent file that is added on the target image through a multiplication process – *Ratio Image x Bd = Bd'*. This threshold helps avoid details that produce lighting distortion on the target image, such as beauty spots, scars, etc derived from the source images.

5.4 Distortion elimination on the illumination transfer approach

If the rectangle that contains the facial features for the 'illumination transfer approach' is bigger than the area of the eyebrows, eyes, nose mouth area, and it contains hair, or ears, or scars, this will produce a distortion because all the head characteristics of the source images have not been equalized with the correspondent characteristics of the target image; therefore, it will transfer the distorted illumination settings to the final result. To correct this, simple user interference is provided.

Figure 8: a) B is the original dot placed by the user. B' is the internal dot of the circle that describes the shortest distance between A and C. **b)** The blue color diagram describes the orthogonal vectors that are used to gradually change the color of the pixels area.

The user can identify the non-distorted area by simply placing any number of dots around the area that contains the proper lighting settings. Afterwards the system will automatically extract that area and paste it back to the original target image, thereby avoiding the distorted parts.

When the user places the dots on the face, a small circle of 20 pixels diameter is drawn with the specified dot at its center. According to the neighbors – dots A and C – the system will calculate a new dot, B', which describes the shortest path from A to C via the circle. B' will be defined as the inner dot (Figure 8 a). This process will be repeated for all the dots placed by the user. The purpose of the internal dots is to achieve a gradual change of color, which takes place in an (approximately) 10 pixels area between the outer and the inner dots, from the color of the original target image to the color of the new synthesized area. Therefore, two borders, ten pixels apart, are created around the extracted facial area. The outer border consists of the line achieved

by connecting the original dots, placed by the user, and the inner border, by connecting the new internal dots created in the above-mentioned procedure.

Orthogonal vectors are used to move from one pixel to another, inwardly, from the outer border lines, in order to gradually change the pixels in the area (Figure 8 b).

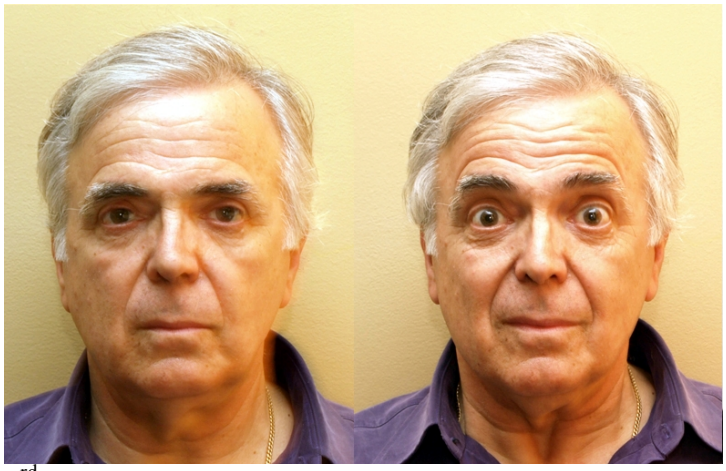
6. Results

The new approach has been applied to detect automatically the facial characteristics of the target image, separate them into two areas, deform them and then transfer the appropriate illumination data from the source images to the final image.

The results will be presented in this section. The source images have been chosen because their facial expressions and illumination settings vary; these are presented in Figure 9. The images are grouped in pairs of neutral and non-neutral facial expressions.

1st Example (Neutral and Smiling expression source images)

2nd Example (Neutral and Sad expression source images)



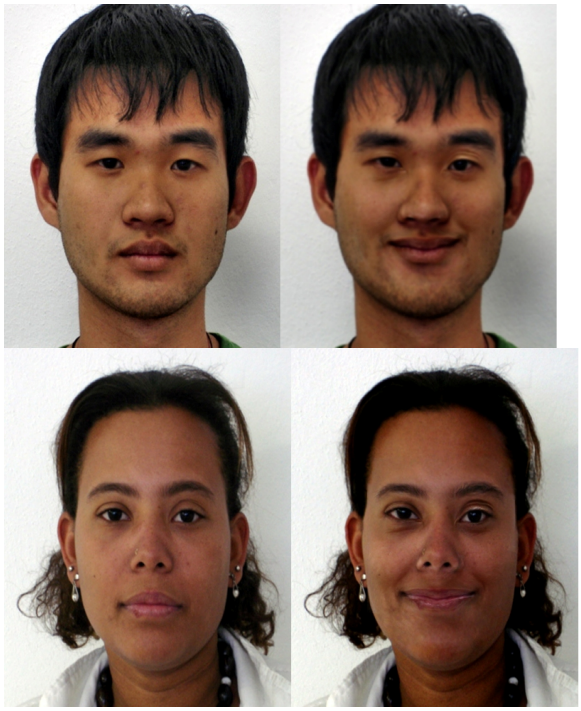
3rd Example (Neutral and Surprised expression source images)

Figure 9: Examples of source images from the library database that are used in order to synthesize new facial expressions.

Figures 10 and 11 show the target images and the resultant images after the deformation process and the illumination setting transfer, having used as source images the pair at Figure 9, 1st example. It can be observed that the wrinkles around the mouth contribute to a realistic result, providing a good level of physical detail. Deformation has been applied only in the areas of the mouth and eyes. As can been seen, the logic of splitting a facial image into areas does not deteriorate the naturalness of the facial expression. Even though no deformation was applied to the area of the nose by the user, this is deformed according to the geometrical deformation of the mouth. Moreover, the illumination settings threshold enables the transfer of a suitable proportion of illumination data that does not produce any distortion, but only physical results.

(a) (b) (c)

Figure 10: Based on source images – Figure 9, 1st example, (a) target image with a neutral expression, (b) after the geometrical deformation (c) the deformed image with a smiling expression and the corresponding wrinkles.



(a) (b) (a) (b)

Figure 11: more results based on Figure 9, 1st example, (a) the target image with a neutral expression, (b) the deformed image with the smiling expression and the corresponding wrinkles.

Figures 12 and 13: show the target neutral and the deformed expression of the process using as source images the pair in Figure 9, 2nd example. The wrinkles of the facial expression of the source image have been transferred to the deformed image, by capturing the illumination settings. The difficulty in this example is the resultant wrinkles in the forehead. However the result illustrates that highly detailed graphics can be achieved, even though the face has been split into areas.

(a) (b) (c) (d)

Figure 12, (a) target image with a neutral expression, (b) automatic edge detection, (c) after the geometrical deformation (d) the deformed image with a sad expression and the corresponding wrinkles between the eyes and the mouth.

Figure 13: more results based on Figure 9, 1st example: (a) the target image with a neutral expression, (b) after the geometrical deformation (c) the deformed image with a sad expression and the corresponding wrinkles.

In Figure 14, the target image has been deformed according to the source image's surprised expression (Figure 9 3rd example). An interesting feature in this figure is the raising eyebrow (surprised) expression and the resultant wrinkles in the forehead. It is very important to note that the source image's model must have a similar age to the target image model. If the target image describes a young person, and the source image an old one, then the algorithm will transfer amount of wrinkles that describe the older age facial characteristics, thereby creating an unwanted distortion on the final target image result.

Figure 14: results based on Figure 9, 3rd example: (a) the target image with a neutral expression, (b) after the geometrical deformation (c) the deformed image with a surprised expression and the corresponding wrinkles.

According to Liu et al [9] approach, the executable time for a facial expression synthesis is approximately thirty minutes. More analytically, the user has to manually place dots that will cover all the facial features on the source and target images. Moreover, the user must be very accurate. If a dot is placed incorrectly, then distortion will certainly occur during the geometrical deformation. According to our approach, the process is primarily automatic. Only in the case of geometrical or illumination distortion does the user interfere by a limited process of interaction.

More specifically, the geometrical deformation is based on two specific facial areas. Therefore there is no need to place dots around all of the facial features, such as the hair, ears, nose, etc. Moreover, by using an expression library database, the system can automatically load all the dot coordinates of the source images.

The number of dots required is therefore reduced from three hundred (Liu's approach) to sixty-nine and the 'executable' time required has been reduced from thirty minutes to less than two minutes – depending on the power system.

The target neutral expression images in Figures 10, 11, 13, 14, 15 belong to FEI Face Database [20].

6.1 Video editing approach

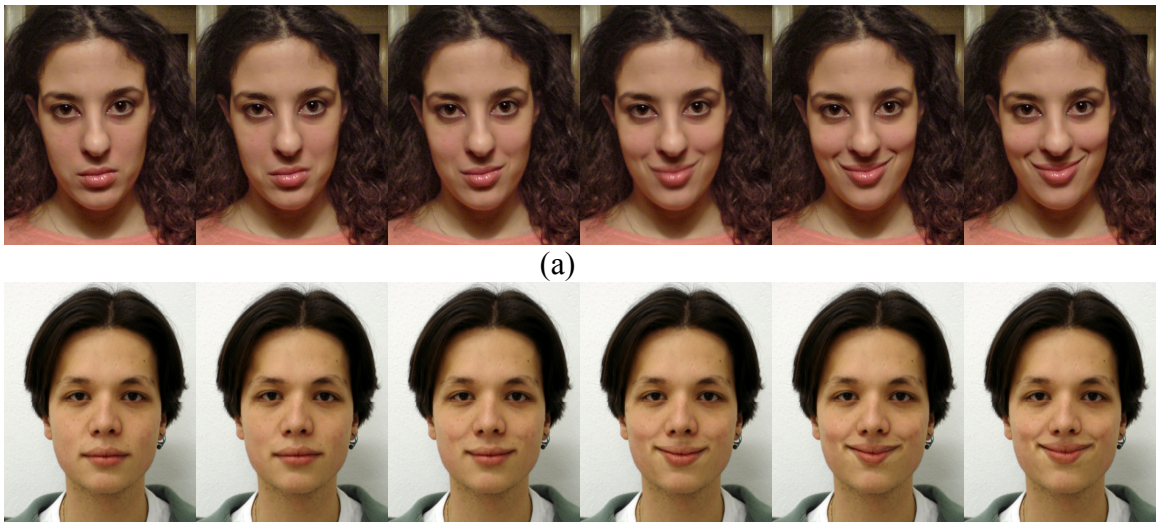


Figure 15: (a) source images, (b) the geometrically deformed target images with the correspondent facial expression and the illumination settings.

Another function that the system can provide is the synthesis of six facial expressions simultaneously for video editing animation purposes. As with the individual facial expression synthesis process, the system contains a set of source animated pictures with the relevant documentations stored in the library. When the process is selected from the user the automatic edge detection of the target image's facial features begins. Where the user interferes by manually correcting the detected dots, the system stores the dots' position in a temporary folder in order for the dots to be re-used in the same positions in order to obtain the remainder of the images. All the synthesized target images are being produced, taking as inputs the neutral expression of both the source and the target images along with the subsequent source image expression (Figure 15).

After having synthesized the sequence of the target facial animation images, a video editing software package can be used to render the images in a video format. By using a "fade in, fade out" effect, very realistic facial video animation can be achieved.

8. Conclusion and Future Work

This paper presents a novel technique to produce natural looking facial expressions in a simple, accurate and highly automated way. It features a highly automated method for facial feature detection, as well as a wrinkle transfer method that allows for the

synthetic wrinkle under different illumination settings between the target and source images. The process has been accelerated by separating parts of the face and extracting two facial areas for the expression synthesis. Moreover, a "facial expressions" library is integrated into the process. Other improvements are the significant reduction in the numbers of dots required to build the whole process and a minimization of necessary interference from the user by deploying an edge detection process for identifying the facial features.

Future work could include the creation of a 3D model that will be generated from the synthesized 2D facial expression. For this purpose, a library of 200 3D heads [18, 19] based on different anthropometric measurements could be used, the heads being categorized by race, age, gender and the sizes of their facial characteristics.

The final deformed image from such a system would contain information about the position and shape of facial characteristics, defined by landmarks and triangles, together with data about desired illumination settings. This would allow the system to search the library by utilizing an efficient algorithm in order to identify the 3D head which most accurately matched the target face; the image could then be adjusted accordingly on the 3D model. The same geometrical deformation that had been used on the 2D images would also have to be incorporated on the 3D model in order for the new expression to be fitted on the 3D head without distortion. The advantage to such a process would be that it would enable the user to take images of the face from different angles.

The results presented in this paper have been accurately created in highly graphical details by using a personal computer. Potentially, applications could be used by computer game designers, or movie animators, in order to allow them to generate various facial animations quickly for their characters.

8. References

1. Badler, N., Platt, S.: Animating facial expressions. ACM SIGGRAPH Comput. Graph. 15(3), 245–252 (1981)

2. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194.

3. Cyberware Laboratory, 3D Scanner with Color Digitizer", Inc, Monterey, California. *4020/RGB*. 1990.

4. DeCarlo, D., Metaxas, D., Stone, M.: An anthropometric face model using variational techniques. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–74. ACM SIGGRAPH, New York (1998)

5. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto, 1978.

6. Green, B., Edge Detection Tutorial
(URL http://www.pages.drexel.edu/~weg22/edge.html), 2002

7. Kähler, K., Haber, J., Yamauchi, H., Seidel, H.-P.: Head shop: generating animated head models with anatomical structure. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 55–63.EUROGRAPHICS, San Antonio, TX (2002) (URL http://www.mpi-inf.mpg.de/~kaehler/slides/sca02-headshop_files/v3_document.htm)

8. Lee, Y., Terzopoulos, D., Waters, K.: Realistic modeling for facial animation. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 55–62. ACM SIGGRAPH, New York (1995)

9. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 271–276. ACM SIGGRAPH, New York (2001)

10. Parke, F.: Computer generated animation of faces. In: Proceedings of the ACM Annual Conference, pp. 451–457. ACM, Boston, MA (1972)

11. Sifakis, E., Neverov, I., Fedkiw, R.: Automatic determination of facial muscle activations from sparse motion capture marker data. ACM Trans. Graph. 24(3), 417–425 (2005)

12. Waters, K.: A muscle model for animating three-dimensional facial expression. Comput. Graph. 22(4), 17–24 (1987)

13. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-driven photorealistic facial expression synthesis. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 177–186. Eurographics, San Diego (2003)

14. Yang, C. K., Chiang, W.T.: An interactive facial expression generation system. In Multimedia Tools and Applications, pp. 41 – 60. Kluwer Academic, Hingham (2008)

15. Leyvand, T., Cohen-Or, D., Dror, D., Lischinski, D.: Data-driven enhancement of facial attractiveness. In: Proceedings of the 2008 ACM SIGGRAPH. Article No. 38. ACM, New York (2008)

16. Bickel, B., Lang, M., Botsch, M., Otaduy, M. A., Gross M.: Pose-Space Animation and Transfer of Facial Details. In: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 57-66. ACM, New York (2008)

17. Golovinskiy, A., Matusik, A., Pfister, H., Rusinkiewicz, S., Funkhouser, T.: A statistical model for synthesis of detailed facial geometry. In ACM Transactions on Graphics (TOG), pp. 1025 – 1034. ACM, New York (2006)

18. (URL http://www.sic.rma.ac.be/~beumier/DB/3d_rma.html)

19. (URL http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb)

20. (URL http://www.fei.edu.br/~cet/facedatabase.html)

Paper 2

Visual Comput (2007)
DOI 10.1007/s00371-007-0175-y

ORIGINAL ARTICLE

Nikolaos Ersotelos
Feng Dong

Building highly realistic facial modeling and animation: a survey

N. Ersotelos (✉) · F. Dong
Department of Information Systems and Computing, Brunel University, Uxbridge, UB8 3PH, UK
Nikolaos.Ersotelos@brunel.ac.uk

Abstract This paper provides a comprehensive survey on the techniques for human facial modeling and animation. The survey is carried out from two different perspectives: facial modeling, which concerns how to produce 3D face models, and facial animation, which regards how to synthesize dynamic facial expressions. To generate an individual face model, we can either perform individualization of a generic model or combine face models from an existing face collection. With respect to facial animation, we have further categorized the techniques into simulation-based, performance-driven and shape blend-based approaches. The strength and weakness of these techniques within each category are discussed, alongside with the applications of these techniques to various exploitations. In addition, a brief historical review of the technique evolution is provided. Limitations and future trend are discussed. Conclusions are drawn at the end of the paper.

Keywords Facial modeling · Facial expression and animation · Generic model adaptation · Morphable modeling · Pseudo muscles · Performance-driven animation · Blend shapes

1 Introduction

Facial modeling and animation have been a research challenge and focus for many years. They play the most substantial role in depicting human characters. The recent advance in facial animation that allows us to produce a rich set of stunning effects on synthetic humans has already brought profound impact on the industry. Meanwhile, within the computer graphics community, new efforts are still emerging and research interests in synthesizing high quality facial animation show no sign of abating.

Examples of early work in facial modeling and animation include [27] and [3]. These works have generated very simple and artificial looking face models and expressions (e.g., models with connected vertical and horizontal lines in [3]). Moreover, the control of facial animation in these works involved a complicated parameterization process and hence appeared to be difficult for untrained users.

However, since the appearance of these pioneer works, significant progress has been materialized by the researchers from the computer graphics community, which have developed a large number of techniques to generate high quality face models and highly realistic facial expressions. However, despite this progress, the existing computer synthesized human facial animation still requires costly resources and sometimes involves considerable manual labors. Furthermore, the outcomes are not yet completely realistic. Therefore, at the current stage, solutions are cost effective, but fully realistic facial animation is still not entirely available.

This survey aims at providing a comprehensive survey for the existing techniques in the area of facial modeling and animation, giving analysis to the strength and weakness for a wide range of techniques. Here we pay special attention to more recent techniques, which allow the production of highly realistic results, as compared to other surveys given in the past [24,26]. In particular, the ana-

lysis on the latest techniques allows us to look into their suitability to different applications and foresee the future research trend in this area. In addition, the survey also provides a historical view on the evolution of these techniques.

The survey is carried out from two perspectives:

- Face modeling, where we introduce the techniques for producing high quality face models up to the latest.
- Facial animation, where we concentrate on the techniques that allow facial animations with high realism.

Facial modeling and facial animation are two strongly interrelated issues. In fact, generating realistic facial animation often involves modeling techniques, for example, to build multiple face layers such as in [20], or to carry out deformation on the face models for desired facial expressions. Therefore, the quality of facial animation is determined by both the employed methods of facial modeling and facial animation. Such a relationship is demonstrated further in our discussions in the rest of the paper.

Figure 1 outlines the structure of the techniques that are covered by this survey. The details of these techniques are given in the following sections.

This paper is structured as follows: Sect. 2 provides a historical view on the technical evolution; Sect. 3 describes two main approaches for face modeling, generic model individualization (Sect. 3.1) and example-based face modeling (Sect. 3.2). Section 4 introduces three main approaches for facial animation, including simulation-based approach (Sect. 4.1), performance-driven animation (Sect. 4.2) and blend shape-based approach (Sect. 4.3). Section 5 provides analysis to their strength and weakness from an application perspective; Sect. 6 gives the limitation of the current techniques and the future research trend. Finally, the conclusion is given in Sect. 7.

2 Brief history

Great interest has been received in computer simulation of human faces and their movements during last few decades. An early example of success was the facial action coding system (FACS), which was introduced by Ekman and Friesen in 1978 to describe primitive facial activities. Early work on computer facial modeling and animation dated back to 1970s, during which the first 3D facial animation was created by Parke [27]. This was followed by a few landmark works in 1980s, which included the deformable face model from [3], the classic work on facial animation using pseudo muscles from [42].

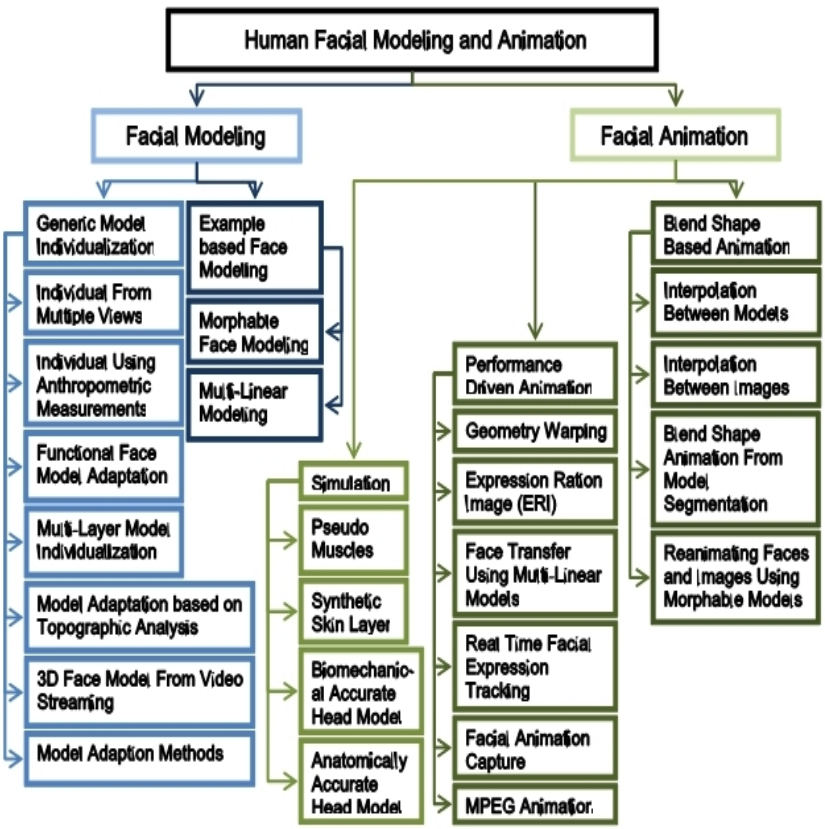


Fig. 1. Categorization of facial modeling and animation methods

Subsequently, a much larger number of significant works were published during the 1990s. Apart from those inherited the pseudo-muscle-based approach, such as in [38], many researchers strived to target high quality face models by adopting natural face measures for the modeling, such as anthropometrics, [8]. Meanwhile, people also started to use scanned 3D models from scanners, which created face models with a great deal of detail and hence allowed them to largely overcame "cartoon styled" artifacts, [4]. To produce facial animation on the scanned models, pseudo muscles needed to be located, [20]. In addition, aiming at a natural looking facial animation, cameras started to be employed to capture facial movements, such as in [12,31], which constituted the early stage of performance-driven facial animation.

More recently, with the rapid advance of hardware, the performance-driven approach has played a more significant role. For example, high quality facial expression with fine details was created from a set of existing photos using different techniques, as in [5,14,23,46,47]. Meanwhile, video analysis on human faces was applied to capture facial movements and hence improve the realism of synthesized facial animation, such as in [6,40].

On the other side, with the aim of improving the comprehension on the mechanism of facial movements, anatomy-based high quality face models have been built in recent years. This has included the multi-layer model from [15], which simulated a head model with skin, muscle, skeleton, etc, and the volumetric model of a human head from [37], which covered a range of head structures including muscles, tissues etc.

However, despite the recent technical advance within this area, the currently available techniques are still not able to meet the requirement from many envisaged applications. To compromise with limited computation resources, considerable amount of simplification has to be made in anatomy-based face models. Also, due to the limitation from the current image and video analysis, the performance-driven approach has to involve a large number of equipments and the accuracy is still subjective to further enhancement. Therefore, facial modeling and animation is still an on-going research issue, and there is a long way before the satisfactory completion of the technology.

3 Face modeling

The aim of face modeling is to generate realistic face models with high visual fidelity. The basic way to represent the shape of a face is to use a triangle mesh. A more complex model involves multiple layers which mimic the anatomical structure of a face.

