Evaluation Metrics for Intelligent Generation of Graphical Game Assets: A Systematic Survey-Based Framework

Kaisei Fukaya, Damon Daylamani-Zad, and Harry Agius

(Survey Paper)

***Abstract*—Generative systems for graphical assets have the potential to provide users with high quality assets at the push of a button. However, there are many forms of assets, and many approaches for producing them. Quantitative evaluation of these methods is necessary if practitioners wish to validate or compare their implementations. Furthermore, providing benchmarks for new methods to strive for or surpass. While most methods are validated using tried-and-tested metrics within their own domains, there is no unified method of finding the most appropriate. We present a framework based on a literature pool of close to 200 papers, that provides guidance in selecting metrics to evaluate the validity and quality of artefacts produced, and the operational capabilities of the method.**

***Index Terms*—Artificial intelligence, evaluation metrics, graphical game assets, PCG.**

I. Introduction

GRAPHICAL assets such as 2D images and textures, or 3D models and environments form a large portion of digital game content. These assets are mostly created by artists, designers and 3D modellers. More recently, content creators have been using techniques such as photogrammetry and 3D scanning to speed-up content creation, especially, at design and prototyping stages. Procedural content generation has also been widely used in digital games from the early years in titles such as Rogue, Maze Craze and River Raid, to more recent examples including. kkrieger and No Man's Sky.

Procedural modelling algorithms [1], shape grammars [2], [3], [4], and deep-learning [5], [6], [7] have all been used to generate graphical assets. When specific content is desired, these approaches can positively impact the effort and time required for creating content, while retaining varying degrees of creative control.

Manuscript received 7 November 2023; revised 1 April 2024; accepted 6 May 2024. Date of publication 9 May 2024; date of current version 5 November 2024. Recommended for acceptance by B. Rosenhahn. *(Corresponding author: Damon Daylamani-Zad.)*

The authors are with the College of Engineering, Design and Physical Sciences, Brunel University London, UB8 3PH London, U.K. (e-mail: kaisei.fukaya@brunel.ac.uk; damon.daylamani-zad@brunel.ac.uk; harry.agius@brunel.ac.uk).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3398998, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3398998

These approaches require quantitative testing and validation, however. While appropriate metrics are applied in assessing these, there is no unified framework for metrics applicable across the gamut of graphical asset generation methods. While deep-learning approaches have a well developed collection of applicable metrics [8] others are fragmented or otherwise context specific.

This is a considerable challenge in using these approaches, as the designers and developers struggle to find the right method(s) for evaluating their approach aside from trial and error [9]. Each approach can be evaluated based on different aspects and various characteristics. Currently, the body of knowledge does not provide a clear guide in identifying suitable evaluation methods for the requirements of designers, developers and their projects.

In this paper a new framework is proposed, with the purpose of providing guidance for practitioners in evaluating graphical asset generation methods, aimed for use in game contexts. A systematic literature search has been conducted, resulting in a framework and analysis of existing metrics. The approach to literature search is presented in Section II, followed by an overview of the framework in Section III, and a breakdown of metrics and their applications in Sections IV, V and VI. Section VII will provide information on how to use the proposed framework.

II. The Approach to Systematic Literature Search

The systematic literature search, shown in Fig. 1, was initially conducted via queries consisting of a selection of key words over the four databases: ACM Digital Library, IEEE Xplore, ScienceDirect and Springer. These initial key words consisted of general terms in the PCG literature, alongside variations and synonyms of the word "generation" or "creation," and terms "asset" and "content." These words, Table I, were separated into three semantic groups, from which queries were formed. Query strings were formed by combining words between groups with AND operators, and words within groups with OR operators. For example: *"(Parametric OR Inverse OR Grammar) AND (Graphic OR Asset OR 3D OR Mesh) AND (Generation OR Synthesis)."* To decrease the number of queries needed, query

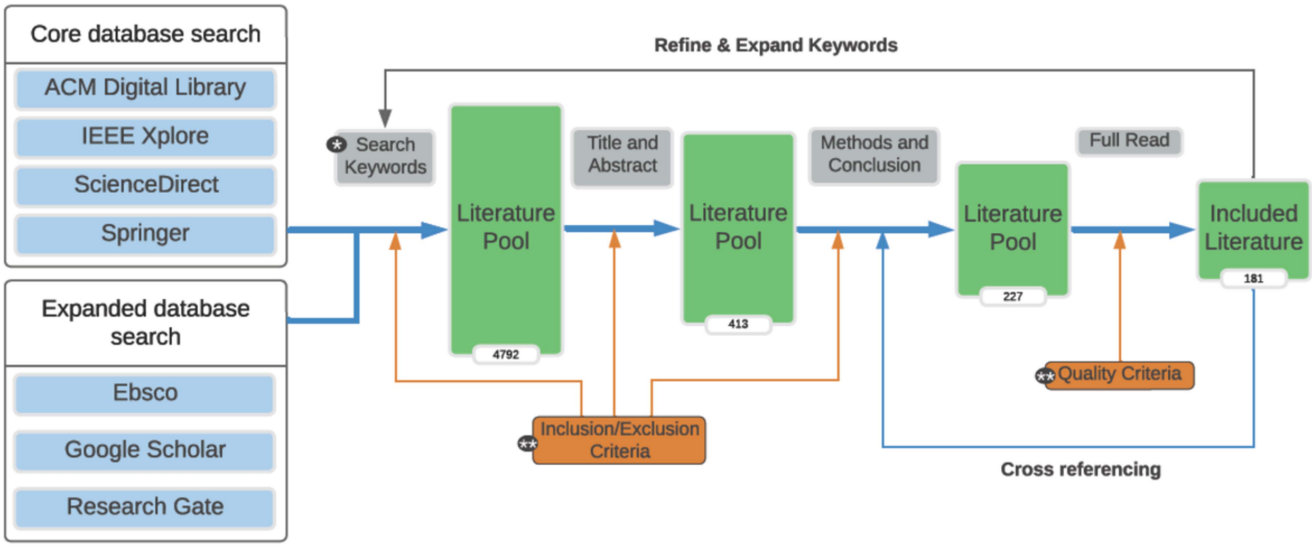


Fig. 1. The systematic literature review process. *Search keywords are listed in Table I, and **inclusion/exclusion and quality criteria are listed in Table II.

TABLE I
THE SEARCH TERMS USED TO QUERY THE CHOSEN DATABASES WITH EXPANDED SEARCH TERMS IN GREY

Group 1			Group 2	Group 3
Procedural*	″Deep Learning″	Grammar	Graphic*	Generation
Algorithmic*	Inverse	Deep	Asset	Synthesi*
″Machine Learning″	Stochastic	Parametric	3D	Modeling
ML	Furniture	Vehicle*	″3D Art″	Modelling
Car*	Building*	Cloud*	Content	Creation
Environment	Road*	Tree*	″3D Model″	Design
Terrain	″Normal map″	″Texture map″	Mesh	Production
Layout	″Height map″	Character	Shape	Assemb*
Face	Hair	Organ	″Text-to-image″	
Sprite	2D			

TABLE II
THE INCLUSION, EXCLUSION AND QUALITY CRITERIA APPLIED TO THE LITERATURE SEARCH

Inclusion Criteria	Exclusion Criteria	Quality Criteria
• Methods for generating graphical assets. • Studies that compare methods for generating graphical assets. • Studies that combine methods for generating graphical assets. • Only the newest version of a publication will be included where different iterations are found. • Literature published between 2016 and 2023.	• Studies concerning techniques for generating assets that are not distinctly graphical assets, e.g., text, audio, animation. • Studies that focus on functional requirements rather than visuals. • Non-procedural methods. • Survey papers. • Review papers. • Posters. • Courses.	• The method is validated. • The method is peer reviewed.

strings each contained multiple terms from each group. In addition to this, a broad search query containing all terms was used. The same queries were applied across all databases. To control the scope of this research we limit our search to static forms of asset by excluding animations and visual effects. As asset types emerged from the first set of searches, these were added in a secondary set of terms. The new terms, shown in grey within Table I, were used to form new queries.

The titles and abstracts of results were evaluated against the inclusion and exclusion criteria seen in Table II, and the results that passed these criteria formed the pool of accepted literature. The process of evaluation, for each query, was continued until the query was exhausted. Queries were considered exhausted once each result had been evaluated, or a full page of results had not passed the criteria. The number of entries per page varied between databases, therefore, per page entry counts were recorded for each database.

Next, the methods and conclusions were evaluated against the inclusion and exclusion criteria, and those that passed were then evaluated against the quality criteria. After the first set of queries were made and evaluated, the search terms were refined. The search process then repeated using these new terms, and

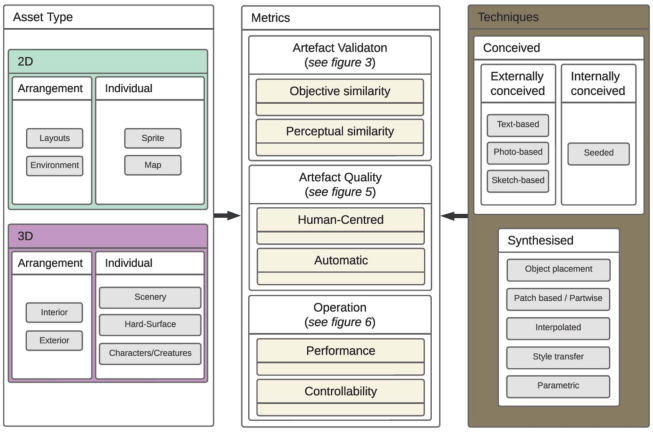


Fig. 2. A high-level view of the metrics framework, with a taxonomy of asset types and techniques found in the literature.

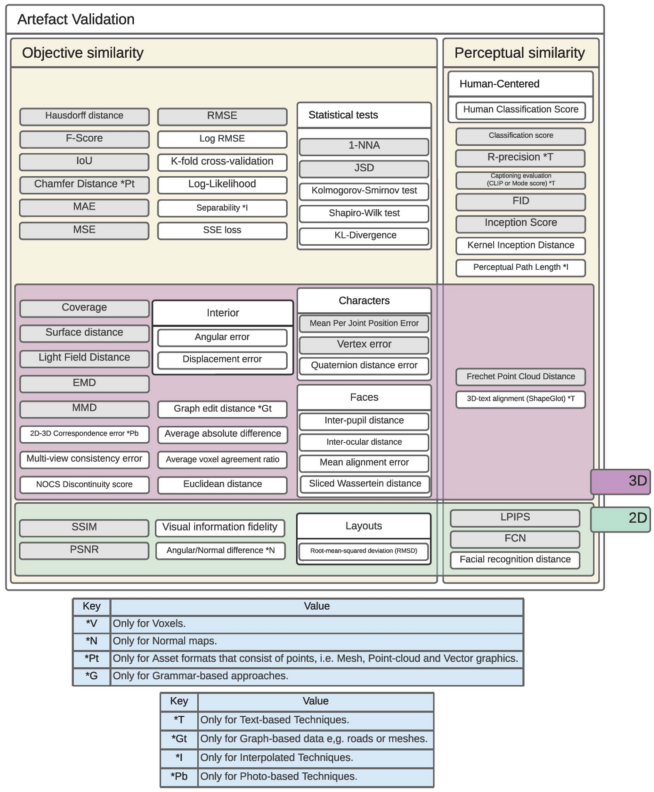


Fig. 3. The low level metric evaluation framework, focusing on operation metrics, and the various contexts in which they are applied. Keys at the bottom indicate their corresponding value within the diagram. White coloured boxes indicate metrics that only occur once within the examined literature, while grey coloured boxes indicate metrics that have more than one occurrence in the examined literature.

additionally, the accepted literature was cross-referenced to find more relevant literature. Following the same process, the search was expanded to the following databases, as well, until results were exhausted: Ebsco, Google Scholar, and ResearchGate to ensure completeness.

III. Evaluation Metrics Framework

There are various metrics that can be applied to graphical asset generation methods, depending on the type or format of the asset, or the generative technique employed. We categorise these metrics into three main classifications: *Operation*, *Artefact Validation*, and *Artefact Quality*.

