DOI: 10.1111/cgf.14024
EUROGRAPHICS 2020
R. Mantiuk and V. Sundstedt
(Guest Editors)

Volume 39 (2020), Number 2
STAR – State of The Art Report

A Survey on Sketch Based Content Creation: from the Desktop to Virtual and Augmented Reality

Sukanya Bhattacharjee and Parag Chaudhuri

Indian Institute of Technology Bombay, India

Abstract

Sketching is one of the most natural ways for representing any object pictorially. It is however, challenging to convert sketches to 3D content that is suitable for various applications like movies, games and computer aided design. With the advent of more accessible Virtual Reality (VR) and Augmented Reality (AR) technologies, sketching can potentially become a more powerful yet easy-to-use modality for content creation. In this state-of-the-art report, we aim to present a comprehensive overview of techniques related to sketch based content creation, both on the desktop and in VR/AR. We discuss various basic concepts related to static and dynamic content creation using sketches. We provide a structured review of various aspects of content creation including model generation, coloring and texturing, and finally animation. We try to motivate the advantages that VR/AR based sketching techniques and systems can offer into making sketch based content creation a more accessible and powerful mode of expression. We also discuss and highlight various unsolved challenges that current sketch based techniques face with the goal of encouraging future research in the domain.

CCS Concepts

• ***Computing methodologies*** → ***Graphics systems and interfaces;*** • ***Human-centered computing*** → *Mixed / augmented reality; Virtual reality;*

1. Introduction

3D modelling is a widely studied area in computer graphics. 3D content creation in computer graphics deals with the creation of 3D models, which can be possibly textured and animated as well. These models are used in various applications ranging from animation movies, computer aided design to virtual and augmented reality experiences. Sketching has always been one of the most naturally used way to pictorially depict anything ranging from physical objects to abstract ideas. It has therefore become a commonly used method for creating 3D content.

Many commercial modelling packages like Blender [Ble17], Maya [Aut17] and 3ds Max [Aut19] have provided easy-to-use yet difficult-to-learn tools for creating 3D contents from sketches. Over the years, the focus for the researchers in this area has been to propose methods that can interpret sketches and create plausible 3D content from them. The sketches can be drawn on a tablet, a phone or a desktop screen which essentially means they are 2D in nature. Though 2D sketches perform well in representing 3D objects but the skill of providing strong depth and shape cues for a complete 3D structure is difficult to master. Virtual and augmented reality have enabled sketching in 3D and this allows the artist to get a better understanding of the depicted object. However, 3D sketching comes with its own challenges as it is quite difficult to create in the free 3D space and stay consistent while drawing. A well studied solution that uses best of both the worlds is to let the user draw a 2D sketch and then the system generates a 3D sketch or model from it. Most of the methods used for this purpose trade ease of use for creative liberty that they afford the artist. Among many recent advancements, the methods based upon this solution approach have been proposed in papers like [IMT99,OSJ11,BJD*12,HGY17] and many others.

Coloring and texturing aspects of the 3D modelling are integral to the content creation process as well. The sketched strokes have always served as a medium to represent various kind of patterns, for example the crease on a skirt can be depicted by straight lines drawn on top of it. Distinguishing the pattern strokes that are used for shading or texturing, from the contour strokes that are used to outline a shape while constructing a model, is a non-trivial task during automated model generation. The coloring and texturing process in general is accomplished by repeated strokes drawn in a region of the sketch. A mapping is computed to choose a color or texture for each part of the 3D model from some part of the 2D sketch. Some of the methods given in [XSS08,CBWG10,BCV*15] and many others focus on these aspects but suffer from the shortcomings inherent to the 2D or 3D nature of the sketches as well as inability to interpret the texture information from the sketches.

While designing static content involves the above mentioned issues, dynamic content design needs to address some additional as-

pects. Traditionally, sequences of the sketches have been used to represent animations. Sketches have also been used to depict the path a model follows while moving. Methods like [TBvdP04] attempt to model such information from sketches.

Advancement in the VR and AR technologies are allowing us to re-imagine the way we look at the classical problems in computer graphics like interactive 3D modelling. Interfaces based upon VR strive to translate the intuitiveness of sketching using a pen-and-paper to 3D content creation. Commercial applications like Tilt Brush [Goo17c], Gravity Sketch [Goo17b], Blocks [Goo17a], ShapeLab [Leo17], Quill [Fac19b] and MasterpieceVR [Mas19] are some examples of such interfaces. Recent work like [XSS08, AHKG*18] also explores the AR/VR platforms to come up with better methods for 3D modelling.

In this state-of-art report, we discuss various aspects of sketch based content creation with the aim of motivating the usefulness of VR or AR assisted sketch based interfaces. We cover some of the fundamental concepts used in sketch based content creation and AR/VR technologies in Section 2. We then present a structured overview in Section 3 for each component of sketch based content creation, which spans various kinds of approaches. We discuss the traditional methods as well as the ones that use the AR/VR platforms. We briefly discuss virtual sculpting in Section 4 as a complimentary paradigm for 3D content creation that is also being translated to VR or AR interfaces from the desktop. In Section 5, we discuss various usability issues with respect to the sketch based interfaces. Finally, we discuss various open problems in Section 6 and conclude by pointing out the future challenges that remain to be addressed in the area.

2. Terminologies and background

In this section, we discuss the relevant concepts and terminologies for sketch based content creation and the related VR/AR technologies.

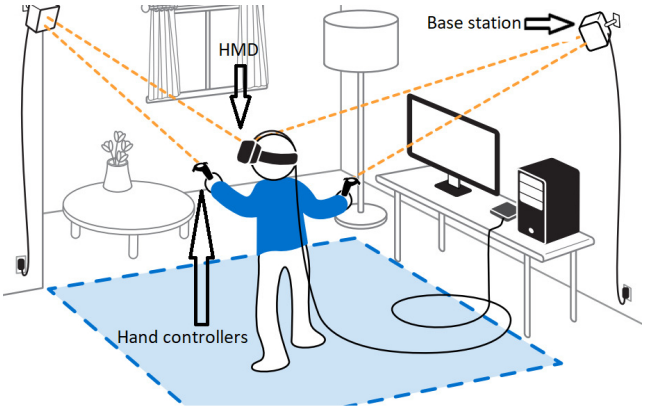


Figure 1: *Generic VR Setup. Based on image from* `www.octopusrift.com`.

2.1. Generic VR setup

Modern VR systems like the HTC Vive [HTC19] and the Oculus Rift [Fac19a] can generally track the hand and gaze (or at least head rotation) of the users while visually immersing them in any simulated environment. The tracking enables the VR system to render any stroke sketched by the artist in free 3D space. As shown in Figure 1, a VR setup can comprise of a pair of base stations, a pair of hand controllers and a head mounted device (HMD), all of them connected to a desktop through cables. The position of the HMD and hand controllers can be found relative to the base stations, thus allowing tracking in all 6 degrees of freedom of movement. The HMD is used to track the viewpoint of the artist, according to which the sketched strokes are displayed. The hand controllers have triggers that are pressed to draw the strokes or select various operations associated with sketching. A detailed description of the display and tracking technologies for the VR devices can be found in [KAS*19].

2.2. Generic AR setup

Augmented reality or AR, in contrast to VR, superimposes virtual content on top of the real world. A camera attached to the user's viewpoint looks at the real world. The movement of the camera with respect to the real world is ascertained by tracking using computer vision algorithms. This allows a real-time renderer to place virtual content within the view of the user that moves coherently with the real world. The display for AR systems can employ optical or video see-through HMD's. Interested users can refer to [SH16] for more detailed information on AR. Sketches made in VR or AR setups are referred to as 3D sketches in this survey. Next, we discuss terminology that lets us describe and work with sketch strokes.

2.3. Sketch strokes

Sketches comprise of one or more strokes in general. This set of strokes in a sketch is used for determining the structure, color, pattern, or path of motion (in case of dynamic content). These strokes can be 2D or 3D in nature depending on the medium used for drawing them as well as how they are interpreted by the interface (Figures 2a and 2c). Each stroke is approximated as a polyline or a curve, for extracting the features represented by it.

Strokes in a sketch are broadly classified as *feature* strokes and *silhouette* strokes. Strokes that define the outer shape of the sketch and bound it, are known as silhouette or contour strokes. These strokes are generally used to infer the overall shape of the corresponding 3D model. On the other hand, feature strokes define the inner structure or the surface detail of the object being sketched. That is, these strokes represent the inherent geometry of the corresponding 3D model (Figure 2b). Annotating a stroke means assigning some semantic information to the stroke that is useful in understanding the intent of the user.

Next, we shall discuss some crucial concepts related to the strokes, which are extensively used in the design of sketch based systems.

2.3.1. Geometric properties

Geometric properties of a stroke define various relations between two or more strokes, which indicate important structural information about the 3D model to be constructed. If the sketch is used for

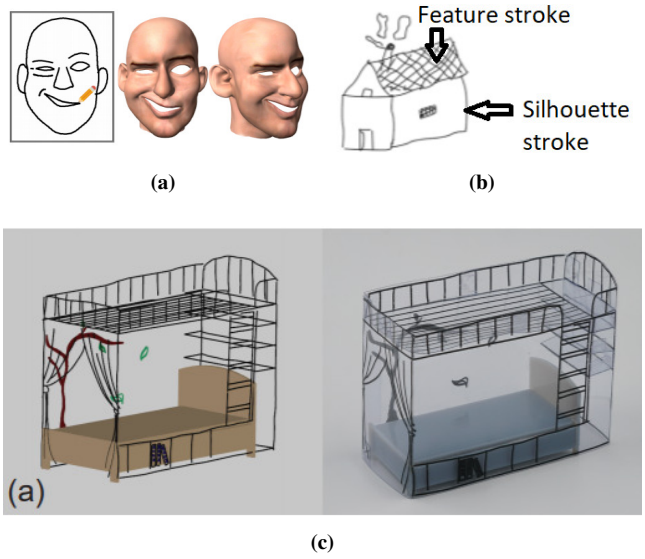


Figure 2: *(a) 2D sketch strokes. Image from [HGY17] (b) Feature and Silhouette strokes. Based on image from [YYL*17] (c) 3D sketch strokes. Image from [XFZ*18].*

creating a 3D model, these properties play a pivotal role in improving the quality of the output 3D model in various ways ranging from helping in closing the sketch boundaries to choosing the required 3D curves for the 3D models. Since sketches are approximate representations, it may be hard to satisfy all the properties depicted in the sketch exactly in a 3D model, but an optimal subset of these properties also help in making the 3D models more accurate. We try to cover few widely used geometric properties of the strokes in this subsection. Figure 3 depicts the collinearity, parallelism and orthogonality properties.

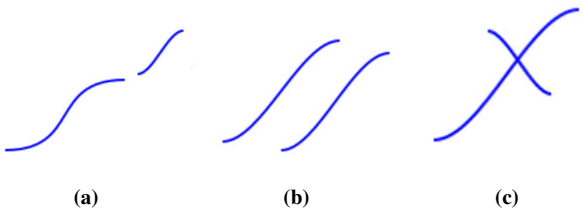


Figure 3: *(a) Collinear strokes. (b) Parallel strokes. (c) Orthogonal strokes.*

Collinearity among two or more strokes implies that the end points of each of these strokes approximately lie on the same straight line. It is useful to understand if the strokes roughly represent straight lines or connected curves.

Parallelism among two or more strokes means that the strokes never intersect each other at any point. A property like this may define two disjoint regions in a 3D model or help infer a surface patch in the model.

Orthogonality among two or more strokes helps to identify the sharp turns in a stroke which translate to the shape features of the 3D model. The strokes may or may not intersect each other.

Coplanarity among two or more strokes means that the strokes lie on the same sketching plane. The strokes satisfying this property help in defining the location of the curves in the 3D model. These strokes are strong indicator of the overall structure of the 3D model. This is one of the most difficult property to capture in case of 3D strokes due to its inherent imprecise nature.

2.4. Curve networks

The curve networks are formed by a set of curves placed together, mostly representing structure of a 3D model (Figure 4a). These curves are formed from the given set of strokes. Depending upon whether the strokes are 2D or 3D, plausible curves are generated from one or more strokes. We describe some major kind of curves that are often encountered in the sketch based 3D models.

Contour curves define the outer boundary of the structure. These curves are generally inferred from the silhouette strokes of the given sketch. The contour curves define the shape of the model being constructed and play a vital role in creation of the appropriate models.

Annotated curves have their semantic meaning associated with them. These may be inferred from the input strokes either by analysing the stroke properties or using the annotations associated with the strokes. In both cases, the annotated curves are useful for defining properties like bump or dent in the structure, which are otherwise difficult to specify.

Surface curves represent the surface of a structure. These are generally generated by sampling the curves on the surfaces obtained from the strokes. These curves can be used to represent the corresponding surfaces visually as well as mathematically.

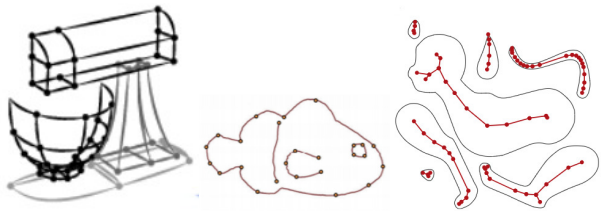


Figure 4: *(a) Curve Network. Image from [XCS*14]. (b) Boundary vertices. Image from [OSJ11]. (c) Axial vertices. Image from [EBC*15].*

2.5. Surface modelling

The sketched strokes can be used to model 3D surfaces, describe their geometry and surface detail like color and texture. Surfaces

can be created by extruding or inflating the strokes, or revolving them about an axis.

Extrusion is the process of translating a stroke perpendicular to its length to create a surface (Figure 5a). Extruding a stroke can be visualized as placing copies of the stroke at uniform intervals such that it covers the required length. This is essentially like sweeping the stroke on two parallel lines defining width of the desired surface. Then a surface is created that is bounded by the original stroke and the last copy of the stroke, ignoring all other copies. This surface can be created using any standard technique of surface creation given its bounding curves.