With the advance of data capturing hardware, many 3D face models have been created via using 3D laser scanner. On the other side, over the last few years, researchers have proposed and developed various techniques on producing quality face models. These techniques can be divided into two categories:

- Generic model individualization, which is based on the idea of creating a face model for a specific subject by carrying out feature-based deformation to a generic model – more details are given in Sect. 3.1.
- Example-based face modeling. This is to create a face model with desired facial features through the linear combinations of an existing face model collection – more details are given in Sect. 3.2.

3.1 Generic model individualization

Generic model individualization generates a facial model for a specific individual through the deformation of a generic model. This is also named as model adaptation. Given the positions of some selected facial features from an individual face, such as eyes corners, mouth and nose positions, the adaptation generates the model for the individual by aligning the corresponding facial features of the generic model towards these given feature positions.

As the basic inputs of the generic model individualization approach, the features the individual face can be positioned in different ways. For example:

1) In [31], the feature positions are given manually through a number of multi-viewed photographs. For more details, see Sect. 3.1.1.
2) In [8] anthropometric measurements are used to describe faces with particular features. For more details, see Sect. 3.1.2, 3.1.3 and 3.1.4.
3) In [45] the individual face is described by a frontal view image, and image analysis is used to detect the face features. For more details, see Sect. 3.1.5.
4) Face features are given by data tracking of video streams using multiple cameras in [49] For more details, see Sect. 3.1.6.

Notably, all the model individualization methods involve deforming a generic model to an individual model. The techniques involved in this process decide the efficiency and effectiveness of the individualization. We will discuss these techniques in Sect. 3.1.7.

3.1.1 Individualization from multiple views

Generating face model constitutes an important step in the work of Pighin et al. [31] which presents a method of generating facial expressions from photographs. To create a model for an individual subject, the method adapts a generic face model to the subject, which is portrayed by a number of photographs (images) taken from different view angles. To assist this adaptation process, features points such as eye corners, nose tip, mouth corners are manually identified in these images – see Fig. 2. A scattered data interpolation technique is used to deform the

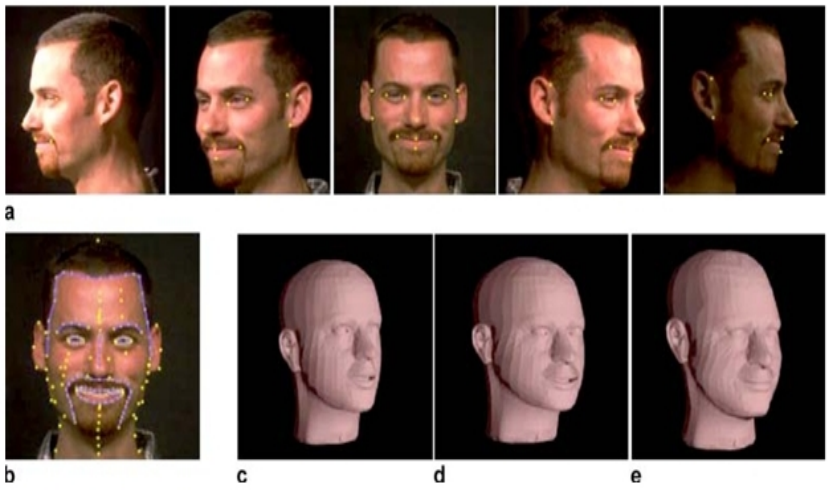


Fig. 2a–e. Model-fitting process: **a** a set of input images with marked feature points, **b** facial features annotated using a set of curves, **c** generic face geometry (shaded surface rendering), **d** face adapted to initial 13 feature points (after pose estimation) **e** face after 99 additional correspondences have been given [31]

generic model to fit into the feature points identified from the images.

Furthermore, the work of [31] goes beyond face modeling towards facial animation. To do this, the above modeling process has to be repeated for several different facial expressions of the same subject. This creates a facial model for each facial expression. Then a 3D shape morphing technique is applied to these facial models to generate transactions in between these models to create facial animations. This issue will be further discussed Sect. 3.3.

However, this proposed method of face modeling involves considerably large amount of manual work. For example, a large number of facial feature points have to be given manually in the multiple viewed images. This is extremely inconvenient and time consuming. Therefore, this algorithm only can work offline.

Lee et al. [21] perform generic face model individualization using photographs from two views: a front view and a side view. To facilitate this, a generic model is also divided into a number of feature regions around each identified feature, and the individualization is based on the facial features obtained from the feature detection, which is carried out in two successive steps: a global matching, which is used to find locations of facial features using statistical data from the images, and a detailed matching, which recognizes the shape for each facial feature using a specific method designed for the feature.

The global feature matching achieves a high accuracy rate. Among the faces that are tested, which cover a wide range of human races, ages, hair colors, and both genders, the authors find the proper position of 201 features out of 203. However, they do not attempt to handle faces with other accessories, such as glasses.

The detailed matching, which employs multi-resolution edge detection methods, achieves different success rate for different features. For example, the forehead recognition is over 90%, the eyes and mouth extraction are around 80%, and the nose identification reaches about 70%.

After the feature detections from the two input views, the features points from these 2D views are combined into 3D points, based on which the deformation of the generic model takes place. The deformation starts from a global transformation to align the 3D points with the generic model, followed by Dirichlet freeform deformation to match the shape of the generic model against the feature points.

3.1.2 Individualization using anthropometric measurements

To create models that match different individuals, DeCarlo et al. [8] propose a facial modeling approach based on facial anthropometric measurements. These measurements are used as the fundamental elements to describe and generate a wide range of geometrical 3D head models.

Anthropometry is a biological science of measuring human body. More specifically, it stores statistical measurements of human body parts in libraries. These libraries contain data which characterizes human bodies by gender, color and age. Data from the anthropometry studies is used in many applications such as plastic surgery planning, human-factors analysis, and 3D human head construction, etc.

The anthropometric measurement – see Fig. 3 involved in [8] includes around 130 feature points and their relative distances, describing the characteristics of a human face. These feature points are based on the Farkas [10] system. Given such a measurement of an individual face, the algorithm generates a static facial model using variational modeling. Variational modeling is an optimization method

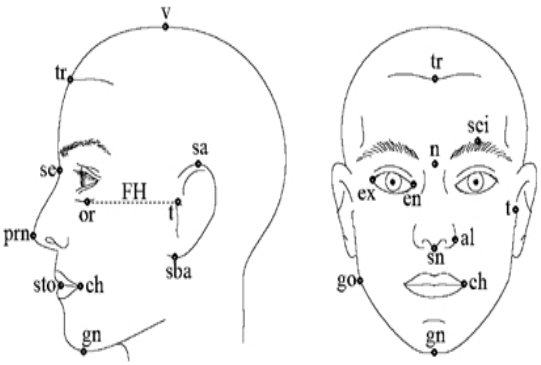


Fig. 3. Anthropometric landmarks on the face [8]

which generates a face model constrained by the features described by the anthropometric measures.

A major disadvantage of using anthropometric measurement as in DeCarlo et al. [8] is that the anthropometric measurements only provide a statistical description of a human face and hence cannot offer more specific human face features such as hooked nose or double chin. Moreover, the employed optimization process involves considerably large computation resource.

3.1.3 Functional face model adaptation

Zhang et al. [48] present an algorithm which allows us to adapt a functional generic face model to an individual face model. The generic model is equipped with a number of pseudo muscles in order to support facial animation – see Sect. 4.1 for more details of pseudo muscle-based facial animation.

The process starts with the specification of a small set of anthropometric landmarks on the 2D images of both the generic and scanned model. The 3D positions of the landmarks are recovered automatically by using a projection mapping approach. A global adaptation is then carried out to adapt the size, position and orientation of the generic model towards the scanned model, referring to a series of measurements based on the recovered 3D landmarks. Following the global adaptation, a local adaptation deforms the generic model to fit all of its vertices to the scanned model. Meanwhile, the underlying muscle structure of the generic model is automatically adapted as well, such that the reconstructed model not only resembles the individual face in shape and color but also allows facial animation from the adapted pseudo muscles.

3.1.4 Multi-layer model individualization

To continue from [8], Kähler et al. [15] present their technique to generate animatable 3D face models. Compared to the models in [8], this work is featured by generating head models with multiple layers to simulate anatomical head structures including skin, muscle, skull, mass-spring, and other separated components (e.g., eyes, teeth, tongue), etc – see Fig. 4. For each model, up to 24 major muscles are used for facial expressions and speech articulations. The skin and muscles are attached to the skull via a mass-spring system. These head models allow real-time animation based on the simulation of facial muscles and elastic skin prosperities.

To facilitate the modeling for individuals, a generic model is provided with the above five layers. Similar to [8], some landmarks are taken from a standard set of the anthropometric literature, which are small dots placed on the model to define the features. These tagged anthropometrically meaningful landmarks allow us to fit the generic model to scanned 3D face models, creating a wide variety of animated face models. Moreover, by using the anthropometric measurements to simulate the growth of a human head, the technique is capable of generating animatable human heads at different ages.

As an extension from this method, Kahler et al. apply the technique to scanned real skull data. As a result, they are able to reconstruct expressive faces from the skull data in an application which is named as "reanimating the dead" [16].

3.1.5 Model adaptation-based on topographic analysis

Besides the use of anthropometric landmark data as introduced in the previous sections, Yin & Weiss [45] suggest the use of topographic representation to give facial features for face model individualization from a generic model. To generate the topographic representation, an

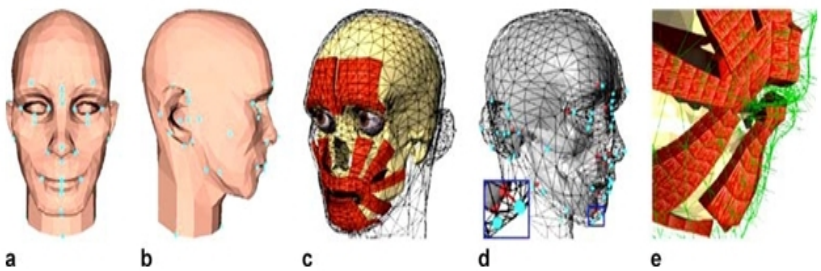


Fig. 4a–e. The reference head: **a** head geometry with landmarks, front view; **b** side view; **c** skull and facial components; **d** skull landmarks related to subset of skin landmarks; **e** facial detail showing spring mesh connecting skin and muscles [15]

input face model is needed. Then a topographic analysis treats the face image as a topographic terrain, and labels each of its pixel as one of the topographic labels, including peak, ridge, saddle, hill, float, ravine, or pit. The hill-labeled pixels are further divided into convex, concave, saddle or slope hill. Following the topographic analysis, the individualization carries out an optimization using the criteria measured by these labels. This optimization adapts a generic model to an individual model.

3.1.6 3D face model from video streaming

The method from Zhang et al. [49] presents an approach of generating facial models and expressions by using multiple video cameras. Video cameras are used to capture model data via stereo matching. A fitting process is then employed to fit a face template to the captured data.

The data capture hardware consists of 6 synchronized video streams (4 monochrome and 2 colors), running at 60 fps (frames per second). This provides a stereo system, which uses three of the cameras to capture from the left side while the other three cameras are from the right. A spacetime stereo algorithm is employed during this data capture, which calculates a time sequence of depth maps using stereo matching.

Then, a template fitting and tracking process is used to fit a face template to the depth maps in the first frame and then perform tracking through the whole sequence. The result of this fitting gives high quality meshes with vertex correspondence over the time. The output of the data capture gives high-resolution 3D meshes at 20 fps sequence, capturing the face geometry, color and motions.

Once acquired, this sequence of models can be interactively manipulated to create expressions using a technique which is named "faceIK" by the authors. More details about the facial animation are given in Sect. 4.3.1.

A main disadvantages of this algorithm is that it is quite resource demanding as it involves special equipments to capture the face geometry and motion. Also, the work is only presented for the animation of a single face without considering variations of different individuals.

3.1.7 Model adaptation methods

All of the above generic model individualization methods involve model adaptation, a technique which deforms a generic model to an individual (target) model. Basically, this is a scatter data interpolation problem, which drives the movement for each vertex on the generic model towards the target model, given only a sparse set of feature point positions as input.

A common approach to solve such a problem is to construct and minimize an interpolation function based on radial basis functions [15,31]. To further improve the quality of the model adaptation, Kahler et al. [15] develop an automatic procedure which allows us to refine the adaptation by using model subdivision without requiring users to input a large number of dense feature points.

In addition, in the work of DeCarlo et al. [8] employs variational modeling for the model adaptation. It works on B-spline face models. The anthropometric measures are used as linear and non-linear surface constraints and subsequently surface fairing with these constraints are applied. This generates smooth surface models which match the anthropometric features.

Zhang et al. [48] deforms a generic model to an individual model through global and local adaptation. The global adaptation involves size and orientation adaptation based on some landmarks that are semi-automatically identified, while the local adaptation includes moving the model vertices locally to fit into the individual model. Essentially, the global adaptation is a process of face pose recovery and model scaling, and the local adaptation provides small adjustments for the vertices positions according to the local geometry.

In [45], second-order differential equations are used to define the adaptation of the generic model. Within these equations, topographic features are used to define the internal and external forces. The external force drives the deformation of the model, while the internal force maintains the shape of the model during the deformation. The result of the deformation matches the generic model to a single frontal view of an individual face.

In [49], the specific adaptation problem is to fit a generic model to a depth map captured from the videos by using Gauss–Newton optimization. The depth map represents the face geometry by a depth value h at each point (x, y). Similar to [45], the fitting metric has two terms, a depth matching, which measures the difference in depth between the generic and target model, and a regularization term, which maintains a good shape for generic model after the deformation.

3.2 Example-based face modeling

The techniques introduced in this section concern creating face model through the combination of existing models. Such methods require support from a collection of face models. Given desired facial features, for example, by using a face photograph as in [4], optimization method is used to find the right combination coefficients. The linear combination of the collected face models with these optimized coefficients provides a close match between the synthetic model and the desired facial features. Noticeably, linear interpolation is also used to create facial animations from a number of example expressions, such as in [31,46,47]. These will be discussed in Sect. 3.

3.2.1 Morphable face modeling

Blanz & Vetter [4] present a face modeling technique named as morphable modeling. A distinct strength of this

modeling approach is that it allows generate a wide variety of face modeling with minimal user input. More specifically, a face model can be created from a single photograph supplied by users and the created model matches the facial features portrayed by the photograph.

The method requires an example set of 3D face models. Morphable face modeling is based on transforming the shapes and textures of these example face models into a vector space representation. The shape and texture of a new face model is represented by a linear combination of these transformed vectors. Moreover, this method allows face manipulations using complex parameters, such as gender, fullness of a face and distinctiveness etc.

However, the implementation of the morphable face modeling is not straightforward, since it requires a large collection of 3D face models within which the dense point to point correspondence between the models have to be established. Also, the algorithm is time consuming – it takes tens of minutes to acquire the geometry of a face from a photograph.

Further, based on the morphable face modeling technique, Blanz et al. [5] propose a face exchange technique, which allows the replacement of an existing face in a target photograph with a new face. To do this, only a single image of the new face is required. By making use of the morphable face modeling, one can create a 3D face model for the new face as a combination of the existing face model collection. By employing optimization method, one renders the model with proper illuminations and postures. This allows the rendered result to fit into the target photograph and thence replace the existing face.

Remarkably, the morphable face modeling has been utilized in the work of [13], in which 3D face model is tracked from a real-time video sequence. It has been found that 3D morphable modeling is extremely well suited to the task of fitting a 3D model to a target video in real time.

3.2.2 Multi-linear modeling

Multi-linear face modeling is another way to create a desired face model from existing face model examples [40]. Similar to the morphable modeling in [4, 5], the multi-linear face modeling requires careful pre-processing of the collected examples to set up full vertex to vertex correspondence between the examples. Then, these examples are organized in the form of a data tensor, which encodes model variations in terms of different attributes, such as identity, expression and viseme. This allows independent variation of each of these attributes. By using the organized data tensor, an arbitrary face model with desired facial expression can be modeled as a linear combination of these examples.

In fact, the multi-linear face modeling was proposed in [40] to make face models for face transfer. The face transfer allows mapping video recorded performance of an individual face to the facial animation of another one. More details of face transfer will be discussed in Sect. 4.2.3.

3.3 Discussion

Comparing between the generic model individualization (GMI) approach and the example-based face modeling (EFM) approach, we can see that a number of strength and weakness of these two approaches:

- GMI only needs one generic model, without involving the support from a face model collection as in EFM. Further, it is necessary that the face models in the collection required by EFM are registered to each other with vertex-to-vertex correspondence. Such a collection might not be accessible for many potential users. As a conclusion, GMI is suitable for applications which have no access to large model collections.
- GMI requires a challenging process of providing facial features. This involves either considerable amount of manual work, or highly cost equipments & vision techniques, as in [49]. In contrast, given a face model selection, EFM allows us to generate face models from one single face image without requiring the identification of facial features. Further, the techniques in EFM such as [4] also allow us to recover face models with facial expressions from single pictures.
- To our best knowledge, EFM only works for face models with single layers, while researcher have used GMI to develop multi-layered anatomical-based models. Extension of EFM to accommodate multi-layered models can be a difficult challenge as potentially it needs to involve multi-layered model collection. Also, the optimization method required in EFM may also be more complicated to deal with multi-layer models.

Table 1 provides a summary for these two approaches and their typical examples.

Table 1. Comparison between GMI and EFM

	Examples	Strength	Weakness
GMI	[31] [21] [8] [15] [49] [48] [45]	1. Only require a generic model 2. Works for models with multiple layers	1. Need to identify facial features 2. Need considerable user inputs
EFM	[5] [4] [40]	1. Do not need to identify facial features 2. Only need a single face image input 3. Allow generate facial expressions	1. Need support of a registered face collection 2. Difficult to generate multi-layer models

4 Facial expression and animation

Driven by the desire of improving visual realism of facial animation and creating naturally looking facial expressions, great effort has been made from the graphics community, which can be categorized as follows:

- Simulation-based approach, which employs simulation methods to generate synthetic facial movements by mimicking the contraction of facial muscles. More details are given in Sect. 4.1.
- Performance-driven animation tries to learn facial expressions from recorded videos or captured face movements and subsequently makes synthetic facial expressions by applying them to a face model. More details are given in Sect. 4.2.
- The blend shape-based approach creates new facial expressions of a face from the linear combination of collected expression examples of the same subject (face). More details are given in Sect. 4.3.

4.1 Simulation-based approach

The motivation of the simulation-based approach is to create synthetic facial expressions by simulating facial muscle actions on a face model. This requires us to define the functionality and locations for a number of pseudo muscles on the face model. The functionality of a pseudo muscle is defined in terms of its influence on the face model, which depends on the employed simulation method. The overall synthetic facial expression is determined by the combination of the pseudo muscle contractions [42]. Further, initiated by the idea of using pseudo muscle simulation, a number of multi-layer models have been developed to simulate the anatomical structure of human face, including skull, muscle, soft tissue, skin etc. [20,37,38]. This greatly improves the visual realism of the synthetic expressions.

4.1.1 Pseudo muscles

The paper from Waters [42] presents a classic work in synthesizing facial animation using pseudo muscles. The movement (extraction) of each pseudo muscle is defined to link to a particular area of the face model. For example, if a facial expression with an open mouth is wanted, only the muscles associated to mouth areas need to be adjusted. This can potentially avoid the facial distortion created by [3], in which a stretched point on the mesh can take effect to the whole mesh shape. In general, the pseudo muscles influence either the upper or lower face. The upper facial muscles are responsible for changing the appearance of the eyebrows, the upper and lower lids of the eyes, while the lower facial muscles determine the appearance of the chin, ears, lips, and the areas around the eyes and the neck.

Further, in [42], pseudo muscles are classified into small groups according to their functionalities. The outcome of this classification is also known as action units (AU), which defines the pseudo muscle actions during various facial movements. In other words, the AU for a specific facial expression (e.g., smile) tells us which muscles need to be activated to synthesize this facial expression. This physics-based simulation greatly reduces the amount of work that must be input by an animator, as he/she can directly specify required facial expressions by controlling the movements of the AUs.

4.1.2 Synthetic skin layers

To improve upon the basic pseudo muscle-based facial animation as presented in [42], the work from Terzopoulos & Waters [38] proposes facial animation by contracting pseudo (synthetic) muscles embedded in an anatomically motivated face skin model.

More specifically, the face model is composed of three synthetic skin layers which are made of spring-mass. The physical simulation propagates the muscle forces through the physics-based synthetic skin thereby deforms the skin to produce facial expressions. This is a combination of the pseudo muscle approach with the anatomy-based facial modeling, which significantly improves the realism of synthetic facial expressions compared to the earlier techniques [27,42].

This approach is also amenable to improvement through the use of more sophisticated biomechanical models and more accurate numerical simulation methods. This is, of course, subject to an increase in computational expense.

4.1.3 Biomechanical skin model

Lee et al. [20] addresses the challenge of automatically creating individual facial models with highly realistic facial expressions based on pseudo muscles. This method allows us to adapt a well-structured generic face model to an individual face model acquired by a 3D Cyberware scanner in a highly automated manner. This adaptation process is similar to the deformation techniques introduced in Sect. 3.1. The outcomes of the algorithm are functional facial models which allow significant amount of facial details and high quality facial animation. The algorithm is applicable to a wide range of individuals.

To allow facial animation, the generic geometric model used by the algorithm consists of a number of different layers. More specifically, on top of the face model, five different layers are used, including the epidermis, dermis, sub-cutaneous connective tissue, fascia and the muscles. Each layer is described as a triangle deformable tissue, which is connected respectively with all the other layers – see Fig. 5. Such a skin tissue modeling is identified as "physically-based modeling of human facial tissue". This

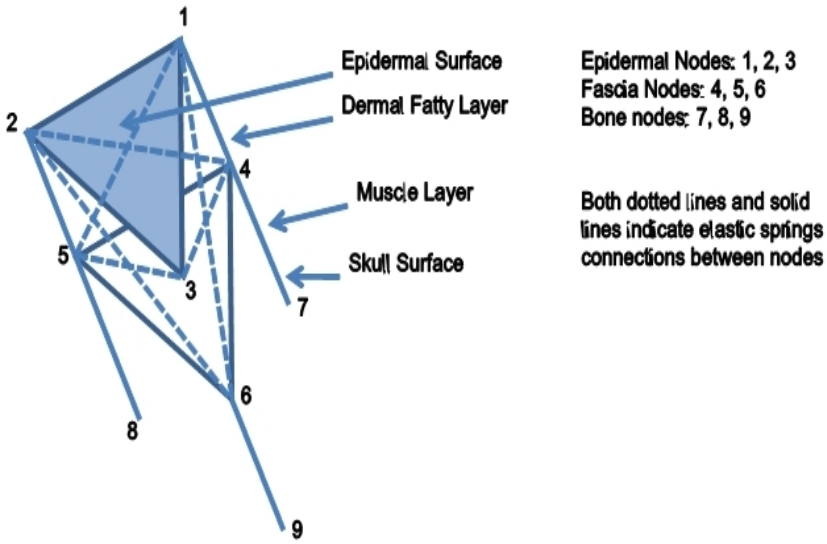


Fig. 5. Triangle deformable tissue [26]

biomechanical face skin model is claimed to be more accurate than those used in the previous methods as in [38].