Methods for generating graphical assets take many forms. These methods largely vary by target asset type and overall technique employed. Fig. 2 shows the taxonomy of asset types and techniques found in the literature, introduced in [10]; related with the high-level metrics framework, contributed here. Methods for generating 2D and 3D assets work with vastly different forms of data. 2D data may consist of bitmaps made up of pixel values, or vector data that represents points in 2D space. Whereas 3D data may consist of meshes, point-clouds or voxels. The technique represents the general task that the generator completes, but also relates to a particular arrangement of input and output data types for a given method. Selecting the correct metrics for evaluating a generative method requires that the asset type and technique is known. For example, a method that uses a sketch-based technique to produce 3D mesh assets will take 2D bitmap or vector data as an input, and will output mesh data. With this knowledge, appropriate *operation*, *artefact validation* and *artefact quality* evaluation metrics can be selected. Low-level diagrams in Figs. 3, 5, and 6 show the expanded list of metrics found in the literature. Items are placed under 3D and 2D groupings, and all ungrouped metrics have been used in both 3D and 2D use cases. Exceptions relating to specific approaches or formats are marked with tags which are defined at the bottom of the figures. Metrics that occur only once in the examined literature are marked white, and metrics that occur more than once in the examined literature are marked grey.

As discussed in [10], the classification of asset and technique type have been derived from the examined literature. 2D and 3D assets are differentiated by the formats used to represent them, as well as the types of asset commonly focused on in the literature. Two broad classifications have been applied to both 2D and 3D asset types, these are *arrangement* and *individual*. Arrangements are groupings of other types of asset, as opposed to *individual* asset types. For example, 2D layouts [11], [12], [13], [14] or environments [15], [16] are arrangements of 2D assets such as sprites. Techniques however, are separated into conceived and synthesised types. The former includes methods that produce assets from scratch, either by re-imagining an input in a different modality (externally conceived), or methods in which the system itself has full creative control (internally conceived). The latter includes methods that manipulate or combine existing assets to form new results. These include object placement algorithms, part-wise synthesis, methods for interpolating between existing assets, style transfer and parametric systems. At a high-level, we group together individual 2D assets as sprites and maps, where sprites consist of graphics that may be used as characters [17] or objects [18] in a 2D world or user interface [13], and maps consist of texture [19], normal [20] or height-maps [21] that are

commonly applied in improving the rendering of mesh based 3D assets. 3D assets may be arranged in an interior context, such as room layouts [22] or in an exterior context such as a city made up of buildings [23]. The high-level 3D categorisations include scenery, hard-surface and characters/creatures. Scenery groups together clouds [24], roads [25], terrain [26], trees [27] and rocks [28]. Hard-surface groups buildings [2], furniture [29], vehicles [5] and props [30]. While characters [31], faces [32], hair [33] and organs [34] are grouped under characters/creatures.

Togelius et al. [35] put forth 5 desirable properties of PCG methods in games. These are: *Speed*, generation time; *Reliability*, the meeting of baseline expectations; *Controllability*, the user's ability to specify or steer content; *Expressivity/Diversity*, the variety of possible artefacts; and *Creativity/Believability*, the quality of artefacts. While it would be invaluable to have the capability of addressing each of these properties via quantitative means, we are able to cover 4 of these properties given the methods found in the literature. *Artefact validation* metrics assess the *Reliability* and to some extent the *Expressivity/Diversity* of the method via statistical aggregation. *Artefact quality* metrics assess the *Creativity/Believability* of the the artefacts, and operation evaluation, namely performance metrics, cover the *Speed* and scalability of the method. On the other hand, *Controllability* is harder to quantify, and thus no metrics were observed in the literature. We posit that this could be mapped to the method's technique classification by means of user degrees of freedom, though it is difficult to determine what constitutes a meaningful degree of freedom. For example, a sketch-based system affords the user a large amount of control over the outcome in exchange for a moderate degree of effort; whereas a seeded approach may require close to zero user input and thus affords very little control. A parametric approach may have varying degrees of freedom depending on the parameters exposed to the user, but whether or not these are meaningful controls is not objectively identifiable. In Fig. 4 of [10] graphical asset generation input types have been ranked in terms of complexity.

Artefact validation consist of *objective* and *perceptual* similarity metrics. *Objective similarity* metrics assess the similarity between outputs and corresponding ground-truths and are often used in aggregate form for evaluation purposes. For these metrics to be applied, the ideal (i.e. ground-truth), corresponding data must exist for each output in the test. As such these metrics are largely seen in deep-learning generative approaches where this data is available as a matter of course. This naturally limits the relevance of objective similarity metrics to *externally conceived* methods, where the generator's job is to reliably interpret data of one form to another e.g. text-to-image or sketch-to-mesh. Conversely, *objective similarity* metrics are counter-productive if at all possible in cases where the generator should have creative agency, as they punish variation and reward conformity. Some common objective similarity metrics include, Mean squared error (MSE) [36], [37], [38], [39], Root mean square error (RMSE) [2], [3], [4], [40], [41] or Intersection over union (IoU) [12], [42], [43], [44].

Perceptual similarity metrics offer an alternative approach to validating artefacts, requiring less correspondence to ground-truth data. This is achieved either through human perception, or

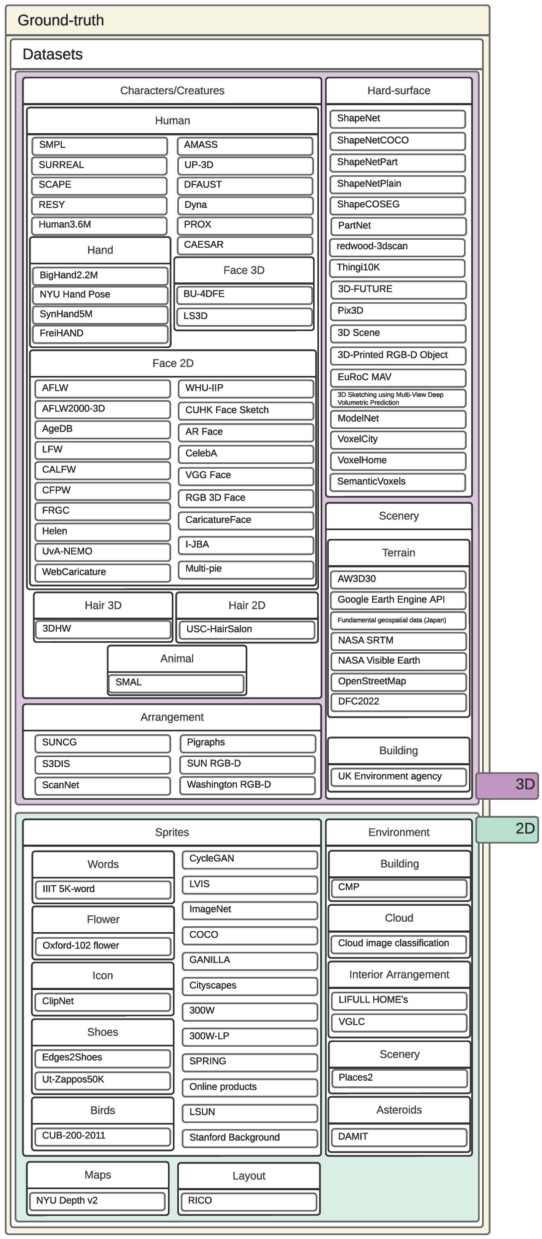


Fig. 4. Ground-truth datasets from the examined literature, arranged by asset type.

standardised deep-learning models that behave as a proxy for human perception such as Inception score (IS) [45] and Frechet inception distance (FID) [46].

Artefact quality metrics assess the generator's ability to produce high quality outputs. This includes *human-centered* measures, such as questionnaires and rating systems [47], [48], [49], [50], and *characteristic* metrics that measure particular attributes of assets.

Operation evaluation includes *performance* and *controllability* measures. Performance metrics assess resources, such as memory [51], [52], [53] or time cost [29], [52], [53], [54], [55] of a method. While these metrics are important during development and testing, they are particularly important when applications go

into production and release and run in real-time, as they have direct implications for user-experience and usability.

It is in the best interest of researchers and practitioners to evaluate the *validity* and the *quality* of artefacts as well as the *operability* of their generative methods, in order to assess and present the capability of their approach, and compare it with existing alternatives. Sections IV, V and VI will cover artefact validation, artefact quality and operation metrics respectively.

IV. Artefact Validation Metrics

A wide breadth of established metrics may be applied in validating the capabilities of generative methods as shown in framework Fig. 3. These artefact validation metrics assess the degree to which generated artefacts match the intended asset type. This can be achieved through objective or perceptual similarity measures, which are presented in Section IV-A and IV-B respectively. Section IV-C will present relevant procedures.

A. Objective Similarity Metrics

Objective similarity testing assumes that there is an exact intended output for every input. Therefore only applying to methods, often deep-learning based, that have the purpose of reliably re-interpreting an input in some way. However, this is not always a complete limitation. For example, with variational autoencoders (VAEs), such metrics can validate how well the method captures the desired features of a dataset, with the method itself still being capable of producing novel and varied outputs by sampling or interpolating between data points [19], [44], [56]. Many games follow a specific art direction, with requirements for the style of assets. Altogether, validating and training a VAE on data that fits these requirements would help ensure style consistency while still producing varied results. In other words, *objective similarity* can help ascertain in these cases, whether a method has successfully captured a target search space.

Objective similarity metrics are applied in the comparison between corresponding data points, such as mean absolute error (MAE) [17], [51], [57], mean squared error (MSE) [36], [37], [38], [39], root-mean-squared error (RMSE) [2], [3], [4], [40], [41] and sum of squared errors (SSE) [23]. MAE aggregates the absolute error of data-points, that is the positive difference between corresponding values, while SSE and MSE square these errors strengthening larger errors and diminishing smaller errors. SSE aggregates via summation while MSE aggregates via mean. The resulting values are in squared units, however. RMSE negates this effect by taking the root of the resulting value [8]. While each of these errors have their uses when training deep-learning methods, MAE and RMSE can be considered more intuitive for evaluation purposes, as they share the same unit as the data they are derived from.

Intersection over union (IoU) measures the overlap between two volumes or regions, measuring the difference in shape and positioning between two assets. This has be applied in 2D for evaluating segmentation tasks [42] and layouts [12] or in 3D for comparing shapes [43] or bounding boxes [44]. Chamfer distance (CD) compares the difference between two sets of points by averaging the difference between each point and its closest point in the other set, thus not requiring a defined pairing or matching set sizes. This can be applied to all point-based asset formats such as point-clouds [58], and meshes [59]. Also used for point-based formats, Hausdorff distance finds the greatest difference between two sets of points [34], [60]. The F-score is a measure that combines the precision and recall for generated and ground-truth counterparts. This is used to score shape similarity between the two, therefore measuring the reconstruction quality of the method. This has been used to evaluate single-view [61], [62], [63] and multi-view [59] reconstruction approaches. For methods of interpolation, a separability score can be obtained to measure how disentangled the latent space is [64]. To test how well a deep-learning model generalises, irrespective to the set of training data, K-fold cross-validation may be employed [24]. This method splits the dataset into K subsets. For each subset, the model is trained on all other subsets, and tested using the subset in question. The mean and variance across tests can then be reported in the chosen evaluation metric. Zhao et al. [65] utilise log-Likelihood analysis in assessing an adversarial autoencoder's ability to capture the distribution of the training data.

Statistical tests may be necessary in the case of large deep-learning models, providing insight into the capability of the method through analysis over output distributions. These tests are purely statistical analysis such as analysis of means and standard deviations, and nearest neighbour classification and regressions. 1-nearest neighbour accuracy (1-NNA) assesses the similarity between two distributions using a 1-nearest neighbour classifier. This classifies each sample as belonging to one of the two groups, thus identical distributions should converge on an accuracy of 50%, therefore the closer the value is to 50% the better [66]. This approach takes into account the similarity in both the quality and the variation or diversity of the two sets, and has been used for 3D deep-learning approaches [67], [68], [69], [70]. Jensen–Shannon divergence (JSD) is used to measure the divergence between a ground truth and output distribution. For example, this is applied in point-cloud evaluation [71], [72]. Ivanov et al. test that their results follow a normal distribution, using Kolmogorov-Smirnov and Shapiro-Wilk tests [73].