Inflation is also a similar process like extrusion with a slight difference. Inflating a surface can be seen as dragging the points enclosed by the boundary strokes outwards, i.e., away from the centre of the closed region formed the boundary strokes (Figure 5b). Extrusion or inflation can be used to construct surfaces from the strokes when the underlying assumption is that the stroke characterizes the overall shape of the surface that is created. It does not assume any symmetry in the shape of the surface in general as the stroke that is extruded or inflated can always be non-symmetric in nature. This method cannot produce the desired surfaces in the cases when the bounding and inner portions of the surface vary structurally.

Surface of revolution is created by revolving a stroke around an axis of rotation. Revolving a stroke can be perceived as moving the stroke around an axis in a circular order, which results into a surface (Figure 5c). It is the same as creating a surface of revolution from the curves, with the curves being replaced by the strokes. This technique can generally be used in constructing surfaces that are assumed to have a symmetrical structure around the axis of rotation. This method cannot capture the asymmetry of the surface represented in the stroke. It differs from the technique of extrusion or inflation in principle as the latter is a sweeping based method, which allows us to create surfaces of non-uniform length and width. In contrast, the surface of revolution is unable to construct the surfaces which may have varied structure across the axis of rotation. The extruded surfaces, inflated surfaces and surfaces of revolution are incapable of capturing structural variance within the surface.

2.6. 3D surface meshes

The 3D surface models can be represented as 3D meshes where a mesh is a collection of triangulated vertices appropriately joined together. There are other representations of surfaces as well like implicit surfaces, parametric surfaces and fair surfaces. A brief description for each of these surfaces and the references of techniques which use them can be found in [OSSJ08]. The techniques used for computing the 3D models from the sketches in general intend to construct the meshes eventually as a tessellated mesh-like representation which is useful for real-time rendering. So in this section, we describe various components and properties related to a 3D mesh that are very often encountered in the literature.

Vertices of a mesh are formed by a set of points capturing the structural information of the model. An axis of a mesh is a vertical line that partitions the mesh or parts of the mesh into two equal halves. The vertices lying on this axis or the vertices forming this

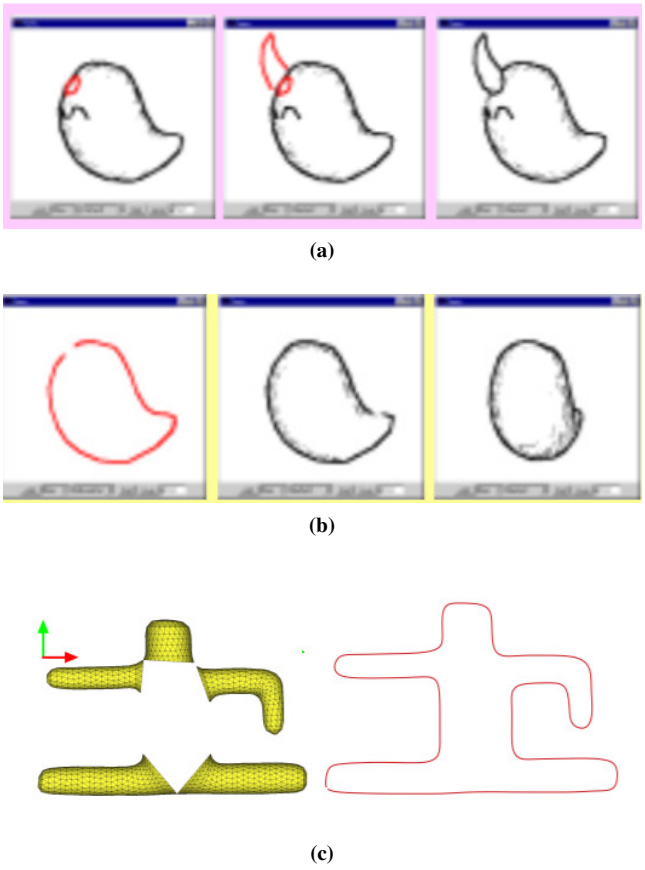


Figure 5: *(a) Extrusion of strokes. Image from [IMT99]. (b) Inflation of strokes. Image from [IMT99]. (c) Surface of revolution from strokes. Image from [BJD*12].*

axis are called axial vertices (Figure 4c). This kind of vertices are important while creating the symmetrical models as well as in rigging a model as the axis and rig of a model are highly correlated. On the other hand, the vertices lying on the bounding curves of a mesh are called boundary vertices (Figure 4b). They define the overall shape of the mesh and are directly related to the sketch strokes given as an input to create the model. These vertices are also helpful in determining the symmetry properties of the mesh as in the symmetrical models these vertices are at approximately same distance from the axial vertices. The axial or boundary vertices can further be classified as terminal and non-terminal vertices. The vertices defining the end points of an axis or boundary are called terminal vertices and the remaining ones are called non-terminal vertices. These terminal vertices are used to join the axes or boundaries of various parts of the mesh together.

Edges of a triangle mesh are created when its vertices are triangulated. They are defined by the sketched strokes provided as the vertices are sampled from the strokes themselves. The edges can be grouped into the ones defining the boundary of the mesh called boundary edges and the remaining ones known as non-boundary edges. Similar to the corresponding vertices, the boundary edges define the shape of the mesh. The non-boundary edges serve as the

connector between various parts of the meshes defined by their vertices.

Connected regions in a mesh are defined by the cluster of vertices that are connected to each other by some sequence of edges. In graph theory terms, if a mesh is considered to be a graph then the connected region is formed by a set of vertices in which there exists a path for every pair of vertices defined by the set of edges between all the vertices of the set. These regions can be directly inferred from the closed and connected regions in the given sketch (Figure 6a). The connected regions can help in defining different parts of a mesh which can be processed separately and then appropriately joined together.

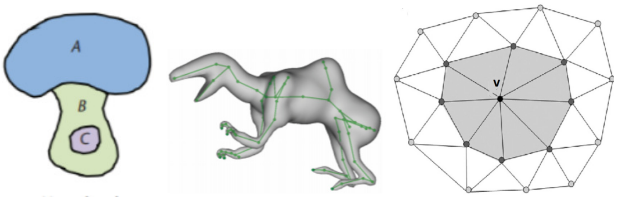


Figure 6: *(a) Connected regions in a sketch. Image from [OSJ11]. (b) Joints in a rig. Image from [BJD*12]. (c) One ring neighbor of a vertex. Image from* `www.researchgate.net`.

Joints are defined by the terminal axial vertices described above. Joints together make up a rig that spans different parts of a mesh (Figure 6b). In order to compute joints from a given sketch, the sharp turns in the strokes defining its various segments can be used. When the parts of the meshes overlap with each other, the joints are defined from the intersection point of their respective axes. In some cases, the axes are computed using the boundary vertices. The joints and rigs are constructed from the computed axes subsequently.

One ring neighbor of a vertex v in a mesh is a set of vertices between a closed edge sequence surrounding it (Figure 6c). These vertices are directly connected to the vertex v by one edge. For many common operations on meshes, this set of vertices are affected by any change made to an attribute of the vertex v, for example to its position or its color. The effect of any change made to v reduces as we go to its farther ring neighbors. Depending on the application and the method used, the number of ring neighbors updated for a vertex is chosen.

Manifoldness A mesh is said to satisfy the property of manifoldness if every edge of the mesh is either a boundary edge otherwise it is common to exactly two triangles of the mesh. This property helps to give the meshes meaningful orientations and is widely preferred for most of the applications.

We conclude the discussion on 3D surface modelling with a brief discussion on various perceptual properties of 3D models which are important irrespective of which representation for the model is chosen.

Conformity is a perceptual property with respect to the 3D models. According to this property, a model in which a set of curves

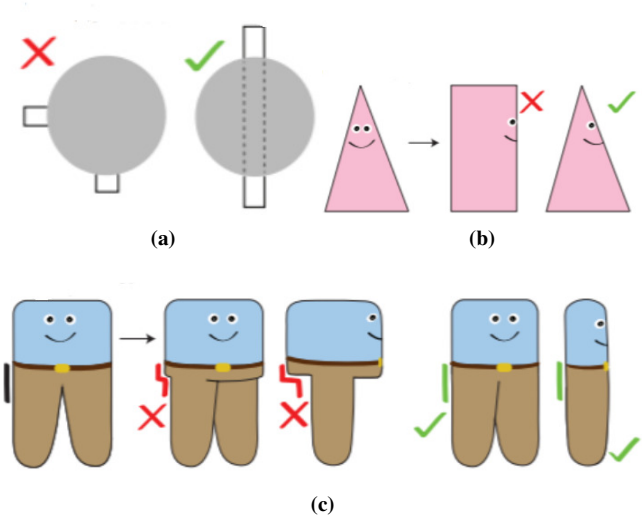


Figure 7: *(a) Conformity. (b) Simplicity. (c) Persistence. Images from [BCV*15].*

are joined smoothly without any sharp transition is perceived to be a better model (Figure 7a). This indicates that smooth transition between different curves of a mesh produces a better model with respect to how a viewer perceives it.

Simplicity is a perceptual property related to the models that prefers a simpler ensemble of parts forming the model. It prefers a model that can be interpreted to have a very similar shape when considered from different sides (Figure 7b). This indicates that a model that has a drastically changing structure when seen from different angles is not perceived to be an appropriate model.

Persistence is another perceptual property of the models that deals with how the shape of a model should look from different viewing angles (Figure 7c). This property forces the model contours to retain their shapes when rotated by small angles. According to persistence property, the models which change their shapes with small rotations are not considered to be appropriate models.

2.7. Footprint and imprint of sketches

With respect to coloring or texturing a 3D model, the footprint and the imprint of the strokes play an important role. A footprint of a set of strokes means the area covered by them on the sketching plane. This actually forms a mask on the plane representing the region that gets affected when these strokes are drawn. On the other hand, an imprint of a set of strokes is the final effect they leave on the sketching plane. This is essentially the final state of the mask formed by the footprint of the strokes after the strokes are drawn. For example, if the strokes represent a brush then its footprint means the area that the strokes would cover on the canvas whenever the brush is used to paint something. In this case, an imprint of the brush is the final color of the painted region after blending the background and brush color appropriately. These concepts are applicable for the methods like [BBS*13] that attempt to preserve the brushing style of the painting like the effect of wetness of the color stroke.

2.8. Keyframing with sketch strokes

The concept of keyframing is quite well known in the context of animation. Using strokes for describing the motion for an animation implies that the strokes carry information about various parameters for animation, for example, the trajectory of motion, duration and shape deformations. Keyframing with respect to the strokes, therefore, implies retrieving this information from the given strokes at the keypoints in the sequence and storing them appropriately. It often requires splitting strokes into some meaningful fragments that can be mapped to the value of some parameter. Interpolation is used to generate the values for generating the information at the intermediate stages between the keypoints.

3. Sketch based content creation

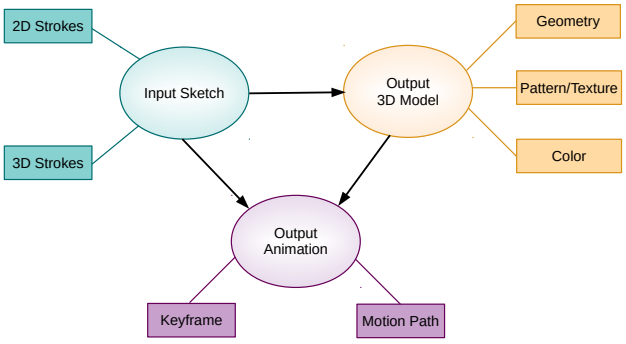


Figure 8: *Schematic diagram for a sketch based content creation framework.*

With the increasing numbers of commercial products providing AR/VR experiences, the need for the assets or content that can be visualized on these platforms is also increasing. But the content creation for this purpose is a non-trivial endeavour. Sketching is one of the natural ways to model on these platforms due to the ease of movement provided by them along with the liberty to inspect the objects by going around them. The content created in the process not only contribute to enrich the AR/VR experience but are also usable for non-AR/VR applications like the movies or computer aided design.

A sketch based content creation framework, in general, takes sketches as input and can produce static 3D models or dynamic animations as output. For creating 3D models we must generate their geometry. Additionally surface color and texture can also be inferred from the sketches. In the case of animated content, the keyframe shapes can be directly generated using the sketches or the generated 3D models. Sketches can also be used to provide motion paths for interpolation during animation. Figure 8 shows a schematic diagram for such a framework and outlines the interrelationships between the various components. We thus consider geometry modelling, color and texture generation and animation to be major aspects of sketch based content creation. We present our survey of sketch based content creation methods aligned to these aspects.

There already exist excellent surveys in literature that provide a thorough and comprehensive discussion on 3D modelling from sketches. They cover various steps involved in 3D modelling from the sketches along with various surface representations used like parametric surfaces, meshes and implicit surfaces. They also discuss the need of having suggestive interfaces for sketch based modelling such that they can provide a prioritized option of actions to be taken upon drawing a stroke while not hindering the creative process of modelling.

In [OSSJ09], which is an extension of [OSSJ08], various input media for acquiring a sketch and the representation of these sketches are also discussed. This is followed by a description of the steps and techniques for pre-processing the input sketch which includes re-sampling, smoothing, fitting and oversketching. An elaborate description on various methods available for 3D modelling from the sketches is given where the methods are categorized based upon whether they construct the 3D model from scratch (evocative and constructive), augment the sketches to the 3D model, or deform the 3D model using the sketches.

Cook et al. [CA09] additionally cover the techniques that infer a volume from the given height fields or shading as well as the sculpting aspect of the modelling task. Moreover, the discussion of various aspects of this task is given with respect to the usability factors involved. Cruz et al. also provide a similar survey for 3D modelling from the sketches in [CV10] but with the emphasis on explaining the overall pipeline of a general sketch based modelling system. The discussion in [DL16] emphasizes data driven approaches that can be used for constructing the 3D models from any input sketch. More recently, Bonnici et al. [BAC*19] reviewed 3D modelling from a sketch based input where they primarily categorized the techniques based upon their usability in an industrial design pipeline.

Even though we cover the topic of 3D modelling from sketches, our work differs from these previous works in two primary aspects. Firstly, we attempt to provide a comprehensive study of a complete sketch based content creation system or framework that includes the aspects of coloring, texturing and animating the 3D models along with modelling their geometry. Secondly, although some of the previous surveys mention the use virtual immersive media and the corresponding input devices in the modelling pipeline but they do not cover in detail the techniques that actually leverage these media for the intended purpose. We cover the relevant techniques that either use VR or AR as the media for sketching or techniques that can potentially be translated to these media.