However, after the creation of a functional facial model, considerable experience is required from users to generate desired facial expressions. For example, to animate the face model, the user has to pull in/out or even to move the elastic angles of the triangles. The disadvantages of that process are the complexities and difficulties to implement a reliable facial expression because it expects from the user a very good consideration of transforming the triangles. Moreover, it is difficult to model the detailed skin deformations, such as expression wrinkles, and consequently the results tend to be less realistic.

4.1.4 Anatomically accurate head model

The recent work from [37] has been claimed as the modeling of a living male subject with high anatomical accuracy. This is a volumetric modeling based on the data acquired from the subject, covering a range of structures including facial musculature, passive tissues and underlying skeletal structures. The rigid articulated cranium and jaw consist of about 30 000 surface triangles, while the flesh is modeled in the form of 850 000 tetrahedral, out of which 370 000 are simulated. 32 facial muscles are included. The data is captured using laser and MRI scans. As claimed by the authors, this model was constructed within a two-month period from 5 undergraduate students – see Fig. 6 for an illustration of the model.

Animating such a complex structure is a highly nonlinear process which involves controllable an-isotropic muscle activations based on fiber directions. To allow for the animation of such a complex structure, the authors propose a performance-based method, which is capable of automatically determining muscle activations by tracking a sparse set of surface landmarks on a performer. Then the resulting animation is obtained by using a non-linear finite element method. Once the controls are reconstructed, the model can be subject to many applications, such as interaction with external objects, dynamic simulation to capture ballistic motion. The facial expressions can be edited in the activation space. The results are claimed to be not only visually plausibly but also anatomically accurate.

Noticeably, the simulation-based approach has also been applied to simulate human body structures and movements. A typical example appears in Wilhelms & Gelder [43], which performs body modeling with multilayers including skeleton, muscles and skin. All these three components are connected in order to create the animation movements between the skin, skeleton and muscles. That process is presented as one of the first succeeded techniques for constructing a 3D body for animating purpose. However, according to Allen Van Gelder [11] by assigning the same stiffness to all springs it appears that

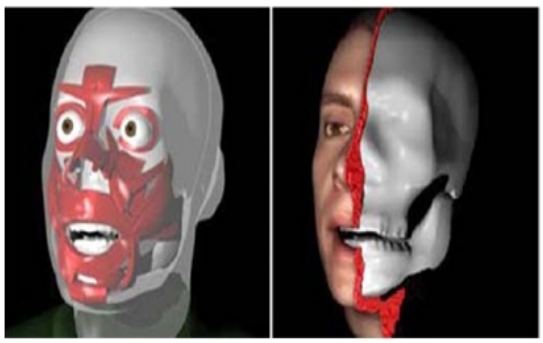


Fig. 6. Anatomically accurate head model [37]

the system fails to simulate a uniform elastic membrane, for equilibrium calculations.

4.2 Performance-driven animation

A simple performance-driven animation is geometry warping based, in which prominent facial features are tracked through a sequential of facial expression images and subsequently their movement is copied to a new face to synthesize facial expression [22, 29, 44]. On top of this, a more delicate algorithm also involves the use of expression ratio image (ERI) in order to create fine details such as face wrinkles [23]. Other examples of performance-driven facial animation include: live facial performance capture [12]; face transfer using multi-linear models [40]; vision-based facial animation control [6].

4.2.1 Geometry warping-based method

Geometry warping-based method gives the simplest form of performance driven animation. The input of a geometry warping-based method consists of two photographs from the same person as source images – the first one is a neutral face while the second one gives a facial expression, plus a target neural face. By manually or automatically locating the facial features, such as eyes, mouth, nose in these images, the algorithm calculates the movements of these features during the facial movement using difference vectors and subsequently applies the difference vectors to the target neural face. This transfers the facial expression from the source image to the target face.

4.2.2 Expression ratio image (ERI)

Normally, facial animation that generates from the geometric deformation of facial models lacks fine details that often appear in real human facial expressions, such as creases and wrinkles. These fine details are normally captured by illumination change during the facial movement. To address the challenge of presenting realistic facial expression, Liu et al. [23] present an image-based approach, which employs the expression ratio image (ERI). ERI allows the capture of the illumination change caused by fine facial details from existing face images. The work is inspired by the research in presenting and transferring lighting and illuminations between images [7, 25, 36].

In fact, a large amount of tiny visible details of a facial expression comes from the change of surface normal on the face, which gives rise to the change of illuminations under a fixed lighting environment. The idea of using ERI comes from the observation that such an illumination change can be extracted in a skin-color independent manner. An ERI is defined as a ratio image which captures the illumination change from example facial expression images. When applying such a ratio image to a neural face, all the fine details of the facial expression from the example images can be preserved – see Fig. 7 as an example.

However, this method works under the assumption that the illumination change only happens when the facial expression changes and the source and target images have similar lighting conditions. If one of the above conditions cannot be held, the ERI-based approach is not able to create a high quality output.

4.2.3 Face transfer using multi-linear models

The work presented by Vlasic et al. [40] allows mapping facial movements from a recorded video to a target face. Such facial movements include visemes (speech-related mouth articulations), facial expressions and head pose. The authors have termed this technique as "Face Transfer". Face transfer extracts the facial movements of an individual subject and subsequently applies them to a target face – see Fig. 8 as an example.

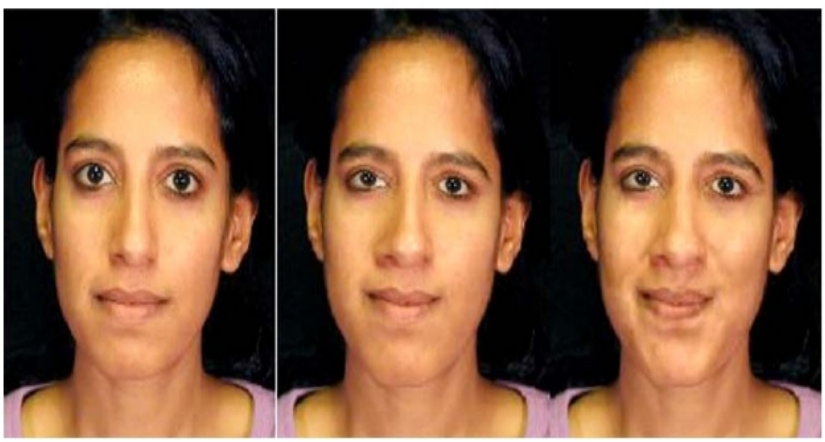


Fig. 7. Expression ratio image approach. *First image* is with a neutral expression (input image). The *second image* is after the geometric warping and the *third one* with ERI. The wrinkles in the *third image* are the result using the ERI [23]

As introduced in Sect. 3.2.2, face transfer is based on a multilinear modeling of 3D face meshes. Multilinear models consist of a collection of face meshes and their estimated variations in terms of different attributes, such as their sizes, identifies and expressions. The modeling allows each of the attributes to be adjusted separately, which facilitates the editing of the facial performance. In principle, if the face collection is sufficiently large, the multilinear modeling allows us to generate any face with any expressions and any visemes. However, as a proof of concept, the collection used in [40] only offers very limited size.

Continuously, the multilinear model can be linked to an optical flow-based tracker. The tracker estimates performance parameters and detailed 3D geometry from video recordings. The mapping from the performance parameters back to the 3D shape is then calculated. Arbitrarily mixture of pose, identity, expressions, and visemes from two or more videos are allowed.

An advantage of the face transfer technique is that it neither requires performers to wear visible facial markers or to be recorded by special face-scanning equipment. Hence, it provides an easy-to-use facial animation system from the users point of view.

4.2.4 Real time facial expression tracking

Chai et al. [6] demonstrate vision-based control of 3D facial animation by presenting a system which allows us to extract animation control parameters from a video and subsequently apply these parameters to a 3D face model to create high quality facial animation. By using this method, users are able to control the animation using recorded face actions. It is expected that low quality videos is sufficient for such a facial animation control.

The input of the algorithm is a single video stream recording the user's facial movement, a preprocessed motion capture database, and a 3D head model captured by laser scanner.

The system has four components. First, video analysis, which tracks the head position in the video, identifies a small number of features and subsequently gives expression control and head pose parameters. Second, preprocessed motion capture data contain a set of head motion & expression data. The head motion and facial deformations are automatically decoupled in the motion capture data during the pre-processing. Third, expression control and animation transform noisy and low resolution control parameters into high quality motions. Fourth, expression retargeting applies the synthesized motion to animate 3D high resolution models.

The expression tracking step involves tracking 19 2D features on the face: one for each point of the upper and lower lip, one for each mouth corner, two for each eye brow, four for each eye and three for the nose. The expression control parameters generated in the first step includes the parameters to describe the movement of the features such as the mouth, nose, eye and eyebrow of the actor in the video. These noisy and low resolution parameters are converted into high resolutions by using an example-based motion synthesis method, which compares the low resolution parameters with the motion data captured in the second step, and subsequently synthesizes a proper high resolution motion. Finally, to map the synthesized motion to a target 3D model, an efficient expression cloning technique is used, which pre-computes a number of bases for the facial deformation of the target model, and then blends them to create run-time facial expressions according to the synthesized motion.

4.2.5 Facial animation capture

Guenter et al. [12] create a system for capturing human facial expression from a live performance and replaying it in a highly realistic manner. The system allows capture the geometry, color and shading information of a face expression and subsequently re-display it using a high quality deformable polygon model decorated by a dynamic texture map.

A 3D scan is used to capture the geometry of the actor's face. To acquire the live performance of the face, multiply cameras (6 calibrated cameras) are used to record the face movement from different positions simultaneously. To help the data capture, a large number of sample points (182 points) are placed on the face. Their positions are reconstructed through the recorded videos from the multiple cameras, which are then used to distort the face geometry in order to create desired facial expressions.

Together with the tracking of the geometric data, the multiple video cameras also capture multiple high reso-

Fig. 8. With the multilinear model, the expression, and the viseme from the *second* and *third left*, respectively, can be transferred to a neural subject face (the *far left*). The *far right* gives the result [40]

lution video images. These images are used to generate a dynamic texture map for the face model which gives a vivid appearance during the re-display of the facial expression. To generate the texture map, the images from the cameras are weighted differently according to their positions.

4.2.6 Principal component-based facial motion capture and analysis

Kshirsagar et al. [19] present their techniques for facial motion capture using principal component analysis (PCA). Through the PCA, a number of principal facial movements are identified from the captured data, which provide insight into the mechanism of general facial movement. This is subsequently used to control the synthesis of new facial expressions.

6 cameras and 27 markers are used for the capture. These markers are compatible to MPEG-4 feature point locations. The work is primarily aimed at speech animation. During the recording of a speech performance, the movements of 14 markers are extracted after removing global head movement.

By applying PCA to the captured data, 6 principal face movements are obtained, which cover the major mouth actions during speech, including open mouth, puckered lips, parted lips, etc. Then, based on the observation that suppressing or enhancing certain principal face movements can generate satisfactory results during the practice of facial expression mixture, various controls are applied to the synthesis of by assigning different weights to different principal movements.

4.3 Blend shape-based approach

The blend shape-based approach creates a desired facial expression through the combination from a set of existing examples. This bears similar idea as the face modeling method presented in Sect. 3.2, which also employs linear combination from a number of exiting face models. The combination can be linear interpolation applied either to images [46,47], or to face models [31,49], or can be morphable modeling based, such as in [4].

4.3.1 Interpolation between models

Given a number face models with different facial expressions, a straightforward idea is to generate facial expressions in-between by using linear interpolation. This has been used frequently in many applications of facial animation. For example, the work from [31] and [49] both generate facial expression following the reconstruction of face models with various expressions.

Noticeably, the work from [32] uses linearly combined 3D face models to recover face position and facial expression from an input video sequence. This recovery process takes place in each frame of the video, and it employs a continuous optimization method in order to find the best matched model at each frame. The 3D model, which is used for the fitting, is based on the linear combination of a set of face models with different facial expressions. This face model set is generated using the technique presented in [31].

However, an obvious disadvantage of this approach is that only facial expressions in between existing examples can be created. Therefore, the technique requires considerable large number of facial expression examples.

Also, linear interpolation does not possess high accuracy and hence is not a prefect solution for generating in between expressions. Remarkably, the recent work from [49] overcomes the weakness of using linear interpolation by presenting a technique named "faceIK". Essentially, faceIK is an inverse kinematics technique, which blends the models to generate different facial expressions under user-specified controls. Moreover, the authors also present a new representation name "face graph", which encodes the dynamics of the face sequence and can be traversed to create desired facial animations.

4.3.2 Interpolation between images

The technique presented in Zhang et al. [46,47] allows us to generate high quality facial expression with significant details such as wrinkles, given a set of example images of different facial expressions. This technique can also be applied to 3D models.

To use these example images, geometry positions of the feature points in the example images need to be identified. Then, a photorealistic facial expression can be obtained from a convex combination of the example expressions based on these positions. Since this technique makes use of high quality expression examples, it can generate photorealistic and natural looking expressions with fine facial details.

Further, to overcome the challenges of automatically recovering feature points from the images, the authors develop a technique to infer missing feature points from the tracked face by using an example-based approach. This allows us to be less demanding on the feature point recovery and tracking technique. In other words, the system should still be well-functioned even if the number of tracked feature points is fewer than what the system requires.

4.3.3 Blend shape animation from model segmentation

To generate high quality facial expression for 3D face models using blend shape methods, Joshi et al. [14] present a method which segments the face models into small regions. The shape blend animation based on this segmentation allows us to handle specific part of a face

without affecting other irrelevant parts and hence preserve significant amount of complexity of human expressions. The face model segmentation method presented in [14] is claimed to be automatic and physically motivated.

4.3.4 Reanimating faces and images using morphable models

Continuing from their previous work [4], Blanz et al. [4] present a technique that allows us to change facial expressions in existing images and videos. The morphable modeling method, which was used previously for face modeling (see Sect. 3.2.1), is extended here to cover facial expressions. To achieve this, instead of only collecting neutral faces as in [4], 35 laser scans of facial expressions are also captured and stored in the face model collections. Morphable modeling using such a face model collection allows 3D reconstruction of non-neutral face models, i.e., it generates face models with expressions. The reconstructed model can be adjusted to change its facial expression and rendered back to its original image or video.

The advantage of using this morphable modeling-based approach is that it can work on any face shown in a single image/video, without requiring example expression data from that particular person.

4.4 Discussion

Comparing the three major facial animation approaches as mentioned above, we can conclude their strength and weakness as follows:

- The simulation-based approach only needs to involve computing resources. Basically, it employs and simplifies a number of biomechanical muscle equations for producing visually correct facial movements. The performance-driven approach, which requires the capture of live facial expressions, involves considerable high cost computer vision equipments for data capturing. Given the state of the art of computer vision, the performance of such a data capture process may not be easy to improve rapidly in the near future. For the blend shape approach, a collection of face models with different facial expression is required. Again, obtaining and accessing such a face collection can be not easy from an ordinary user point of view.
- The performance-driven approach has the most potential for achieving more visual realism. In fact, the performance of the simulation-based approach is always limited by the underlying biomechanical simulation method. Unfortunately, most of the biomechanical simulations, including those claimed with high accuracy as in [37], take a great simplification from the real world model. In contrast, the methodology of the performance-driven approach, which learns facial expression from real performance, offers a mechanism to be close to the real world animation as much as possible.
- The shape blend approach is only capable of creating animation in-between the existing examples, which is a great limitation. Also, its performance heavily relies on the quality of the example models, as well as the employed interpolation methods.
- In principle, the simulation-based approach bears great potential in medical applications. It can work in conjunction with anatomical and medical data, medical simulation, which provides high quality medical models. Such a simulation can potentially provide guidelines for medical professionals in their practice.

Table 2 gives a summary of comparison of the performance between these three major approaches.

Table 2. Comparison between the simulation, performance-driven and shape blend approach

	Examples	Strength	Weakness
Simulation-based approach	[42] [38] [20] [37]	1. Only require computing resources 2. Great potential in medical applications	Artificial-looking facial expression
Performance-driven approach	[44] [23] [40] [6] [12] [19]	Great potential to achieve visual realism equipments	Need to involve motion capture
Shape blend approach	[31] [49] [46,47] [14] [5]	Easy implementation	Need high quality facial expression examples

5 Applications

Facial modeling and animation faces a wide range of demands from industry nowadays. This section will discuss how the techniques described within this paper meet these demands, providing their suitability towards different areas of exploitation. Here, we will focus on major applications including movie industry, computer games, medicine and telecommunication. Of course, it is well understood that the actual potential of facial modeling and animation goes beyond these limited number of applications discussed here. In addition, as a typical example of applying facial animation to multimedia, we also introduce facial animation in MPEG-4 in Sect. 5.5.

5.1 Movie industry

The movie industry has received huge benefit from the advance of facial modeling and animation techniques. The increasing number of computer-made movies, such as Toy Story, Shrek, Monsters Inc, Monster House, King Kong, has demonstrated that the current techniques are well suitable to create cartoon styled movies. To this end, traditional techniques, such as geometric face modeling and simulation-based facial animation [20,38,42], are quite competent for the job.

However, major work still needs to be done to achieve results with high realism. Due to its post-processing nature, movie production usually allows considerably long processing in order to achieve the desired quality. Hence, methods involving high cost resource are permitted, such as large face models with significant details captured from 3D scanners, facial motion captured using costly equipments, etc. However, given the state of the art, there is still a long way to go before we are able to completely remove artificial looking from computer synthesized faces and produce outcomes that are indistinguishable with those from real faces.

5.2 Computer games

The nature of computer games implies that rapid response speed and, in many occasions, the power of real-time processing is essential. Given the fact that a large number of computer games take place in hand-held devices nowadays, many of them employ face models with moderate quality and artificial looking facial movement in order to compromise with the speed issue.

Highly accurate computation for facial animation, such as anatomy-based simulation or high degree of shape blending (interpolation), is normally not necessary for computer games. To maintain a good balance between the speed and image quality, preprocessed or captured facial movement data can be stored so as to take off the burden from real time computing. Although this only supports a limited number of facial expressions, it should be suitable in the context of computer game design at present stage as many current games only involve limited facial expressions.

Hence, given the limitation of the current techniques and computing power, there is an immense barrier to overcome before we achieve completely freeform and realistic facial animation in computer games.

5.3 Medicine

Computer-based simulation can help medical society to enhance their knowledge to the underlying problem of facial movement and expressions by creating insight views for facial anatomy and structures. The aim of such an application is to achieve highly accurate simulation rather than just to gain a stunning looking. While early effort on pseudo muscle-based simulation [42] did not provide required precision, recent work on anatomy-based simulation [37] makes a significant step towards this direction. Noticeably, this application also overlaps with the research in medical visualization, which allows us to display the data of human anatomy in a highly accurate manner.

5.4 Telecommunication

The capability of displaying a talking face is always a desirable feature in telecommunication. Here the challenge is to reduce the time lag while the display quality is improved. Due to bandwidth limit, transmitting high resolution face images is usually practically prohibitive. A feasible solution is to store high-resolution generic face models locally, which can be adapted to different individuals involved in the communication. Given these models, only movements of the models need to be transmitted during the communication, which cost less bandwidth. However, as we introduced above, a fully automatic model adaptation is still not available, hence manual identification of facial features is still necessary. Further, completely traceless, fast and reliable facial performance tracking is still a challenge. Therefore, large progress is still required before we are allowed to view each other vividly via computer facial animation techniques during a telecommunication.

5.5 MPEG-4 facial animation

A superb feature of MPEG-4 is the support of integration of natural and synthetic scenes through an object-based audiovisual representation. It was also the first time to standardize the tools supporting 3D facial animation. To achieve high quality face modeling and animation outcomes, three types of facial data are specified [1] – see Fig. 9:

1) *Facial animation parameters (FAPs).* are designed to allow the animation of a 3D facial model, reproducing

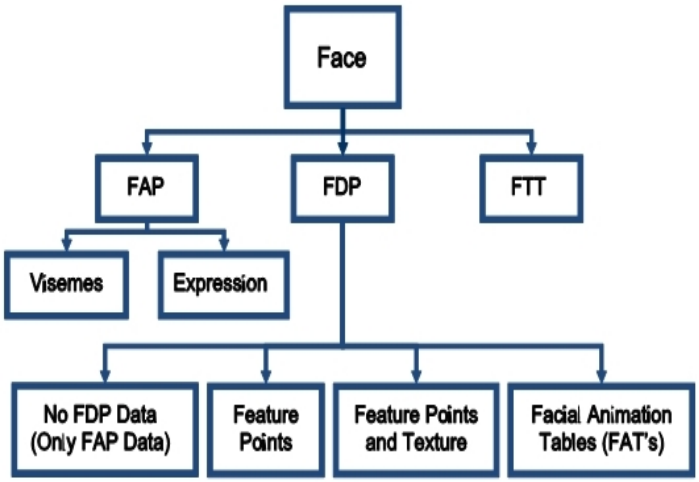


Fig. 9. MPEG-4 facial animation data

facial movements, expressions, emotions and speech pronunciation. The FAP set includes 68 parameters, 66 of which are low-level parameters related to the movements of lips, jaw, eyes, mouth, cheek, nose, etc., and the rest two are high-level parameters related to expressions and visemes. By using the FAPs, one can generate different expressions, including joy, sadness, anger, fear, disgust and surprise. Visemes are the visual analog to phonemes and allow the efficient rendering of visemes for better speech pronunciation.

2) *Facial Definition Parameters (FDPs).* FDPs allow us to configure a 3D facial model, which is to be animated by means of FAPs as described above. This can be done by either adapting a previously available model or by sending a completely new model alongside with the information about how to perform its animation. A MPEG file can potentially carry A) no FDP data, or B) feature points, which constitute a set of 3D features represented by their coordinates, or C) feature points plus textures, which include additional texture information for the model, or D) facial animation tables (FATs), which are designed to define full animation control of a complete new model.
3) *FAP Interpolation Table (FIT).* FIT allows interpolation for the FAPs. This facilitates the definition of facial animation as only a small set of FAPs needs to be included in the MPEG file to describe a dynamic facial animation. This small set of FAP is used to determine the values of other FAPs during the animation using the interpolation based on FIT.

6 Limitations and future trend

Rapid progress has been made in facial modeling and animation in recent years. Comparing recent results with those from early years, we can achieve much more vivid and realistic looking faces synthesized from computers. However, on the other side, great limitations still exist for future improvement. Here, we provide a brief summary on the major limitations of the current technology and discuss how to address these challenges in future:

1) Although we can capture extremely high quality face models with great deal of fine details through 3D scanning, how to efficiently represent and effectively manipulate such a large model has become a critical issue. These manipulations are important if we wish to modify or apply facial animation to the model.
2) While the simulation-based facial modeling and animation provides valuable insights to human facial structure and its movement, the current computational models have made significant simplification and hence lack sufficient accuracy. One potential approach to improve the accuracy is to combine facial animation with Medical Visualization technology, which offers truthful display of facial structures through captured medical volumetric data (CT & MRI).
3) The performance-driven approach for facial animation is greatly limited by current computer vision technology. Due to these limitations, particular arrangements, such as face marks and fixed camera positions, need to be made during facial movement capture, and subsequently complex data processing and analysis is required. However, frequently, many of these captures still fail to provide convincing results.
4) Currently, most of the face-rendering techniques require at least two photographs in order to allow multiple views of a face. However, in many applications users desire to create multiple views from a single input image. The existing techniques which allow us to do so [4] need support from a considerably large face model database, in which the models have to be registered. This poses as a great limitation.