Many similarity metrics are specific to 3D assets. For example, earth mover's distance (EMD) is used to measure the difference between an output and a ground truth distribution. This is used for point-clouds [74], voxels [75], and meshes [76]. Coverage scores the amount of similarity between two sets of point-clouds or meshes, for example, between generated results and a reference set [5], [69], [77]. Minimum matching distance (MMD) [5], [69], [78] instead gives a better representation of difference between the two sets, matching items by minimum distance and yielding the average of these distances [79]. Mesh reconstruction similarity can be calculated using point-wise euclidean distance between target and generated meshes [80], [81].

Surface distance metrics such as those used in [36], [82], [83] densely compare 3D points as a measure of surface similarity. Light field distance metrics use multi-view renderings of the 3D assets to calculate shape similarity invariant to rotation [5], [84]. Whereas multi-view consistency error observes the distance

between the same points at different viewing angles [85]. Here, a normalised object coordinate space (NOCS) representation is used. A NOCS discontinuity score is also introduced, to measure the connectivity of the surface [85]. Öngün and Temizel [86] introduce average absolute difference (AAD) and average voxel agreement ratio (AVAR) metrics, which measure the agreement between paired voxel shapes at different angles. For assessing data in graph form, graph edit distance [87] can be used. This represents the minimum amount of change required to transform one graph to another, and thus how similar their topologies are. A key aspect when generating characters is the joint angle. Mean per joint position error (MPJPE), vertex error and quaternion distance error may be used to measure this [31]. For interior object placement, bounding box displacement and angular errors have been used as metrics for correct positioning and rotation [44].

In the task of reconstructing caricature faces in 3D, inter-pupil and inter-ocular distance metrics have been used [88]. While sliced Wasserstein distance is used to compare the difference in ground-truth and generated face patch distributions [89], and mean alignment error may be used to find the average difference between vertex positions [90].

Alternatively, some similarity metrics are specific to 2D assets. For example, peak signal-to-noise ratio (PSNR) can be used to determine the strength of noise within an image, and therefore can be a measure of visual quality [47], [91], [92]. Structural similarity (SSIM) measures the perceptual quality based on high-level structure, comparing the output to a ground-truth image [93]. This has seen widespread use across image and texture generation approaches [19], [38], [47], [91]. Yadav et al. [94] use the visual information fidelity (VIF) metric [95] to assess the visual quality of their outputs. VIF measures the difference in visual information for the output image by assuming the ground-truth image to be a perfect signal [95]. Alternatively, when evaluating methods for generating normal maps, an angular or normal difference metric may be applied [20], [96]. These are similar to other pixel-wise error metrics, in that distances in pixel-wise values between two images are assessed. Angular error is reported as angles in degrees, as each pixel on a normal map represents a direction [20], while the normal difference metric reports non-angular values [96].

Root-mean-squared deviation (RMSD) has been applied in evaluating 2D layout generation by computing the discrepancy between the positioning of elements in generated and ground truth layouts [13].

B. Perceptual Similarity Metrics

While objective similarity is not always possible due to requiring one-to-one ground-truth data, perceptual similarity can be used to classify or score an artefact based on *perceived similarity* to a visual reference. Perceptual similarity in its simplest form may involve human-centered classification of artefacts. Here, a human evaluator will be given visual references to compare generated artefacts with, and tasked with judging whether the artefact is ”real” or ”fake” [19], [97], or whether it belongs to the target classification [47], [98], as mentioned in Section V-C. Alternatively, a variety of automatic perceptual similarity evaluators may be used. The *Inception* CNN model has been shown to perform well at image classification and detection tasks [99]. Since its introduction, subsequent versions of the *Inception* model have been used for evaluating generative models, such as generative adversarial networks (GANs). Metrics such as inception score (IS) [100], frechet inception distance (FID) [46] and kernel inception distance (KID) [101] each make use of the inner layers of the *Inception* model to compare latent similarities. Though IS uses this to evaluate the perceptual similarity of images with their expected class, FID and KID compare a distribution of generated artefacts with a ground-truth distribution. As a result, IS is limited to the categories that *Inception* is trained on, (typically ImageNet ILSVRC [102]) but requires no ground-truth data. However, FID and KID are not limited to assessing pre-defined categories but require a reference or ground-truth dataset for comparison. FID utilises Frechet distance between the two image distributions, and assumes that the two are Gaussian. This is applied to 2D assets [7], [64], [103], [104], [105], [106], [107], 3D assets via 3D classifiers [37], [108] or rasterisation to 2D form [109], [110].The Frechet point cloud distance extends FID for applications in assessing the similarity of point-based 3D shapes [111]. This has been used to evaluate many deep-learning based 3D point-cloud [71], [111], [112] and mesh [67] generators. KID instead utilises the maximum mean discrepancy between the image distributions, and does not assume a Gaussian form [101]. This has been used for evaluating faces in 2D [113] and for evaluating 3D assets rendered in 2D form [114]. IS however, does not take into account the statistical distribution of the data. It has been used primarily in the evaluation of 2D approaches [50], [113], [115], [116], [117], though it has been adapted for 3D as the 3D Inception Score, which uses a 3D classification network instead [118]. The 2D IS has also been applied to 3D assets that have been rendered in 2D form [114].

For VAE based interpolation, a perceptual path length metric is proposed [64]. This metric measures the perceptual distance at points along a path in the latent space, determining how smooth the interpolation is. For text-based generation methods, CLIP score [119] or R-precision may be used to measure the alignment between the output and the text prompt used to produce it. The latter is calculated using a Deep Attentional Multimodal Similarity Model (DAMSM) or CLIP-R-precision [120]. Extending this concept to 3D, the ShapeGlot dataset [121] can be used to train a similar alignment score model for text-3D generation methods [122]. A similar metric for faces is proposed [89], making use of features from a facial recognition network. For 2D tasks, learned perceptual image patch similarity (LPIPS) can be applied. This metric is trained on the Berkeley-Adobe Perceptual Patch Similarity (BAPPS) dataset, which consists of human perceptual judgements across many sets of images [123]. In a similar way, some approaches use alternative pre-trained fully convolutional networks (FCNs) as perceptual measures for 2D images [124], [125].

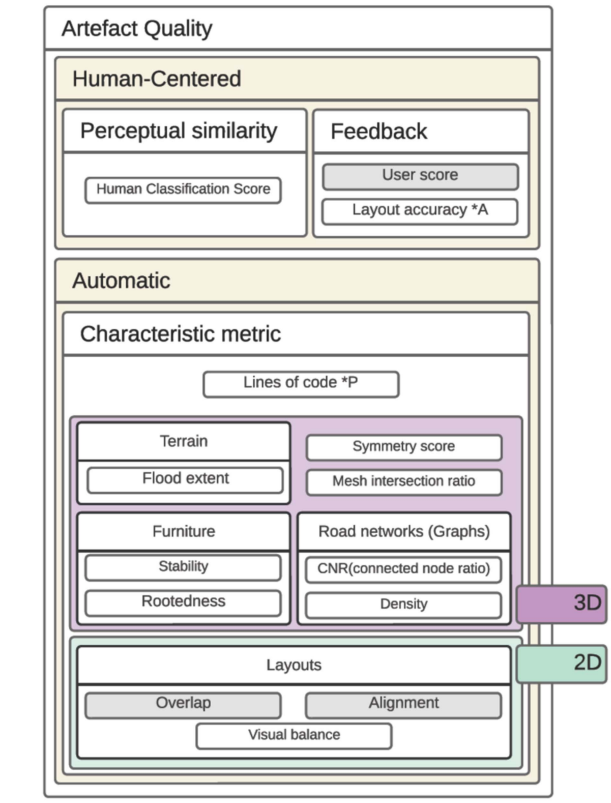
C. Evaluation Procedures

For methods that contain multiple sub-processes, e.g. deep-learning approaches. Ablation studies are a common procedure for assessing the effectiveness of these sub-processes or

components [5], [18], [32], [44], [65], [126]. In an ablation study, components are systematically assessed for their contribution to the effectiveness of the method. This is achieved by removing the component and evaluating the rest of the method without it.

When developing generative AI systems, developers require a way to evaluate the quality and performance of their iterations against each other and against other existing methods. Benchmark datasets provide a controlled input to objectively compare outputs and performance. The challenge of benchmark datasets is that it might be difficult to always find a dataset that closely matches the intended input/output of the method being developed. Hence, may require slight adjustments to how the method functions. For methods that require image data, a popular dataset to use is ImageNet [190]. While general 3D datasets include: ShapeNet [191], ModelNet [192] and Pix3D [193]. Part-wise 3D methods may use PartNet [194]. Datasets are also available for certain specific object types, such as shoes (UT Zappos50K [195]) and birds (CUB-200-2011 [196]). Fig. 4 shows the ground-truth datasets found in the literature, separated into 2D and 3D data types, and grouped by asset type.

When conducting objective similarity validation, the dataset must meet two requirements: 1) it must contain data that can be used as an input into the generative system and 2) it must contain corresponding ground-truth data in the same form as the artefacts. When conducting validation via perceptual similarity, wherein most methods compare 2D image data, as long as the data can be arranged as such it can be used. For example, mesh data could be rendered to create 2D data for perceptual comparison, as seen in [67]. There are exceptions to this approach, namely with captioning evaluation and 3D-text alignment in which artefacts are scored on closeness to text descriptions. Alternatively, when using IS no dataset is required, because scoring is based on a pre-trained classifier.



Key	Value
*T	Only for Text-based Techniques.
*A	Only for Arrangement asset types.

Key	Value
*I	Only for Interpolated Techniques.
*P	Only for Parametric Techniques.

Fig. 5. The low level evaluation metrics framework, focusing on artefact quality metrics used in the literature, and the various contexts in which they are applied. Keys in the bottom-most table are used to indicate their corresponding value within the diagram. White coloured boxes indicate metrics that only occur once within the examined literature, while gray coloured boxes indicate metrics that have more than one occurrence in the examined literature.

V. Artefact Quality

Artefact quality evaluation methods are categorised as *human-centered*, or *automatic quality metrics*, shown in framework Fig. 5. *Human-centered* quality metrics rely on human assessment, while *automatic quality metrics* assess objective characteristics of assets. The following Section V-A and V-B will cover human-centered and automatic quality metrics respectively. Section V-C will cover relevant evaluation procedures.

A. Human-Centered Quality Metrics

Human perception and opinion can be used to measure asset quality. Such metrics are particularly valuable given the purpose of graphical assets, and their integration into games or other digital media where end-user experience and immersion are paramount. However, these measures can be inconsistent, and dependent on many factors. Thus, are best paired with more objective forms of evaluation i.e. characteristic metrics. Human perception is a common approach to evaluation in text-guided generative methods for example [48], [49], [50].

A preference metric may be used, where human evaluators choose the result that they prefer out of a selection of examples. These examples can consist of outputs from the method being evaluated and outputs from existing alternative methods. These questions can be posed as a general preference [47], or preferences for certain aspects of the result [48]. Users may alternatively rate or score the method on a scale [149], [197].

In assessing interior room layouts, [22] introduce a layout accuracy metric, which observes the number of furniture pieces a human evaluator chooses to move in a generated layout. In other words, this is the number of placements that the human evaluator is unsatisfied with.

B. Automatic Quality Metrics

Automatic characteristic metrics may be applied in examining particular characteristics of generated assets. These metrics tend to be specific to the type of asset and format used. To choose or develop a characteristic metric, desired characteristics of an artefact must first be determined. This will be be dependent on the intended use case, and whether the characteristic can be objectively measured. For example, symmetry score [118] and mesh intersection ratio [62] are applied in quantifying the symmetry and self-intersection of 3D assets respectively. For

road networks, or graphs in general, connected node ratio and density metrics [25] can be employed to examine the properties of the networks. Jones et al. [132] introduce stability and rootedness metrics for evaluating generated furniture assets, which use physics interactions such as gravity and pushing forces to test the generated shapes. While a measure of flood extent is used for measuring the realism of terrains, as natural formations tend to have a degree of drainage [131]. For 2D layouts a measure of visual balance has been used; considering the distribution of elements across the layout [14]. While the overlap and alignment of elements have been used as measures for layout quality, where the positioning of text and visuals are key. Here, a good layout will have minimal overlap between elements and maximal alignment [11], [12]. Where programming languages for shape creation are concerned [1], the number of lines of code has been used as a metric to determine the the language's efficiency or writing speed.

Depending on the use case, there may be certain desirable quality requirements for the generated artefacts. For example, to consider a 3D model "game ready", it may need to meet geometric standards, such as having low self-intersection [62]. If the realism or functionality of a design is a concern, metrics similar to stability or rootedness will be of more relevance [132].