We shall review the methods and the techniques that are proposed to solve the problems corresponding to each aspect of a sketch based content creation system. Table 1 summarizes various techniques highlighting the aspect of sketch based content creation that they address and the platforms used by them. The table also gives details about which class of technique is used for a particular aspect of the content creation process. Table 2 defines the terminology used in Table 1 for describing various classes of techniques. A brief description of every paper in the Table 1 can be found in this survey in appropriate sections.

We dedicate Section 3.1 to review the methods for generating the geometry and shape of 3D models from the 2D and 3D sketches. Section 3.2 reviews the techniques used to generate the pattern and

color for the surface of a 3D model from the sketched strokes. The methods dealing with the animated content are reviewed in Section 3.3. Additionally, we also present a review of methods that assist the process of sketching in Section 3.4. This kind of assistance is helpful and often crucial in improving the quality of output generated by reducing the inconsistencies in the input sketches.

3.1. Generating 3D models

Generating the geometry or shape of 3D models from given 2D or 3D strokes involves interpreting each of these stroke and building a compatible and feasible feature set that can be used to construct the model. Over the years, several approaches have been designed for constructing the 3D models using inputs varying from the annotated 2D sketched strokes to the freehand 3D sketched strokes.

Deering et al. [Dee95] proposed one of the earliest VR based sketching system called HoloSketch. Users are allowed to create 3D shapes in the mid-air and then edit and animate the models created. The resulting animated 3D model can be saved. The system uses the head-tracked stereo shutter glasses and a desktop for display. Many approaches like [NISA07, SKČ*14, EBC*15] are designed to minimize the user input by modeling from a single view sketch and creating corresponding models with their parts meaningfully positioned at appropriate depths automatically. In contrast to the above, methods like [BBS08, XSS08] require the user to sketch strokes from multiple views and are able to create more complex models. Another category of approaches [AS11] minimizes the required user input to only two strokes but significantly reduces the set of objects that can be modelled from the strokes. Complementary to these methods, some techniques [XCS*14] can create a large variety of models at the cost of the user being forced to sketch a large number of strokes.

The output of various methods also varies from computing the models belonging to a pre-defined set to any model satisfying the feature set created from the input. A major factor in designing an interface for sketch based modelling is to find a sweet-spot between the amount of user input required by the system and the creative liberty the system offers. Various approaches based upon the kind of sketch that are input to the method, categorized broadly by the techniques used for creating the model, are reviewed in the subsequent subsections.

3.1.1. From 2D sketches

Extrusion or Inflation based methods The Teddy system, introduced by [IMT99] was one of the first to construct 3D models from 2D sketches. The method in [IMT99] works on the principle of draw-rotate-draw, where the artist sketches some strokes then rotates the sketch and draws other strokes. The sketch is approximated with a closed planar polygon that is triangulated while removing some unnecessary triangles. An axis joining the mid points of the side of the triangles lying inside the polygons is computed. As the method assumes rounded shape for the generated 3D model, all the axial vertices are elevated, the non-boundary edges are made quarter oval, and appropriately mirrored about the axis. For extruding one surface from the other, the stroke corresponding to the surface to be extruded is swept over the other following the shape of the latter. Operations like removing some portions of the 3D model or, erasing, coloring and smoothing the model by adding some strokes to the sketch are also supported. The method given in [IH03] is focused on creating smooth meshes with even triangulation for the models generated from the 2D sketches like the one described above. A similar strategy to the Teddy system is also employed by Andre et al. [AS11]. In order to generate a 3D model using their method, the artist needs to draw at least two strokes called *construction strokes*, corresponding to the latitudinal and longitudinal surface curves, and an optional stroke called *silhouette stroke* corresponding to the outer shape of the model. The construction strokes are used to create the 3D model, where one of the strokes is swept over the other following the shape of the silhouette stroke. Other operations like over-sketching to modify the output model, and designing a scene with various 3D models at different depths are also supported. Cherlin et al. [CSSJ05] also attempt to model 3D objects from few 2D strokes inspired by the technique of progressively refining a minimalist sketch when using the pen-and-paper medium. Bergig et al. [BHESB09] as well propose a technique to author a scene in AR with the 3D models in the mechanical systems from their sketches or printed images captured live from a video feed. The input sketch is pre-processed to separate its individual objects. The system recognises different annotations that are provided by the user to specify either the geometry of the object or some physics based parameters like friction. Each object is converted to a vector image to determine the location of each of its vertices which is by fitting lines to the strokes of the sketch. The lines from the image corresponding to the occluded and non-occluded parts of the 3D model is determined. The surface inflation technique is applied suitably to compute the model, where the vertex positions for the occluded part of the model is inferred from the most plausible set of values. Overlapping objects in the sketches are composed by culling them appropriately to produce a consistent view.

Symmetrical modelling methods Many approaches are designed by assuming that there exists some symmetry in the complementary sides of the 3D model, for example the front and the back side, or the left and the right side. Olsen et al. [OSJ11] propose a method to model the 3D objects while retaining the natural look of the object as much as possible according to the sketch. The sketched strokes are traced to extract the contour curve for the individual sub-parts of the model while removing the points where more than two strokes meet by extending the shorter curves to meet the farther ones. The connected closed regions in the sketch are used to define a *stroke-region graph* with the regions as nodes and the strokes associated with them as edges. This graph is traversed to segregate various connected closed regions of the sketch that correspond to different parts of the final 3D model. The sketch is approximated by a triangulated polygon and each vertex of this polygon is elevated by a value depending upon its distance from the boundary vertices. The sketching style of the artist is preserved by using parameters like the speed and pressure applied by the user to draw the stroke. An augmented reality (AR) based method using a mobile phone as the interactive device, proposed in [FYX17], uses a similar approach as the above method to create a 3D model. The sketch contour is identified by separating it from the background using the black colored boundary. The closed connected regions formed

Method	Input	Modelling	Color/texture	Animation	Assistance	VR/AR based
Surface Drawing [SPS01]	3D	Inference	Direct	✗	✗	✓
Smooth Meshes [IH03]	2D	Inflation	✗	✗	✗	✗
Articulated figure animation [DAC*03]	2D	✗	✗	Pose estimate	✗	✗
Motion doodle [TBvdP04]	2D,3D	✗	✗	Identify strokes	✗	✗
View-Dependent Animation [CKB04]	2D	✗	✗	Pose estimate	✗	✗
Parameterized Object model [YSvdP05]	2D	Retrieval	Fixed	✗	✗	✗
Few strokes model [CSSJ05]	2D	Extrusion	✗	✗	✗	✗
Beautification techniques [WSP05]	2D	✗	✗	✗	Per subpart	✗
Dynamic 2D Patterns [BSM*07]	2D	✗	Transform	✗	✗	✗
Magic Canvas [SI07]	2D	Retrieval	Fixed	✗	✗	✗
Multi-view sketching [LGS07]	2D	Symmetric	✗	✗	✗	✗
FiberMesh [NISA07]	2D	Inference	✗	✗	✗	✗
Napkin Sketch [XSS08]	3D	On grid	Direct	✗	✗	✓
ILoveSketch [BBS08]	2D,3D	On plane	✗	✗	✗	✗
Thor [ALS08]	2D	Symmetric	✗	✗	✗	✗
Beautify sketch with suggestion [MSR09]	2D	✗	✗	✗	Full sketch	✗
Hand Animators to 3D animation [JSH09]	2D	✗	✗	Annotate	✗	✗
Author mechanical system [BHESB09]	2D	Inflation	✗	Annotate	Full sketch	✓
Content creation in AR [HGB*10]	2D	Annotate	Fixed	✗	✗	✓
Detailed Paint [CBWG10]	2D	✗	Blend	✗	✗	✗
LifeSketch [YW10]	2D	Inflation	✗	Pose estimate	✗	✗
NaturaSketch [OSJ11]	2D	Symmetric	Symmetric	✗	✗	✗
Singleview modeling [AS11]	2D	Extrusion	✗	✗	✗	✗
ShadowDraw [LZC11]	2D	✗	✗	✗	Per stroke	✗
RigMesh [BJD*12]	2D	Symmetric	✗	✗	✗	✗
3D proxies for characters [JSMH12]	2D	✗	✗	Annotate	✗	✗
Painterly characters [BBS*13]	2D	✗	✗	Pose estimate	✗	✗
Geosemantic Snapping [SAG*13]	2D	Annotate	✗	✗	✗	✗
Mosaic [AGYS14]	2D	✗	Grow	✗	✗	✗
True2Form [XCS*14]	2D	Inference	✗	✗	✗	✗
Ink-and-Ray [SKČ*14]	2D	Inference	Symmetric	✗	✗	✗
Cartoon Drawings model [BCV*15]	2D	Symmetric	Symmetric	✗	✗	✗
SecondSkin [DPS15]	2D	Inference	Symmetric	✗	✗	✗
Live Texturing [MNZ*15]	2D	✗	Symmetric	✗	✗	✓
Sideview animal model [EBC*15]	2D	Symmetric	✗	✗	✗	✗
Space-time animation [GRGC15]	2D	✗	✗	Identify strokes	✗	✗
3D sketch based retrieval [LLG*15]	3D	Retrieval	Fixed	✗	✗	✗
ShipShape [FAS15]	2D	✗	✗	✗	Per curve	✗
TraceMove [PGC16]	2D	Inference	✗	Pose estimate	✗	✗
Learning to simplify [SSISI16]	2D	✗	✗	✗	Full sketch	✗
Drawing Texture [ZHY17]	2D	✗	Symmetric	✗	✗	✗
Magic Toon [FYX17]	2D	Inflation	Symmetric	✗	✗	✓
Sketch-a-Net [YYL*17]	2D	✗	✗	✗	Per stroke	✗
DeepSketch2Face [HGY17]	2D	Inference	✗	✗	✗	✗
Window-shaping [HR17]	3D	Inflation	Symmetric	✗	✗	✓
SheetAnim [GC18]	2D	Inference	Symmetric	Pose estimate	✗	✗
SymbiosisSketch [AHKG*18]	2D,3D	On canvas	Direct	✗	✗	✓
Model-Guided sketch [XFZ*18]	3D	On plane	✗	✗	✗	✗
Learning to Sketch [SPS*18]	NA	✗	✗	✗	Per stroke	✗
3D sketch based retrieval in VR [GJS18]	3D	Retrieval	Fixed	✗	✗	✓
Sketch based man-made shapes [SBS19]	2D	Inference	✗	✗	✗	✗
SurfaceBrush [RRS19]	3D	Inference	✗	✗	✗	✓
Magical Hand [AKK*19]	3D	✗	✗	Identify strokes	✗	✓

Table 1: *Comparison of various methods (post 2000) for sketch based content creation according to the kind of their input, the aspects of content creation they address and the method used for it, and platform on which they are implemented.*

Method	Description
Inference	Solving the constraints defined by the sketched strokes for determining various model properties.
Inflation	Refer Section 2.5 for definition.
Retrieval	Finding the most similar model to the sketch from the database.
Extrusion	Refer Section 2.5 for definition.
Symmetric (modelling)	Assuming symmetry about an axis to create surfaces of revolution for modelling.
On grid	Sketching enabled by 3D grids, defined by the user using the sketched strokes.
On plane	Sketching enabled by 3D planes, inferred from the user input or reference models.
Annotate (modelling)	Interpreting sketch strokes as symbols with special meanings to describe various properties of the model (Refer Section 2.5 for more details).
On canvas	Sketching enabled by user defined 3D non-planar grids or by their orthographic view in 2D.
Direct	Assigning the same color from the 3D sketch to corresponding parts in a 3D model.
Fixed	Constructing models with pre-defined colors and textures irrespective of the sketched strokes.
Transform	Computing the geometric transform between two sketches to understand temporal changes in the sketches.
Blend	Compositing the temporal changes in the colors of the sketched strokes.
Symmetric (coloring)	Assigning same color to the corresponding occluded and non-occluded parts of the model.
Grow	Expanding the sketched strokes following different constraints to get the desired texture effects.
Pose estimate	Computing the pose of the model from sketch(es), to animate it.
Identify strokes	Recognizing the sketched strokes to retrieve the animation parameters represented by them.
Annotate (animation)	Using symbols with special meanings to describe various parameters for animating a model.
Per subpart	Modifying the individual parts of a sketch to rectify it.
Full sketch	Modifying the entire sketch simultaneously to rectify it.
Per stroke	Modifying the individual strokes of a sketch to rectify it.

Table 2: *Description of various terms used in Table 1.*

by the strokes are used to segment the sketch. A method similar to the one given in [OSJ11] is used to generate a 3D model from the extracted contour curves and the connected regions. A circular mapping is used to determine the elevation value for each vertex x and is given by the following equation:

$$H(x) = \sqrt{D_{max} \cdot D_{max} - (D_{max} - D(x))} \quad (1)$$

Here, $D(x)$ is the distance of the current vertex from the nearest boundary vertex and D_{max} is the maximum of all such distances in the region to which x belongs. It is possible to have multiple copies of the 3D models generated by this method at different locations to design a scene. This method also allows the artist to rig the model by specifying the end points of some predefined nine joints of the skeleton explicitly and the others being inferred by the system. A joint position $\vec{J}$ is computed using the equation

$$\vec{J} = \vec{J}_{start} + r_1 \cdot (\vec{J}_{end} - \vec{J}_{start}) + r_2 \cdot \|\vec{J}_{end} - \vec{J}_{start}\| \cdot \vec{D} \quad (2)$$

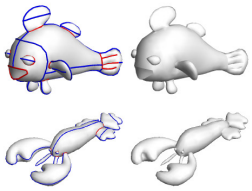
Here, $\vec{J}_{start}$ and $\vec{J}_{end}$ are two dependent joints, $\vec{D}$ is the direction of the offset, r_1 and r_2 are the ratios of their lengths. Borosan et al. [BJD*12] also construct a rigged 3D model from the given sketch. The sketch is approximated by a triangulated polygon and an axis is formed for it using a method similar to the one used for the same purpose in [IMT99]. The 3D models for the parts, excluding the area bounded by the edges introduced due to triangulation, is computed using generalized cylinders about the corresponding axis with circular cross sectional areas of the cylinder being perpendicular to the axis. The 3D model is completed by triangulating a set of points sampled from each line segment joining centre of the neighbouring circular cross sections. A simplified axis is used as the skeleton with the joints being placed at centre of each triangle formed by a set of three edges introduced due to the triangulation. An appropriate skinning algorithm is applied to complete the rigging process. The removal and merge of the parts of the 3D model is also supported.