Correspondingly, in future, more work will be carried out in conjunction with anatomy and medical visualization to enhance the quality and accuracy of face

models to meet the demands from medical science. On the other side, although the performance-driven approach has proved its great potential through its initial results, its future depends much on the progress of motion capture techniques in computer vision, including software analysis and hardware performance. In addition, further efforts will be needed to develop more robust techniques for face rendering from a single input of a face image.

The techniques covered in this survey are mainly designed to create face shapes and facial movements with respect to time. To carry out a comprehensive simulation to real face, combining these techniques with the modeling and animation of other associated components is necessary, such as hair, eyes, and tongue. In particular, hair modeling has been consistently challenging in computer graphics, due to its complicated and dynamic nature. Although already being adopted by many commercial software packages, hair modeling is only at its infant stage. At present, computer-synthesized results are still generally artificial looking and have large difference with real hair, especially with respect to hair animation. For more details of hair modeling, please see a recent survey paper [41].

7 Conclusion

This paper has presented a comprehensive survey for the techniques of human facial modeling and animation. We have carried out the survey from two perspectives: facial model and facial animation.

Facial modeling techniques concern how to make 3D face models. This can be further categorized into the generic individualization approach and the example-based face modeling approach. The generic individualization approach is less resource demanding – it only requires one generic model, while the example-based face modeling approach needs a considerably large model collections. On the other hand, the example-based face modeling can create a face model from one frontal face image without identifying face features, while the generic individualization approach needs considerable input to identify facial features.

For facial animation, it can be done by either the simulation-based approach, or the performance-driven approach, or the shape blend approach. The simulation-based approach only requires computing resources, while the performance-driven approach involves considerable motion capture equipments. However, the simulation-based approach only provides artificial looking animations, while the performance-driven approach has the potential to be more realistic. The performance of the shape blend approach depends largely on the existing facial expression examples. Although it is easy to implement, it only allows synthetic expression in between the existing examples.

Facial modeling and animation face many demands from various applications, including movie industry, computer games, medicine and telecommunication. Given the current state of the art of the technology, great effort is required before we are able to completely remove artificial looking from computer synthesized faces and produce outcomes that are indistinguishable from those from real faces. This is particularly true in computer game manufacture where the capability for real time computation is often significant.

Future computer facial animation depends largely on the progress of computer vision technology, based on which we can potentially produce vivid facial animation via the capture of facial movements. On the other hand, from the perspective of medical science, computer simulation helps to enhance the knowledge to the underlying problem of facial movement and expressions. In future, more work will be needed in conjunction with anatomy and medical visualization to enhance the quality and accuracy of face models.

References

1. Abrantes, G.A., Pereira, F.: MPEG-4 facial animation technology: survey, implementation, and results. IEEE Trans. Circuits Syst. Video Technol. **9**(2), 290–305 (1999)
2. Allen, B., Curless, B., Popović, Z.: The space of human body shapes: reconstruction and parameterization from range scans. ACM Trans. Graph. **22**(3), 587–594 (2003)
3. Badler, N., Platt, S.: Animating facial expressions. ACM SIGGRAPH Comput. Graph. **15**(3), 245–252 (1981)
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 187–194. ACM SIGGRAPH, New York (1999) (URL: http://www.kyb.tuebingen.mpg.de/bu/people/volker/)
5. Blanz, V., Scherbaum, K., Vetter, T., Seidel, H.-P.: Exchanging faces in images. Comput. Graph. Forum **23**(3), 669–676 (2004)
6. Chai, J.-X., Xiao, J., Hodgins, J.: Vision-based control of 3D facial animation. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 193–206. EUROGRAPHICS, San Diego (2003) (URL: http://faculty.cs.tamu.edu/jchai/projects/face-animation/)
7. Debevec, P.E.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 189–198. ACM SIGGRAPH, New York (1998)
8. DeCarlo, D., Metaxas, D., Stone, M.: An anthropometric face model using variational techniques. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–74. ACM SIGGRAPH, New York (1998)
9. Du, Y., Lin, X.: Emotional facial expression model building. Pattern Recognit. Lett. **24**, 2923–2934 (2003)
10. Farkas, L.: Anthropometry of the Head and Face, 2nd edn. Raven Press, New York (1994)

11. Gelder, A.V.: Approximate simulation of elastic membranes by triangulated spring meshes. J. Graph. Tools **3**(2), 21–42 (1998)
12. Guenter, B., Grimm, C., Wood, D., Malvar, H., Pighin, F.: Making faces. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 55–66. ACM SIGGRAPH, Boston, MA (1998)
13. Hiwada, K., Maki, A., Nakashima, A.: Mimicking video: real-time morphable 3D model fitting. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 132–139. ACM SIGGRAPH, Osaka (2003)
14. Joshi, P., Tien, W.C., Desbrun, M., Pighin, F.: Learning controls for blend shape based realistic facial animation. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 187–192. EUROGRAPHICS, San Diego (2003)
15. Kähler, K., Haber, J., Yamauchi, H., Seidel, H.-P.: Head shop: generating animated head models with anatomical structure. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 55–63. EUROGRAPHICS, San Antonio, TX (2002) (URL http://www.mpi-inf.mpg.de/~kachler/slides/sca02-headshop_files/v3_document.htm)
16. Kähler, K., Haber, J., Seidel, H.-P.: Reanimating the dead: reconstruction of expressive faces from skull data. ACM Trans. Graph. **22**(3), 554–561 (2003)
17. Kalra, P., Garchery, S., Kshirsagar, S.: Facial deformation models. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) Handbook of Virtual Humans, chap. 6. John Wiley & Sons, West Sussex, England (2004)
18. Koch, A.: Structured design implementation – a strategy for implementing regular data paths on FPGAs. In: Proceedings of the 1996 ACM 4th International Symposium on Field Programmable Gate Arrays, pp. 151–157. ACM SIGDA, Monterey, CA (1996)
19. Kshirsagar, S., Egges, A., Garchery, S.: Expressive speech animation and facial communication. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) Handbook of Virtual Humans, chap. 10. John Wiley & Sons, West Sussex, England (2004)
20. Lee, Y., Terzopoulos, D., Waters, K.: Realistic modeling for facial animation. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 55–62. ACM SIGGRAPH, New York (1995)
21. Lee, W., Goto, T., Kshirsagar, S., Molet, T.: Face cloning and face motion capture. In: Magnenat-Thalmann, N., Thalmann, D. (eds.) Handbook of Virtual Humans, chap. 2. John Wiley & Sons, West Sussex, England (2004)
22. Litwinowicz, P., Williams, L.: Animating images with drawings. In: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, pp. 409–412. ACM SIGGRAPH, New York (1994)
23. Liu, Z., Shan, Y., Zhang, Z.: Expressive expression mapping with ratio images. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 271–276. ACM SIGGRAPH, New York (2001)
24. Magnenat-Thalmann, N., Thalmann, D.: Handbook of Virtual Humans. John Wiley & Sons, West Sussex, England (2004)
25. Marschner, S.R., Greenberg, D.P.: Inverse lighting for photography. In: IST/SID 5th Colort Imaging Conference, pp. 262–265. IS&T, Scottsdale (1997) (URL: http://www.graphics.cornell.edu/pubs/1997/MG97.html)
26. Noh, J., Neumann, U.: A Survey of Facial Modeling and Animation Techniques. USC Technical Report 99-705, Integrated Media Systems Center, University of Southern California (1998)
27. Parke, F.: Computer generated animation of faces. In: Proceedings of the ACM Annual Conference, pp. 451–457. ACM, Boston, MA (1972)
28. Parke, F.I.: A Parametric Model for Human Faces. PhD Thesis, University of Utah, Salt Lake City, UTEC-CSc-75-047, USA (1974)
29. Parke, F.I., Waters, K.: Computer Facial Animation. AK Peters, Wellesley, MA (1996)
30. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM Trans. Graph. **23**(3), 664–672 (2004)
31. Pighin, F., Hecker, J., Lischinski, D., Szeliski, R., Salesin, D.H.: Synthesizing realistic facial expressions from photographs. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 75–84. ACM SIGGRAPH, New York (1998)
32. Pighin, F., Szeliski, R., Salesin, D.: Resynthesizing facial animation through 3D model-based tracking. In: Proceedings of the 7th IEEE International Conference on Computer Vision, vol. 1, pp. 143–150. IEEE Computer Society, Los Alamitos, CA, USA (1999)
33. Pratscher, M., Coleman, P., Laszlo, J., Singh, K.: Outside-in anatomy based character rigging. In: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 329–338. Eurographics, Los Angeles (2005)
34. Seo, H., Magnenat-Thalmann, N.: An automatic modeling of human bodies from sizing parameters. In: Proceedings of the 2003 Symposium on Interactive 3D Graphics, pp. 19–26. ACM SIGGRAPH, Monterey, CA (2003)
35. Seo, H., Cordier, F., Magnenat-Thalmann, N.: Synthesizing animatable body models with parameterized shape modifications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 120–125. Eurographics, San Diego (2003)
36. Shashua, A., Riklin-Raviv, T.: The quotient image: class-based re-rendering and recognition with varying illuminations. IEEE Trans. Pattern Anal. Mach. Intell. **23**(2), 129–139 (2001)
37. Sifakis, E., Neverov, I., Fedkiw, R.: Automatic determination of facial muscle activations from sparse motion capture marker data. ACM Trans. Graph. **24**(3), 417–425 (2005)
38. Terzopoulos, D., Waters, K.: Physically-based facial modeling, analysis, and animation. Vis. Comput. Animation **1**, 73–80 (1990)
39. Tu, P.-H., Lin, I.-C., Yeh, J.-S., Liang, R.-H., Ouhyoung, M.: Surface detail capturing for realistic facial animation. J. Comput. Sci. Technol. **19**(5), 618–625 (2004)
40. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. ACM Trans. Graph. **24**(3), 426–433 (2005)
41. Ward, K., Bertails, F., Kim, T.Y., Marschner, S.R., Cani, M.P., Lin, M.C.: A survey on hair modeling: styling, simulation, and rendering. IEEE Trans. Vis. Comput. Graph. **13**(2), 213–234 (2007) (URL: http://www.cs.unc.edu/~wardk/research.html)
42. Waters, K.: A muscle model for animating three-dimensional facial expression. Comput. Graph. **22**(4), 17–24 (1987)
43. Wilhelms, J., Gelder, A.V.: Anatomically based modeling. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, pp. 173–180. ACM Press/Addison-Wesley Publishing Co., New York (1997)
44. Williams, L.: Performance driven facial animation. Comput. Graph. **24**(4), 235–242 (1990)
45. Yin, L., Weiss, K.: Generating 3D views of facial expressions from frontal face video based on topographic analysis. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp. 360–363. ACM SIGMULTIMEDIA, New York (2004)
46. Zhang, Q., Liu, Z., Guo, B., Shum, H.: Geometry-driven photorealistic facial expression synthesis. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 177–186. Eurographics, San Diego (2003)
47. Zhang, Q., Liu, Z., Guo, B., Terzopoulos, D., Shum, H.: Geometry-driven photorealistic facial expression synthesis. IEEE Trans. Vis. Comput. Graph. **12**(1), 48–60 (2006)
48. Zhang, Y., Sim, T., Tan, C.L.: Rapid modeling of 3D faces for animation using an efficient adaptation algorithm. In: Proceedings of the 2nd International

Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, pp. 173–181. ACM SIGGRAPH, Singapore (2004)

49. Zhang, L., Snavely, N., Curless, B., Seitz, S.M.: Spacetime faces: high resolution capture for modeling and animation. ACM Trans. Graph. **23**(3), 548–558 (2004) (URL: http://grail.cs.washington.edu/projects/stfaces)

NIKOLAOS ERSOTELOS is a PhD student in computer science at the Department of Information Systems and Computing of Brunel University. His research targets on generating new algorithms for constructing new modeling and rendering techniques for facial synthesized expressions.
He was awarded the BSc degree ('99) in music technology from Hertfordshire University, UK. In 2005 he was awarded the MSc degree in media production and distribution with distinction from Lancaster University, UK.

FENG DONG is a lecturer in computer graphics at the Department of Information Systems and Computing, Brunel University, UK. His research interests include fundamental computer graphics algorithms, medical visualization, volume rendering, human modeling, and virtual reality. Dong received a PhD in computer science from Zhejiang University, China.
He is a member of the UK Virtual Reality Special Interest Group (VRSIG).

Paper 3

Image Based Synthesis for Human facial Expression

Nikolaos Ersotelos Feng Dong

Department of Information Systems and Computing, Brunel University, UB8 3PH

Nikolaos.Ersotelos@brunel.ac.uk

Department of Computing & Information Systems, Bedfordshire University, LU1 3JU

Feng.Dong@beds.ac.uk

Abstract

The synthesis of realistic facial expressions has been one of the most important issues in computer graphics for the last two decades. Comparing the face with other objects or human figures has always been one of the most complicated subjects due to the change of the illumination settings which are configured according to the movement, position and expression. Since then, several methodologies have been developed providing good results based on modelling and animation techniques. The main aspect of this paper is to improve the existed algorithm of Liu et all [1] and construct new modelling and rendering techniques. Moreover, the most important methodologies on modelling and animation techniques will be presented and analysed with reference to their main strengths and weaknesses from an application perspective.

The purpose of this research is to provide a faster, low cost and accurate 2D facial expression synthesis which can be expanded on 3D models, for animation purposes.

1. Introduction

Recently, research in computer graphics and computer vision has focused on synthesizing moods and emotions of human faces which is one of the most difficult and highly applicable aspects since it can be used in important industry areas such as game and entertainment, medical science and telecommunications. Scientific effort has already established many modelling and animation techniques which resulted in creating realistic facial expressions on 2D or 3D format. One of the major benefits of these techniques is synthesizing or capturing real world illumination and graphic details and transferring them to the computer graphics area.

Facial characteristics such as creases and wrinkles can only be captured by the illumination changes during facial movement. Liu et al. [1] has presented an image based approach which employs the expression ratio image (ERI). ERI is a method for capturing, transferring and lighting illumination changes from two source images of the same person to another different person image.

The current study aims at providing a novel approach of that existing technique. It will discover its limitations and will provide new techniques for better results. At this paper, several approaches of significant contribution in the computer vision area will be presented, along with their advantages and disadvantages. Afterwards, emphasis will be given on the Liu et all [1] approach on which the current study is based. The weaknesses will be categorised and corresponding improvements will be proposed, in order to provide realistic results in a faster and automated way that requires minimum interaction from the user. At section 4 some experimental results will be presented, followed by a short discussion and proposals for future work.

1.2 Aim and Objectives

This paper targets in a novel approach which allows the synthesis of accurate facial expressions using a small set of input images. This is motivated by the large amount of existing work in facial animation and modeling. The major goal is to be capable of generating different views and facial expressions of human face, requiring very limited interactions from users and limited input images. That will be achieved by a number of key techniques which are of great scientific interest:

- Facial animation and facial modeling.
- Split process in small and fast sections.
- Human facial geometrical expression deformation.
- Image colour transfer techniques to balance image colours.

2. Previous work

Since Parke [2, 3] innovative work on 3D facial model animation in 1970, several other approaches have been developed on 3D most of them are categorized in two sections, such as facial modelling and animation [4].

- Facial modelling is the section which includes all the techniques regarding the synthesis of high quality 3D head.
- Facial animation concerns with techniques that produce facial animation with high realism.

Despite the different approaches developed at these two sections, several techniques have been based on combined use of both of them [4].

The basic methodology for a 3D head construction is to use a triangle mesh. The triangle mesh describes the facial characteristics and consists by dots connected with each other by the common edges of the triangles. Another method for obtaining an accurate 3D head is by using a laser cylindrical scanner such as those produced by Cyberware [13]. Yuencheng Lee and Demetri Terzopoulos [5] presented a technique where the 3D head construction was established with the use of Cyberware scanner. They created highly realistic models with facial expressions based on pseudo muscles. Those pseudo muscles are consisted from several layers of triangles which describe the skin, nerves, skull, etc. By changing the settings of the triangle meshes (pseudo muscles) new expressions can be synthesized.

A radically different approach is the 'performance based animation', in which measurements from real actors are used to drive synthetic characters [17, 18, 19].

Douglas DeCarlo [6] has presented a technique for modelling a face based on anthropometric measurements. Anthropometry is a science which processes, collects, categorises and stores in libraries statistical data regarding the race, gender and age of real human heads. These data can be exploited in order to treat facial characteristic of a 3D face model. The developed algorithm can synthesize the best surface that satisfies the geometric constraints which the measurements impose, using variational modeling.

Pighin [7] has presented a method for generating facial expressions based on photographs. The inputs of the process are a sufficient number of images of the faces, each one captured from difference angle, and an appropriate 3D model which will be used as a base for the digitized pictures to be adjusted on. In order to succeed in matching the 3D head with the facial pose of the pictures, the position, sizes and facial characteristics are allocated on the pictures by manually placing several points as landmarks on them. The output of the process is the facial model which has been appropriately adjusted, as far as pose and facial characteristics is regarded, to the pictures. In order to create new facial expressions, 2D morphing techniques were combined with transformations.

Another face modelling technique was presented by Blanz and Vetter [8]. A face model can be created from a single picture. This technique requires a library of several 3D models. The final 3D model is based on a process which transforms the shape and texture of the example 3D model in a vector shape representation.

In a newer version [9] of Blanz approach, an algorithm was presented which allows us to change facial expressions in existing images and videos. This study proposes a face exchange method which replaces the existing face of a 3D model with a new face from a 2D image. An algorithm can estimate a textured 3D face model from a 2D facial image. Moreover, by employing optimization methods, the 3D model is rendered with proper illumination and postures.

Another approach for creating a realistic facial expression was presented by Sifakis et al [21] who used a 3D head consisted by 30 thousand surface triangles. More analytically, the model is consisted by 850 thousand thresholds with 32 muscles. Continuously the 3D head is controlled by the muscle activations and the kinematic bones degrees of freedom. The 3D model is marked with different colour landmarks which specify the muscles identity, which will be activated to generate the new expressions or even facial animations.

The morphable modeling method which had been used for face modelling is extended in order to cover facial expressions as well. The library of the system contains 3D head models with several expressions. The system, after isolating the neutral face from the photograph and synthesizing it as a 3D model, is able of changing the expression and rendering it back to the original image or video.

Expressive expression mapping [1] is a technique for facial animation based on capturing the illumination alternances of one's person's expression and mapping it to another person's face. This technique was applied at the renewed film "Tony de Peltie" to animate facial expressions of the characters. This technique has also been employed and improved in this paper. The advantage of this method is that it is a low cost process producing realistic results that accurately preserves facial details. Several approaches were introduced in that area of transferring illumination settings such as [14, 15, 16, 20].

Zhang et al [10] introduced a technique which automatically synthesizes the corresponding expression facial image retaining photorealistic and natural looking expression details. This method exploits the feature point positions of the facial characteristics and divides the face into 14 sub-regions. Those sub divisions are necessary for the system in order to change specific parts of the face according to the expression that is to be created. The system infers the feature points of expression from the subset of tracked points through an example based approach. When the feature points change position, then geometrically deformation is deployed in order new facial expressions to be generated.

3. Discussion of existing technique and new approach implementation.

Liu et al [1] has presented an approach which generates facial expressions based on geometric deformation and facial illumination settings. This image based approach employs the Expression Ratio Image (ERI), which allows the capture of the illumination settings of a face. Those data are transferred between source and the imported images.

The requisite inputs of the process are two source photographs of the same person, one in neutral face position and the other with an expression. The positions of the face features such as mouth, eyes, eye brows, ears, nose and shape of the head are manually located from the user by placing dots.

The geometric deformation is produced by calculating the difference of points' position between the source images and by transferring this difference to an imported image. Those dots are connected by triangles. The internal areas of the triangles are deformed according to the points' position changes and give the geometrically deformed facial expression (figure 1).

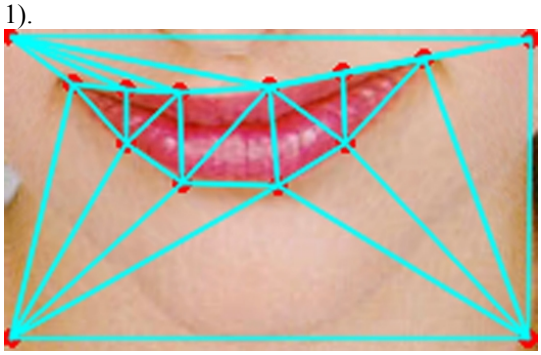


Figure 1: Dots have been placed around the face, lips, nose, eyes, eye brows, etc to describe the facial features. These dots are then connected by triangles. By moving the triangles new deformed expressions can be generated.

Afterwards, by aligning the source images with the deformed imported image (B') through image warping the system can calculate the ratio image.

$$R(u,v)=A'(u,v)/A(u,v)$$

Where R is the ratio image and A', A are the warped source images. Finally by multiplying the R(u,v) with B' the system can transfer the wrinkles of the source images on the imported image. The above calculation works only if there is a match in the illumination settings of the source and the imported images. This process needs to be developed in order to produce results with more realistic graphic details.

Current study aims to handle and improve the below mentioned issues:

The process requires from the user to manually place an efficient amount of dots in order to specify the head shape and facial features. Taking into consideration that a wrong placement of a dot will result in distorted geometric deformation, the accuracy with which the dots will be placed is a crucial factor, affecting the quality of the results. Moreover, the amount of the dots analogically increases the system requirements in terms of system memory and processor's capability; every additional dot inserted implies considerable increase in consequent calculations. It also needs to be noticed that if the amount of dots in the imported image is e.g. 100 dots the same amount of dots with the same order and position must be placed in the source images as well. Therefore, for each expression to be created, 300 dots are required to be manually inserted.

All the dots are connected with triangles. According to the amount of dots, the more triangles will be produced. Because the triangles are connected one with the other the more of them in the images the more distortion will generate on the final result. The distortion will be produced because the area which the triangles cover is deformed. The user can not choose the amount of the triangles which will be created.

The fact that the process is manual affects the duration of the procedure. The executable time for a facial expression synthesis is around 30 minutes, depending on the experience of the user.

If the image warping process between the geometrically deformed imported image and the source images is generated with errors, those errors will be transferred later on, on the imported image through ERI. According to Liu et al [1] there is not any process which could remove that distortion from the final result.

The purpose of the research is to make this algorithm more functional by altering it to an automatic process, modified in order to produce more accurate results through a faster procedure.

3.1 Split face to areas

In order to have the distortion on the geometric eliminated the amount and size of the produced triangles needs to be reduced. As a result, process has been focused on two isolated facial areas which primarily contribute to facial expressions. These are the areas around mouth and around eyebrows. These areas get extracted from the images as layers in order to be deformed separately (figure 2). After the deformation of each individual area, the deformed result is imported to the images. The major advantage of the above procedure is that having a limited area to be deformed, the executable time is considerably reduced. The necessity to insert dots in order to define areas such as the chin, ears or hairline is no longer apparent in order to generate the new facial expression. The system is able to automatically detect and handle the areas of interest. At the present stage of the study, there is a limitation apparent regarding the size of the imported image which needs to be equal to the size of source images.