C. Evaluation Procedures

For the task of collecting human feedback or observing human perceptions, questionnaires are the primary method employed. When receiving quantitative feedback from users, it is common to obtain a scoring or ranking from participants. This is obtained through a Likert-scale for example [166]. In many cases Likert-scales are used to collect ordinal ratings, or binary choice questions may be used for classification. For text-to-image generation methods, DrawBench [48] may be used to benchmark and compare the performance of one method with another, providing a systematic list of prompts. For example, an experiment may ask users to compare two images generated via the same prompt, from different methods, and rate them in terms of fidelity or image-text alignment [48].

When developing a questionnaire for evaluating a generative method, consider general questionnaire design principles [198] and practices for implementing Likert scale questions [199]. As an alternative to rating, a questionnaire may present participants with a reference asset (i.e. image or 3D model) and ask them to choose between a number of assets based on similarity to the reference [47], [98], where one image is the output of the chosen method, and others are from comparative methods. Similarly, a set of images from various methods can be ranked [197]. A human classification score, or fooling rate may also be obtained by presenting participants with a "real" asset and a generated result, and asking them to select the one that is "real" [19], [97]. Here, the "real" asset does not necessarily have to be a photograph or an exact match to the generated result. The purpose is to see if the quality of the artefact can fool the participant into believing that it was not generated. This also acts as a form of artefact validation as, if the participant believes the artefact is a "real" example of a particular asset type, then it must be identifiable as such.

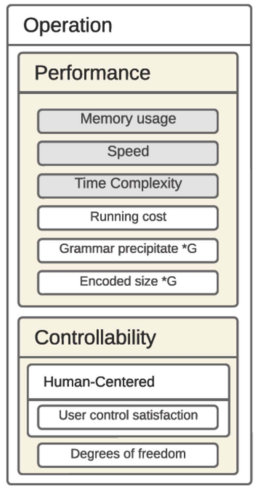


Fig. 6. The low level metric evaluation framework, focusing on operation metrics used in the literature. White coloured boxes indicate metrics that only occur once within the examined literature, while gray coloured boxes indicate metrics that have more than one occurrence in the examined literature.

VI. Operation Metrics

Fig. 6 presents the operation metrics used in the literature. These consist of various *performance* metrics and controllability. There are four performance metrics that can be applied to most methods, these are: memory usage, speed, time complexity and running cost. Alternatively, for grammars specifically, there are two performance metrics that can be applied, *encoded size*, and *grammar precipitate*. Memory usage and speed can be observed by monitoring the hardware usage or efficiency during runtime. In general, these metrics are dependent on the relevant specifications of the machine used, so these should be reported in any performance evaluation. It is often necessary to measure the speed of an implementation. A faster approach can allow for more content to be produced in a shorter amount of time. An example of speed testing can be found in the work of [152], where the method's performance with different numbers of inputs, and different hardware are compared. A common finite hardware resource is volatile memory such as dynamic random access memory (DRAM) for CPU based computation, and video random access memory (VRAM) for GPU based computation. These have been used as evaluation metrics in [51], [52], [53]. Memory usage provides a benchmark for the minimum system memory required to run the method, determining the type of device it may operate on. The scalability of a system can be assessed by evaluating the time complexity of a method given the inputs, as seen in [200] for example. This provides an indication for the speed of a method depending on its scale or number of inputs.

Running cost may be relevant for commercial projects or content generation services, for example, Bhatt et al. [151] report potential running cost of their generative method as a cloud-based tool. For grammar based methods, an encoded

TABLE III
USAGE OF 2D METRICS IN LITERATURE, ORGANISED BY GENERATIVE TECHNIQUE USED

Quantitative Evaluation Metrics									
Metrics		Text-based	Image-based	Seeded	Object placement	Patch based/ Partwise	Interpolated	Style transfer	Parametric
Perceptual similarity	LPIPS	[92]	[19], [95]					[108]	
	FCN	[126]	[127]						
	Facial recognition distance			[90]					
Characteristic	Visual balance				[14]				
Objective similarity	SSIM	[92], [129]	[47], [93], [130], [17], [19], [95], [131]	[38]					
	PSNR	[92]	[47], [93], [130], [95], [131]		[15]				
	Visual information fidelity		[95]						
	Angular/Normal difference		[20], [98], [128]		[44]				
	Overlap				[11], [12], [14]				
	Alignment				[11], [12]				
	RMSD				[13]				

size [156] or grammar precipitate metric [150] may be used, determining the efficiency of the grammar encoding, and the versatility of extracted rules respectively.

Characteristics such as parallelizability, distributability and access to intermediate results [54], are contributing factors to performance. Though the review did not yield any quantitative measures for these, they should be considered per user need and expertise. Memory usage, speed, time complexity and running cost can all be impacted by these characteristics. Hence their impact can be measured through these metrics. While not observed in the literature, the controllability of a method could be measured in user degrees of freedom. While more degrees of freedom does not necessarily equate to more control, it could suggest more variability of input. But more controls can come at the cost of usability [201]. Instead, controllability could be indicated via user studies or indirectly through general usability assessment, such as System Usability Scale (SUS) [202].

VII. EVALUATION METRICS FRAMEWORK: USAGE

This section will present how to select appropriate evaluation metrics from those found within the literature. Figs. 3, 4, 5 and 6 show the categorisation of validation metrics, ground-truth datasets, quality metrics and operation metrics, while Tables III, IV and V, show the literature organised by metric and technique used for 2D, 3D and shared metrics respectively. It is evident that some metrics are very popular for evaluating specific techniques, while others are popular regardless of technique. For example, PSNR has been used in five instances for evaluating image-based 2D reconstruction tasks, while FID has been used in a total of 30 instances evaluating text-based, image-based, seeded, style-transfer and parametric generation tasks. There are many metrics that, due to being context specific, are only used in one or two instances, e.g. flood extent, stability, rootedness, inter-pupil and inter-ocular distances.

Fig. 7 presents the selection procedure for operation, validity and quality metrics. This process is about finding available methods to consider for all three metric types. In this process, relevant metrics are first narrowed down by category based on various limiting factors, and then the dimensionality of the asset type. Visiting the relevant figure a list of metrics can be obtained from headings related to the asset type, and more general metrics: Figs. 3, 4, 5 and 6. These metrics can then be found in the relevant tables, where the choice can be narrowed down based on the technique and by observing their application in existing examples cited: Tables III, IV and V. Metrics applied to the same technique may be considered more relevant. If the intention is to compare with existing methods, then a popular metric for that technique should be chosen. To identify the technique a method's inputs and functionality should be considered, as discussed in Section III. For artefact validation metrics, human-centered or automatic approaches can be used. The purpose of artefact validation is to assess whether artefacts are of the intended asset type. Artefact validation via objective similarity is only possible when ground-truth data is available. In most cases, evaluation of a method via ground-truth datasets is only possible with externally conceived approaches in which there is a known intended result for a given input e.g. conversion from sketch-to-mesh. Many of

TABLE IV
USAGE OF 3D METRICS IN LITERATURE, ORGANISED BY GENERATIVE TECHNIQUE USED

Quantitative Evaluation Metrics									
	Metrics	Text-based	Image-based	Seeded	Object placement	Patch based/ Partwise	Interpolated	Style transfer	Parametric
Perceptual similarity	Frechet Point Cloud Distance	[110], [132]	[67]	[71], [113], [114], [56]					
	3D-text alignment (ShapeGlot)	[124]							
Characteristic	flood extent					[133]			
	Stability								[134]
	Rootedness								[134]
	Symmetry score			[120]					
	Mesh intersection ratio		[62]						
	CNR					[25]			
	Density					[25]			
Objective similarity	Coverage		[67], [70], [135]	[68], [74], [114], [71], [79], [113], [72], [77], [136], [5]			[56], [69]		
	Surface distance		[36], [83], [84]						
	Light Field Distance		[5], [85]						
	EMD	[132]	[70], [137], [138], [34], [63], [76], [18]	[68], [74], [79], [71], [139], [140], [72], [113]			[56], [69]	[75]	
	MMD	[37]	[5], [67], [70]	[68], [79], [140]			[69]		
	Correspondence error		[86]						
	Multi-view consistency error		[141]						
	NOCS Discontinuity score		[141]						
	Graph edit distance								[112]
	Average absolute distance								[87]
	Average voxel agreement ratio								[87]
	Euclidean distance		[81], [82], [142]						
	Angular error		[20]		[44]				
	Displacement error				[44]				
	Mean Per Joint Position error		[31], [142], [143]						
	Vertex error		[144]	[39]					
	Quaternion distance error		[31]						
	Inter-pupil distance		[89]						
	Inter-ocular distance		[89]						
	Mean alignment error		[91]						
	Sliced Wasserstein distance			[90]					

TABLE V
USAGE OF METRICS THAT DO NOT REQUIRE 2D OR 3D DATA SPECIFICALLY, WITHIN THE LITERATURE, ORGANISED BY GENERATIVE TECHNIQUE USED

Quantitative Evaluation Metrics									
	Metrics	Text-based	Image-based	Seeded	Object placement	Patch based/ Partwise	Interpolated	Style transfer	Parametric
Performance	Memory usage		[53]		[51]				[52]
	Speed		[2], [32], [33], [41], [53], [84], [89], [145], [146], [147], [148], [149], [150], [151], [152], [153], [154], [155], [156], [157]	[54], [158], [159]	[11], [13], [22], [55], [160], [161]	[133], [162]			[1], [26], [27], [29], [52], [163], [164], [165], [166]
	Running cost		[153]						
	Encoded size			[158]					
Feedback	Grammar precipitate		[152]						
	User score	[37], [42], [48], [49], [50], [167], [168], [169]	[47], [127]						
	Layout accuracy				[22]				
Perceptual similarity	Classification score		[58]	[170]					
	Captioning evaluation	[49], [50]							
	FID	[6], [37], [48], [49], [50], [92], [110], [111], [119], [124], [171]	[5], [67], [95], [105], [116], [131], [145]	[7], [115], [172]				[106], [107], [108], [109], [173]	[60], [64], [112], [117]
	Inception Score	[6], [50], [92], [119], [129], [132], [168], [169], [171], [174]	[151]	[116], [118], [120]					[117]
	Kernel Inception Distance		[116]	[115]					
	Perceptual Path Length						[64]		
Characteristic	Lines of code								[1]

TABLE V
(CONTINUED.)

Objective similarity	Hausdorff distance	[175]	[34], [176]				[60]
	F-Score		[34], [59], [61], [62], [63], [175], [177]	[140]			
	R-precision	[132], [168], [169], [171], [178]					
	IoU	[37], [42], [132]	[30], [34], [36], [43], [59], [61], [83], [98], [142], [154], [175], [179], [180], [181], [182], [183], [184], [185], [186], [187], [188]		[12], [15], [44]	[69]	
	Chamfer Distance		[5], [18], [30], [34], [43], [53], [58], [59], [63], [70], [76], [85], [128], [137], [138], [142], [177], [179], [181], [189]	[56], [68], [71], [72], [74], [79], [113], [140]			
	MAE		[17], [57]		[51]		
	MSE	[37]	[33], [36]	[38], [39]			
	RMSE		[2], [3], [17], [23], [40], [41], [57], [190], [191]		[4]		
	Log RMSE				[4]		
	K-fold cross-validation			[24]			
	Log-Likelihood						[65]
	Separability			[64]			
	SSE loss		[23]				
	1-NNA		[67], [70]	[68]		[69]	
	JSD		[135]	[74], [79], [140]		[71], [72], [77], [113], [136]	
	Kolmogorov-Smirnov test			[73]			
	Shapiro-Wilk test			[73]			