In a complementary method Bessmeltev et al. [BCV*15] propose a method that requires the user to provide a rig with the sketch to create the 3D model. Each bone of the rig is iteratively associated with some part of the sketch while traversing the bones in the depth order from the viewer's point of view. The assignment is discarded if any mapping violates the conditions of Gesalt continuity [Kof13] with respect to how a bone-mesh association is perceived by a viewer and the required reassignment is done. The 3D model for the part of the sketch which is associated to a bone is generated as a surface of revolution using that bone as the axis and is appropriately joined with the other parts. The 3D models formed using this method are intended to satisfy the conformity, simplicity and persistence principles (see Section 2.6).

The technique given in [EBC*15] is specifically targeted for computation of the 3D models of animals from their side-view sketches. Each stroke is approximated with the half edges, which are traversed in different orders to identify various regions of the sketch discarding the occluded ones. To close the contours of each region, the potential points that need to be connected are identified using the edges corresponding to these regions. For connecting these points, the curve between these points which minimizes the following energy equation corresponding to the two possibilities of the curve $(i = 0, 1)$ is selected.

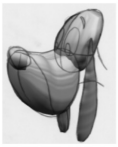
$$E_i = ({p^i}_0 \cdot {p^i}_1)(E_o + {E_c}^i) \quad (3)$$

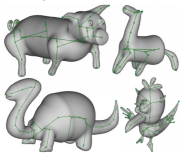

(a) Teddy [IMT99]


(b) FiberMesh [NISA07]


(c) Magic Canvas [SI07]

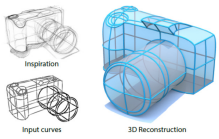

(d) NaturaSketch [OSJ11]


(e) Singleview modelling [AS11]

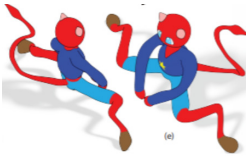

(f) RigMesh [BJD*12]


(g) Geosemantic snapping [SAG*13]


(h) Ink-and-Ray [SKČ*14]

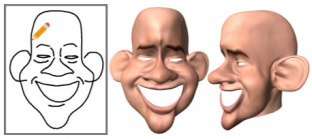


(i) True2Form [XCS*14]


(j) Cartoon drawing model [BCV*15]


(k) SecondSkin [DPS15]


(l) Sideview animal model [EBC*15]


(m) DeepSketch2Face [HGY17]


(n) MagicToon [FYX17]


(o) Sketch based man-made shape [SBS19]

Figure 9: *Models generated from various methods discussed in Section 3.1.1. The caption contains reference of the method that was used to generate the corresponding result.*

Here, ${p_j}^i$ denotes the probability of the points being connected correctly and is given by $\min(\frac{{\alpha_j}^i}{\beta_j}, 1)$, where ${\alpha_j}^i$ and β_j denote the angles the curve makes with the region boundaries. E_o and E_c are the energies of the curves connecting pairs of the terminal points and non-terminal points, respectively, and are given by $E = \frac{1}{l}\sum_i |k_{i+1} - k_i|\delta l$, where l is the curve length and $k_i = \frac{\|\dot{c}_i \times \ddot{c}_i\|}{\|\dot{c}_i{}^3\|}$ is the curvature at a sample point i. The axis with the minimal number of vertices and branches for each closed region is computed. A surface around the axis of each part is generated to construct the individual parts of the model. A *depth order graph* is computed with the root node as the main body of the animal and the other nodes corresponding to its other body parts placed according to their relative order with respect to the main body. It is used for placing every part of the model in its appropriate position. The method proposed in [ALS08] falls into this category of modelling as well. It creates the models using the skeleton of the object sketched by the user as the axis to create a surface of revolution around different parts of skeletons.

Levet et al. use multiple sketches of an object from different views to create the 3D models in [LGS07]. The sketch in each view represents different information like one view for sketching profile and the other for skeleton, and these are combined together to form a 3D model. Levi and Gotsman [LG13] present ArtiSketch as a system that can recover the 3D model from sketches showing a character from different views. The users also have to input a skeleton sketched on the character in each view. A skeleton consistent across all views is recovered first. Thereafter, a triangulated mesh is recovered such that the silhouette of the rigged and posed mesh matches the sketch in the said views.

Annotation based methods Some of the methods also need the annotations from the artist about the strokes or the objects being sketched to infer the 3D model. Method in [OSJ11] needs the artist to specify various kinds of geometric constructs like bumps and holes, using some special symbols. In order to create the 3D structures using the method given in [SAG*13], the artist needs to specify the kind of primitive like sphere, cone, cylinder, or cube they are drawing along with the sketch. The strokes of the sketch are matched with the curves of the specified primitive using the bipartite matching and the matching is fine-tuned using some primitive specific heuristics based upon the distinctive characteristics of each primitive. The system solves an optimization problem spanned in two phases - the first phase best fits the specified primitive curves to the matching sketched strokes and the second phase correctly places the different parts of the model together. Geometric relations between the strokes and the primitive curves like collinearity and orthogonality (see Section 2.3.1) are used in the second phase of optimization. Hagbi et al. [HGB*10] devised a method to use sketch based annotations to design a game in AR. Various annotations ranging from the dots to specify a starting point to a blue colored scribble to describe the water bodies are used to create a virtual scene. The 3D objects, called *assets* in context of game designing, are registered with the real world. The assets are augmented on any planar physical entity like a napkin or a whiteboard and the entire scene can be seen in AR using an HMD or a mobile phone.

Retrieval based methods Few methods use the sketch based retrieval techniques to generate a 3D model from the sketch. One such method is proposed in [SI07], which considers the most similar 3D model from the database corresponding to the sketch as output. Sixteen different views for each model in the database is considered for computing the retrieval score between the sketch and the particular model. For the input sketch and each view of the model, the angular distance between the centre of the sketch or the model and the sample points on their respective contours is converted to the frequency domain. The retrieval score in frequency domain is computed as

$$FV = \sum_{i=1}^{n} |Freq_{sketch,i} - Freq_{model,i}| \tag{4}$$

Here, $Freq_{sketch,k}$ denotes the value of the k^{th} sample point of the sketch in the frequency domain and $Freq_{model,k}$ denotes the value of the k^{th} sample point of the corresponding view of the model in the frequency domain.

The low frequency term is converted back to a smooth contour and the retrieval score is computed as

$$SV = \sum_{i=1}^{n} |Rad_{sketch,i} - Rad_{model,i}| \tag{5}$$

Here, $Rad_{sketch,k}$ denotes the value of the k^{th} sample point from the smooth contour of the sketch and $Rad_{model,k}$ denotes the value of the k^{th} sample point from the smooth contour of the corresponding view of the model.

The final retrieval score for the corresponding view is $\alpha(1 - normFV) + \beta(1 - normSV)$, where normFV and normSV are the respective normalized scores, and α and β are pre-defined constants. The average of the top two scores among the sixteen scores corresponding to a model is used as its retrieval score. To orient the retrieved model according to the sketch, the re-projection error between the image formed by projecting the retrieved model on the image plane and the sketch is minimized.

Recent deep learning based retrieval techniques like the one given in [YYL*17] can replace the retrieval algorithm in this kind of approaches. Yu et al. use an ensemble of Convolutional Neural Networks (CNNs) [KSH14] in [YYL*17] to identify the object being sketched. This ensemble is trained with the sketches at various abstract levels and having different deformations. This may be combined with the 3D model generation method proposed by Shin et al. as described above to produce better results.

Inference based methods Few methods use some form of inference to construct the 3D model without making any assumptions about its structure. Methods presented in [MSK00, YSvdP05, NISA07, XCS*14, DPS15] belong to this category of approaches for 3D modelling where they infer a feature set for constructing the 3D model. In [NISA07], the authors present a sketching interface inspired from [IMT99], in which a 3D model can be created and edited using the *control curves* defining the surface geometry. Differential equation based optimizations are used to appropriately deform the control curves (computing position and rotation of its vertices) according to the changes made by the user to them. A similar optimization technique with appropriate changes is used for computing the surfaces from these control curves and the corresponding mesh is constructed. Re-meshing of the model is done only when the changes in geometry of the surface is above some pre-defined threshold.

Xu et.al construct a 3D wireframe model (3D curve network) from the given 2D sketch in [XCS*14]. Each 2D stroke and the corresponding output 3D curve are represented as Bézier splines with one-to-one correspondence between their control points, $\left\{b_j{}^i\right\}$ and $\left\{B_j{}^i\right\}$, respectively. An energy function representing the *sketch fidelity*, which is a combination of the projection accuracy, affine variation and minimal foreshortening, is minimized in an iterative manner with respect to the shape regularities, C_k, applied selectively. Each shape regularity captures one of the geometric relations like orthogonality, coplanarity (see Section 2.3.1) between the 2D strokes that has to be maintained between their corresponding 3D curves as well. A likelihood score $L(\alpha_i)$ for each of shape regularity C_k represents its applicability to the final model. This is computed in each iteration. A score of 1 makes the corresponding regularity a hard constraint for the minimization problem from the next iteration.

Sykora et al. [SKČ*14] proposed a method to construct *bas-relief meshes* [BKY99] that retain the properties from the input sketches. The given sketch is segmented into some logical regions. Unfinished contours, either drawn by the user or resulting from articulation, are completed. A depth ordering for each of these regions is computed using diffusion curve based techniques [OBW*08] and it is refined by the user inputs. Each of these regions is inflated using a Poisson equation representing the amount of buckle for the inflated surface. The boundary constraints for the surface can be changed to get the desired amount of roundness. These inflated parts are stitched together with the constraints de-

rived from occlusion of their corresponding regions in the input sketch. The texture from the input sketch is applied to the resultant mesh assuming a symmetry among the complementary parts in mesh. This mesh can either be rendered using full global illumination or using a cartoon shader, with an orthographic camera.

A 3D model guided sketching method to build a layered model on top of a base model is presented in [DPS15]. Each of the sketched stroke is classified as one of the four curve types - shell contour, shell projection, normal plane, tangent curve. Depending upon the curve type of the stroke, various strategies are used to project the sample points of the 2D stroke to either the reference layer of sketching or a minimum skew plane. A *minimum skew plane* is a plane defined by the end points of the curve and the normals at those points that minimizes the deformation caused due to constructing a 3D curve from a 2D curve. The generated curve primitives are represented using a graph where loops indicate a surface. The surfaces are modelled as a three or four sided Coons patch with supported operations like mirroring the surfaces from front to the back or constructing an envelope like surface. Volume generation is also possible by detecting a closed loop in a single view or extruding one surface from the other.

More recently, Han et al. designed a method in [HGY17] specifically for 3D face construction from the sketch using an AlexNet [KSH14] based framework. The set of vertices V of a 3D face is computed by the system by multiplying a tensor C, representing all the faces in the dataset, with appropriate bases u and v corresponding to the identity and expression of the face.

$$V = C \times_2 u^T \times_3 v^T \tag{6}$$

Here, the symbol $\times_n$ denotes the operation of multiplying a vector with the n^{th} dimension of a tensor, where $n = 2, 3, \ldots$ The network predicts values of the coefficients of these bases. To ensure that the system has the global shape information as well, the *silhouette strokes* and *feature strokes* of the sketch are also expressed in terms of two bases similar to the above ones. These two bases corresponding to both these strokes are also fed to the network. The loss function minimizes the difference between the ground truth value g_i and the computed value V of each vertex. A gesture based editing of the sketch is also possible, where the gesture is recognized using a small convolutional neural network (CNN) [KSH14] and the required action is applied.

Another recently proposed method by Smirnov et al. [SBS19] constructs the 3D models for four categories of man made objects, namely airplane, knife, bathtub and guitar. The proposed method uses a convolutional neural network (CNN) [KSH14] for inferring a series of parametric surfaces to fit into a template model for the given sketch. Smirnov et al. create a synthetic dataset for training the model to avoid inaccuracy of the resultant models due to lack of data. A vectorized sketch of any object from the above four categories is augmented with its contour strokes. Newer sketches are created by rasterizing the above augmented sketch infused with different stroke widths, sketching styles, and realistic textures. Each patch of the mesh is represented by a Coons patch. A template for each category of objects is created by a collection of Coons patches joined together appropriately. The network is trained with a loss function which biases the output model to be non-self intersecting while optimally fitting the inferred patches to the template corresponding to the sketch. Figure 9 shows example outputs of various methods discussed in this section.

3.1.2. From 3D sketches

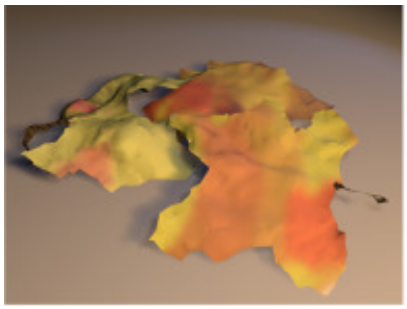
In [SPS01], a semi-immersive platform based system for creating 3D models in proposed that allows the user to create the model with *repeated markings*. Various interactive tools like tongs and magnets are provided to manipulate the models easily along with the standard tools of erasing and marking terminal points of strokes. The system uses the semi immersive environment provided by the Responsive Workbench, as described in [KF94].

Xin et al. devised an AR based method in [XSS08] which uses a mobile phone to enable 3D sketching. Initially, a sketching grid is displayed on detecting a visual marker and then the camera position and the hands of the artist are tracked. Before drawing the strokes the artist specifies the actual sketching grid on which she intends to draw. For this the artist sketches a line perpendicular to the current sketching grid which lies on the intended sketching grid. A sketching grid perpendicular to the current one is generated, which can be rotated to obtain a grid of the required orientation. The colored strokes are drawn on these grid planes and it is possible to edit the strokes in previous grid planes as well.