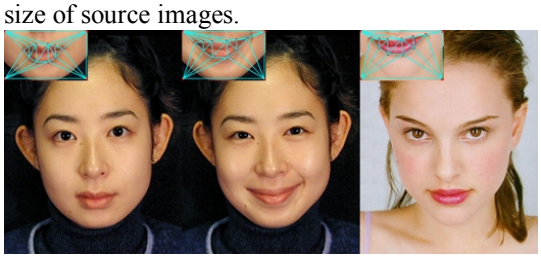


Figure 2: First step of the process: the mouth has been extracted from the main image to be geometrically deformed and afterwards paste it back on the top of the original image

3.2 Elimination of geometrical distortion

Another issue that needs to be referred is the possibility the geometrical deformation process to lead to geometrical distortion of the imported image. This effect particularly affects the shape of the face around the chin. More specifically, if the face in the source image has a big mouth, then the rectangle necessary to cover the mouth area (chapter 3.1) must be of comparative size. The size of this rectangle is possible to cover the chin or part of the contour of the face at the imported image. If then a geometrical deformation process is applied, it may insert significant distortion to the imported image. In order to avoid this distortion, the user can specify and copy the correctly deformed area, excluding the affected face contour or other distorted parts, and place it on the top of imported image with the neutral expression. The correctly deformed area gets defined by placing a sufficient amount of dots. There is no restriction on the amount of dots or on the shape of the area which these dots describe. The system adjusts and copies this user defined area on the original imported image, constructing the deformed image with no distortion at the face's contour.

3.3 Illumination transfer

In order to have the wrinkles of a facial expression calculated, transferred and adjusted to the imported image, the method that Lee has presented is being analysed and improved at this paper so that better results are being achieved regardless to the illumination conditions. As it has been already presented, Lee's algorithm uses a warping process to have the source images aligned with the imported deformed image. The following stage is the calculation of the ratio image A/A', by dividing the resultant warped source images (A for the neutral expression's pose and A' for smiling expression's pose), and the multiplication of the ratio image with the imported deformed image. In this way, the illumination settings are transferred form the source image to the imported geometrically deformed image. The disadvantage of such a process is that the ratio image is affected by the colour of the skin and by the illumination settings of the source images, which influences the wrinkle values that are eventually transferred to the imported deformed image. The resultant image can be further deteriorated if the source images are of bad quality since this will insert hard colours or artificial results in the areas of wrinkles. In order to alleviate these distortions, Lee proposes the usage of a filter, such a Gaussian filter, which will normalise the specific areas.

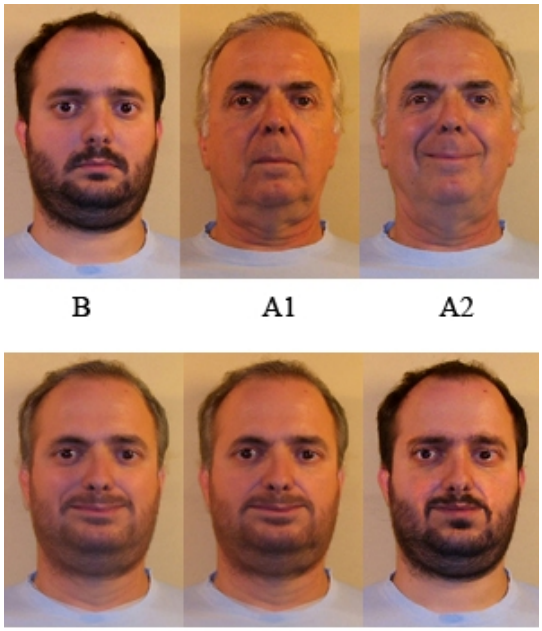


Figure 3: new approach in calculation of the ratio image

In this paper, the methodology has been changed to eliminate the above disadvantage. More specifically, the way that ratio image is calculated has changed. Instead of dividing the source images with each other, each of them is divided with the imported deformed image. The purpose of this change is to keep the wrinkles of the source images but also to adjust them with the illumination and colour skin settings of the imported image. The results are two new source images *Bg/Ag* and *Bg/A'g* where *Bg* is the imported deformed image (figure 3). The ratio image is afterwards calculated by dividing these two resultant images. In case that the skin colour between the source and imported image significantly differ, there is an option provided, allowing the user to define a threshold in the percentage of wrinkles' data that would be transferred to the final result.

3.4 Facial expression database.

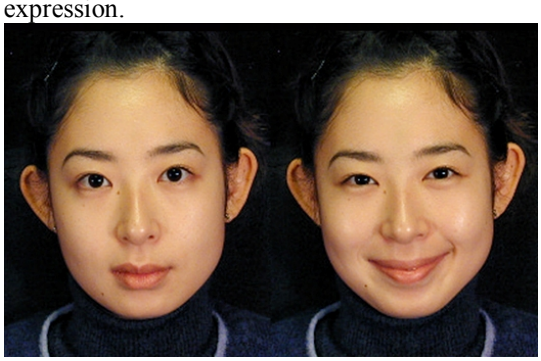
Models for several facial expressions of a specific face have been stored in a database. These models have been stored along with the coordinates of the points that define all the main facial characteristics. These data are available for the user as a plain text and can be loaded any time the user needs to accordingly modify the expression of the imported image. For each picture in the database, three types of data sets need to be maintained in order to store the mouth perimeter, the eyes and eyebrows' perimeter and also special characteristics e.g. wrinkles on the abovementioned areas that need to be described. This library eliminates the need for manual placement of dots in order to define the facial characteristics. Moreover, a user friendly interface can provide to the user many options for facial expressions to be inserted in a source image for future use. The main advantage of the previous process is that it is dependent only on the facial expressions since it is not affected by the image part surrounding the head e.g. ears, hair, neck or clothes.

3.5 Copy facial area – Noise reduction.

Subsequent to geometrical deformation process it is the wrinkles' processing. Details like wrinkles contribute to a great extent to the production of realistic results therefore it is very important to be copied and transferred from the source image. These fine details are normally captured by illumination changes during the facial movement. However, previous approaches encountered difficulty in discriminating distortion caused from hair, neck and face shape from wrinkles and useful details. At the present work, the option to place a rectangle on the face which covers and copies the mouth area and eyes-eyebrows' area is examined which isolates the specific areas and eliminates similar distortions. Facial expression and overall picture quality gets improved by applying noise reduction techniques.

4. Results

The new approach has been applied to deform facial images and create synthesized facial expressions. In this section, the results will be presented. All the source images have been provided from Liu et al. [1]. The source images are presented in figure 4, chosen in order to vary as far as facial expressions and illumination settings are concerned. The images are grouped in pairs of a neutral and a non neutral facial expression.



1rst Example (Neutral and Smiling expression source images)

2nd Example (Neutral and Sad expression source images)

3rd Example (Neutral and Surprised expression source images)

Figure 4: Source images from Liu et al. [1], which are used in order to synthesize new facial expressions.

In Figure 5, the imported image and the resultant image after the deformation process are presented, having used as source images the pair of Figure 4 (1st example). It can be noticed that the wrinkles around the mouth contribute to a realistic result, providing a good level of physical details. Deformation has been applied only in the areas of mouth and eyes. As it can been seen from the result, the logic of splitting face to areas does not deteriorate the naturalness of the facial expression. Even though no deformation was applied to the area of the nose from the user, this is deformed according to the geometrical deformation of the mouth. It is noticed that the width of the smile affects the area at the bottom of the nose; this also appears at Figure 4 (example 1) which is an original image.

Figure 5: On the left the imported image with a neutral expression and on the right the deformed image with the smiling expression and the corresponding wrinkles.

In figure 5 the imported and the deformed image of the process using as source images the pair of Figure 4 (2nd example) are presented. The wrinkles of the facial expression of the source image have been transferred to the deformed image by capturing the illumination settings (Figure 5). The difficulty at this example is encountered at the eyes' deformation. The fact that the eyes in the source image are almost closed, introduces by itself a high level of distortion. However the result illustrates that highly detailed graphics can be achieved, even though the face has been splitted to areas.

Figure 6 On the left an imported image with a neutral expression and on the right the deformed image with a sad expression and the corresponding wrinkles between the eyes and the mouth.

In figure 7, source image has been deformed according to Figure 4 (3rd example). Interesting in this figure is the raising eyebrow (surprised) expression and the resultant wrinkles in the forehead. In order to insert these wrinkles, the user has to include this area, by defining an appropriately sized rectangle, in the wrinkles calculation procedure, as described at 3.3.

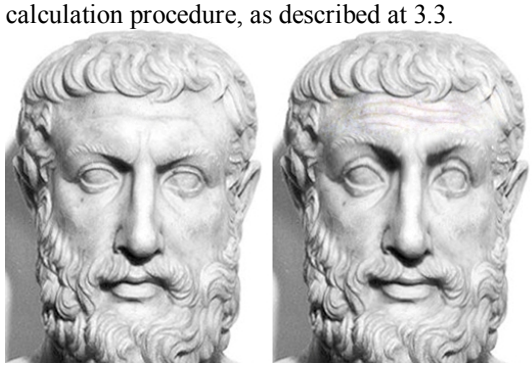


Figure 7: On the left an imported image with a neutral expression and on the right the deformed image with the raising eyebrow expression and the appropriate wrinkles on the forehead.

5. Discussions and Future Plans

In this paper, a process to deform images using warping and geometric deformation is presented.

This process is not completely automatic since a user interactive process is deployed. The next step of this research is to make the procedure fully automated. In order to have this accomplished, the facial characteristics need to be automatically identified. To this regard, an edge detection technique could be employed so that the system could detect facial characteristics, place correctly a sufficient number of dots around them and proceed with geometric deformation. In this way the user does not interfere with the system, thus eliminating the possibility of human errors.
Furthermore, new approaches can be established in order to enable the system to detect and handle facial expressions with open mouth. This could fairly complicate the process since the system would need to generate teeth or tongue by using the settings of the source images.
Future work could also include the creation of a 3D model that will be generated from the synthesized 2D facial expression. For this purpose, a library of 200 3D heads [11, 12] based on different anthropometric measurements could be used. The 3D heads on the database are categorised by the race, age, gender and the size of the facial characteristics.
More specifically, the final deformed image contains info about the position and shape of the facial characteristics, defined by the landmarks and triangles, along with data about the illumination settings. Having this info as an input, the system could search through the library, utilising an efficient algorithm, in order to identify the appropriate 3D head which better matches the imported face. Continuously the system adjusts the image on the 3D model. The same geometrical deformation of the 2D images must take place also on the 3D model in order to have the new expression fitted on the 3D head with no distortion. The advantage of this process is that it enables the user to have shots of the face from different angles.

6. Conclusion

In this paper, several facial modeling and animation techniques have been presented and comparatively analyzed according to their advantages and disadvantages. The main aim of this paper is to introduce a new method for reproducing natural looking human 2D or 3D facial expressions based on Liu et al. [1] approach, which has been comprehensively presented.
This research intends to develop a novel technique to produce accurate, natural looking facial expressions in a simple and fast way. For this purpose, facial expression mapping techniques have been investigated and used along with human face rendering, as described at the third part of this study. The technique is developed to create effectively 2D or 3D facial expressions, taking into consideration parameters such as distortion, interference, cost, complexity which need to be minimized.
The results presented in the paper are accurately built with high graphic details. The distortion caused either by geometrical deformation or by transferring the illumination settings has been confined.
This method has a great potential of being used since potential applications could be employed by computer game designers and movie animators to quickly generate expressive characters or for low bandwidth telecommunications e.g. videoconferencing.

References

[1] Zicheng Liu, Ying Shan, Zhengyou Zhang, "Expressive expression mapping with ratio images". *In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, New York 2001, pp. 271–276.

[2] F. Parke. "Computer generated animation of faces". *In Proceedings of the ACM Annual Conference*, ACM, Boston, 1972, pp. 451–457.

[3] F.I. Parke.A. "Parametric Model for Human Faces". *PhD Thesis*, University of Utah, Salt Lake City, USA, 1974. UTEC-CSc-75-047

[4] N. Ersotelos and F. Dong. "Building highly realistic facial modeling and animation: a survey". *The Visual Computer: International Journal of Computer Graphics,* Springer-Verlag, New York, November 2007, pp.13 – 30.

[5] Y. Lee, D. Terzopoulos, K. Waters, "Realistic Modeling for Facial Animation". *In Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, New York, 1995.

[6] D. DeCarlo, D. Metaxas, M. Stone, "An Anthropometric Face Model using Variational Techniques". In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, New York, July 1998, pp. 67-74.

[7] F. Pighin, R. Szeliski, D. Salesin, "Resynthesizing facial animation through 3D model-based tracking". *In Proceedings of the 7th IEEE International Conference on Computer Vision*, IEEE Computer Society, Los Alamitos, 1999, pp. 143–150.

[8] V. Blanz, T. Vetter., "A morphable model for the synthesis of 3D faces". *In Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ACM SIGGRAPH, New York 1999, pp. 187–194.
(URL:http://www.kyb.tuebingen.mpg.de/bu/people/volker/)

[9] V. Blanz, K. Scherbaum, T. Vetter, H.-P. Seidel, “Exchanging faces in images”. In Computer Graphics Forum, 2004, **23**(3), pp.669–676

[10] Q. Zhang, Z. Liu, B. Guo, H. Shum, “Geometry-driven photorealistic facial expression synthesis”. *In Proceeding of the ACM symposium on Computer Graphics*, ACM SIGGRAPH, July 2003, pp. 48-60.

[11]http://www.sic.rma.ac.be/~beumier/DB/3d_rma.html

[12]http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb

[13] Cyberware Laboratory, 3D Scanner with Color Digitizer”, Inc, Monterey, California. *4020/RGB.* 1990.

[14] S. R. Marschner and D. P. Greenberg. “Inverse lighting for photography”. *In Proceedings of IS&T/SID Fifth Color Imaging Conference,* Scottsdale, November 1997, pp. 262-265.

[15] P. E. Debevec. “Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography”. *In Computer Graphics, Annual Conference Series*, SIGGRAPH, July 1998, pp. 189–198.

[16] J. Chai, J. Xiao and J. Hodgins, “Vision-based Control of 3D Facial Animation”, *In Eurographics/Siggraph Symposium on Computer Animation*, Eurographics Association, San Diego-California, July 2003, pp. 193-206.

[17] P. Bergeron and P. Lachapelle. “Controlling Facial Expressions and Body Movements in the Computer-Generated Animated Short “Tony De Peltrie””. *In Advanced Computer Animation seminar notes*, SIGGRAPH 85. July 1985.

[18] I. Essa, S. Basu, T. Darrell, and A. Pentland. “Modeling, Tracking and Interactive Animation of Faces and Heads Using Input from Video”. *In Computer Animation Conference*, June 1996, pp. 68–79.

[19] L. Williams. “Performance-Driven Facial Animation”. In *Conference Proceedings*, SIGGRAPH 90, August 1990, v. 24, pp. 235–242.

[20] P. Litwinowicz and L. Williams, “Animating Images with Drawings,” In *Computer Graphics*, Aug. 1990, pp. 235-242.

[21] E. Sifakis, I. Neverov, R. Fedkiw, Automatic Determination of Facial Muscle Activations from Spare Motion Capture marker data, In ACM Transaction of Graphics, July 2005,pp.417-425

Paper 4

Second International Workshop on Semantic Media Adaptation and Personalization

Image-Based Rendering for Computer Synthesized Human Figures

Nikolaos Ersotelos
Department of Information Systems and Computing, Brunel University, UB8 3PH
Nikolaos.Ersotelos@brunel.ac.uk

Abstract

Human figures synthesis has been one of the most complicated problems in computer graphics. In particular, facial animation is the most significant issue. Two decades have passed since Parke's pioneering work in animating faces [1]. In the span of time, significant effort has been devoted to the development of computational models of human face for applications in diverse areas such as entertainment, low bandwidth teleconferencing, surgical facial planning and virtual reality. However, the task of accurately modeling the expressive human face by computer remains a major challenge. The main aspect of this research is to construct a new modeling and rendering technique. There will be brief analysis of IBMR (Image Based Modeling and Rendering) techniques for human face synthesis in computer graphics, research results obtained to date and future aims with discussions. The outcome of this research is to provide a faster, low cost and accurate 3D model that enables realistic facial expressions for animation purposes.

1. Introduction

Given the high interest of computer graphics community in generating human models and facial expressions, human face rendering and animation triggered the interest of research community. Image/video editing in the computer game and movie industry, which involves synthesizing of 3D expressive human views concentrated considerable effort of researches and resulted in presentation of various approaches and techniques.

Early work employed physical and anatomical model based approaches, such as [2]. A large amount of recent work is in favour of a performance driven approach, in which the fidelity of realistic human faces and animation are enhanced by images, videos and motion data. As these captured images and video data are originating from real world, they preserve more naturalism than those techniques derived from purely synthetic approaches.

Image-based modeling (IBM) and rendering (IBR) techniques have recently concentrated research interest due to the enhancements in realism of the resulting 3D models. They make use of images, instead of polygons, as modeling and rendering primitives. IBR allows synthetic views through the combination and interpolation of existing real world images. In this way, it is able to capture subtle real-world effects, imperfections and details that conventional graphic rendering failed to reproduce, eliminating at the same time the demands in labour effort and time. As a result, IBM and IBR covered the need for simpler and accelerated modeling and rendering techniques suitable for representing complex human faces.

Aiming towards new contributions to facial animation by using image based approach, this research will take the challenge of investigating and combining novel image based modeling and rendering techniques with existing Computer Graphics in order to create high quality 3D human face models with fast processing algorithms. Emphasis will be put on naturalism of the resulting 3D models by preserving real-world details.

2. Aim and Objectives

This research aims to a novel approach in creating 3D facial views and animations, having a small set of images as input. This is motivated by the large amount of existing work in facial animation, especially by the recent advance of IBR. The main objective is to implement a fast algorithm capable of creating different views and expressions of the human head, requiring limited interactions from users. Emphasis is put on key techniques such as image based rendering for human faces, human facial expression generator, image colour transfer and human pose estimation techniques.

95-3040-0/07 $25.00 © 2007 IEEE
10.1109/SMAP.2007.47

3. Previous work

A number of approaches have been developed to model and animate realistic 3D facial expressions. Parke and Waters [3] work introduced simple geometric interpolation between face models which had been digitized by hand. A radically different approach is performance based animation, at which measurements from real actors are used to drive synthetic characters [4]. Today, face models can also be obtained using laser based cylindrical scanners, such as those produced by Cyberware. A method based on Cyberware scanner has been presented by Yuencheng Lee and Demetri Terzopoulos [6]. Their algorithm begins with cylindrical range and reflective data acquired by that scanner and automatically constructs an efficient and fully functional model of the subject head. The main advantage of this method is that it overcomes the need for repeated manual modification of control parameters which is necessary in order to compensate for geometric variations of facial features from person to person.

Expression mapping [5] has been a popular method for generating facial animations. This technique was applied at the renewed film "Tony de Peltie" to animate facial expressions of the characters. Given two photographs of the same person, the first with neutral face and the second with facial expression, this method uses points to identify the position of facial features (eyes, eye brows, mouth, etc.) on both photographs. The points can be inserted either manually or automatically. The differentiation between respective points is calculated and applied on the features' position of a new neutral face. This results in a facial expression for that face through geometry controlled image warping.

Another approach, introduced by Zicheng Liu [7] captures the illumination change of one face's expression and maps it on another face, using geometric warping. The illumination change gets captured by what is introduced as the "Expression Ratio Image" (ERI). Even if this method generates more realistic facial expressions since it is able to capture facial details, the difficulty to obtain the ERI can make the method complicated to be applied.

Marschner [8] used the colour ratio between rendered image pairs. By that approach the user takes pictures of an object with different lighting and angles each time. Afterwards, lighting condition of the object can be modified by generating the average colour ratio and applying it to the object. Debevec [9] introduced the color difference approach. The developed algorithm calculates the color difference between the synthesized image pairs and then modifies the original pictures. Similarly with the above technique Jin-xiang Chai [10] implemented a real time facial tracking system which extracts small sets of animation control parameters from a video stream. The introduced algorithm captures motion data from a video input and translates the 2D low quality animation control signals into high quality 3D facial expressions. For the video analysis purposes, a generic cylinder model is used to approximate the head geometry in the monocular video stream.

Another technique introduced by Quingshan Zhang [11] is the geometry driven facial expression synthesis. Given the points of features' positions at a facial expression, the system automatically synthesises the corresponding expression image with photorealistic and natural looking expression details. This technique does not require the collection of a large number of feature points from the face image which had been necessary for the tracking process. The system infers the feature points of motion from the subset of tracked points through an example based approach. Moreover, while the user drags the feature points, the system is able to interactively generate facial expressions with skin deformation details.

Douglas DeCarlo [12] invented a technique based on anthropometric measurements. Anthropometry is a science dedicated to human face and body measurements. The form and values of these measurements derive from facial characteristics such as the position and shape of the mouth, eyes, nose etc. Using these measurements the system can identify the genre, the age and the race of the pictured person based on a statistical database. Continuously, the system synthesizes the best surface, using variational modeling. Variational modeling is a framework for building surfaces by constrained optimization.

Another tool for generating a realistic animated 3D head is the MPEG-4 [13]. The MPEG-4 is a representation of a human facial structure for visual and audio construction and synchronization of a 3D model. To establish accurate graphic results three types of facial data are specified.

Facial Animation Parameters (FAPs) are designed to produce animation of a 3D model reproducing facial movements, expressions and speech pronunciation. The FAP is consisted by 68 parameters.

Facial Definition Parameters (FDPs): allow to change the 3D model settings in order to be used by the receiver by adding either new expression characteristics to a FAP model or by sending a completely new model with the information describing how to perform its animation.

FAP Interpolation Table (FIT): FIT allows interpolation of the FAPs. This facilitates the synthesis

process of a facial animation as it is needed only a small amount of FAP information. MPEG-4 offers a standard to accommodate facial animation. However, it does not support highly realistic and naturally looking image-based facial animation. The current approach is designed to achieve more visual realism.

4. New model synthesis approach and anticipated results

Several approaches have been presented in the past, others with accurate results but with complicated executable process and other based on simple algorithms requiring expensive hardware equipments. The main purpose of this research is to investigate an algorithm which will give a quick, low cost and accurate in high graphic details facial expression synthesis in 2D and in 3D format. The hardware equipment required to implement the algorithm will be only a personal computer. The executable time for generating a 3D facial expression from a 2D source image will be less than two minutes. More specifically, this research intends to construct a novel IBMR technique to display the face of the inserted character at the same pose and facial expression as the replaced character. Then, by identifying and comparing the positions of eyes, noses, mouths, we shall be able to synthesize new facial expression on the inserted image. The novel IBR technique will need only a passport photograph as an input to create the 3D model. In order to achieve these aspects the research will be split in the following phases:

4.1 Facial expression mapping

In the first phase of the research, a number of different potential approaches to generate facial expression have been investigated. Among them this research focuses on the expression mapping methodology which can transfer facial expressions, as described by a few existing images, to a neutral face. In that way a rich set of expressions can be produced while the limitation of morph-based approach which carries out in-between interpolations from a set of captured expressions has been overcome.

The Zicheng Liu [7] methodology has been chosen as the main approach according to which ERI and several other techniques will be combined for more realistic deformed expressions. This algorithm is based on parameterization process (Figure 1) and on the illumination changes which a facial expression can give. By using ERI the user can copy the weight of each pixel of the source picture and adjust accordingly the deformed image. By this process realistic wrinkles and expression details can be achieved on the derived deformed face.

Additionally, other methods will be combined such as colour ratio or anthropometric measurements for the improvement of the parameterization and illumination process.

Figure 1. Dots have been placed around the face, lips, nose, eyes, eye brows, etc to describe the facial features. These dots are then connected by triangles. By moving the triangles new deformed expressions can be generated.