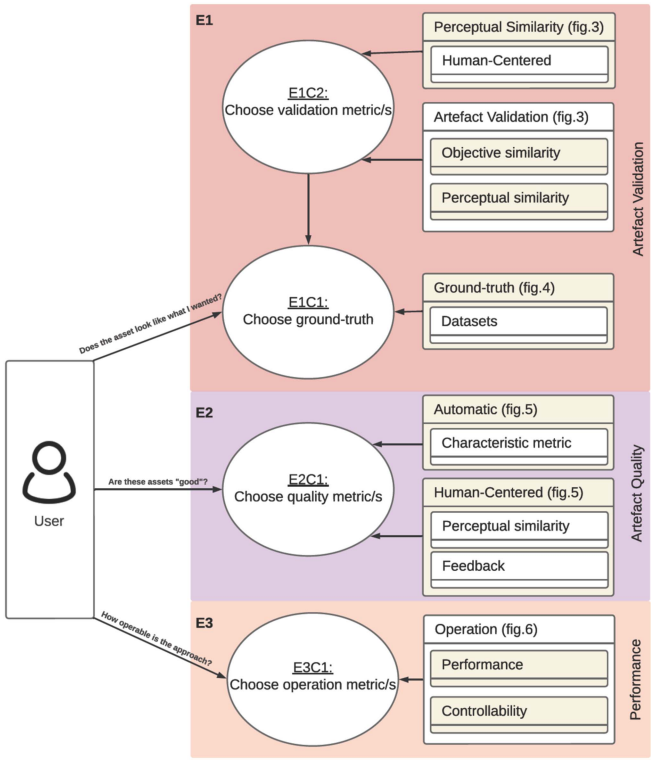


Fig. 7. The process for selecting appropriate operation, validity and quality metrics. Box colours match counterparts in Figs. 3–6.

such methods, often deep-learning based, will already use this form of validation as a means to optimise the generative model itself. Naturally, applying these metrics to the final generative system using data previously unseen to the model will assess its ability to perform the externally conceived task. To compare a method with existing approaches a benchmark dataset may be used, though the availability of a relevant dataset depends on how popular the asset type is for other generative systems. For example a generic 3D reconstruction task has many options for benchmark datasets, while a method that produces a highly specific type of artefact may require a bespoke dataset. Perceptual metrics require less direct conformity to a ground-truth. Here, artefacts can be compared with datasets that represent the general intended appearance of the artefacts. Metrics such as FID can be used to assess visual similarity between generated artefacts and a reference dataset without one-to-one correspondence. Alternatively, IS does not require any ground-truth data, but can only be used if the asset type belongs to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) categorisation [102]. Human-centered validation is achievable with any method, though it may not always be feasible due to the time it takes to recruit and access participants. The creator of a method designs or trains their system based on their conception of what attributes define the asset type. Human-centered perceptual similarity testing may be used to confirm if others deem the artefacts to meet their own conceptions. Typically a description, example or reference will be provided as a point of comparison for the assessor. This is not an assessment of quality, but a subjective assessment of whether artefacts are perceived as what they should be, i.e. is the method for generating chairs producing what can be deemed a chair. Where possible, both automatic and human validation metrics should be applied, though the former is only possible with externally conceived methods.

Artefact quality evaluation metrics can either be automatic or human-centered. These can be selected based on relevance to a use case. Automatic quality evaluation can be achieved through characteristic metrics. These are metrics that apply to specific characteristics of an artefact, and will be dependent on the asset type, data format or end use case of the artefact. For example, assessing the stability of furniture under physics simulation [132], or the quality of a mesh by the amount that it intersects with itself [62]. These make assumptions about what makes a quality asset e.g. a good chair will remain upright when exposed to gravity, and a good mesh surface will not intersect with itself. As such, it is difficult to systematically choose characteristic metrics as it is highly context specific. Alternatively, human-centered feedback through scoring can provide a more subjective, opinion based assessment of the quality of artefacts. Though finding participants and collecting this data will take more time than automated testing. Both should be used where possible.

Operation evaluation includes performance and controllability measures. Memory usage, speed and time complexity can be applied to any generative algorithm. These measures are beneficial for determining the resources required to use a given method. Additionally, in the case of generative methods as a cloud service, running cost may be assessed [151], while grammar based methods may benefit from assessing grammar precipitate [150] and encoded size [156]. Controllability should be considered based on the context of use. While there are no direct quantitative measures, users may be asked to indicate their satisfaction with the controls, whether they are too limited to achieve their goals, or too complex to comfortably use.

In most cases, multiple metrics of each category will be relevant. Where feasible, applying every relevant metric will provide the most precision in evaluating a method. This is not always possible due to time constraints, therefore, in such cases individual judgement should be used to prioritise which metrics are used.

Algorithm 1 presents the process for selecting all three types of metric based on the artefact asset type, the input type of the method, and the evaluation aim.

To illustrate the use of this algorithmic approach, we present a scenario where game developer A has produced a method for generating 3D models of castles using a GAN architecture. The method is using a seeded technique as the input is a randomised vector. First, game developer A validates if the outputs of the method are in fact ”castles”. They choose a dataset and validation metric (E1) by first examining if there are any ground-truth datasets available for buildings (E1C1). Looking at buildings in Fig. 4, there is LiDAR data from the U.K. environment agency, which has been used for finding roof shapes [185]. This is not so relevant to their use case. Not finding an existing similar dataset, they can use an unused portion of the dataset they used for training. They proceed with selecting a

Algorithm 1: Algorithm for Selecting Metrics.

procedure Selecting Evaluation Metricsasset_type, input_type, AIM ▷Prioritise metrics applied to same technique/$input\_type$, consider relevance of metrics based on the example uses in the literature.
  if AIM = Does the generated asset look like what I wanted? **then**
    return $Dataset, ValidationMetrics \leftarrow \mathbf{E1}(\mathbf{asset\_type}, \mathbf{input\_type})$
  ▷Choosing a validation metric.
  if AIM = Are these assets "good"? **then**
    return $QualityMetrics \leftarrow \mathbf{E2}(\mathbf{asset\_type}, \mathbf{input\_type})$
  ▷Choosing a quality metric.
  if AIM = How operable is this approach? **then**
    return $OperationMetrics \leftarrow \mathbf{E3}(\mathbf{asset\_type})$
  ▷Choosing an operation metric.

function E1: Artefact Validationasset_type, input_type
  $Dataset \leftarrow \mathbf{E1C1}(\mathbf{asset\_type}, \mathbf{input\_type})$
  $ValidationMetrics \leftarrow \mathbf{E1C2}(\mathbf{Dataset}, \mathbf{asset\_type}, \mathbf{input\_type})$
  return $Dataset, ValidationMetrics$

function E1C1: Choose ground-truthasset_type, input_type
  if Relevant dataset in Fig. 4 for given $asset\_type$ **then**
    if $input\_type$ matches dataset input type **then**
      returnDataset ← dataset matching $asset\_type$ and $input\_type$
    returnDataset ← dataset matching $asset\_type$
  if Ground-truth dataset can be created or has been used in training **then**
    returnDataset ← created or unused portion of training dataset

function E1C2: Choose validation metricdataset, asset_type, input_type
  if there is an expected output for every $input\_type$ **then**
    if $dataset$ contains $asset\_type$ and $input\_type$ **then**
      if $asset\_type$ is 3D **then**
        Metric options ← Relevant 3D Objective similarity metrics in Fig. 3 based on $asset\_type$.
        Metrics ← Find metric options in Tables IV and V.
      else if $asset\_type$ is 2D **then**
        Metric options ← Relevant 2D Objective similarity metrics in Fig. 3 based on $asset\_type$.
        Metrics ← Find metric options in Tables III and V.

  if $dataset$ matches the intended general appearance of $asset\_type$ **then**
    if $asset\_type$ is 3D **then**
      Metric options ← Relevant 3D Perceptual similarity metrics in Fig. 3 based on $asset\_type$.
      Metrics ← Find metric options in Tables IV and V.
    else if $asset\_type$ is 2D **then**
      Metric options ← Relevant 2D Perceptual similarity metrics in Fig. 3 based on $asset\_type$.
      Metrics ← Find metric options in Tables III and V.
  if consider human participation **then**
    Metrics ← Human Classification Score (Table V)
  if $asset\_type$ belongs to an ILSVRC [102] category **then**
    Metrics ← Inception Score (Table V)
  if appearance of $asset\_type$ can be put into words **then**
    if $asset\_type$ is 3D **then**
      Metrics ← ShapeGlot (Table IV)
    else if $asset\_type$ is 2D **then**
      Metrics ← ClipScore (Table III)

```
55:  returnMetrics
56:
57: function E2: Artefact Qualityasset_type, input_type
58:  if consider human participation then
59:  Metrics ← Human classification or feedback (Table V)
60:  if asset_type has measurable characteristics then
61:  if asset_type is 3D then
62:  Metric options ← Relevant characteristic metrics in Fig. 5 based on asset_type
63:  Metrics ← Find metric options in Tables IV and V.
64:  Metrics ← Devise new characteristic metric/s based on use case.
65:  else if asset_type is 2D then
66:  Metric options ← Relevant characteristic metrics in Fig. 5 based on asset_type
67:  Metrics ← Find metric options in Tables III and V.
68:  Metrics ← Devise new characteristic metric/s based on use case.
69:  returnMetrics
70:
71: function E3: Operationasset_type
72:  Metrics ← Memory usage, speed and time complexity (Table V)
73:  if consider human participation then
74:  Metrics ← User control satisfaction
75:  if intended to be run as a service then
76:  Assess running cost (Table V)
77:  if asset_type is grammar based then
78:  Assess grammar precipitate and encoded size (Table V)
```

validation metric (E1C2). As their technique is seeded, there is no clear expected output for every input. Once they validate that their dataset matches their intended artefact appearance, they consider general or 3D perceptual similarity metrics as their asset type is 3D. Due to the challenges of human participation, they decide not to use human classification. Therefore, they move on to other perceptual similarity methods which includes Inception Score, where "castles" is a category of ILSVRC, Fig. 3 and Table V. Then, developer A proceeds to to assess the quality of their artefacts (E2). As before, they reject human participation and use an automatic characteristic metric. They choose to measure mesh intersection ratio based on Fig. 5 and Table IV, as self-intersecting geometry can look bad and waste resources. Finally, to evaluate the operability of their method (E3), developer A assesses the memory usage, speed and time complexity. They once again reject human participation, so do not assess controllability. They run their GAN solution locally, not on cloud, and their method does not use grammars, therefore they do not assess running cost or grammar metrics.

VIII. Conclusion

To evaluate graphical asset generation methods, appropriate metrics are required. In this paper, a systematic survey of the literature has been conducted, and a framework for metric selection has been introduced. This framework addresses the need for a centralised point of reference for quantitatively evaluating graphical asset generation methods. In this framework, there are three types of evaluation, artefact validation, artefact quality and operation evaluation. These three types of evaluation include quantitative metrics that measure the efficacy, and facilitate comparison and benchmarking for methods that generate graphical game assets. These metrics map onto five desirable properties of PCG methods in games: Speed, Reliability, Expressivity/Diversity and Creativity/Believablility. Though no quantitative measures were found for assessing the controllability of methods in the literature, we suggest the consideration of user degrees of freedom, based on a method's input type. Though further research should be conducted into how controllability can be compared between methods.

References

[1] A. R. Willis, P. Ganesh, K. Volle, J. Zhang, and K. Brink, "Volumetric procedural models for shape representation," *Graph. Vis. Comput.*, vol. 4, 2021, Art. no. 200018.

[2] G. Nishida, A. Bousseau, and D. G. Aliaga, "Procedural modeling of a building from a single image," *Comput. Graph. Forum*, vol. 37, no. 2, pp. 415–429, 2018.

[3] J. Cao et al., "Facade geometry generation from low-resolution aerial photographs for building energy modeling," *Building Environ.*, vol. 123, pp. 601–624, 2017.

[4] C. Jiang et al., "Configurable 3D scene synthesis and 2D image rendering with per-pixel ground truth using stochastic grammars," *Int. J. Comput. Vis.*, vol. 126, pp. 920–941, 2018.

[5] J. Gao et al., "Get3D: A generative model of high quality 3D textured shapes learned from images," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 31 841–31 854, 2022.

[6] A. Ramesh et al., "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 8821–8831.

[7] R. Karp and Z. Swiderska-Chadaj, "Automatic generation of graphical game assets using GAN," in *Proc. 7th Int. Conf. Comput. Technol. Appl.*, 2021, pp. 7–12.

[8] A. Botchkarev, "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology," 2018, *arXiv: 1809.03006*.

[9] S. Risi and J. Togelius, "Increasing generality in machine learning through procedural content generation," *Nat. Mach. Intell.*, vol. 2, no. 8, pp. 428–436, 2020.