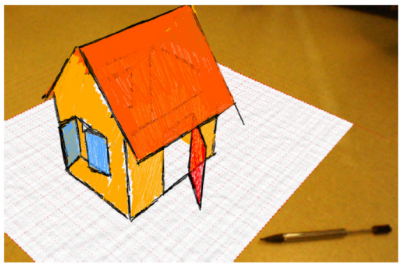
Li et al. [LLG*15] present a method to retrieve the 3D models from the 3D sketches using a Microsoft Kinnect [Azu19] for tracking the movement of the user. Kalman filtering [Kal60] is used to remove the noise from the input sketch. The 3D outlines for the 3D models stored in the dataset are computed by first normalizing them using Principle Component Analysis (PCA) [Hot33] and applying appropriate translation and scaling, and then integrating the contour points from their six principle views and uniformly resampling the 3D points. A 3D shape histogram [AKKS99] is computed for the input sketch and the 3D outlines for each of the 3D models in the dataset. The top 10 3D models with the least Euclidean distance between their respective histogram and the histogram of the input sketch are displayed.

Huo et al. [HR17] present a tangible AR based interaction technique that allows direct 3D modelling around physical objects using sketched strokes. The user sketches the silhouette strokes for the intended shape from which a mesh is created and the mesh is inflated using the function chosen by the user. The physical object is used to determine the 3D location of the shape being created.

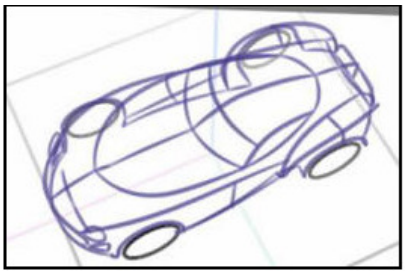
Xu et al. devise a different approach for 3D sketching in [XFZ*18] that is guided by a reference 3D model. The reference model is used to determine the sketching plane for the 3D strokes. From the polygonal approximation of the outer contour of the reference model, distinctive edges and planes are extracted. Every sketched stroke is also approximated by some polylines. A set of geometric relations like parallelism and collinearity (see Section 2.3.1) between each stroke and the edge or plane of the reference model as well as attachment of the stroke to these edges or planes are determined. The geometric relations are marked as valid if the information given by them is sufficient to infer the sketching plane for the corresponding 3D stroke. If no set of valid geometric relations is found, then the sketching plane is determined by



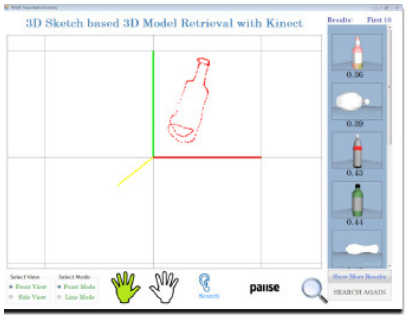
(a) SurfaceDraw [SPS01]



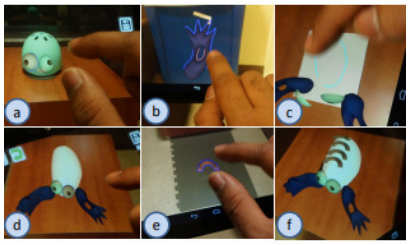
(b) NapkinSketch [XSS08]



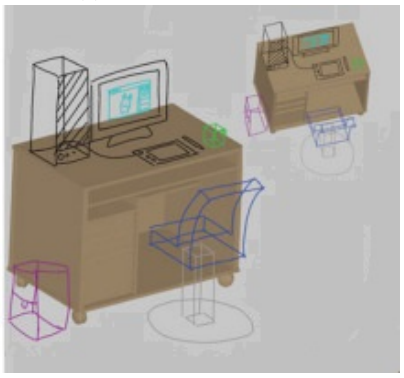
(c) ILoveSketch [BBS08]



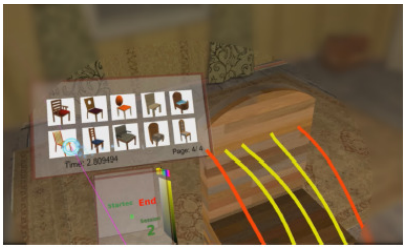
(d) 3D model from 3D sketch [LLG*15]



(e) Window-Shaping [HR17]



(f) Model-guided sketch [XFZ*18]



(g) Retrieval from 3D sketch in VR [GJS18]

(h) SymbiosisSketch [AHKG*18]

(i) SurfaceBrush [RRS19]

Figure 10: *Models generated from various methods discussed in Sections 3.1.2 and 3.1.3. The caption contains reference of the method which generated the corresponding image.*

the major planes of the model and the previously inferred planes. Among all the possibles planes, the final sketching plane is the one for which maximum number of geometric relations that were determined in 2D are preserved in 3D.

In [GJS18], a CNN [KSH14] based framework is proposed to retrieve the required 3D model of a chair from a 3D sketch made on top of an initial chair model. The models and the sketched strokes are displayed in a virtual environment and a dataset of the 3D models of chairs is used for training. Twelve different views of each model in the dataset is fed to a VGG-M network [CSVZ14] aggregated using max pooling to compute the initial feature descriptor for the model, which is passed to another fully connected network to get its final feature descriptor. The same feature descriptor is computed for the sketch augmented on the base model and the top 40 3D models with minimum Euclidean distance between their feature descriptors and that of the input sketch is displayed. The features like structural similarity, curve pattern matching and color similarity are considered. An iterative refinement of the sketch and changing the base model according to the retrieval results is necessary to get required results.

Another recently proposed method by Rosales et al. [RRS19] computes manifold surfaces from the sketches drawn in a VR platform. The work analyses characteristic nature of the 3D strokes drawn in a VR environment which is significantly different from

the ones drawn in a traditional desktop environment. A major characteristic of the strokes defining their approach is that the adjacent strokes have similar tangential direction along the polylines approximating the strokes if they are intended to be on the same surface. The method works on the idea of clustering the strokes to form the mesh strips and finally clustering these strips to form the required surface. The strokes are clustered by matching their vertices pairwise where the matching is categorized by the proximity, tangential direction and normal orientation of the strokes while avoiding certain outliers. The mesh strips are created from the matched strokes obtained above. The strokes which might not be adjacent but are intended to be part of the same surface are accounted by matching the boundary vertices of the respective mesh strips they belong to. The manifoldness property is satisfied by restricting the candidate set of the matching strokes. Finally the constructed mesh strips are joined together using the same matching algorithm as above while consistently orienting their normals. The surfaces created by this method can be passed to any available modelling tool or technique to get a 3D model for it. Figure 10 shows example of outputs generated by the methods described in this section.

3.1.3. 2D-3D sketch hybrid methods

Bae et al. propose a method in [BBS08], which tries to mimic the pen-and-paper sketching style both in 2D and 3D. Multi-stroke sketching with ink drying metaphor resembles the natural way in which the artists generally sketch. A weighted average of Non-Uniform Rational B-Splines (NURBS) [Far92] is fitted to the stroke to form the final curve. The sharp features of the strokes are retained. Flipping and zooming of the sketching plane is enabled in a particular mode with the gestures similar to the ones used while doing the same on a paper. The 3D sketching is provided by drawing two strokes on a single plane or on the two epipolar [HZ03] planes, where one of the planes is specified explicitly. The sketching planes are specified using a 3D curve and a plane passing through it or the extrusion axis. The gesture sets corresponding to the 2D and 3D modes are provided for the operations like rotating, deleting, extruding surface from the curves. The system orients the sketching plane to a view which it considers to be suitable for sketching, depending upon the time taken by the artists to draw subsequent strokes.

The technique given in [AHKG*18] also provides a sketching interface both in 2D and 3D, using the AR/VR platform for 3D sketching. A sketching canvas is specified by drawing the curves representing the required canvas. Each of these curves is sampled with a fixed number of points from which a best fit plane is computed and then these points are fitted with the height field of the computed plane. The 3D sketching in mid air is enabled by an AR/VR platform and the users sketch directly on these canvases. The 2D sketching uses a tablet and the user sketches on the orthographic projections of the canvases. The models are scaled to a canonical volume representing a workspace to facilitate sketching at any desired zoom level. Sketching on top of the planar physical objects by bringing them into the workspace is also possible which enables the user to create any augmented reality scene as well. Figure 10 shows example of outputs generated by the methods discussed above.

3.2. Coloring and Texturing 3D models

Coloring or Texturing a 3D model requires a mapping between the model and the sketch that assigns a color to the parts of a model from some part of the sketch. Some methods like [IMT99, XSS08, OSJ11, SKČ*14, BCV*15, DPS15, FYX17] which are mainly focused on generating the 3D models from the sketches also address the coloring task for a model. The method given in [IMT99] finds the projection of the polyline approximation of the sketched stroke on the 3D model surface. Splicing together all these projections forms the 3D colored stroke for the input sketch stroke. On the other hand, as the techniques in [XSS08, AHKG*18] provide 3D sketching, so a direct mapping of color between the corresponding 3D stroke and the model is used. The methods proposed in [OSJ11, SKČ*14, BCV*15, DPS15, FYX17, HR17] accomplish the coloring task by assuming symmetry in appearance between front and back of the model and assign the same color to the front and back portions. The method given in [GC18] is mainly focused on creating an animation sequence for the sketched models but it also provides a technique to preserve the stroke style as well as a technique to color the models. It preserves the stroke style by updating the contour curves of the model corresponding to a given sketch stroke such that the former matches the latter. It also allows coloring by assuming symmetry and assigning the same color to the front and back portions of the model.

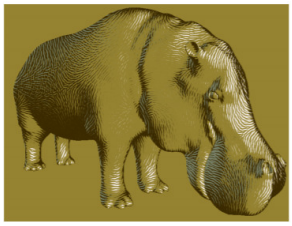
In [HGB*10], the annotations or symbols are used to specify an object and the corresponding 3D model of the object is placed in the scene. In this kind of a method, the sketched strokes serve only to represent an object. The color and texture assignment is fixed, independent of the strokes.

Another aspect with respect to coloring or texturing of a model revolves around creating patterns or texture on surface on the models. Unlike the assignment of color, this aspect is generally not taken care of when designing methods for generating the 3D models. In this section, we shall review the techniques which are mainly focused on these two aspects of the coloring and texturing of 3D model as discussed above. The output generated by various techniques reviewed in the following section is given in Figure 11.

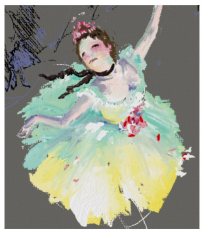
Symmetrical coloring methods Magnenat et al. [MNZ*15] propose an AR based method for producing a colored 3D model from the live coloring of one of the 2D sketches from the coloring book. A fixed 3D model is pre-generated for every sketch outline present in the book. This forms a template, i.e., a 3D mesh corresponding to the 2D sketch. At runtime, the 3D model corresponding to the current sketch is identified by a voting mechanism where all feature points of the colored sketch from the video feed vote for a model, irrespective of the shape of the page. A linear system of equations given below is solved to identify the 3D model from the sketching book page. This equation minimizes the re-projection error between the 3D model chosen and the 2D sketch (encoded by M) while keeping the 3D mesh deformation plausible (encoded by A).

$$\min_{c} \|MPc\|^2 + {w_r}^2 \|APc\|^2, s.t. C(Pc) \leq 0 \tag{7}$$

Here, P is a constant parameterization matrix, C is a constant and c is the set of control vertices such that all the other vertices of the mesh are the linear combination of these vertices. In order to color

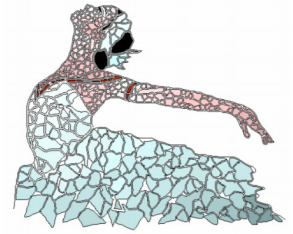


(a) Dyn. 2D patterns [BSM*07]



(b) Detailed Paint [CBWG10]

(c) Sketchaser [HGB*10]



(d) Mosaic [AGYS14]

(e) Live Texturing [MNZ*15]

(f) Drawing Texture [ZHY17]

Figure 11: *Coloring and texturing of models generated from various methods discussed in Section 3.2. The caption contains reference of the method which generated the corresponding image.*

and texture the 3D model, a UV mapped mesh, a color segmented map of the model, and a mapping between the sketch and the UV mapped mesh is created. An energy function corresponding to a spring system is minimized to color the corresponding patches at front and back side of the 3D model. This function considers the neighboring points along the seam as being connected by a spring. This reduces irregularities in the texture of the 3D model. The final 3D model is augmented on top of the sketching book page and is visualized using the AR framework.

Method given in [ZHY17] also uses an AR based approach, which is very similar to the above method, to color a 3D model generated from a 2D colored sketch. The 3D model is represented as a triangular mesh with the mapping between the vertices of the sketch and that of the 3D model being stored. The color for visible side of the model is assigned by looking up the UV map of the 2D sketch. The color applied to a patch at the occluded side of the model is same as the one applied to its corresponding patch at the visible side.

Pattern designing methods Some methods are designed for preserving the textures corresponding to some particular design pattern and style, for example the fading brush strokes. One such method is proposed by Abdrashitov et al. [AGYS14] to create the mosaic styled 2D models. A tile of any shape and color is sketched and copied to form the desired mosaics. The copying and coloring is done by sketching a stroke, which specifies the copying direction and the color shading pattern, respectively. For creating the mosaic, the tiles are placed such that each tile is at a uniform distance from all of its neighboring tiles. To solve this problem, a tile growing algorithm is devised where the tile contour is uniformly sampled and all the sample points are shrunk to the tile centroid. Then all the sample points are moved outward towards the tile boundary such that the uniform distance constraint is maintained by updating every point p_i with the following value:

$$p_i \longleftarrow p_i + G_i + R_i + S_i \tag{8}$$

All points grow outward according to G, repel other points using R and constraint the shape by S. Here,

$$G_i = g \cdot (o_i - c)$$

where, $c = \frac{\sum_i (o_i)}{N}$ is the centroid, o_i is the origin of the tile, g is frictional constant.

$$R_i = \sum_j vdw(\|q_j - p_i\|) \cdot (q_j - p_i)$$

where, vdw is the van der Waals force between p_i and other tile boundaries q_j.

$$S_i = p_i + rigid \cdot ((Av_i + T) - p_i)$$

where, $v_i = c + k \cdot g(o_i - c)$, $rigid$ is rigidity parameter and $T = \frac{\sum_i p_i}{N} - c$ is the translation.