4.2 A novel IBR method for human face rendering.

A library of 200 3D heads [14, 15] based on different anthropometric measurements will be used in the second phase. This database contains a large range of heads categorized by races, ages or gender and will be used to generate the face of the picture on the top of the face of the 3D model. More analytically, this is to develop a novel IBR technique which is capable of rendering the face of the inserted 2D character at the same pose as the replaced 3D character. Given a passport photograph (deformed image) of the inserted character as a sole input, this rendering will choose the face model from the library that fits more accurately the input photo, by using facial recognition methods. By that method the system will be capable to find through the database the closest 3D head according to the facial features.

Continuously, this will be followed by a geometry deformation which modifies the face model to fit the face of the inserted character. Then, by using view dependent texture mapping, we will be able to map the photograph to the deformed model and this textured face model will allow 3D views from different angles.

5. Conclusion

The aim of this paper is to introduce the main issues on which this research has focused. Several image-based modeling and rendering techniques have been presented and compared. This research intends to develop a novel technique associated with two challenging areas in 3D modeling: human facial figures and expressions. For this purpose, facial expression mapping techniques have been investigated and will be used along with human face rendering, as described at the forth part of this study. The anticipated result will be a 3D model based on a 2D image. Therefore, the user of this product will have the opportunity to develop 3D heads with several facial expressions from 2D images as well as the corresponding 3D views from different angles using a cost effective and comparably simple method.

The ambition of this research is to establish a powerful and effective image tool useful for image editing. The results of this algorithm have great potential for further exploitation and development in areas such as entertainment industry or telecommunications. Possible applications could be computer game design or movie animations, where characters could be easily created and adopt natural looking facial expressions. Instead of synthesizing 3D models for each frame this approach can generate fast and accurate models using MPEG-4 tools for speech and visual animation purposes. The capability of displaying talking faces has always been under investigation in the telecommunications' area. The growing demand for video streaming applications such as video-conferencing along with the demand for low-bandwidth transmission makes 3D face synthesis a hot research topic. The expected algorithm could be used in order to automatically generate a talking face at the receiver by synthesizing facial expressions and adjusting them in an appropriate chosen 3D model. Since only a small amount of data would be sent e.g. a 2D photo and some parameters, the requirements for bandwidth would be decreased.

References

[1] F. Parke. "Computer generated animation of faces", *Proceedings of the ACM annual conference,* ACM, New York, August 1972, pp. 451–457.

[2] K. Waters. "A muscle model for animating three-dimensional facial expressions", *Proceedings of the 14th annual conference on Computer graphics and interactive techniques,* ACM SIGGRAPH Computer Graphics, New York, July 1987, pp. 17-24.

[3] Frederic I. Parke and Keith Waters. *Computer Facial Animation*, AK Peters, Wellesley, Massachusetts, 1996.

[4]Irfan Essa, Sumit Basu, Trevor Darrell, and Alex Pentland. "Modeling, Tracking and Interactive Animation of Faces and Heads Using Input from Video". *Proceedings In Computer Animation Conference*, IEEE Computer Society Washington, June 1996, pp. 68.

[6] Y. Lee, D. Terzopoulos and K. Waters, "Realistic Modeling for Facial Animation", *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH, Los Angeles, August 1995, pp.55-61.

[7] Zicheng Liu, Ying Shan and Zhengyou Zhang, "Expressive Expression Mapping with Ratio Images" , *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH, New York, August 2001, pp. 271-276.

[8] S. R. Marschner and D. P. Greenberg. "Inverse lighting for photography". *In Proceedings of IS&T/SID Fifth Color Imaging Conference,* Scottsdale, November 1997, pp. 262-265.

[9] P. E. Debevec. "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography". In *Computer Graphics, Annual Conference Series*, SIGGRAPH, July 1998, pp. 189–198.

[10] J. Chai, J. Xiao and J. Hodgins, "Vision-based Control of 3D Facial Animation", *Eurographics/Siggraph Symposium on Computer Animation*, Eurographics Association, San Diego- California, July 2003, pp. 193-206.

[11] Qingshan Zhang, Z.Liu, B.Guo and H. Shum, "Geometry-Driven Photorealistic Facial Expression Synthesis", *IEEE Transactions on Visualization and Computer Graphics*, IEEE, Piscataway, January 2003, pp. 48-60.

[12] D. DeCarlo, D. Metaxas and M. Stone, "An Anthropometric Face Model using Variational Techniques", *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM press, New York, July 1998, pp. 67-74.

[13] G. A. Abrantes and F. Pereira, "MPEG-4 Facial Animation Tehnology: Survey, Implementation, and Results", *IEEE Trans. on Circuits and Systems for Video Technology*, IEEE vol. 9 no.2, 1999, pp. 290-305

[14] http://www.sic.rma.ac.be/~beumier/DB/3d_rma.html

[15] http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb

Paper 5

Incremental Synthesis for Generating Large Texture Images

Nikolaos Ersotelos, Feng Dong

Department of Information Systems and Computing, Brunel University, UB8 3PH, United Kingdom
Nikolaos.Ersotelos@brunel.ac.uk

Abstract. This paper presents a fast method for synthesizing large textures from small samples using incremental texture patches and samples. In contrast to the existing work in the area, it offers fast speed and high image qualities through efficient computations. The heart of the proposed method is a multi-level synthesis framework which supports the increase of texture patches and samples. By applying the method to a wide range of texture samples, we have demonstrated the effectiveness of the method through the results and discussions.

1. Introduction

Textures have proved to be the most effective approach to enhance surface appearance. In the past decade several approaches have been established providing highly detailed textures in Computer Graphics. Remarkably, a large amount of recent research attention has been focusing on texture synthesis, which generates new texture image bearing the same visual characteristics as a given texture sample.

The recent success in texture synthesis has dramatically improved the performance of texture technology in terms of reducing algorithm complexity and widening applicability. However, despite these great efforts and progresses, the current available texture synthesis techniques are still facing critical challenges particularly in processing speed. Considerable processing time is required even for generating medium sized results. For example, in some of the latest works, Kwatra et al [Kwatra05] required 1-3 minutes to generate a 256x256 texture; Wu & Yu [Wu04] needed around two minutes to generate a 256x256 texture from a sample at 128x128; The LILIES image in [Kwatra03] took about 5 minutes, etc.

This paper is to present a method for synthesizing large textures. Our aim is at textures whose sizes are at least over 1024x1024. For the large textures, processing speed is particularly important as obviously they involve more computation. In addition, we shall also pay our attentions to the result qualities since generally larger result implies higher chances to generating mismatch between neighbouring patches and consequently brings defects into the results.

The main contribution of this paper is to present a methodology to perform the synthesis within a novel multi-level framework, within which we increase both the size of sample texture image and texture patches. Our research has demonstrated that using larger sample texture images and texture patches improves both synthesis speed and result quality.

This method is particularly beneficial to applications within which large texture synthesis is required. Our results have shown that the proposed method is capable of generating a 1024x1024 texture with good quality in just over 30 seconds. To our knowledge, this is one of the fastest methods in the area.

The rest of this paper is organised as follows: Section 2 gives an overview of previous work; Section 3 presents the motivation of the our methodology; Section 4 provides a

detailed description on the proposed method; Finally, results and comparisons with previous works are given in Section 5, and the conclusion is drawn in Section 6.

1.1 Previous work

The early 2D texture synthesis methods compute global statistics from sample textures and then generate new texture images that possess the same statistics [DeBonet97, Heeger95, Portilla00, Zhu98]. However, recent papers have suggested that enforcing local statistics is a more sensible approach and gives better results. These methods can be either *pixel-based* [Ashikhmin01, Hertzmann01, Wei00], which generate a synthesised image pixel by pixel, or *patchbased* [Efros01, Liang01, Xu00, Kwatra03, Wu04], which make a new texture image by taking patches from the sample texture and pasting them together in a consistent way.

The pixel-based methods offer control over the texture properties at a pixel level, e.g. each pixel can be given different texture orientations, but have a critical weakness – errors from previously synthesised pixels can percolate into the rest of the results and hence cause significant defects.

A patch-based method generally provides better results, since the texture structures inside the pasted texture patches are maintained. However, it required a lot of work to ensure that all the patches are compatible with their neighbours. Some recent solutions that employ irregularly-shaped texture patches [Praun00, Dischler02, Kwatra03] have considerably improved the output quality.

Also, some recent work has used the idea of tiling for fast on-line texture synthesis [Cohen03]; or near-regular texture analysis and manipulation [Liu04]; or texture optimization using Markov Random Field-based similarity metric[Kwatra05]; or parallel controllable texture synthesis based on neighborhood matching[Lefebvre05].

Motivation

Our target in this research is to generate large textures within good timing. To achieve this, we propose a multi-level framework to speed up the synthesis process without losing quality. The proposed method is based on patch-based texture synthesis, considering the general high qualities from patch-based methods.

Our motivation is based on the following observations on patch size used in a general patch based texture synthesis method:

1. Patch size, which can be a parameter decided by user input, has large influence on the results in terms of quality and speed. For the speed issue, larger patch size gives higher processing speed.
2. Since normally the size of input texture sample is small, we can not use large sized texture patches. Otherwise, it would only allow small number of candidate patches available and hence generate visible repetitions in the results – see more detailed explanations in Section 4.

Given these observations, with conventional patch based texture synthesis methods, the result texture contains a large number of small texture patches selected from the input texture sample. This not only requires large amount of processing time to match these patches, but also increases the risk of generating mismatch between the patches.

The general methodology of this paper is a multi-level based framework, in which both sample texture and patch size increase during the synthesis process. The increased texture samples come from the intermediately synthesized texture images within the multilevel framework. Because of the increase of the patch sizes, our final result consists of large texture patches chosen from the increased texture samples. In addition, the computation overhead due to the increase of the texture samples is

compensated by the use of the larger texture patches. This brings fast synthesis speed and good quality.

Texture Synthesis using Incremental Patches

In this section, we will start with an analysis on the computation required for a general patch based texture synthesis. This is followed by the description of our texture synthesis using incremental texture patches, which offers fast performance as well as good image qualities.

Computation Analysis

Generating a large texture image using fixed patch size is extremely time-consuming. Here we will present the timing problem by conducting the following quantitative analysis on the processing time for patch based texture synthesis. Without losing generality, we will use texture synthesis via rectangular texture patches, such as in [Efros01], as an example. Similar problems also occur in other patch based methods.

In fact, the processing time for a patch based texture synthesis relies heavily on the following three parameters:

1. The size of the input texture sample image;
2. The size of the adopted texture patches;
3. The size of the output image.

These three sizes decide how much computation is required to perform a texture synthesis. Given an input texture sample, the bottle neck of the computation is to scan the texture sample in order to identify texture patches which are eligible for "stitching" together.

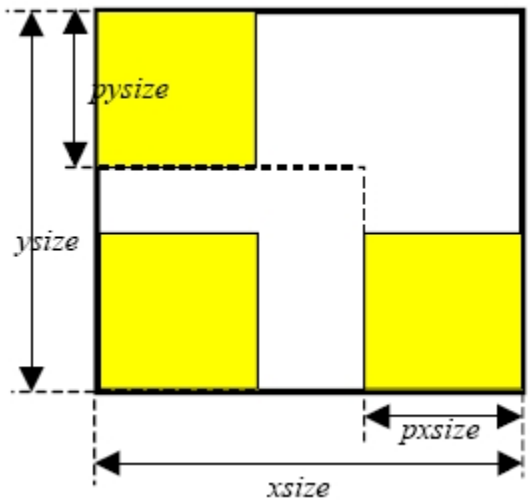


Figure 1 Candidate patches in a texture sample Image: if *xsize* x *ysize* is the size of the texture sample, *pxsize* x *pysize* is the size of the patch, then *(xsize-pxsize)* x *(ysize-pysize)* gives the number of all possible patches.

Increasing the above first and third parameters increases the amount of the scans while increasing the second parameter decreases the scan times involved. Obviously, larger amount of scans implies more computation.

If we assume the sample image size is 128x128, and we use texture patches sized at 32x32, then to generate a result 1024x1024 texture image, we need approximately, (1024/32)2=1024 patches. This implies scanning the sample texture image for 1024 times. Each time of the scan approximately involves checking (128-32)2 = 9216 candidate patches – see Figure 1 for more detailed explanations. Therefore, in total,

we need to carry out patch similarity checking for 1024x9216 > 9 million times. Bear in mind that this number is going up dramatically either we increase the input sample image or the output image. Due to the heavy computation, typically considerable long time is required to generate a texture image of 1024x1024. And this is subject to further increase for larger output images.
Given the above analysis, to bring the large texture synthesis to a reasonable speed, we have to greatly drop the number of the scans involved in the patch based texture synthesis. To this end, a straightforward solution is to increase the patch size, e.g. to 64x64, to reduce the amount of the patch scanning in the texture sample image.
However, using larger texture patches without increasing texture sample size leads to the heavy reduce of candidate patches. For example, if the patch size is 64x64, the available candidate patches become (128- 64)2=4096, which is half of the number for 32x32 – see Figure 1 for more detailed explanations. This creates a risk of generating result image with clearly visible repetitions and consequently affects the quality of the result.
Visible repetitions can be avoided by using larger input texture sample to increase of the number of candidate patches. However, in a typical texture synthesis problem, the input texture sample is fixed.

Our Solution: Multi-level Synthesis
To address the challenges mentioned above, our solution offers the great elimination of the processing time while maintain the quality. Our general methodology is to make use of incremental texture patches to bring fast performance and incremental texture sample image source to avoid visible repetition

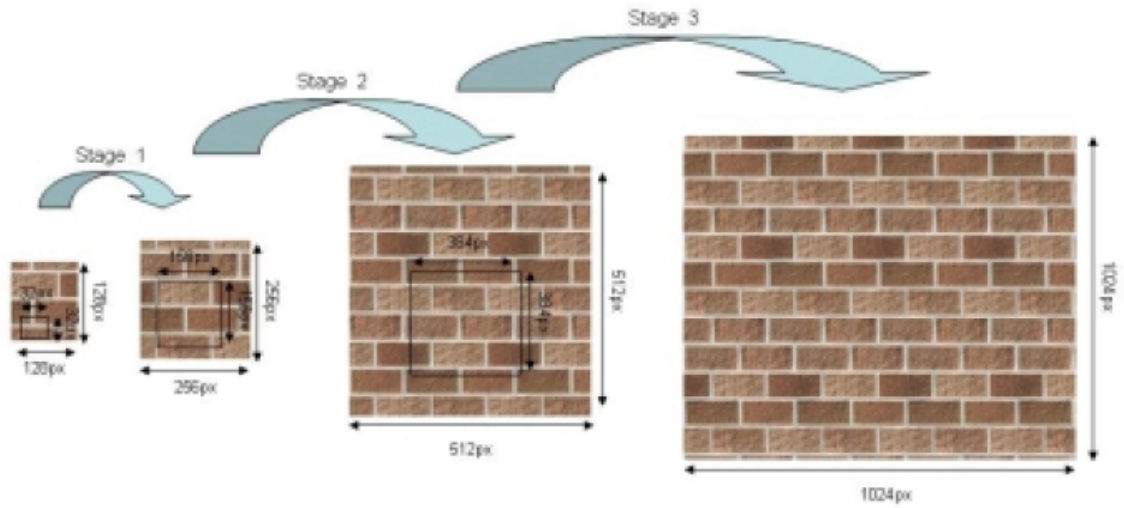


Figure 2 An example of generating large synthesized image in a multi-level framework problem mentioned in Section 4.1.
More specifically, we propose a multi-level framework to perform the synthesis. It works as follows:

- At the bottom level, patch based texture synthesis is carried out from the given input texture sample.
- Then the synthesized result, which presumably sizes as *d size*, is used as a texture sample for the next level.
- From the texture sample at this level, synthesize a new texture image which sizes as 2x*dsize*, using largely sized texture patches.
- The synthesized result texture is again then passed onto the next level and the above process is repeated until the size of final result is reached.

As illustrated above, the main idea of the proposed multi-level framework is that, instead of generating result in one go as in previous methods such as in [Efros01], we use multiple levels and at each level and the synthesized result is passed to the next level as the input sample texture image. Consequently, the double-sized synthesis result at each level also doubles the sample texture image for the next level. Meanwhile, to overcome the computation overheads brought by the increase of the sample texture image and achieve the same processing time as in the previous level, we have to increase the size of the texture patches in order to keep the same number of the patches as used in the previous level.

As an example, we assume the original input texture image is 128x128, and we set the result texture at the bottom level at 256x256 synthesized using patches sized 32x32. This takes approximately 8x8 patches (and hence 8x8 scans) to make the texture. Then, in the next level, the texture sample is sized 256x256, and we also increase the size of the patch to 156x156. Again this approximately takes the same amount of scans (and hence the same processing time) as in the previous level to make a new 512x512 texture. This continues to the next levels until we reach the desired output size. For a 1024x1024 output, three levels are required – See Figure 2 for the illustration.

Compared with the standard patch-based approach, this method takes less amount of scanning and patch similarity check and therefore saves a lot of computation.

In addition, this multi-level framework also bears positive effects on the synthesized quality. Due to the increase of the path sizes, we have only a small number of patches in the final result. This reduces the risk of generating mismatches between large number of small patches in the result. Therefore, despite the fact that we are targeting large textures, as we will show in the next Section, our results do not bear large amount of defects.

Results and Discussions

We have implemented the algorithm on a machine with Intel Pentium 4 3.4GHz, 1GB DDR SDRAM. Some of our results have been presented in Figure 3.

For most of the images that we have experimented, it generally took between 30 – 60 seconds to generate a 1024x1024 texture image with considerable good qualities. Compared to the timing that we have previously mentioned in [Kwatra03, Kwatra05, Wu04], this appears to be significantly fast and hence a good advance.

Figure 3 presents some of our results. The large images are from our results and the small images are samples.

Figure 4 provides some results from related work for comparison, including those [Efros01], [Kwatra03] and [Kwatra05]. From Figure 4 we can see that our results are comparable or even better than others in terms of the quality.

As we have demonstrated in these results, the proposed method is generally faster than the existing methods. Also, due to the use of the multi-level framework, which allows the increase of the texture patches and samples, the quality of our large results (1024x1024) are comparable with the smaller results from others.

Although the algorithm appears to work well in a wide range of textures, the drawback is that it is not straightforward to make it parallel due to the multi-level framework involved. The images in the multi-level framework have to be synthesized in an order. Therefore, the speed we have achieved is not as fast as those from parallel synthesis [Lefebvre05].

Conclusions and Future Work

We have presented a novel patch-based method for synthesising large texture using incremental texture patches and samples. The heart of the method is a multi-level synthesis framework which allows the increase of texture patches and samples. By using incremental texture patches and samples, we have achieved better results both in terms of processing speed and quality.

In future, we can extend this work to controllable texture synthesis, which allows more synthesis controls from users. In addition, we will also investigate texture synthesis from multiple texture sources.

Acknowledgement

This work was supported by grant no EP/C006623/1 from the Engineering and Physical Sciences Research Council of the UK.

References

[Ashikhmin01] Ashikhmin, M. 2001. Synthesizing natural textures. Proceedings of the ACM Symposium on Interactive 3D Graphics, 2001: 217-226.

[Cohen03] Cohen, M. F., Shade, J., Hiller, S., Deussen, O., 2003: Wang Tiles for Image and Texture Generation. ACM Trans on Graphics, 22(3): 287-294

[DeBonet97] De Bonet, J. S., 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. Proc SIGGRAPH 97, 361-368.

[Dischler02] Dischler, J. M., Maritaud, K., Levy, B., and Ghazanfarpour, D., 2002: Texture Particles. Computer Graphics Forum, 21(3): 401-410

[Efros01] Efros, A. A., and Freeman, W. T., 2001, Image quilting for texture synthesis and transfer. Proc. SIGGRAPH 01: 341-346.

[Heeger95] Heeger, D. J., and Bergen, J. R., 1995 Pyramid-based texture analysis/synthesis. Proc. SIGGRAPH 95, 229- 238.

[Hertzmann01] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., Salesin, D., 2001. Image Analogies. Proc. SIGGRAPH 01, 327- 340.

[Kwatra03] Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A., 2003. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. ACM Trans on Graphics, 22(3): 277-286.

[Kwatra05] Kwatra, V., Essa, I., Bobick, A., Kwatra, N., 2005. Texture optimization for example-based synthesis. ACM Trans on Graphics, 24(3): 795-802.

[Lefebvre05] Lefebvre, S., Hoppe, H., 2005. Parallel controllable texture synthesis. ACM Trans on Graphics, 24(3): 777-786.

[Liang01] Liang, L., Liu, C., Xu, Y, Guo, B., Shum, H., 2001. Real-time texture synthesis by patch-based sampling. ACM Trans. On graphics, Vol.20(3): 127-150.

[Liu04] Liu, Y., Lin, W., Hays, J., 2004. Near regular texture analysis and manipulation. ACM Trans on Graphics, 23(3): 368 -376

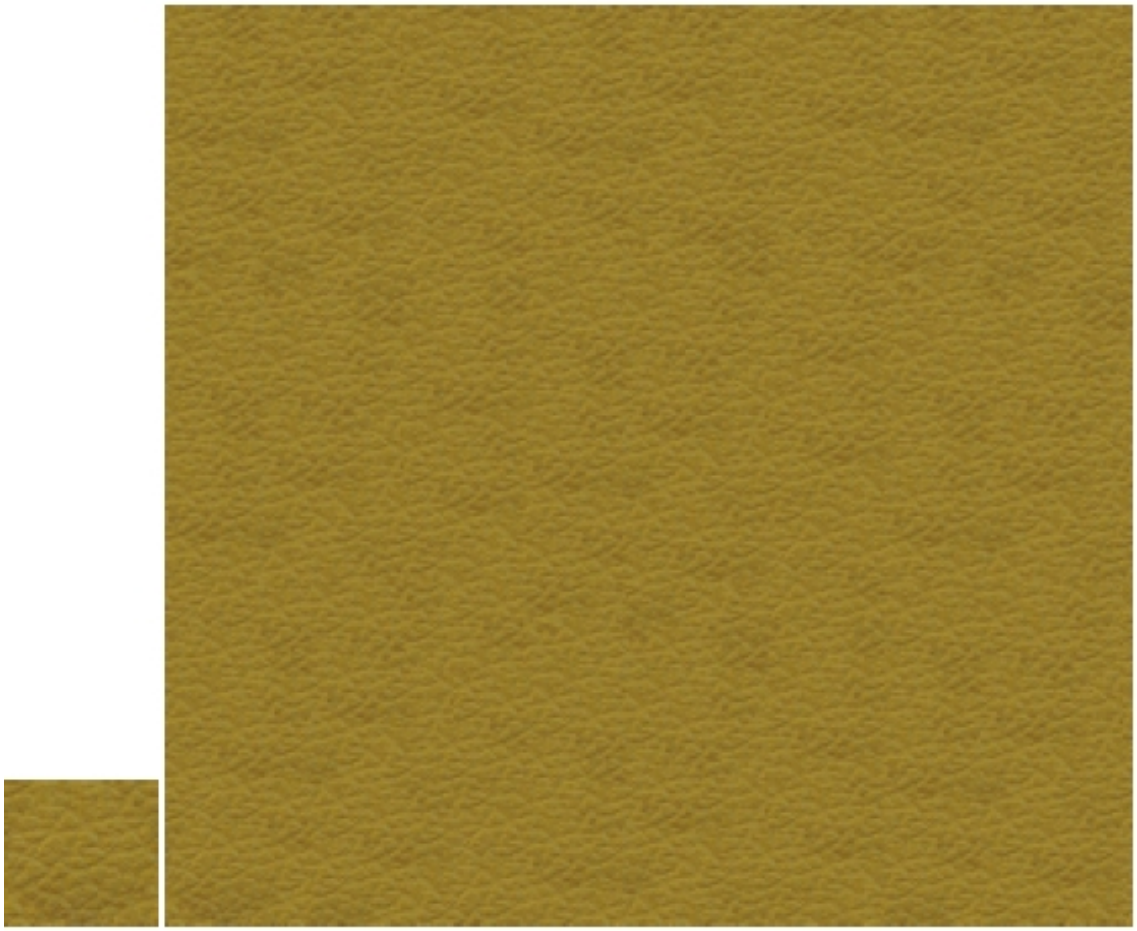
[Portilla00] Portilla, J., and Simoncelli, E. P., 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision, Vol.40(1), 49-71.