[10] K. Fukaya, D. Daylamani-Zad, and H. Agius, "Intelligent generation of graphical game assets: A conceptual framework and systematic review of the state of the art," 2023, *arXiv:2311.10129*.

[11] J. Li, J. Yang, J. Zhang, C. Liu, C. Wang, and T. Xu, "Attribute-conditioned layout GAN for automatic graphic design," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 10, pp. 4039–4048, Oct. 2021.

[12] J. Li, J. Yang, A. Hertzmann, J. Zhang, and T. Xu, "LayoutGAN: Synthesizing graphic layouts with vector-wireframe adversarial networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2388–2399, Jul. 2021.

[13] D. Lukes, J. Sarracino, C. Coleman, H. Peleg, S. Lerner, and N. Polikarpova, "Synthesis of web layouts from examples," in *Proc. 29th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2021, pp. 651–663.

[14] H. Hu, C. Zhang, and Y. Liang, "A study on the automatic generation of banner layouts," *Comput. Elect. Eng.*, vol. 93, 2021, Art. no. 107269.

[15] D. Smirnov, M. Gharbi, M. Fisher, V. Guizilini, A. Efros, and J. M. Solomon, "Marionette: Self-supervised sprite learning," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 5494–5505, 2021.

[16] O. Korn, M. Blatz, A. Rees, J. Schaal, V. Schwind, and D. Görlich, "Procedural content generation for game props? A study on the effects on user experience," *Comput. Entertain.*, vol. 15, no. 2, pp. 1–15, 2017.

[17] Y. R. Serpa and M. A. F. Rodrigues, "Towards machine-learning assisted asset generation for games: A study on pixel art sprite sheets," in *Proc. IEEE 18th Braz. Symp. Comput. Games Digit. Entertainment*, 2019, pp. 182–191.

[18] I.-C. Shen and B.-Y. Chen, "ClipGen: A deep generative model for clipart vectorization and synthesis," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 12, pp. 4211–4224, Dec. 2022.

[19] L. Gao, T. Wu, Y.-J. Yuan, M.-X. Lin, Y.-K. Lai, and H. Zhang, "TM-NET: Deep generative networks for textured meshes," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–15, 2021.

[20] W. Su, D. Du, X. Yang, S. Zhou, and H. Fu, "Interactive sketch-based normal map generation with deep neural networks," in *Proc. ACM Comput. Graph. Interactive Techn.*, vol. 1, no. 1, pp. 1–17, 2018.

[21] T. Wang and S. Kurabayashi, "Sketch2map: A game map design support system allowing quick hand sketch prototyping," in *Proc. IEEE Conf. Games*, 2020, pp. 596–599. [Online]. Available: https://api.semanticscholar.org/CorpusID:225050404

[22] P. Song, Y. Zheng, J. Jia, and Y. Gao, "Web3D-based automatic furniture layout system using recursive case-based reasoning and floor field," *Multimedia Tools Appl.*, vol. 78, no. 4, pp. 5051–5079, 2019.

[23] S. Kim, D. Kim, and S. Choi, "CityCraft: 3D virtual city creation from a single image," *Vis. Comput.*, vol. 36, no. 5, pp. 911–924, 2020.

[24] C. E. V. Muniz and W. L. O. dos Santos, "Generative Design applied to Cloud Modeling," in *Proc. IEEE 20th Braz. Symp. Comput. Games Digit. Entertainment*, 2021, pp. 79–86.

[25] E. Teng and R. Bidarra, "A semantic approach to patch-based procedural generation of urban road networks," in *Proc. 12th Int. Conf. Found. Digit. Games*, 2017, pp. 1–10.

[26] R. Dey, J. G. Doig, and C. Gatzidis, "Procedural feature generation for volumetric terrains using voxel grammars," *Entertainment Comput.*, vol. 27, pp. 128–136, 2018.

[27] R. Ratul, S. Sultana, J. Tasnim, and A. Rahman, "Applicability of space colonization algorithm for real time tree generation," in *Proc. 22nd Int. Conf. Comput. Inf. Technol.*, 2019, pp. 1–6.

[28] P. Kuang, D. Luo, and H. Wang, "Masked 3D conditional generative adversarial network for rock mesh generation," *Cluster Comput.*, vol. 22, no. 6, pp. 15 471–15 481, Nov. 2019.

[29] Y. Guan et al., "FAME: 3D shape generation via functionality-aware model evolution," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 4, pp. 1758–1772, Apr. 2022.

[30] C. Lin, T. Fan, W. Wang, and M. Nießner, "Modeling 3D shapes by reinforcement learning," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, UK, Springer, Aug. 23–28, 2020, pp. 545–561.

[31] S. Zhang and N. Xiao, "Detailed 3D human body reconstruction from a single image based on mesh deformation," *IEEE Access*, vol. 9, pp. 8595–8603, 2021.

[32] T. Shi, Z. Zou, Z. Shi, and Y. Yuan, "Neural rendering for game character auto-creation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1489–1502, Mar. 2022.

[33] Y. Shen, C. Zhang, H. Fu, K. Zhou, and Y. Zheng, "DeepSketchHair: Deep sketch-based 3D hair modeling," *IEEE Trans. on Vis.ization Comput. Graph.*, vol. 27, no. 7, pp. 3250–3263, Jul. 2021.

[34] Y. Wang, Z. Zhong, and J. Hua, "DeepOrganNet: On-the-fly reconstruction and visualization of 3D/4D lung models from single-view projections by deep deformation network," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 960–970, Jan. 2020.

[35] J. Togelius, N. Shaker, and M. J. Nelson, *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Berlin, Germany: Springer, 2016.

[36] J. K. Pontes, C. Kong, S. Sridharan, S. Lucey, A. Eriksson, and C. Fookes, "Image2Mesh: A learning framework for single image 3D reconstruction," *Lecture Notes Comput. Sci.*, vol. 11361, pp. 365–381, 2019.

[37] A. Sanghi et al., "Clip-forge: Towards zero-shot text-to-shape generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 603–18 613.

[38] R. R. Spick and J. Walker, "Realistic and textured terrain generation using GANs," in *Proc. 16th ACM SIGGRAPH Eur. Conf. Vis. Media Prod.*, 2019, pp. 1–10.

[39] Q. Tan, L. Gao, Y.-K. Lai, and S. Xia, "Variational autoencoders for deforming 3D mesh models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5841–5850.

[40] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong, "Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 285–295.

[41] G. Nishida, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and A. Bousseau, "Interactive sketching of urban procedural models," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, 2016.

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251. [Online]. Available: http://arxiv.org/abs/1703.10593

[43] S. Naritomi and K. Yanai, "3D mesh reconstruction of foods from a single image," in *Proc. Proc. ACM 3rd Workshop AIxFood, Co-Located*, pp. 7–11, 2021.

[44] P. Purkait, C. Zach, and I. Reid, "SG-VAE: Scene grammar variational autoencoder to generate new indoor scenes," *Lecture Notes Comput. Sci.*, vol. 12369, pp. 155–171, 2020.

[45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," *Adv. Neural Inf. Process. Syst.*, vol. 29, pp. 2234–2242, 2016.

[46] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 6629–6640.

[47] J. Lin, Y. Yuan, and Z. Zou, "MeinGame: Create a game character face from a single portrait," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 311–319.

[48] C. Saharia et al., "Photorealistic text-to-image diffusion models with deep language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 36 479–36 494.

[49] J. Yu et al., "Scaling autoregressive models for content-rich text-to-image generation," 2022, *arXiv:2206.10789*.

[50] M. Ding et al., "CogView: Mastering text-to-image generation via transformers," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 19 822–19 835, 2021.

[51] J. Zhang, C. Wang, C. Li, and H. Qin, "Example-based rapid generation of vegetation on terrain via CNN-based distribution learning," *Vis. Comput.*, vol. 35, no. 6/8, pp. 1181–1191, 2019.

[52] W. H. DosSantos, P. Ivson, and A. B. Raposo, "CAD shape grammar: Procedural generation for massive CAD model," in *Proc. 30th Conf. Graph. Patterns Images*, 2017, pp. 31–38.

[53] P.-S. Wang, C.-Y. Sun, Y. Liu, and X. Tong, "Adaptive O-CNN: A patch-based deep representation of 3D shapes," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–11, 2018.

[54] J. Urban Davis, F. Anderson, M. Stroetzel, T. Grossman, and G. Fitzmaurice, "Designing co-creative AI for virtual environments," in *Proc. 13th Conf. Creativity Cogn.*, 2021, pp. 1–11.

[55] M. Li et al., "Grains: Generative recursive autoencoders for indoor scenes," *ACM Trans. Graph.*, vol. 38, no. 2, pp. 1–16, 2019.

[56] J. Yang, K. Mo, Y.-K. Lai, L. J. Guibas, and L. Gao, "DSG-NET: Learning disentangled structure and geometry for 3D shape generation," *ACM Trans. Graph.*, vol. 42, no. 1, pp. 1–17, 2022.

[57] A. Khan et al., "Learning-detailed 3D face reconstruction based on convolutional neural networks from a single image," *Neural Comput. Appl.*, vol. 33, pp. 5951–5964, 2021.

[58] A. Yuniarti, A. Z. Arifin, and N. Suciati, "A 3D template-based point generation network for 3D reconstruction from single images," *Appl. Soft Comput.*, vol. 111, 2021, Art. no. 107749.

[59] C. Wen, Y. Zhang, C. Cao, Z. Li, X. Xue, and Y. Fu, "Pixel2Mesh++: 3D mesh generation and refinement from multi-view images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2166–2180, Feb. 2023.

[60] R. Konečný, S. Syllaiou, and F. Liarokapis, "Procedural modeling in archaeology: Approximating ionic style columns for games," in *Proc. 8th Int. Conf. Games Virtual Worlds Serious Appl.*, 2016, pp. 1–8.

[61] Y. Lu, Y. Wang, and G. Lu, "Single image shape-from-silhouettes," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 3604–3613.

[62] H. Li, W. Ye, G. Zhang, S. Zhang, and H. Bao, "Saliency guided subdivision for single-view mesh reconstruction," in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 1098–1107.

[63] N. Wang et al., "Pixel2Mesh: 3D mesh model generation via image guided deformation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3600–3613, Oct. 2021.

[64] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4396–4405.

[65] Y. Zhao, B. Deng, J. Huang, H. Lu, and X. S. Hua, "Stylized adversarial autoencoder for image generation," in *Proc. ACM Multimedia Conf.*, 2017, pp. 244–251.

[66] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "Pointflow: 3D point cloud generation with continuous normalizing flows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4541–4550.

[67] X. Zheng, Y. Liu, P. Wang, and X. Tong, "SDF-StyleGAN: Implicit SDF-based styleGAN for 3D shape generation," in *Computer Graphics Forum*, vol. 41. Hoboken, NJ, USA: Wiley Online Library, 2022, pp. 52–63.

[68] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu, "Neural wavelet-domain diffusion for 3D shape generation," in *Proc. SIGGRAPH Asia Conf. Papers*, 2022, pp. 1–9.

[69] X. Zeng et al., "LION: Latent point diffusion models for 3D shape generation," in *Proc. Adv. Neural Inf. Process. Syst.*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., Curran Associates, Inc., 2022, pp. 10 021–10 039.

[70] A.-C. Cheng, X. Li, S. Liu, M. Sun, and M.-H. Yang, "Autoregressive 3D shape generation via canonical mapping," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 89–104.

[71] Y. Li and G. Baciu, "HSGAN: Hierarchical graph learning for point cloud generation," *IEEE Trans. Image Process.*, vol. 30, pp. 4540–4554, 2021.

[72] Y. Li and G. Baciu, "SG-GAN: Adversarial self-attention GCN for point cloud topological parts generation," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 10, pp. 3499–3512, Oct. 2022.

[73] G. Ivanov, M. H. Petersen, K. Kovalský, K. Engberg, G. Palamas, and K. Eng-Berg, "An explorative design process for game map generation based on satellite images and playability factors," in *Proc. Int. Conf. Found. Digit. Games*, 2020, pp. 1–4.

[74] C. Öngün and A. Temizel, "LPMNet: Latent part modification and generation for 3D point clouds," *Comput. Graph.*, vol. 96, pp. 1–13, 2021.