Another method aimed at retaining the oil pastel and brush effects in a painting is given in [CBWG10]. The goal of this method is to avoid the loss of color details due to the repeated re-sampling of the color values whenever a color stroke is drawn. To simulate the canvas to brush color transfer and vice-versa, 2D maps corresponding to them are maintained. The key idea is that the maps are of the same resolution as the canvas, so as to avoid the re-sampling error. The footprint (see Section 2.7) of the brush acts as a mask for the *canvas to brush color map*, which is rigidly transformed to match the brush positions. Before each imprint (see Section 2.7), the *brush to canvas color map* is updated except the region covered by the canvas to brush color map. The strokes are rendered as blend of the source color and the corresponding color in the brush to canvas map depending upon the height field of the brush footprint. Wet and dry blending and color streaks can be created by this method. The above two methods are designed for 2D models but can potentially be extended for 3D models as well.

Breslav et al. provides a way to appropriately deform the pattern strokes on a 3D model when the model transforms in [BSM*07]. The points from the current and previous frames are densely sampled, and a 2D similarity transform is computed, which best maps the points from the previous frame to the ones in the current frame in a least square sense. It is formulated by taking the points as complex numbers since complex number multiplication represents rotation and uniform scaling. The equation for computing the above 2D similarity transform is

$$E = \sum_i w_i |z p_i - c_i|^2 \tag{9}$$

Here, $c_i = x_i - \bar{c}$ and $p_i = x_i - \bar{p}$, where x_i is the sample point in corresponding frames (current or previous), $\bar{c}$ and $\bar{p}$ is the centroid

of the corresponding frames. To find the texture coordinate for each pattern in the current frame, the origin and bases of its UV map is updated using the similarity transform computed while maintaining the range variation and tone values. The similarity transform is computed patch wise and blended appropriately at the boundary to avoid sliding effect in successive frames.

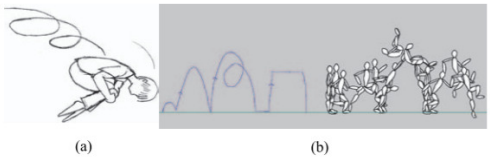
3.3. Animating 3D models

Animating a 3D model includes figuring out the path of motion and the keyframes shapes, i.e., deformations that the model undergoes in a period of time. In the context of sketch based animation, there are two variants for inferring the path and deformation - inferring from multiple sketches of the model, or inferring from a sketch stroke and applying it to the model. The choice between these variants depends upon the artists as well as the kind of output expected. Specifying animations using the sketched strokes has been the theme of quite a few sketch based content creation methods. The animation properties which are specified with the strokes can range from the motion path to the shape deformation for the 3D model. This section is dedicated for reviewing those methods which address the above issues.

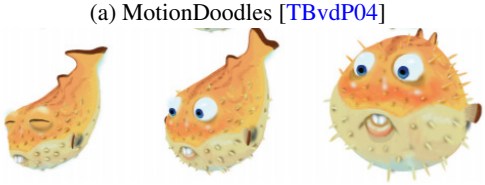
An approach for specifying the motion path for animating a model in both 2D and 3D environment is devised in [TBvdP04]. A basic human skeletal model is generated from a set of continuous closed strokes representing the salient parts of a human body joined together. The major axis corresponding to this ensemble of sketch strokes is used for placing the corresponding bones forming the skeleton. *Cursive strokes*, drawn to represent the motion path, are segmented using the vertical directions at corner points. Each of these segments is assigned to one of the eight possible token values depending upon the kind of curve it resembles. 18 different motion types can be represented through the stroke that is identified using the regular expression formed by sequence of tokens assigned to the segments. The corresponding motion is applied to the 3D model while appropriately keyframing and interpolating the frames. Similarly, a motion path can be sketched in a 3D scene as well.

A lot of prior work deals with the creation of 3D animation from the 2D sketches. Davis et al. [DAC*03] present a method to parse a sequence of sketches to create articulated 3D animation. Chaudhuri et al. [CKB04] present view-dependent character animation as a way to incorporate the relationship between the character pose and the camera viewpoint as inferred from a sequence of sketches. Jain et al. [JSH09] propose a method to utilize the skills of traditionally trained 2D sketch based animators to create 3D animation of a human motion. They use captured motion to create the 3D animation from the input 2D sketches, while maintaining the timing of the original animation. Further work by Jain et al. [JSMH12] allow the animation of 3D proxies like simulated 3D clothing to be influenced by the sketched 2D animation.

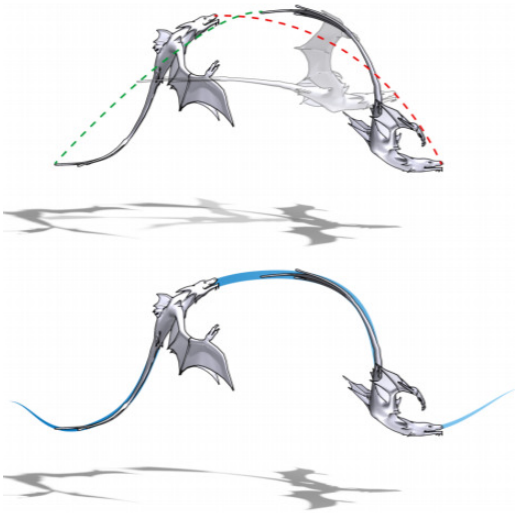
Bergig et al. [BHESB09] proposed a method to animate a mechanical system in AR from 2D sketches using a physical simulation system which processes the annotations given by the user to specify parameters like force and friction. Methods like the one given in [YW10] attempt to provide automatic rigging and skinning



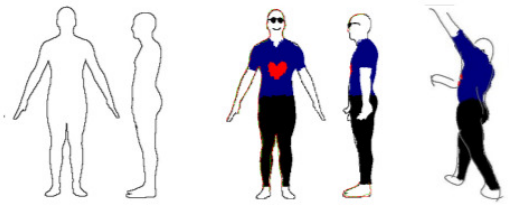
(a) MotionDoodles [TBvdP04]



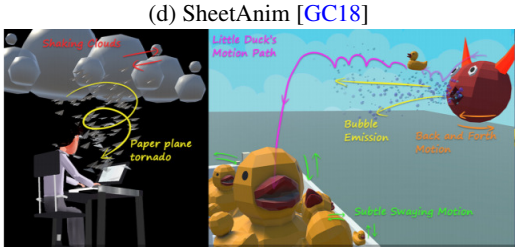
(b) Painterly characters [BBS*13]



(c) Space-time animation [GRGC15]



(d) SheetAnim [GC18]



(e) Magical Hand [AKK*19]

Figure 12: *Animating models from various methods discussed in Section 3.3. The caption contains reference of the method which generated the corresponding image.*

for simple models created using the sketches to which any motion data can be applied.

Bassett et al. [BBS*13] interpolate sketched keyframes, thus, integrating sketching to the traditional animation pipeline. A 3D model is loaded which provides a premise for sketching the strokes. The rigged 3D model is posed for the keyframes using any traditional modelling software. The sketch strokes are transformed accordingly using a convex combination of a set of matrices given by $M = \sum_i w_i M_i$. Each matrix M_i is computed corresponding to each vertex of the 3D model such that it best aligns the vertex and its one ring neighbour (see Section 2.6) to the required target position in a least square sense. An appropriate blend of the temporal keyframes from the traditional animation pipeline and the *configuration-space keyframes* produced from this method is used for the interpolation which creates the final sequence of animation. The configuration-space keyframes is defined using the rig parameters, light and camera positions, opacity and position of sketched strokes. A Voronoi classification based interpolation technique is used to generate these sequence of frames.

Guay et al. formulated a technique in [GRGC15] which infers both the temporal and spacial poses required to animate a 3D model from a sketched stroke. These poses are specified as a surface parameterized by time and a line parameter. A moving window approach is used to compute this surface, where the ratio of the length in rest pose to the full length of the path is used to compute these two parameters in each window. Singular points in the strokes and the velocity of drawing are used to make the motion specific computations like the take off and landing times and the time of flight, respectively. To make the model follow a particular shape at a given time, the method computes the optimal rotation angle for each bone of the skeleton constrained to retain the length of the mode. Various operations like over-sketching to edit current path, changing shape of out of plane features of the model, applying periodic motion to the model, and specifying out of plane motion path are supported.

Another method for creating animation sequence for characters from sketches is proposed in [GC18]. The user provides two sketches, one from the front view and the other from the side view. An approximate template mesh model is implicitly deformed to fit the mesh. Subsequently, the user needs to drag and adjust a desired skeleton into the model. The pose of the model is matched to that of the sketch by using a standard 2D-3D joint matching technique, where the ambiguity related to scale is resolved by using the two input sketches. Any motion data can be applied to this model and the interface deforms the model accordingly using a standard skinning technique. This is then used to generate animation in the form of deformed sketches where novel sketched views are generated by matching sketch strokes to silhouettes of the posed mesh.

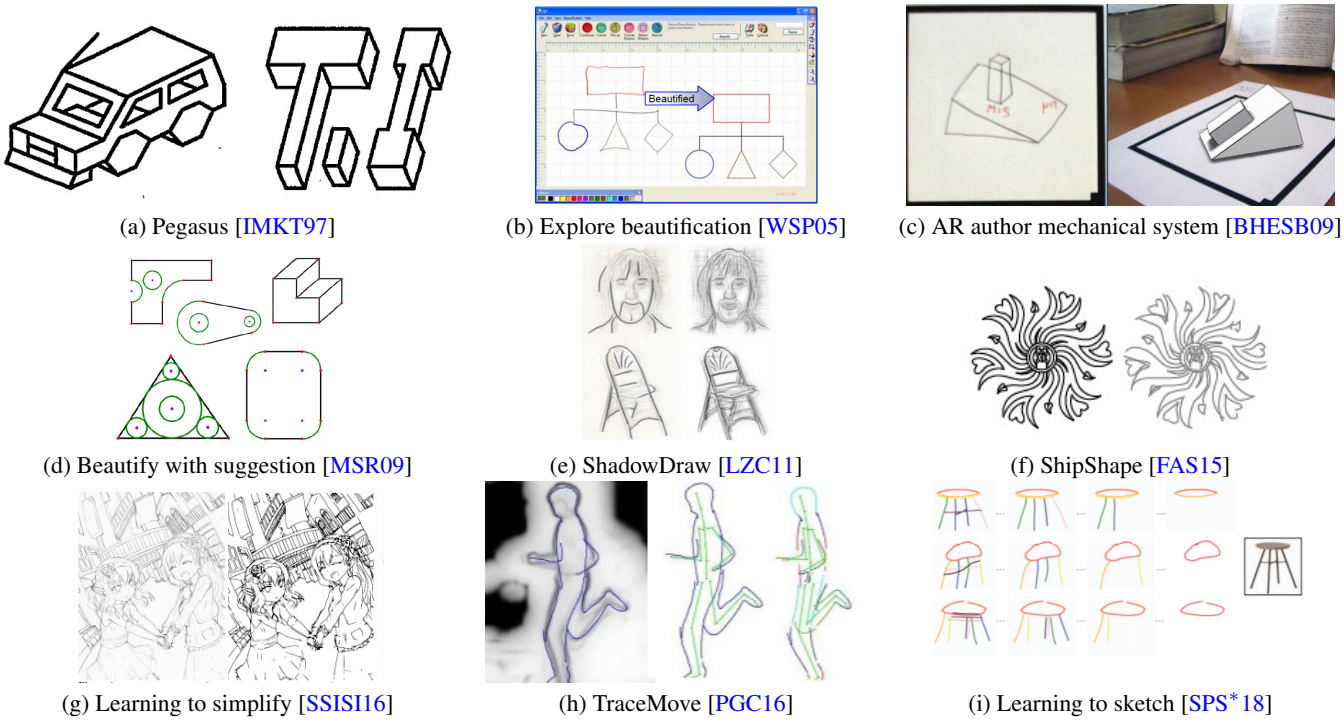
Recently, Arora et al. propose a method in [AKK*19] to animate 3D objects using the hand gestures in the AR/VR platforms. Based upon a thorough user study, they formulated a taxonomy of a subset of significant hand gestures that can suffice for specifying the actions corresponding to animations. As a proof-of-concept tool, they develop an interactive system, which tracks the hands of the user and interprets the intended gesture. The action mapped with the interpreted gesture is applied to the scene. The attributes ranging from the pose of the hand to the speed of its movement are used to compute various animation related parameters. The keyframing and interpolations are taken care of by the system internally. Figure 12 shows images of the interfaces and their respective sketch inputs for the methods discussed in this section.

3.4. Assisting sketching

Assistance in sketching is not an output of the sketch based content creation pipeline, but it proves to be very helpful at each step of the pipeline. The process of sketching becomes much more easier if the artists are provided with some possible set of strokes to guide them while drawing any object. This results in more precise sketches, thus, improving the quality of the output generated by any method. A category of methods designed for the purpose of providing assistance generally identifies the object the artist intends to sketch after each stroke and infers the possible set of strokes that can be drawn to complete the sketch. One such method for guiding 2D sketching is given in [LZC11], where the probable strokes to be drawn to complete an ongoing sketch are displayed as a shadow. A dataset of multiple images belonging to forty different categories of objects is considered. The edge map for each image of the dataset is computed using the length and curvature of each edge and is split into multiple patches represented using the image descriptors. Every patch has a predefined number of encoded descriptors that are stored in a structure indexed using the image number and patch number. Each of these descriptors are encoded using the min-hashing technique. A similar processing is applied to the ongoing sketch and a voting technique is applied to determine the top 100 candidate images. Each candidate is appropriately aligned and a weight $W_i = \alpha \frac{v_i V_i}{\sum_j v_j V_j}$ is associated with each of them depending upon the number of strokes drawn as well as the edge correlation of a candidate image with the sketched strokes. Here, v_i captures the tradeoff between the positively and negatively correlated edges of the strokes and the candidates images and V_i captures the spatial variation of the positive correlation of these edges. The probable strokes to be drawn is the weighted combination $S = \sum_i W_i E_i^{'}$ of all the candidate images represented by its edges $E_i^{'}$.