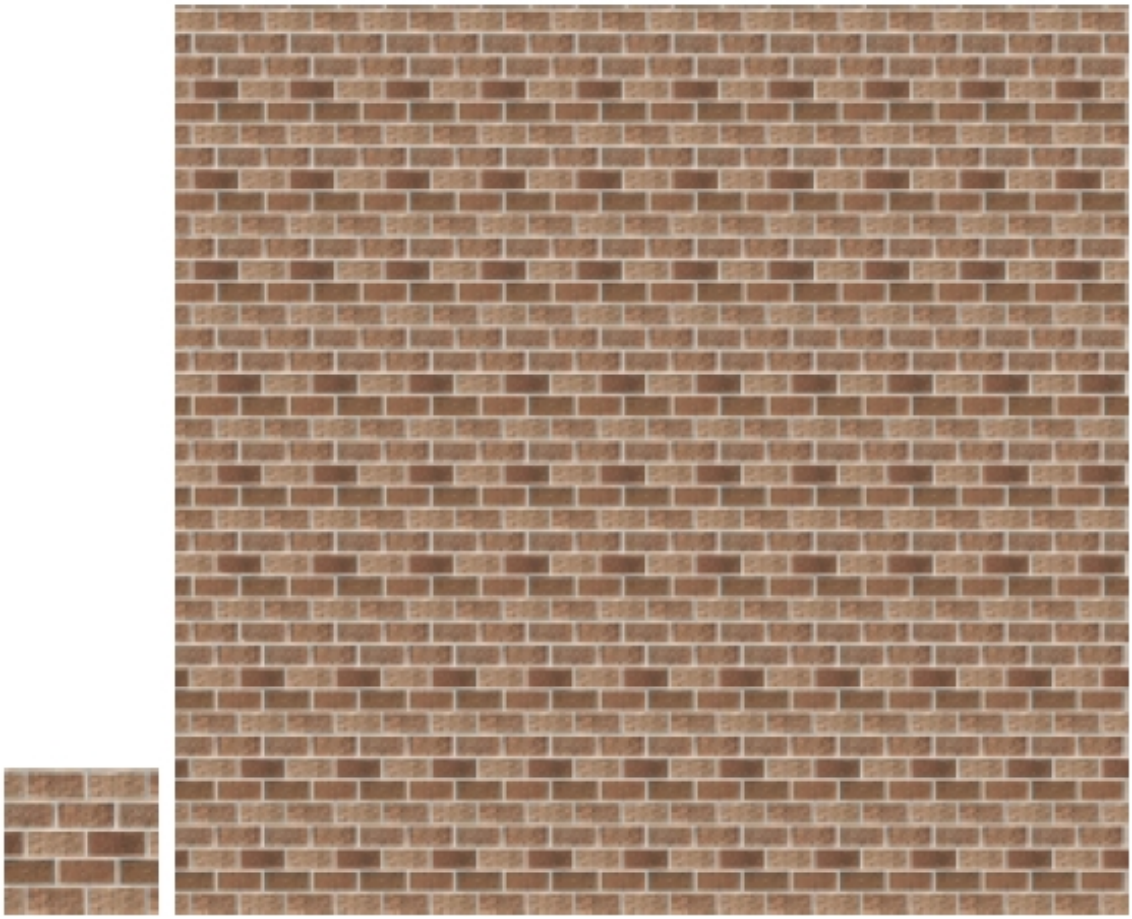
[Praun00] Praun, E., Finkelstein, A., Hoppe, H., 2000. Lapped Textures. Proc. SIGGRAPH 00, 2000: 465-470.

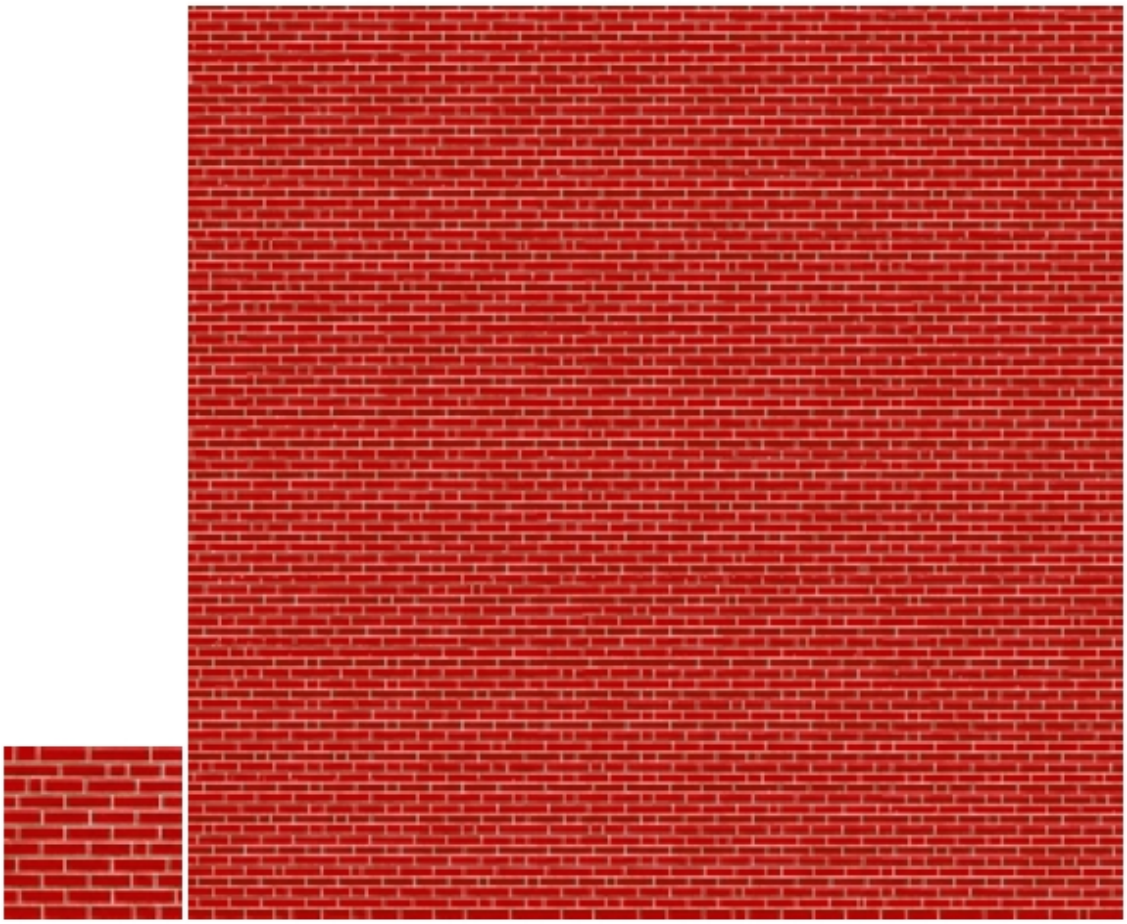
[Wei00] Wei, L. Y., and Levoy, M., 2000. Fast texture synthesis using tree-structured vector quantization. Proc. SIGGRAPH 00: 479-488.

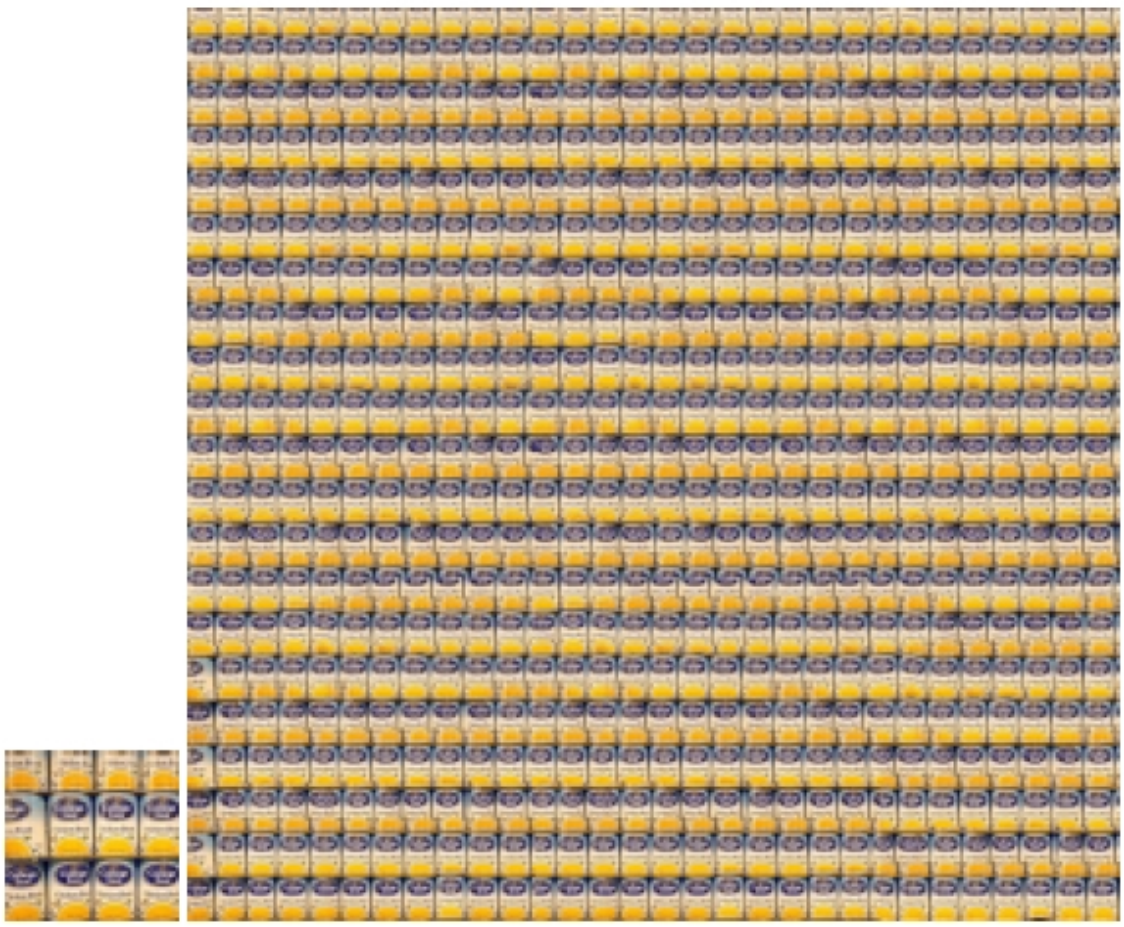
[Wu04] Wu. Q., Yu, Y., 2004. Feature Matching and deformation for texture synthesis. ACM Trans on Graphics, 23(3): 364 - 367

[Xu00] Xu, Y., Guo, B., and Shum, H., 2000. Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR- 2000-32, Microsoft Research, April 2000.
[Zhu98] Zhu, S. C., Wu, Y., and Mumford, D., 1998. Filters, random fields and maximum entropy (frame). International Journal of Computer Vision, 27(2): 1-20.









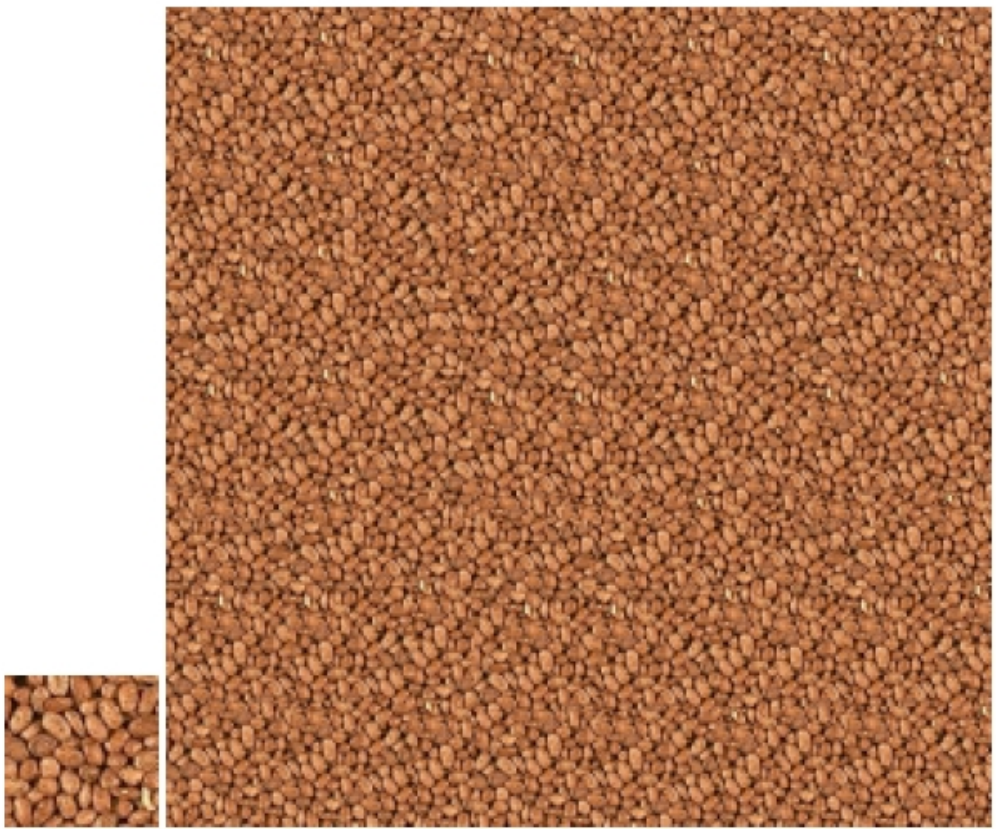
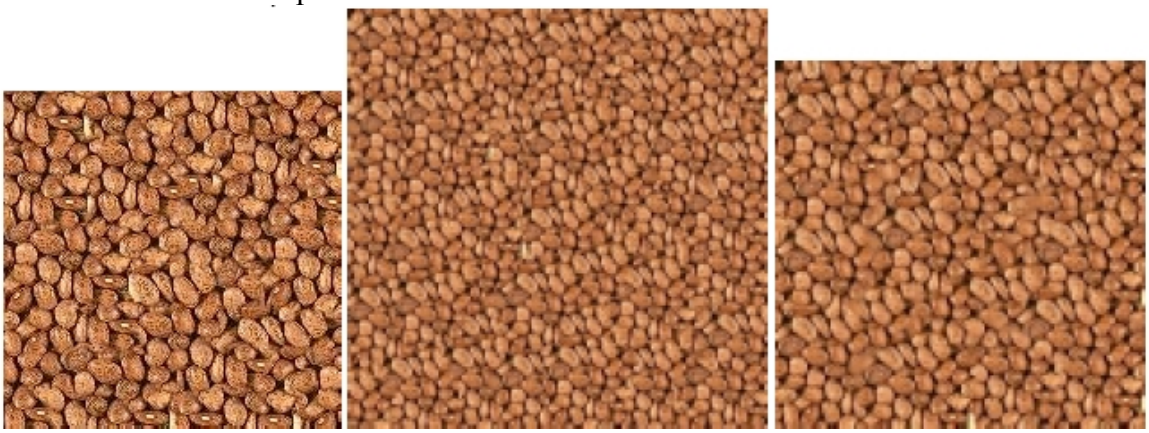


Figure 3 Patch Based Texture Synthesis. Process average time for each result 40 – 45 seconds for 1024x1024px



[Efros01] [Kwatra03] [Kwatra05]

[Efros01] [Kwatra03]

Figure 4 Results from related work for comparisons

Paper 6

Incremental Synthesis for Generating Large Texture Images

Nikolaos Ersotelos*, Feng Dong

Department of Information Systems and Computing, Brunel University, UK, UB8 3PH

Abstract

This paper presents a fast method for synthesizing large textures from small samples using incremental texture patches and samples. In contrast to the existing work in the area, it offers fast speed and high image qualities through efficient computations. The heart of the proposed method is a multi-level synthesis framework which supports the increase of texture patches and samples. By applying the method to a wide range of texture samples, we have demonstrated the effectiveness of the method through the results and discussions.

Keywords: Texture Synthesis, Patch based Methods, Incremental Patches

1. Introduction

Textures have proved to be the most effective approach to enhance surface appearance. In the past decade several approaches have been established providing highly detailed textures in Computer Graphics. Remarkably, a large amount of recent research attention has been focusing on texture synthesis, which generates new texture image bearing the same visual characteristics as a given texture sample.

The recent success in texture synthesis has dramatically improved the performance of texture technology in terms of reducing algorithm complexity and widening applicability. However, despite these great efforts and progresses, the current available texture synthesis techniques are still facing critical challenges particularly in processing speed. Considerable processing time is required even for generating medium sized results. For example, in some of the latest works, Kwatra et al [Kwatra05] required 1-3 minutes to generate a 256x256 texture; Wu & Yu [Wu04] needed around two minutes to generate a 256x256 texture from a sample at 128x128; The LILIES image in [Kwatra03] took about 5 minutes, etc.

This paper is to present a method for synthesizing large textures. Our aim is at textures whose sizes are at least over 1024x1024. For the large textures, processing speed is particularly important as obviously they involve more computation. In addition, we shall also pay our attentions to the result qualities since generally larger result implies higher chances to generating mismatch between neighbouring patches and consequently brings defects into the results.

The main contribution of this paper is to present a methodology to perform the synthesis within a novel multi-level framework, within which we increase both the size of sample texture image and texture patches. Our research has demonstrated that using larger sample texture images and texture patches improves both synthesis speed and result quality.

This method is particularly beneficial to applications within which large texture synthesis is required. Our results have shown that the proposed method is capable of generating a 1024x1024 texture with good quality in just over 30 seconds. To our knowledge, this is one of the fastest methods in the area.

The rest of this paper is organised as follows: Section 2 gives an overview of previous work; Section 3 presents the motivation of the our methodology; Section 4 provides a detailed description on the proposed method; Finally, results and comparisons with previous works are given in Section 5, and the conclusion is drawn in Section 6.

2. Previous work

The early 2D texture synthesis methods compute global statistics from sample textures and then generate new texture images that possess the same statistics [DeBonet97,

* Corresponding author:
Nikolaos.Ersotelos@brunel.ac.uk

Heeger95, Portilla00, Zhu98]. However, recent papers have suggested that enforcing local statistics is a more sensible approach and gives better results. These methods can be either pixel-based [Ashikhmin01, Hertzmann01, Wei00], which generate a synthesised image pixel by pixel, or patch-based [Efros01, Liang01, Xu00, Kwatra03, Wu04], which make a new texture image by taking patches from the sample texture and pasting them together in a consistent way.

The pixel-based methods offer control over the texture properties at a pixel level, e.g. each pixel can be given different texture orientations, but have a critical weakness – errors from previously synthesised pixels can percolate into the rest of the results and hence cause significant defects.

A patch-based method generally provides better results, since the texture structures inside the pasted texture patches are maintained. However, it required a lot of work to ensure that all the patches are compatible with their neighbours. Some recent solutions that employ irregularly-shaped texture patches [Praun00, Dischler02, Kwatra03] have considerably improved the output quality.

Also, some recent work has used the idea of tiling for fast on-line texture synthesis [Cohen03]; or near-regular texture analysis and manipulation [Liu04]; or texture optimisation using Markov Random Field-based similarity metric[Kwatra05]; or parallel controllable texture synthesis based on neighbourhood matching[Lefebvre05].

3. Motivation

Our target in this research is to generate large textures within good timing. To achieve this, we propose a multi-level framework to speed up the synthesis process without losing quality. The proposed method is based on patch-based texture synthesis, considering the general high qualities from patch-based methods.

Our motivation is based on the following observations on patch size used in a general patch based texture synthesis method:

1) Patch size, which can be a parameter decided by user input, has large influence on the results in terms of quality and speed. For the speed issue, larger patch size gives higher processing speed.
2) Since normally the size of input texture sample is small, we can not use large sized texture patches. Otherwise, it would only allow small number of candidate patches available and hence generate visible repetitions in the results – see more detailed explanations in Section 4.

Given these observations, with conventional patch based texture synthesis methods, the result texture contains a large number of small texture patches selected from the input texture sample. This not only requires large amount of processing time to match these patches, but also increases the risk of generating mismatch between the patches.

The general methodology of this paper is a multi-level based framework, in which both sample texture and patch size increase during the synthesis process. The increased texture samples come from the intermediately synthesized texture images within the multi-level framework. Because of the increase of the patch sizes, our final result consists of large texture patches chosen from the increased texture samples. In addition, the computation overhead due to the increase of the texture samples is compensated by the use of the larger texture patches. This brings fast synthesis speed and good quality.

4. Texture Synthesis using Incremental Patches

In this section, we will start with an analysis on the computation required for a general patch based texture synthesis. This is followed by the description of our texture synthesis using incremental texture patches, which offers fast performance as well as good image qualities.

4.1 Computation Analysis

Generating a large texture image using fixed patch size is extremely time-consuming. Here we will present the timing problem by conducting the following quantitative analysis on the processing time for patch based texture synthesis. Without losing generality, we will use texture synthesis via rectangular texture patches, such as in [Efros01], as an example. Similar problems also occur in other patch-based methods.

In fact, the processing time for a patch based texture synthesis relies heavily on the following three parameters:

1) The size of the input texture sample image;
2) The size of the adopted texture patches;
3) The size of the output image.

These three sizes decide how much computation is required to perform a texture synthesis. Given an input texture sample, the

bottle neck of the computation is to scan the texture sample in order to identify texture patches which are eligible for "stitching" together.

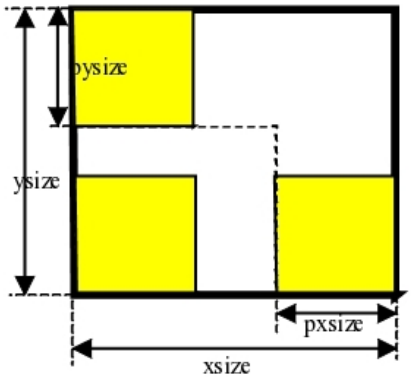


Figure 1 Candidate patches in a texture sample Image: if xsize x ysize is the size of the texture sample, pxsize x pysize is the size of the patch, then (xsize-pxsize) x (ysize-pysize) gives the number of all possible patches.

Increasing the above first and third parameters increases the amount of the scans while increasing the second parameter decreases the scan times involved. Obviously, larger amount of scans implies more computation.