[75] T. Friedrich, B. Hammer, and S. Menzel, "Voxel-based three-dimensional neural style transfer," in *Proc. 16th Int. Work-Conf. on Artif. Neural Netw. Adv. Comput. Intell.*, Springer, 2021, pp. 334–346.

[76] J. Pan, X. Han, W. Chen, J. Tang, and K. Jia, "Deep mesh reconstruction from single RGB images via topology modification networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9964–9973.

[77] S. Lim, M. Shin, and J. Paik, "Point cloud generation using deep adversarial local features for augmented and mixed reality contents," *IEEE Trans. Consum. Electron.*, vol. 68, no. 1, pp. 69–76, Feb. 2022.

[78] S. Li, M. Liu, and C. Walder, "EditVAE: Unsupervised parts-aware controllable 3D point cloud shape generation," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1386–1394.

[79] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Learning representations and generative models for 3D point clouds," in *PRoc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 40–49.

[80] F. Tong, M. Nakao, S. Wu, M. Nakamura, and T. Matsuda, "X-ray2Shape: Reconstruction of 3D liver shape from a single 2D projection image," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2020, pp. 1608–1611.

[81] X. Sun and Z. Lian, "EasyMesh: An efficient method to reconstruct 3D mesh from a single image," *Comput. Aided Geometric Des.*, vol. 80, 2020, Art. no. 101862.

[82] V. A. Knyaz, V. V. Kniaz, and F. Remondino, "Image-to-voxel model translation with conditional adversarial networks," *Lecture Notes Comput. Sci.*, vol. 11129, no. 1, pp. 601–618, 2019.

[83] S. N. Silva Junior, F. C. Chamone, R. C. Ferreira, and E. R. Nascimento, "A 3D modeling methodology based on a concavity-aware geometric test to create 3D textured coarse models from concept art and orthographic projections," *Comput. Graph.*, vol. 76, pp. 73–83, 2018.

[84] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu, "Neural template: Topology-aware reconstruction and disentangled generation of 3D meshes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 572–18 582.

[85] J. Lei, S. Sridhar, P. Guerrero, M. Sung, N. Mitra, and L. J. Guibas, "Pix2Surf: Learning parametric 3D surface models of objects from images," *Lecture Notes Comput. Sci.*, vol. 12363, pp. 121–138, 2020.

[86] C. Öngün and A. Temizel, "Paired 3D model generation with conditional generative adversarial networks," *Lecture Notes Comput. Sci.*, vol. 11129, pp. 473–487, 2019.

[87] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau, "An exact graph edit distance algorithm for solving pattern recognition problems," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods - Volume 1*, 2015, pp. 271–278.

[88] H. Cai, Y. Guo, Z. Peng, and J. Zhang, "Landmark detection and 3D face reconstruction for caricature using a nonlinear parametric model," *Graphical Models*, vol. 115, 2021, Art. no. 101103.

[89] G. Shamai, R. Slossberg, and R. Kimmel, "Synthesizing facial photometries and corresponding geometries using generative adversarial networks," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 15, no. 3s, pp. 1–24, 2019.

[90] P. Ji, M. Zeng, and X. Liu, "View consistent 3D face reconstruction using Siamese encoder-decoders," *Commun. Comput. Inf. Sci.*, vol. 1314, pp. 209–223, 2020.

[91] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 684–10 695.

[92] Y. Fan, Y. Liu, G. Lv, S. Liu, G. Li, and Y. Huang, "Full face-and-head 3D model with photorealistic texture," *IEEE Access*, vol. 8, pp. 210 709–210 721, 2020.

[93] Z. Wang, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[94] N. K. Yadav, S. K. Singh, and S. R. Dubey, "CSA-GAN: Cyclic synthesized attention guided generative adversarial network for face synthesis," *Appl. Intell.*, vol. 52, pp. 12704–12723, 2022.

[95] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

[96] N. Xiang, L. Wang, T. Jiang, Y. Li, X. Yang, and J. Zhang, "Single-image mesh reconstruction and pose estimation via generative normal map," in *Proc. 32nd Int. Conf. Comput. Animation Social Agents*, 2019, pp. 79–84.

[97] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Springer International Publishing, 2016, pp. 649–666.

[98] Z. Lun, E. Kalogerakis, R. Wang, and A. Sheffer, "Functionality preserving shape style transfer," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–14, Dec. 2016.

[99] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[100] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[101] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying MMD GANs," 2018, *arXiv: 1801.01401*.

[102] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[103] A. Das, Y. Yang, T. M. Hospedales, T. Xiang, and Y.-Z. Song, "Cloud2curve: Generation and vectorization of parametric sketches," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7088–7097.

[104] X. Gao, Y. Tian, and Z. Qi, "RPD-GAN: Learning to draw realistic paintings with generative adversarial network," *IEEE Trans. Image Process.*, vol. 29, pp. 8706–8720, 2020.

[105] Y. Shu et al., "GAN-based multi-style photo cartoonization," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 10, pp. 3376–3390, Oct. 2022.

[106] H. Kazemi, S. M. Iranmanesh, and N. Nasrabadi, "Style and content disentanglement in generative adversarial networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 848–856.

[107] C. Sun, Y. Zhou, and Y. Han, "Automatic generation of architecture facade for historical urban renovation using generative adversarial network," *Building Environ.*, vol. 212, 2022, Art. no. 108781.

[108] Z. Liu, P. Dai, R. Li, X. Qi, and C.-W. Fu, "ISS: Image as stetting stone for text-guided 3D shape generation," 2022, *arXiv:2209.04145*.

[109] Y. Fukatsu and M. Aono, "3D mesh generation by introducing extended attentive normalization," in *Proc. 8th Int. Conf. Adv. Inform. Concepts Theory Appl.*, 2021, pp. 1–6.

[110] N. Nauata, K.-H. Chang, C.-Y. Cheng, G. Mori, and Y. Furukawa, "House-GAN: Relational generative adversarial networks for graph-constrained house layout generation," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, Aug. 23–28, 2020, pp. 162–177.

[111] D. W. Shu, S. W. Park, and J. Kwon, "3D point cloud generative adversarial network based on tree structured graph convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3859–3868.

[112] R. Li, X. Li, K.-H. Hui, and C.-W. Fu, "SP-GAN: Sphere-guided 3D shape generation and manipulation," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–12, 2021.

[113] R. Or-El, X. Luo, M. Shan, E. Shechtman, J. J. Park, and I. Kemelmacher-Shlizerman, "StyleSDF: High-resolution 3D-consistent image and geometry generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 503–13 513.

[114] P. Henderson, V. Tsiminaki, and C. H. Lampert, "Leveraging 2D data to learn textured 3D mesh generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7498–7507.

[115] Y. Wang, A. Dantcheva, and F. Bremond, "From attribute-labels to faces: Face generation using a conditional generative adversarial network," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 692–698.

[116] A. Fadaeddini, B. Majidi, and M. Eshghi, "A case study of generative adversarial networks for procedural synthesis of original textures in video games," in *Proc. IEEE 2nd Nat. 1st Int. Digit. Games Res. Conf. Trends Technol. Appl.*, 2018, pp. 118–122.

[117] H. Schulze, D. Yaman, and A. Waibel, "CAGAN: Text-to-image generation with combined attention generative adversarial networks," in *Proc. DAGM German Conf. Pattern Recognit.*, Springer, 2021, pp. 392–404.

[118] H. Wang, N. Schor, R. Hu, H. Huang, D. Cohen-Or, and H. Huang, "Global-to-local generative model for 3D shapes," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–10, 2018.

[119] J. Hessel, A. Holtzman, M. Forbes, R. Le Bras, and Y. Choi, "CLIPScore: A reference-free evaluation metric for image captioning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 7514–7528.

[120] D. H. Park, S. Azadi, X. Liu, T. Darrell, and A. Rohrbach, "Benchmark for compositional text-to-image synthesis," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track*, 2021, pp. 1–13.

[121] P. Achlioptas, J. Fan, R. Hawkins, N. Goodman, and L. J. Guibas, "ShapeGlot: Learning language for shape differentiation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8938–8947.

[122] R. Fu, X. Zhan, Y. Chen, D. Ritchie, and S. Sridhar, "ShapeCrafter: A recursive text-conditioned 3D shape generation model," in *Proc. Adv. Neural Inf. Process. Syst.*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022, pp. 8882–8895.

[123] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.

[124] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[125] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.

[126] G. Gkioxari, J. Johnson, and J. Malik, "Mesh R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9784–9794.

[127] K. Yue, Y. Li, and H. Li, "Progressive semantic image synthesis via generative adversarial network," in *Proc. IEEE Vis. Commun. Image Process.*, 2019, pp. 1–4.

[128] J. Kim, H. Kim, H. Nam, J. Park, and S. Lee, "TextureMe: High-quality textured scene reconstruction in real time," *ACM Trans. Graph.*, vol. 41, no. 3, pp. 1–18, 2022.

[129] W. Xia, Y. Yang, and J.-H. Xue, "Cali-sketch: Stroke calibration and completion for high-quality face image generation from human-like sketches," *Neurocomputing*, vol. 460, pp. 256–265, 2021.

[130] Z. Liu, Y. Wang, X. Qi, and C.-W. Fu, "Towards implicit text-guided 3D shape generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17 896–17 906.

[131] S. Bangay, "Deterministic procedural generation of mesh detail through gradient tiling," in *Proc. Australas. Comput. Sci. Week MultiConf.*, 2017, pp. 1–10.

[132] R. K. Jones et al., "ShapeAssembly: Learning to generate programs for 3D shape structure synthesis," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–20, 2020.

[133] R. Klokov, E. Boyer, and J. Verbeek, "Discrete point flow networks for efficient point cloud generation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 694–710.

[134] L. Gao et al., "SDM-NET: Deep generative network for structured deformable mesh," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–15, 2019.

[135] Y. Lin, J. Guo, Y. Gao, Y.-F. Li, Z. Wang, and L. Khan, "Generating point cloud from single image in the few shot scenario," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 2834–2842.

[136] J. Yang, Y. Li, and L. Yang, "Shape transformer nets: Generating viewpoint-invariant 3D shapes from a single image," *J. Vis. Commun. Image Representation*, vol. 81, 2021, Art. no. 103345.

[137] T. Kimura, T. Matsubara, and K. Uehara, "Chartpointflow for topology-aware 3D point cloud generation," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 1396–1404.

[138] Z. Du, H. Shen, X. Li, and M. Wang, "3D building fabrication with geometry and texture coordination via hybrid GAN," *J. Ambient Intell. Humanized Comput.*, vol. 13, pp. 5177–5188, 2022.

[139] J. Lei, S. Sridhar, P. Guerrero, M. Sung, N. Mitra, and L. J. Guibas, "Pix2Surf: Learning parametric 3D surface models of objects from images," in *Proc. 16th Eur. Conf. Comput. Vis.*, Glasgow, U.K., Springer, Aug. 23–28, 2020, pp. 121–138.

[140] H. Yu, C. Cheang, Y. Fu, and X. Xue, "Multi-view shape generation for a 3D human-like body," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 19, no. 1, pp. 1–22, 2023.

[141] K. Yang, J. Lu, S. Hu, and X. Chen, "Deep 3D modeling of human bodies from freehand sketching," in *Proc. 27th Int. Conf. MultiMedia Model.*, Prague, Czech Republic, Springer, Jun. 22–24, 2021, pp. 36–48.

[142] J. Malik et al., "HandVoxNet: Deep voxel-based network for 3D hand shape and pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7113–7122.

[143] T. Shi, Y. Yuan, C. Fan, Z. Zou, Z. Shi, and Y. Liu, "Face-to-parameter translation for game character auto-creation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 161–170.

[144] J. Jeon, Y. Jung, H. Kim, and S. Lee, "Texture map generation for 3D reconstructed scenes," *Vis. Comput.*, vol. 32, pp. 955–965, 2016.

[145] M. Slavcheva, W. Kehl, N. Navab, and S. Ilic, "SDF-2-SDF: Highly accurate 3D object reconstruction," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Springer, Oct. 11–14, 2016, pp. 680–696.

[146] T. T. Niemirepo, M. Viitanen, and J. Vanne, "Open3DGen: Open-source software for reconstructing textured 3D models from RGB-D images," in *Proc. 12th ACM Multimedia Syst. Conf.*, 2021, pp. 12–22.