Patel et al. [PGC16] extend the method in [LZC11] to provide assistance in sketching animated characters. For a static pose of a character, they used the Shadowdraw method described above, using a database of recorded videos of people performing particular actions. They modify the kind of edge map generated for each image from the database of the videos by biasing it towards long edges for better results. Once the user is done with sketching the character, assistance is also provided for animating it. A motion capture database is used for mapping various poses of the sketched character to a known pose. Initially, the user explicitly marks the joints of skeleton of the sketched character for the interface to map it to a best matching skeleton from the database. The skeleton of the character is scaled appropriately to match those from the database. A shadow hint about the pose next to the current pose in the motion capture data of the matched skeleton is shown to the user. The character sketch is then rigged appropriately to be deformed according to the prediction to produce the sketch for the next pose in the animation.

A method based on a hybrid deep neural network is proposed



(a) Pegasus [IMKT97] (b) Explore beautification [WSP05] (c) AR author mechanical system [BHESB09]

(d) Beautify with suggestion [MSR09] (e) ShadowDraw [LZC11] (f) ShipShape [FAS15]

(g) Learning to simplify [SSISI16] (h) TraceMove [PGC16] (i) Learning to sketch [SPS*18]

Figure 13: *Assistance provided in sketching by methods discussed in Section 3.4. The caption contains reference of the method which generated the corresponding image.*

recently in [SPS*18], which can be used to guide an artist while sketching though it has been developed to mimic the human sketching process. A combination of two supervised models, which generates a photo from the given sketch and vice versa, and two unsupervised models, which generates a sketch from sketch and photo from photo, is used. The photo encoder-decoder is a traditional CNN [KSH14] downsampler and upsampler whereas the sketch encoder-decoder is a bidirectional LSTM [HS97] sequence model. Both the encoders produce a latent vector which is used to condition the decoders while generating their respective results. All the four sub models are trained together and the loss function is a weighted sum of two terms, one intends to minimize the differences between corresponding predicted results and the ground truth values and the other forces all the sub models to create latent vectors with similar distributions.

Another way of providing assistance in sketching is to *beautify* it i.e., allow the user to draw free-hand strokes and rectify them by inferring the intended geometry to produce cleaner sketches. The methods like [IMKT97, WSP05, BHESB09, MSR09, FAS15, SSISI16] attempt to beautify the free-hand sketches. In [IMKT97], the geometric relations like connectivity and symmetry are determined between the input stroke and the segments or curves present in the on-going sketch. A set of all possible constraints with respect to the input stroke are inferred using the parameters of previous strokes and this set of constraints is solved to find multiple candidates for the desired stroke. These candidates, ranked by their likelihood of being correct, are displayed in an ascending order of their rank from which the user can select the most suitable corrected stroke.

Wang et al. [WSP05] design an interface to integrate various contrasting beautification modes like manual or continuous, pen-up triggered or corner triggered, or formal shadowing or sketch shadowing in a grid based interface. It works by identifying the basic primitive like circles (full, half, quarter), rectangles and triangles in the sketch and modify it accordingly. Bergig et al. [BHESB09] provide a two-stage beautification technique. In the first stage, a sketch of an object in a mechanical system is modified to align the straight lines properly using *angular histograms* to reduce deviations. In the second stage, the model constructed from the sketch is modified to counter the possible displacement of the vertices due to reconstruction. For this, the vertices are shifted to the intersection point of the three faces sharing it and then the vertices are snapped to a 3D grid.

Complementary to a method which beautifies individual strokes, the method proposed in [MSR09] is focused on beautifying the entire sketch with addition of each new stroke. Multiple disjoint sets of the geometric constraints are inferred between the input stroke and previous strokes satisfying various perceptual properties and Gesalt continuities [Kof13]. Each set of constraints is solved separately to produce different suggestions. These suggestions are evaluated using the cost function associated with each of the constraints and their difference with other suggestions to avoid redundancy and clutter in the user selection display.

In [FAS15], a technique to beautify the curved paths in sketches

represented as Bézier curves is proposed. A tree structure is maintained with each node representing the path to be modified where each edge represents a unique constraint to be applied, and the leaf nodes represent the suggested curves. This tree is traversed in *best-fit search* manner till some pre-defined number of leaf nodes are encountered. These set of leaf nodes represent the set of suggestions from which user can choose the desired one or can continue sketching. For recognising the similar curves for the relations like symmetry or reflection, previous paths and input path re-sampled with lesser number of points are matched with each other.

A deep learning based technique to automatically produce clean vector images from the rough sketches is proposed in [SSISI16]. They use a CNN based framework using only convolutional layers clubbed with a loss map, which is computed using the histogram around each pixel in the ground truth label to avoid unnecessary bias towards thicker strokes in the sketch. The network is trained with a dataset having pairs of a rough sketch and its corresponding clean image. The dataset is constructed using the inverse construction technique of letting the artists draw the rough sketch for a given clean sketch for better results. This technique works on the entire sketch simultaneously and not on a stroke-by-stroke basis. Results of various assistance techniques described above are given in Figure 13.

4. Virtual sculpting - Another paradigm for content creation

Sketching in 3D using AR or VR is almost akin to sculpting, as the user can walk around the sketch, inspect and work on it from any direction.

Virtual sculpting has been a popular way of creating 3D models. As compared to the sketch based modelling, virtual sculpting still needs the user to learn some set of skills to be able to use it proficiently. Moreover, for designing an interface for virtual sculpting, the haptic feedback also plays an important role along with all other factors related to the 3D modelling.

In this section, we provide a brief discussion of some of the interfaces proposed over the years to create 3D models using the virtual sculpting. However, since it is a rich field of research in itself, an elaborate discussion on it is beyond the scope of this survey.

McDonnell et al. [MQW01] propose an interactive sculpting framework based upon *subdivision solids* and *physics-based modeling*. Due to the support for natural response to the applied forces, their interface gives a user the illusion of sculpting with a virtual clay. The user can change the topology of the object with a wide range of sculpting tools that are provided. In [SCC11], a method is proposed to sculpt *triangular* manifold meshes with arbitrary deformations made to the surface in the real-time using *quasi-uniform* meshes. The user can make the changes to the mesh using various gestures selected from a tool set to apply operations ranging from smoothing the surface to inflating an area of the surface. The mesh is deformed according to the user input while ensuring the output mesh to be closed and non-self intersecting. De Goes and James [DGJ17] present *regularized Kelvinlets* as a way to perform physically-based virtual sculpting where the deformations are obtained by regularization of fundamental solutions of linear elasticity. The method can produce realistic simulations of elastic materials during interactive sculpting.

An AR based 3D sculpting system is proposed by Jang et al. [JKWW14] to enable sculpting objects in a real world scene with mid-air hand gestures. A head mounted display and a head mounted RGBD camera is used to track the hands and head of the user in the 3D space as well as to render the virtual sculpted object seamlessly blended with the real world scene. Lu et al. [LSL*15] attempt to provide an interface for a *non-photorealistic* rendering of a scene augmented with the virtual content created using sculpting in real-time. The surfaces are deformed using haptic interaction and the boundaries for each element of the scene is highlighted using the techniques for edge detection. Appropriate *post-processing* is applied to the scene for better visualization and the coherence of the rendered scene is maintained over time and space.

In [CDCG18], virtual sculpting of a mesh is done using a *tangible* surface with pressure sensors attached to it. The idea behind providing such an interface is to give the user an illusion of touching the mesh while manipulating it. The interface coupled to a VR platform and, thus, immerses the user with respect to both the visual and haptic senses.

5. Usability factors for sketch based interfaces

Our work is mainly focused on the technical aspects of various components in a sketch based content creation pipeline with respect to the kind of input and output, and the methods used to generate the output from the input. However, as is the case with any human-computer interface, usability, human computing factors and perception issues provide vital feedback and guidelines for their design. These aspects are the defining factors in deciding the effectiveness of such an interface to a variety of users. They also describe the kind of applications in which each category of these methods can be suitably used.

A detailed discussion on these aspects revolving around usability and human computing factors of the sketch based interface can be found in the various existing case studies and surveys like [CCNA06, dAMGW09, WIMB10, IMG13, JH13, MVN13].

In particular, an experimental study and analysis on various usability related factors for sketching based interfaces in the VR platform is presented in [AKA*17]. Dudley et al. [DSK18] also present an elaborate discussion on different techniques available for bare hand sketching in the mid-air. They design different tasks which broadly cover the spectrum of the major operations involved in creating the contents using mid-air hand gestures. Each of the interaction technique is evaluated on multiple parameters ranging from the completion time to the accuracy of the final output. This work provides a set of principles that needs to be followed in general while designing a mid-air hand gestures based system. It also analyses the useful of general mid-air interaction techniques to a user depending upon the task at their hands. Once these usability factors are known, strategies to evaluate the interfaces with respect to these usability aspects also needs to be considered to ensure their competency with the available traditional methods. The effectiveness of any interactive interface including the sketch based interfaces

with respect to its human computing factors can be formally evaluated using various strategies specifically designed for this purpose. A detailed discussion on these strategies and their applicability in different situations is presented in [LHV*18].

Fernando et al. [FWK19] study different characteristics of strokes in the drawing process which solely depend on the techniques and perception of the artist. They discuss various user aspects that need to be taken into consideration regarding the interpretation and visualization of the sketched strokes while designing any interactive interface for sketching. Arora et al. [AKK*19] present a detailed study and analysis of the different interpretations of numerous hand gestures made in the mid-air by a user to depict various actions for animating an object. They attempt to understand the preferences of different users for the various gestures that they would use to specify a particular action of animation. Their user study provides a direction for making a VR platform based animation interface more intuitive for any user.

6. Future work and open problems

The field of sketch based content creation has seen a variety of research over the past two decades. However, there are numerous challenges and open problems that are yet to be addressed.

Generating 3D models from sketches is still a challenging problem. Directly modelling contents in 3D on the desktop, using professional tools like ZBrush [Pix19] requires a significant amount of skill. Therefore, there is considerable interest in creating the sketch based tools that can directly parse the sketches to create the 3D contents. The 2D sketch based methods are limited by the proficiency of the artist to represent 3D on a flat surface. They either need multiple views of the objects to be sketched or they assume that the model being sketched is symmetrical about some plane. Otherwise, a matching 3D object is retrieved or heavily detailed 2D or 3D sketch is drawn to create accurate 3D models. This makes the process of 3D modelling unnecessarily complicated. On the other hand 3D sketches are too difficult to be drawn accurately, especially by the novice users. Existing methods like [AHKG*18, XFZ*18] need to constrain the user significantly to function. This often hampers the creative liberty of the user and makes them spend more time in handling the technical issues.

There is need for methods and techniques that can automatically understand where to place each stroke in a free 3D space, as they are drawn by the user. This requires the method to know or learn the semantic meaning of each stroke or set of strokes using which a plausible a 3D model can be constructed.

An open problem is therefore to parse the general intent of the artist from a sparse set of sketched strokes. The method should be amenable to providing a feedback to help the novice artist and also allow intuitive control to allow the expert artist easy handles to create as they want. We postulate that a combination of learning based methods to infer the intended structure being drawn by the user and methods leveraging the benefits offered by the AR/VR platforms can potentially solve this issue.

Distinguishing pattern strokes from the contour strokes is another major issue that has not been solved convincingly. Many of the existing methods that infer the color or texture a 3D model from sketches, directly apply the same color from a sketch to the model. This causes loss of the drawing style that the user might want to reflect in the final 3D model. Most of the methods do not try to understand the underlying geometry of the object while coloring it which provides essential cues about the surface of an object. Some methods like [BBS*13, AGYS14] model the artists' intent better however, they do not support simultaneous modelling and coloring. This problem is exacerbated during 3D sketching because of poorer control on the sketched strokes. There is need for methods that can understand the kind of texture, material, and color that should be given to a particular 3D model or a part of it based upon the strokes corresponding to fine grained details.

3D sketch assisted animation has a lot of potential to be a very intuitive medium for creation of the 3D animated contents. 2D sketching, though intuitive for traditionally trained animators, requires hours of painstaking work to create the illusion of 3D movement. Popular 3D animation techniques on the desktop do not leverage all the benefits of sketching. 3D sketching in the AR/VR platforms can naturally provide the input for animation in a 3D space by allowing a sketch based access to the entire domain of movement. The recent works like [HAC*16, HDAC*16, ZDG*17] also demonstrate the usefulness of the AR/VR platforms in scene planning, that is crucial to the animated film making. Recently, Arora et al. [AKK*19] presented a method that tries to make use of the benefits of virtual reality to author 3D animation. We believe that the sketch based methods assisted by virtual and augmented reality can provide much richer interfaces for creation of animated 3D content.

7. Conclusions

Sketch based content creation is an area of wide interest and active research. The advancements in VR and AR technologies in recent times potentially provide more ways in which we can look into the classical problems like sketch based modelling and animation. In this work, we provide a detailed and structured review of all aspects of sketch based content creation. We begin by explaining some fundamental concepts relevant to this study. Then we discuss the details of various sketch based methods that have been proposed over the years for creating the coloured and textured 3D models as well as animating them. We describe methods which can be potentially combined with the AR/VR technologies helping them overcome many of their shortcomings due to being restricted to the traditional desktop or tablet platforms. We also review recent methods that actually leverage some of the advantages offered by the AR and VR platforms in the context of content creation.

However, in spite of all the developments in this area over the past decades, there remain many challenges which are unsolved. We believe efforts made towards better understanding the information offered by the sketches in the VR and AR platforms will be of great benefit to modelling and animation of 3D content. We hope this work encourages researchers to work on these challenges and further advance the state of the art.