If we assume the sample image size is 128x128, and we use texture patches sized at 32x32, then to generate a result 1024x1024 texture image, we need approximately, $(1024/32)^2$=1024 patches. This implies scanning the sample texture image for 1024 times. Each time of the scan approximately involves checking $(128\text{-}32)^2$ = 9216 candidate patches – see Figure 1 for more detailed explanations. Therefore, in total, we need to carry out patch similarity checking for 1024x9216 > 9 million times. Bear in mind that this number is going up dramatically either we increase the input sample image or the output image. Due to the heavy computation, typically considerable long time is required to generate a texture image of 1024x1024. And this is subject to further increase for larger output images.

Given the above analysis, to bring the large texture synthesis to a reasonable speed, we have to greatly drop the number of the scans involved in the patch based texture synthesis. To this end, a straightforward solution is to increase the patch size, e.g. to 64x64, to reduce the amount of the patch scanning in the texture sample image.

However, using larger texture patches without increasing texture sample size leads to the heavy reduce of candidate patches. For example, if the patch size is 64x64, the available candidate patches become $(128\text{-}64)^2$=4096, which is half of the number for 32x32 – see Figure 1 for more detailed explanations. This creates a risk of generating result image with clearly visible repetitions and consequently affects the quality of the result.

Visible repetitions can be avoided by using larger input texture sample to increase of the number of candidate patches. However, in a typical texture synthesis problem, the input texture sample is fixed.

4.2 Our Solution: Multi-level Synthesis

To address the challenges mentioned above, our solution offers the great elimination of the processing time while maintain the quality. Our general methodology is to make use of incremental texture patches to bring fast performance and incremental texture sample image source to avoid visible repetition

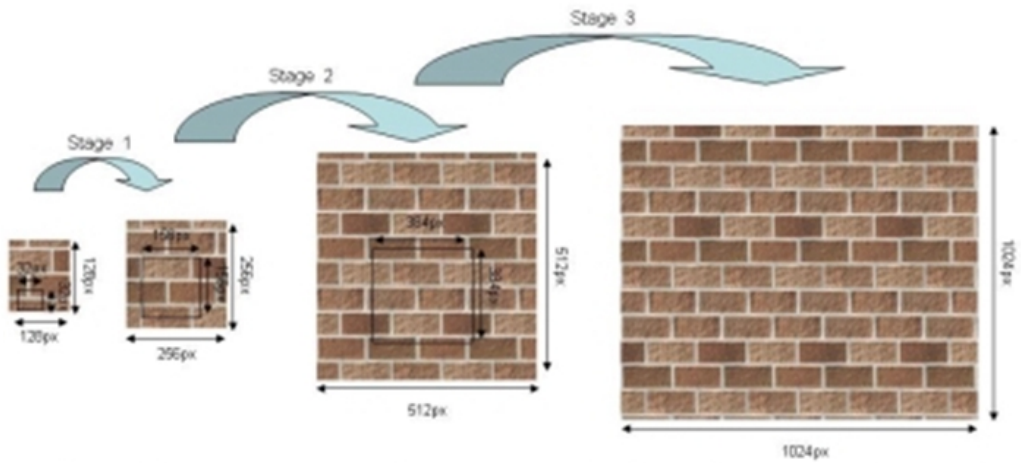


Figure 2 An example of generating large synthesized image in a multi-level framework

problem mentioned in Section 4.1.

More specifically, we propose a multi-level framework to perform the synthesis. It works as follows:

- At the bottom level, patch based texture synthesis is carried out from the given input texture sample.
- Then the synthesized result, which presumably sizes as dsize, is used as a texture sample for the next level.
- From the texture sample at this level, synthesize a new texture image which sizes as 2xdsize, using largely sized texture patches.
- The synthesized result texture is again then passed onto the next level and the above process is repeated until the size of final result is reached.

As illustrated above, the main idea of the proposed multi-level framework is that, instead of generating result in one go as in previous methods such as in [Efros01], we use multiple levels and at each level and the synthesized result is passed to the next level as the input sample texture image. Consequently, the double-sized synthesis result at each level also doubles the sample texture image for the next level. Meanwhile, to overcome the computation overheads brought by the increase of the sample texture image and achieve the same processing time as in the previous level, we have to increase the size of the texture patches in order to keep the same number of the patches as used in the previous level.

As an example, we assume the original input texture image is 128x128, and we set the result texture at the bottom level at 256x256 synthesized using patches sized 32x32. This takes approximately 8x8 patches (and hence 8x8 scans) to make the texture. Then, in the next level, the texture sample is sized 256x256, and we also increase the size of the patch to 156x156. Again this approximately takes the same amount of scans (and hence the same processing time) as in the previous level to make a new 512x512 texture. This continues to the next levels until we reach the desired output size. For a 1024x1024 output, three levels are required – See Figure 2 for the illustration.

Compared with the standard patch-based approach, this method takes less amount of scanning and patch similarity check and therefore saves a lot of computation.

In addition, this multi-level framework also bears positive effects on the synthesized quality. Due to the increase of the path sizes, we have only a small number of patches in the final result. This reduces the risk of generating mismatches between large number of small patches in the result. Therefore, despite the fact that we are targeting large textures, as we will show in the next Section, our results do not bear large amount of defects.

5. Results and Discussions

We have implemented the algorithm on a machine with Intel Pentium 4 3.4GHz, 1GB DDR SDRAM. Some of our results have been presented in Figure 3.

For most of the images that we have experimented, it generally took between 30 – 60 seconds to generate a 1024x1024 texture image with considerable good qualities. Compared to the timing that we have previously mentioned in [Kwatra03, Kwatra05, Wu04], this appears to be significantly fast and hence a good advance.

Figure 3 presents some of our results. The large images are from our results and the small images are samples.

Figure 4 provides some results from related work for comparison, including those [Efros01], [Kwatra03] and [Kwatra05]. From Figure 4 we can see that our results are comparable or even better than others in terms of the quality.

As we have demonstrated in these results, the proposed method is generally faster than the existing methods. Also, due to the use of the multi-level framework, which allows the increase of the texture patches and samples, the quality of our large results (1024x1024) are comparable with the smaller results from others.

Although the algorithm appears to work well in a wide range of textures, the drawback is that it is not straightforward to make it parallel due to the multi-level framework involved. The images in the multi-level framework have to be synthesized in an order. Therefore, the speed we have achieved is not as fast as those from parallel synthesis[Lefebvre05].

6. Conclusions and Future Work

We have presented a novel patch-based method for synthesising large texture using incremental texture patches and samples. The heart of the method is a multi-level synthesis framework which allows the increase of texture patches and samples. By using incremental

texture patches and samples, we have achieved better results both in terms of processing speed and quality.

In future, we can extend this work to controllable texture synthesis, which allows more synthesis controls from users. In addition, we will also investigate texture synthesis from multiple texture sources.

7. Acknowledgement

This work was supported by grant no. EP/C006623/1 from the Engineering and Physical Sciences Research Council of the UK.

References

[Ashikhmin01] Ashikhmin, M. 2001. Synthesizing natural textures. Proceedings of the ACM Symposium on Interactive 3D Graphics, 2001: 217-226.

[Cohen03] Cohen, M. F., Shade, J., Hiller, S., Deussen, O., 2003: Wang Tiles for Image and Texture Generation. ACM Trans on Graphics, 22(3): 287-294

[DeBonet97] De Bonet, J. S., 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. Proc SIGGRAPH 97, 361-368.

[Dischler02] Dischler, J. M., Maritaud, K., Levy, B., and Ghazanfarpour, D., 2002: Texture Particles. Computer Graphics Forum, 21(3): 401-410

[Efros01] Efros, A. A., and Freeman, W. T., 2001, Image quilting for texture synthesis and transfer. Proc. SIGGRAPH 01: 341-346.

[Heeger95] Heeger, D. J., and Bergen, J. R., 1995 Pyramid-based texture analysis/synthesis. Proc. SIGGRAPH 95, 229-238.

[Hertzmann01] Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., Salesin, D., 2001. Image Analogies. Proc. SIGGRAPH 01, 327-340.

[Kwatra03] Kwatra, V., Schodl, A., Essa, I., Turk, G., Bobick, A., 2003. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. ACM Trans on Graphics, 22(3): 277-286.

[Kwatra05] Kwatra, V., Essa, I., Bobick, A., Kwatra, N., 2005. Texture optimization for example-based synthesis. ACM Trans on Graphics, 24(3): 795-802.

[Lefebvre05] Lefebvre, S., Hoppe, H., 2005. Parallel controllable texture synthesis. ACM Trans on Graphics, 24(3): 777-786.

[Liang01] Liang, L., Liu, C., Xu, Y, Guo, B., Shum, H., 2001. Real-time texture synthesis by patch-based sampling. ACM Trans. on graphics, Vol.20(3): 127-150.

[Liu04] Liu, Y., Lin, W., Hays, J., 2004. Near-regular texture analysis and manipulation. ACM Trans on Graphics, 23(3): 368 -376

[Portilla00] Portilla, J., and Simoncelli, E. P., 2000. A parametric texture model based on joint statistics of complex wavelet coefficients. International Journal of Computer Vision, Vol.40(1), 49-71.

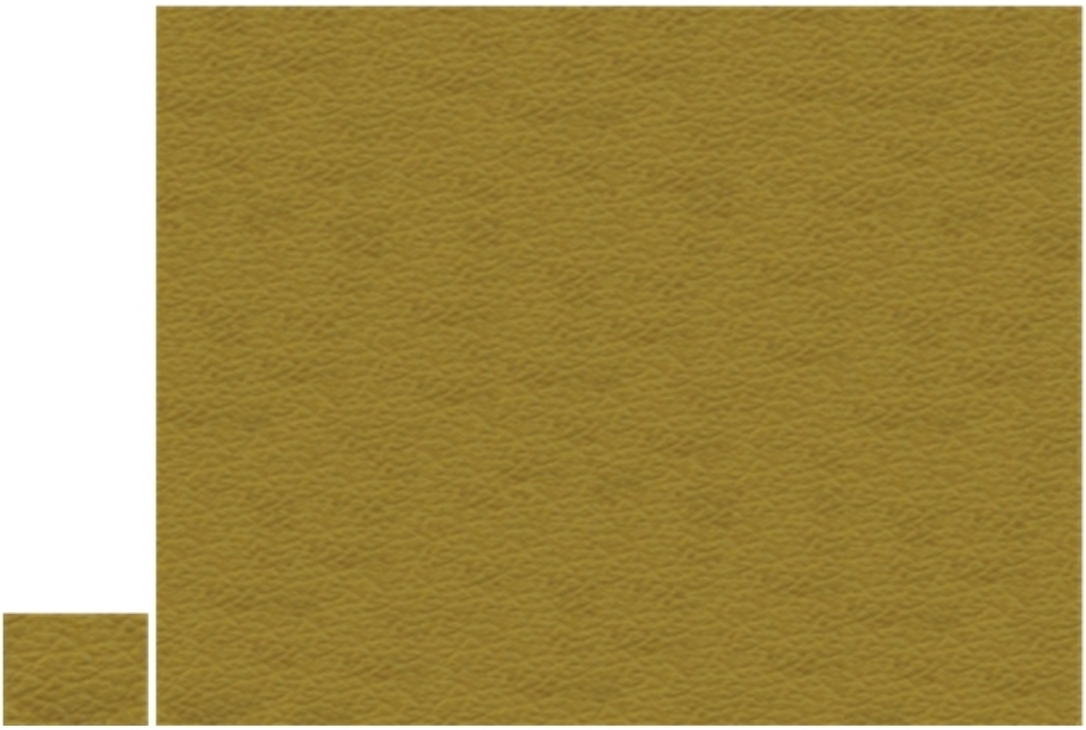
[Praun00] Praun, E., Finkelstein, A., Hoppe, H., 2000. Lapped Textures. Proc. SIGGRAPH 00, 2000: 465-470.

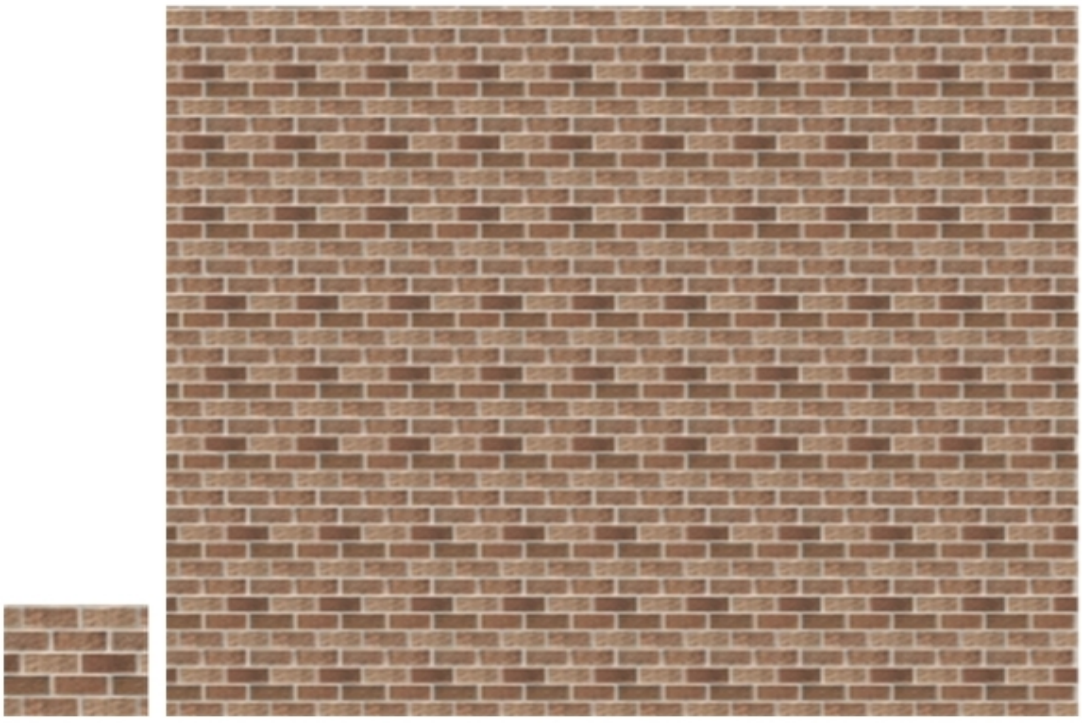
[Wei00] Wei, L. Y., and Levoy, M., 2000. Fast texture synthesis using tree-structured vector quantization. Proc. SIGGRAPH 00: 479-488.

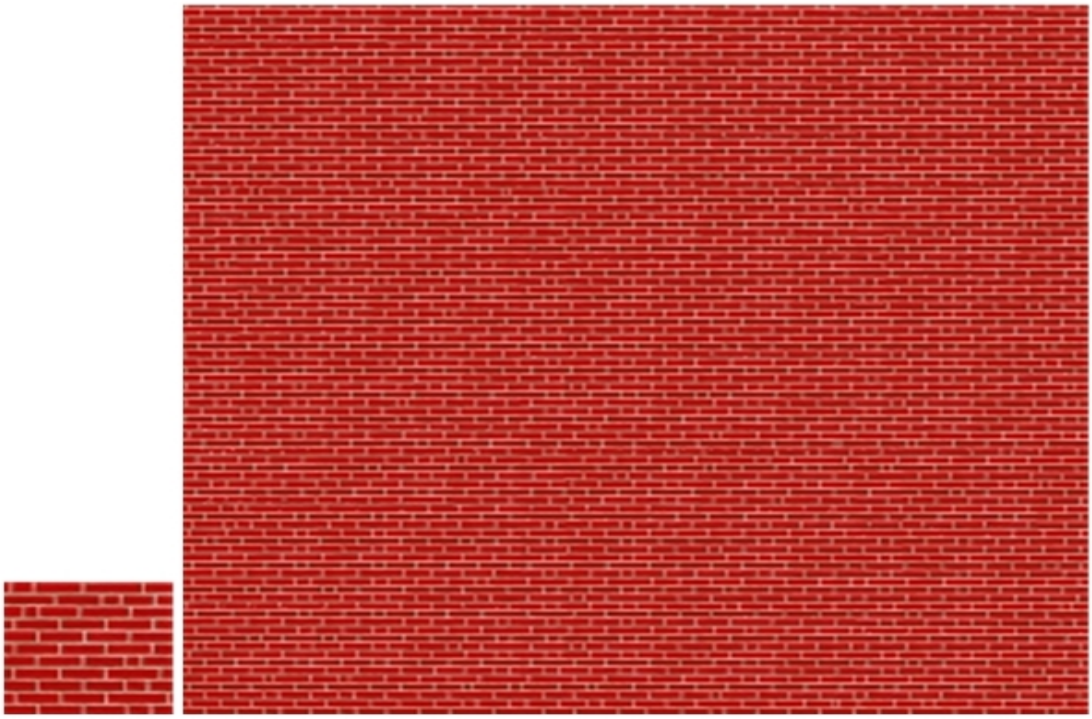
[Wu04] Wu. Q., Yu, Y., 2004. Feature Matching and deformation for texture synthesis. ACM Trans on Graphics, 23(3): 364 - 367

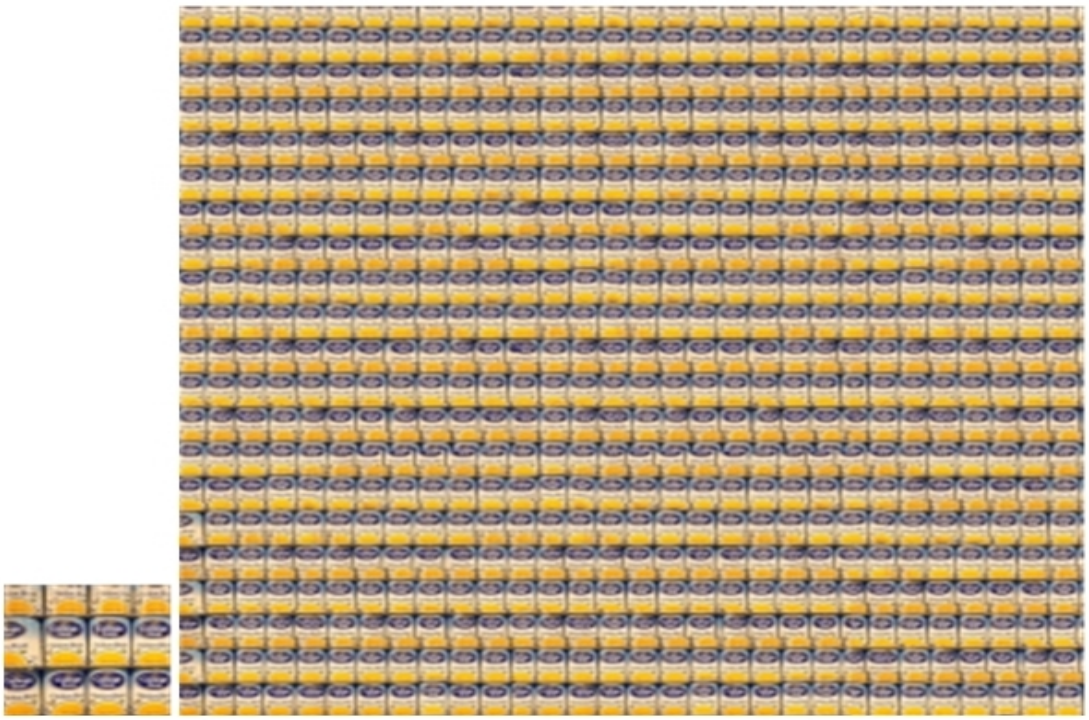
[Xu00] Xu, Y., Guo, B., and Shum, H., 2000. Chaos mosaic: Fast and memory efficient texture synthesis. Technical Report MSR-TR-2000-32, Microsoft Research, April 2000.

[Zhu98] Zhu, S. C., Wu, Y., and Mumford, D., 1998. Filters, random fields and maximum entropy (frame). International Journal of Computer Vision, 27(2): 1-20.









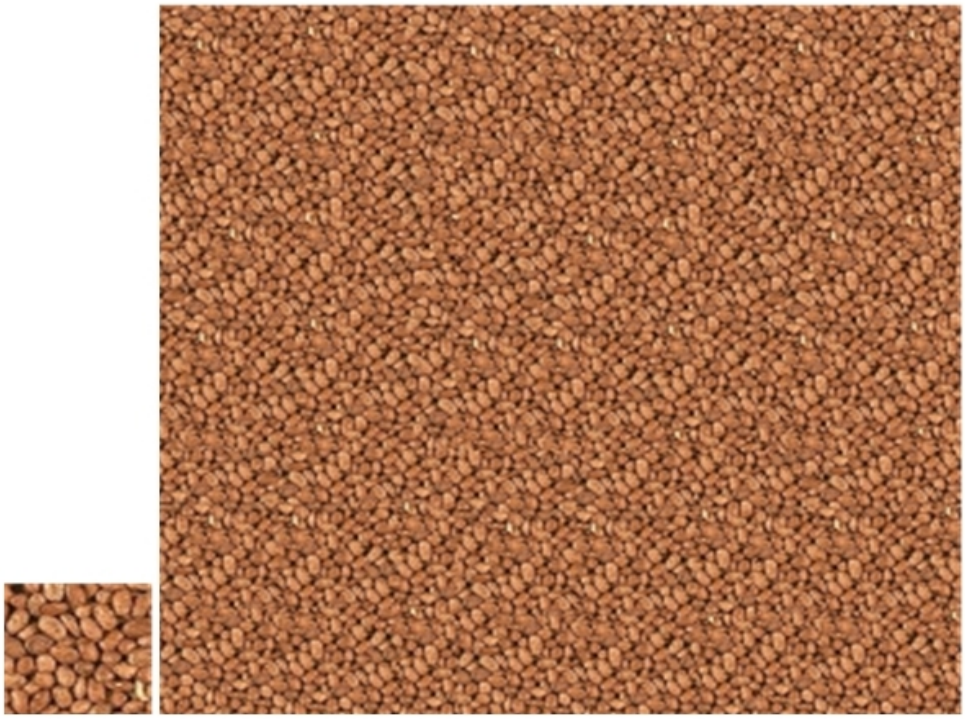
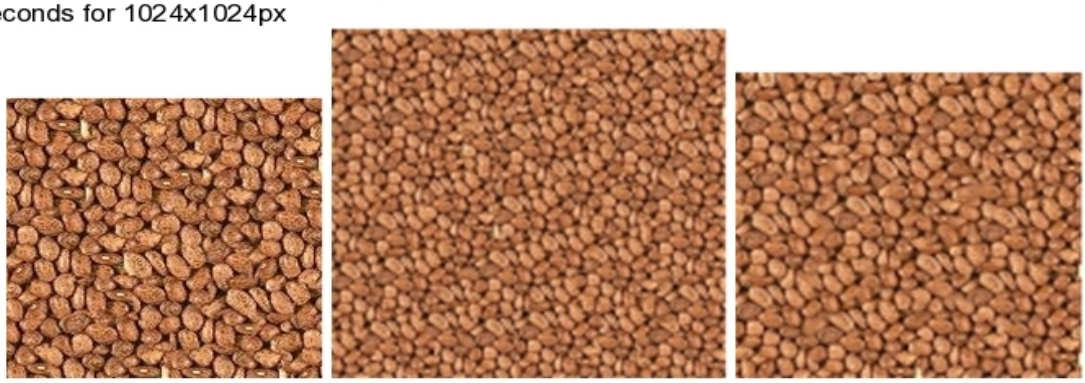


Figure 3 Patch Based Texture Synthesis. Process average time for each result 40 – 45 seconds for 1024x1024px



[Efros01] [Kwatra03] [Kwatra05]

[Efros01] [Kwatra03]

Figure 4 Results from related work for comparisons