[147] Y. Gao, Y. Yao, and Y. Jiang, "Multi-target 3D reconstruction from RGB-D data," in *Proc. 2nd Int. Conf. Comput. Sci. Softw. Eng.*, 2019, pp. 184–191.

[148] J. Ren et al., "Intuitive and efficient roof modeling for reconstruction and synthesis," 2021, *arXiv:2109.07683*.

[149] S. Li, S. Su, J. Lin, G. Cai, and L. Sun, "Deep 3D caricature face generation with identity and structure consistency," *Neurocomputing*, vol. 454, pp. 178–188, 2021.

[150] D. Li, D. Hu, Y. Sun, and Y. Hu, "3D scene reconstruction using a texture probabilistic grammar," *Multimedia Tools Appl.*, vol. 77, no. 21, pp. 28 417–28 440, 2018.

[151] M. Bhatt et al., "Design and deployment of Photo2Building: A cloud-based procedural modeling tool as a service," in *Proc. Pract. Experience Adv. Res. Comput.*, 2020, pp. 132–138.

[152] J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau, "3D sketching using multi-view deep volumetric prediction," in *Proc. ACM Comput. Graph. Interactive Techn.*, vol. 1, no. 1, pp. 1–22, Jul. 2018.

[153] J. Delanoy, D. Coeurjolly, J.-O. Lachaud, and A. Bousseau, "Combining voxel and normal predictions for multi-view 3D sketching," *Comput. Graph.*, vol. 82, pp. 65–72, 2019.

[154] M. Becher, M. Krone, G. Reina, and T. Ertl, "Feature-based volumetric terrain generation," in *Proc. 21st ACM SIGGRAPH Symp. Interactive 3D Graph. Games*, 2017, pp. 1–9.

[155] R. H. Kazi, T. Grossman, H. Cheong, A. Hashemi, and G. Fitzmaurice, "Dreamsketch: Early stage 3D design explorations with sketching and generative design," in *Proc. 30th Annu. ACM Symp. User Interface Softw. Technol.*, 2017, pp. 401–414.

[156] M. Fiser, B. Benes, J. G. Galicia, M. Abdul-Massih, D. G. Aliaga, and V. Krs, "Learning geometric graph grammars," in *Proc. 32nd Spring Conf. Comput. Graph.*, 2016, pp. 7–15.

[157] M. C. Green, C. Salge, and J. Togelius, "Organic building generation in Minecraft," in *Proc. 14th Int. Conf. Found. Digit. Games*, 2019, pp. 1–7.

[158] P. Song, Y. Zheng, and J. Jia, "Web3D learning platform of furniture layout based on case-based reasoning and distance field," in *Proc. 11th Int. Conf. E-Learn. Games*, Bournemouth, U.K., Springer, Jun. 26–28, 2017, pp. 235–250.

[159] R. Fischer, P. Dittmann, R. Weller, and G. Zachmann, "Autobiomes: Procedural generation of multi-biome landscapes," *Vis. Comput.*, vol. 36, pp. 2263–2272, 2020.

[160] V. Krs, R. Měch, M. Gaillard, N. Carr, and B. Benes, "PICO: Procedural iterative constrained optimizer for geometric modeling," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 10, pp. 3968–3981, Oct. 2021.

[161] K. Golubev, A. Zagarskikh, and A. Karsakov, "Dijkstra-based terrain generation using advanced weight functions," *Procedia Comput. Sci.*, vol. 101, pp. 152–160, 2016.

[162] I. Antoniuk and P. Rokita, "Generation of complex underground systems for application in computer games with schematic maps and l-systems," in *Proc. Int. Conf. Comput. Vis. Graph.*, Warsaw, Poland, Springer, Sep. 19–21, 2016, pp. 3–16.

[163] K. Franke and H. Müller, "Procedural generation of 3D karst caves with speleothems," *Comput. Graph.*, vol. 102, pp. 533–545, 2022.

[164] X.-Z. Li, R. Weller, and G. Zachmann, "Astrogen–procedural generation of highly detailed asteroid models," in *Proc. IEEE 15th Int. Conf. Control Automat. Robot. Vis.*, 2018, pp. 1771–1778.

[165] Z. Canfes, M. F. Atasoy, A. Dirik, and P. Yanardag, "Text and image guided 3D avatar generation and manipulation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2023, pp. 4421–4431.

[166] T. Qiao, J. Zhang, D. Xu, and D. Tao, "Learn, imagine and create: Text-to-image generation from prior knowledge," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 887–897, 2019.

[167] T. Qiao, J. Zhang, D. Xu, and D. Tao, "MirrorGAN: Learning text-to-image generation by redescription," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1505–1514.

[168] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2016, pp. 318–335.

[169] D. Peng, W. Yang, C. Liu, and S. Lü, "SAM-GAN: Self-attention supporting multi-stage generative adversarial networks for text-to-image synthesis," *Neural Netw.*, vol. 138, pp. 57–67, 2021.

[170] X.-M. Du, F. Li, H.-R. Yan, R. Fu, and Y. Zhou, "Terrain edge stitching based on least squares generative adversarial networks," in *Proc. IEEE 16th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, 2019, pp. 157–161.

[171] H. Hou, J. Huo, J. Wu, Y.-K. Lai, and Y. Gao, "MW-GAN: Multi-warping GAN for caricature generation with multi-style geometric exaggeration," *IEEE Trans. Image Process.*, vol. 30, pp. 8644–8657, 2021.

[172] B. Li, X. Qi, T. Lukasiewicz, and P. Torr, "Controllable text-to-image generation," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 2065–2075, 2019.

[173] P. Mittal, Y.-C. Cheng, M. Singh, and S. Tulsiani, "AutoSDF: Shape priors for 3D completion, reconstruction and generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 306–315.

[174] F. Kong, N. Wilson, and S. Shadden, "A deep-learning approach for direct whole-heart mesh reconstruction," *Med. Image Anal.*, vol. 74, 2021, Art. no. 102222.

[175] X. Li, K. Ping, X. Gu, and M. He, "3D shape reconstruction of furniture object from a single real indoor image," in *Proc. IEEE 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, 2020, pp. 101–104.

[176] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 867–876.

[177] W. Ling and D. He, "A deep learning method based on structure inference for single-view 3D reconstruction," in *Proc. IEEE 5th Adv. Inf. Technol. Electron. Automat. Control Conf.*, 2021, pp. 2240–2244.

[178] M. Zhao, G. Xiong, M. Zhou, Z. Shen, and F.-Y. Wang, "3D-RVP: A method for 3D object reconstruction from a single depth view using voxel and point," *Neurocomputing*, vol. 430, pp. 94–103, 2021.

[179] Y. Dongsheng, K. Ping, and X. Gu, "3D reconstruction based on GAT from a single image," in *Proc. 17th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process.*, 2020, pp. 122–125.

[180] Z. Weng and S. Yeung, "Holistic 3D human and scene mesh estimation from single view images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 334–343.

[181] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D mesh renderer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3907–3916.

[182] H. Xu and J. Bai, "ARShape-net: Single-view image oriented 3D shape reconstruction with an adversarial refiner," in *Proc. Int. Conf. Artif. Intell.*, Springer, 2021, pp. 638–649.

[183] H. Xie, H. Yao, X. Sun, S. Zhou, and X. Tong, "Weighted voxel: A novel voxel representation for 3D reconstruction," in *Proc. 10th Int. Conf. Internet Multimedia Comput. Service*, 2018, pp. 1–4.

[184] V. V. Kniaz, V. A. Knyaz, F. Remondino, A. Bordodymov, and P. Moshkantsev, "Image-to-voxel model translation for 3D scene reconstruction and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 105–124.

[185] H. Zeng, J. Wu, and Y. Furukawa, "Neural procedural reconstruction for residential buildings," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 737–753.

[186] S. Saito, L. Hu, C. Ma, H. Ibayashi, L. Luo, and H. Li, "3D hair synthesis using volumetric variational autoencoders," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–12, 2018.

[187] J. Pan, J. Li, X. Han, and K. Jia, "Residual MeshNet: Learning to deform meshes for single-view 3D reconstruction," in *Proc. IEEE Int. Conf. 3D Vis.*, 2018, pp. 719–727.

[188] H. Kuang, Y. Ding, X. Ma, and X. Liu, "3D face reconstruction with texture details from a single image based on gan," in *Proc. IEEE 11th Int. Conf. Measuring Technol. Mechatronics Automat.*, 2019, pp. 385–388.

[189] C. Li, T. Guan, M. Yang, and C. Zhang, "Combining data-and-model-driven 3D modelling (CDMD3DM) for small indoor scenes using RGB-D data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 180, pp. 1–13, 2021.

[190] L. Fei-Fei, J. Deng, O. Russakovsky, A. Berg, and K. Li, "ImageNet dataset," 2021. [Online]. Available: https://image-net.org/

[191] A. X. Chang et al., "ShapeNet dataset," 2015. [Online]. Available: https://shapenet.org/

[192] Z. Wu et al., "ModelNet dataset," 2015. [Online]. Available: https://modelnet.cs.princeton.edu/

[193] X. Sun et al., "Pix3D dataset," 2018. [Online]. Available: http://pix3d.csail.mit.edu/

[194] K. Mo et al., "PartNet dataset," 2019. [Online]. Available: https://www.shapenet.org/download/parts

[195] A. Yu and K. Grauman, "UT Zappos50K dataset," 2017. [Online]. Available: https://vision.cs.utexas.edu/projects/finegrained/utzap50k/

[196] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200–2011 dataset," California Inst. Technol., 2011. [Online]. Available: http://www.vision.caltech.edu/datasets/cub_200_2011/

[197] Z. Zhang et al., "Customizable GAN: Customizable image synthesis based on adversarial learning," in *Proc. 27th Int. Conf. Neural Inf. Process.*, Bangkok, Thailand, Springer, Nov. 18–22, 2020, pp. 336–344.

[198] J. A. Krosnick, "Questionnaire design," in *The Palgrave Handbook of Survey Research*. Berlin, Germany: Springer, 2018, pp. 439–455.

[199] L. South, D. Saffo, O. Vitek, C. Dunne, and M. A. Borkin, "Effective use of likert scales in visualization evaluations: A systematic review," *Comput. Graph. Forum*, vol. 41, no. 3, pp. 43–55, 2022.

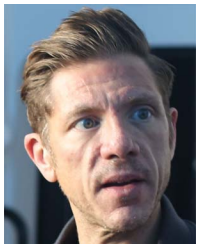
[200] C.-F. Chen and E. S. Rosenberg, "Dynamic omnidirectional texture synthesis for photorealistic virtual content creation," in *Proc. IEEE Int. Symp. Mixed Augmented Reality Adjunct*, 2018, pp. 85–90.

[201] M. Resnick et al., Design principles for tools to support creative thinking, 2005. [Online]. Available: https://doi.org/10.1184/R1/6621917.v1

[202] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, pp. 4–7, 1995.

Kaisei Fukaya received the BSc degree in game design and development from the University of Greenwich. He is currently working toward the PhD degree with the College of Engineering, Design and Physical Science, Brunel University London. His research research interests include applications of AI, machine learning, tools for game development, procedural asset generation and serious games. He has published his research findings widely in journals and presented his work at several conferences.

Damon Daylamani-Zad received the BSc degree in software engineering from the University of Tehran, the MSc degree in multimedia computing and the PhD degree in electronic and computer engineering both from Brunel University London where he has also been an EPSRC research fellow. He is a senior lecturer in AI and Games with the College of Engineering, Design and Physical Science at Brunel University London. He is a fellow of the British Computing Society. His research interests focus on applications of artificial intelligence and machine learning, collaborative games, serious gaming, and user modelling and personalisation, as well as application of evolutionary algorithms in creative computing. He has published his research findings widely in journals, edited books and presented his work at several conferences including those hosted by the IEEE.



Harry Agius received the BSc, MSc, and PhD degrees in computing and information systems from LSE, University of London, U.K. He is a senior lecturer in computing with the Brunel Design School, Brunel University London, U.K., and a fellow of the British Computer Society. His research more than the past three decades has focused on various areas of creative computing involving digital media and AI, most notably games and personalisation. He currently serves as section editor for the "Digital Games, Virtual Reality, and Augmented Reality" track of the "*Multimedia Tools and Applications*" journal (Springer) and was coeditor of the "Handbook of Digital Games" (IEEE Press, 2014).