References

[AGYS14] ABDRASHITOV R., GUY E., YAO J., SINGH K.: Mosaic: sketch-based interface for creating digital decorative mosaics. In *Proceedings of the 4th Joint Symposium on Computational Aesthetics, Non-Photorealistic Animation and Rendering, and Sketch-Based Interfaces and Modeling* (2014), ACM, pp. 5–10. 8, 15, 20

[AHKG*18] ARORA R., HABIB KAZI R., GROSSMAN T., FITZMAURICE G., SINGH K.: Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), ACM, p. 185. 2, 8, 13, 14, 20

[AKA*17] ARORA R., KAZI R. H., ANDERSON F., GROSSMAN T., SINGH K., FITZMAURICE G. W.: Experimental evaluation of sketching on surfaces in vr. In *CHI* (2017), vol. 17, pp. 5643–5654. 19

[AKK*19] ARORA R., KAZI R. H., KAUFMAN D., LI W., SINGH K.: Magicalhands: Mid-air hand gestures for animating in vr. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (2019), ACM, pp. 463–477. 8, 16, 17, 20

[AKKS99] ANKERST M., KASTENMÜLLER G., KRIEGEL H.-P., SEIDL T.: 3d shape histograms for similarity search and classification in spatial databases. In *International symposium on spatial databases* (1999), Springer, pp. 207–226. 12

[ALS08] ARCILA R., LEVET F., SCHLICK C.: Thor: Sketch-based 3d modeling by skeletons. In *International Symposium on Smart Graphics* (2008), Springer, pp. 232–238. 8, 10

[AS11] ANDRE A., SAITO S.: Single-view sketch based modeling. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (2011), ACM, pp. 133–140. 7, 8, 10

[Aut17] AUTODESK: Maya, 2017. https://www.autodesk.com/products/maya/overview Last accessed on 15-10-2019. 1

[Aut19] AUTODESK: Max, 2019. https://www.autodesk.in/products/3ds-max/overview Last accessed on 15-10-2019. 1

[Azu19] AZURE M.: Microsoft kinnect, 2019. https://azure.microsoft.com/en-us/services/kinect-dk/ Last accessed on 21-01-2020. 12

[BAC*19] BONNICI A., AKMAN A., CALLEJA G., CAMILLERI K. P., FEHLING P., FERREIRA A., HERMUTH F., ISRAEL J. H., LANDWEHR T., LIU J., ET AL.: Sketch-based interaction and modeling: where do we stand? *AI EDAM 33*, 4 (2019), 370–388. 6

[BBS08] BAE S.-H., BALAKRISHNAN R., SINGH K.: Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (2008), ACM, pp. 151–160. 7, 8, 13, 14

[BBS*13] BASSETT K., BARAN I., SCHMID J., GROSS M., SUMNER R. W.: Authoring and animating painterly characters. *ACM Transactions on Graphics (TOG) 32*, 5 (2013), 156. 5, 8, 16, 17, 20

[BCV*15] BESSMELTSEV M., CHANG W., VINING N., SHEFFER A., SINGH K.: Modeling character canvases from cartoon drawings. *ACM Transactions on Graphics (TOG) 34*, 5 (2015), 162. 1, 5, 8, 9, 10, 14

[BHESB09] BERGIG O., HAGBI N., EL-SANA J., BILLINGHURST M.: In-place 3d sketching for authoring and augmenting mechanical systems. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality* (2009), IEEE, pp. 87–94. 7, 8, 16, 18

[BJD*12] BOROSÁN P., JIN M., DECARLO D., GINGOLD Y., NEALEN A.: Rigmesh: automatic rigging for part-based shape modeling and deformation. *ACM Transactions on Graphics 31*, 6 (2012), 198. 1, 4, 5, 8, 9, 10

[BKY99] BELHUMEUR P. N., KRIEGMAN D. J., YUILLE A. L.: The bas-relief ambiguity. *International journal of computer vision 35*, 1 (1999), 33–44. 11

[Ble17] BLENDER: Blender 2.79b, 2017. https://www.blender.org/ Last accessed on 15-10-2019. 1

[BSM*07] BRESLAV S., SZERSZEN K., MARKOSIAN L., BARLA P., THOLLOT J.: Dynamic 2d patterns for shading 3d scenes. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 20. 8, 15

[CA09] COOK M. T., AGAH A.: A survey of sketch-based 3-d modeling techniques. *Interacting with computers 21*, 3 (2009), 201–211. 6

[CBWG10] CHU N., BAXTER W., WEI L.-Y., GOVINDARAJU N.: Detail-preserving paint modeling for 3d brushes. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (2010), ACM, pp. 27–34. 1, 8, 15

[CCNA06] COMPANY P., CONTERO M., NAYA F., ALEIXOS N.: A study of usability of sketching tools aimed at supporting prescriptive sketches. In *Proceedings of the Third Eurographics conference on Sketch-Based Interfaces and Modeling* (2006), pp. 139–146. 19

[CDCG18] CALLENS E., DANIEAU F., COSTES A., GUILLOTEL P.: A tangible surface for digital sculpting in virtual environments. In *International Conference on Human Haptic Sensing and Touch Enabled Computer Applications* (2018), Springer, pp. 157–168. 19

[CKB04] CHAUDHURI P., KALRA P., BANERJEE S.: A system for view-dependent animation. *Computer Graphics Forum 23*, 3 (2004), 411–420. 8, 16

[CSSJ05] CHERLIN J. J., SAMAVATI F., SOUSA M. C., JORGE J. A.: Sketch-based modeling with few strokes. In *Proceedings of the 21st spring conference on Computer graphics* (2005), ACM, pp. 137–145. 7, 8

[CSVZ14] CHATFIELD K., SIMONYAN K., VEDALDI A., ZISSERMAN A.: Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531* (2014). 13

[CV10] CRUZ L. M. V., VELHO L.: A sketch on sketch-based interfaces and modeling. In *2010 23RD SIBGRAPI-Conference on Graphics, Patterns and Images Tutorials* (2010), IEEE, pp. 22–33. 6

[DAC*03] DAVIS J., AGRAWALA M., CHUANG E., POPOVIĆ Z., SALESIN D.: A sketching interface for articulated figure animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), SCA '03, Eurographics Association, pp. 320–328. 8, 16

[dAMGW09] DE ARAUJO MACHADO T. L., GOMES A. S., WALTER M.: A comparison study: Sketch-based interfaces versus wimp interfaces in three dimensional modeling tasks. In *2009 Latin American Web Congress* (2009), IEEE, pp. 29–35. 19

[Dee95] DEERING M. F.: Holosketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction (TOCHI) 2*, 3 (1995), 220–238. 7

[DGJ17] DE GOES F., JAMES D. L.: Regularized kelvinlets: Sculpting brushes based on fundamental solutions of elasticity. *ACM Transactions on Graphics 36*, 4 (July 2017). 19

[DL16] DING C., LIU L.: A survey of sketch based modeling systems. *Frontiers of Computer Science 10*, 6 (2016), 985–999. 6

[DPS15] DE PAOLI C., SINGH K.: Secondskin: sketch-based construction of layered 3d models. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 126. 8, 10, 11, 12, 14

[DSK18] DUDLEY J. J., SCHUFF H., KRISTENSSON P. O.: Bare-handed 3d drawing in augmented reality. In *Proceedings of the 2018 Designing Interactive Systems Conference* (2018), pp. 241–252. 19

[EBC*15] ENTEM E., BARTHE L., CANI M.-P., CORDIER F., VAN DE PANNE M.: Modeling 3d animals from a side-view sketch. *Computers & Graphics 46* (2015), 221–230. 3, 7, 8, 9, 10

[Fac19a] FACEBOOK: Oculus, 2019. https://www.oculus.com Last accessed on 15-10-2019. 2

[Fac19b] FACEBOOK: Quill, 2019. https://quill.fb.com/ Last accessed on 22-10-2019. 2

[Far92] FARIN G.: From conics to nurbs: A tutorial and survey. *IEEE Computer Graphics and applications*, 5 (1992), 78–86. 14

[FAS15] FIŠER J., ASENTE P., SỲKORA D.: Shipshape: a drawing beautification assistant. In *Proceedings of the workshop on Sketch-Based Interfaces and Modeling* (2015), Eurographics Association, pp. 49–57. 8, 18

[FWK19] FERNANDO P., WEILER J., KUZNETSOV S.: A rough sketch of the freehand drawing process: Blending the line between action and artifact. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–13. 20

[FYX17] FENG L., YANG X., XIAO S.: Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. In *2017 IEEE Virtual Reality (VR)* (2017), IEEE, pp. 195–204. 7, 8, 10, 14

[GC18] GUPTA H., CHAUDHURI P.: Sheetanim-from model sheets to 2d hand-drawn character animation-. In *VISIGRAPP (1: GRAPP)* (2018), pp. 17–27. 8, 14, 16, 17

[GJS18] GIUNCHI D., JAMES S., STEED A.: 3d sketching for interactive model retrieval in virtual reality. In *Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering* (2018), pp. 1–12. 8, 13

[Goo17a] GOOGLE: Blocks, 2017. https://vr.google.com/blocks/ Last accessed on 24-04-2019. 2

[Goo17b] GOOGLE: Gravity Sketch, 2017. https://www.gravitysketch.com Last accessed on 15-10-2019. 2

[Goo17c] GOOGLE: Tilt Brush, 2017. https://www.tiltbrush.com Last accessed on 15-10-2019. 2

[GRGC15] GUAY M., RONFARD R., GLEICHER M., CANI M.-P.: Space-time sketching of character animation. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 118. 8, 16, 17

[HAC*16] HENRIKSON R., ARAUJO B., CHEVALIER F., SINGH K., BALAKRISHNAN R.: Multi-device storyboards for cinematic narratives in vr. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (2016), ACM, pp. 787–796. 20

[HDAC*16] HENRIKSON R., DE ARAUJO B., CHEVALIER F., SINGH K., BALAKRISHNAN R.: Storeoboard: Sketching stereoscopic storyboards. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), ACM, pp. 4587–4598. 20

[HGB*10] HAGBI N., GRASSET R., BERGIG O., BILLINGHURST M., EL-SANA J.: In-place sketching for content authoring in augmented reality games. In *2010 IEEE Virtual Reality Conference (VR)* (2010), IEEE, pp. 91–94. 8, 11, 14, 15

[HGY17] HAN X., GAO C., YU Y.: Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 126. 1, 3, 8, 10, 12

[Hot33] HOTELLING H.: Analysis ofa complex statistical variables into principal components. *Journal ofEducational Psychology, Sept. and Oct., 1933HotellingSept. Journal of Educational Psychology1933* (1933). 12

[HR17] HUO K., RAMANI K.: Window-shaping: 3d design ideation by creating on, borrowing from, and looking at the physical world. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction* (2017), pp. 37–45. 8, 12, 13, 14

[HS97] HOCHREITER S., SCHMIDHUBER J.: Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780. 18

[HTC19] HTC: HTC Vive, 2019. https://www.vive.com/in/ Last accessed on 15-10-2019. 2

[HZ03] HARTLEY R., ZISSERMAN A.: *Multiple View Geometry in Computer Vision*, 2 ed. Cambridge University Press, USA, 2003. 14

[IH03] IGARASHI T., HUGHES J. F.: Smooth meshes for sketch-based freeform modeling. In *Proceedings of the 2003 symposium on Interactive 3D graphics* (2003), ACM, pp. 139–142. 7, 8

[IMG13] ISRAEL J. H., MAUDERLI L., GRESLIN L.: Mastering digital materiality in immersive modelling. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling* (2013), pp. 15–22. 19

[IMKT97] IGARASHI T., MATSUOKA S., KAWACHIYA S., TANAKA H.: Interactive beautification: A technique for rapid geometric design. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1997), UIST âĂŹ97, Association for Computing Machinery, p. 105âĂŞ114. URL: https://doi.org/10.1145/263407.263525, doi:10.1145/263407.263525. 18

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH Conference on Computer graphics and interactive techniques* (1999), ACM, pp. 409–416. 1, 4, 7, 9, 10, 11, 14

[JH13] JANKOWSKI J., HACHET M.: A Survey of Interaction Techniques for Interactive 3D Environments. In *Eurographics 2013 - State of the Art Reports* (2013), The Eurographics Association. 19

[JKWW14] JANG S.-A., KIM H.-I., WOO W., WAKEFIELD G.: Airsculpt: A wearable augmented reality 3d sculpting system. In *International Conference on Distributed, Ambient, and Pervasive Interactions* (2014), Springer, pp. 130–141. 19

[JSH09] JAIN E., SHEIKH Y., HODGINS J.: Leveraging the talent of hand animators to create three-dimensional animation. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), ACM, pp. 93–102. 8, 16

[JSMH12] JAIN E., SHEIKH Y., MAHLER M., HODGINS J.: Three-dimensional proxies for hand-drawn characters. *ACM Trans. Graph. 31*, 1 (Feb. 2012), 8:1–8:16. 8, 16

[Kal60] KALMAN R. E.: A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D (1960), 35–45. 12

[KAS*19] KOULIERIS G. A., AKŞIT K., STENGEL M., MANTIUK R., MANIA K., RICHARDT C.: Near-eye display and tracking technologies for virtual and augmented reality. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 493–519. 2

[KF94] KRUEGER W., FROEHLICH B.: Responsive workbench. In *Virtual RealityâĂŹ94*. Springer, 1994, pp. 73–80. 12

[Kof13] KOFFKA K.: *Principles of Gestalt psychology*. Routledge, 2013. 9, 18

[KSH14] KRIZHEVSKY A., SUTSKEVER I., HINTON G.: Imagenet classification with deep convolutional neural. In *Neural Information Processing Systems* (2014), pp. 1–9. 11, 12, 13, 18

[Leo17] LEOPOLY: ShapeLab, 2017. https://shapelabvr.com/ Last accessed on 22-10-2019. 2

[LG13] LEVI Z., GOTSMAN C.: Artisketch: A system for articulated sketch modeling. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 235–244. 10

[LGS07] LEVET F., GRANIER X., SCHLICK C.: Multi-view sketch-based freeform modeling. In *International Symposium on Smart Graphics* (2007), Springer, pp. 204–209. 8, 10

[LHV*18] LEDO D., HOUBEN S., VERMEULEN J., MARQUARDT N., OEHLBERG L., GREENBERG S.: Evaluation strategies for hci toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–17. 20

[LLG*15] LI B., LU Y., GHUMMAN A., STRYLOWSKI B., GUTIERREZ M., SADIQ S., FORSTER S., FEOLA N., BUGERIN T.: 3d sketch-based 3d model retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (2015), pp. 555–558. 8, 12, 13

[LSL*15] LU P., SHENG B., LUO S., JIA X., WU W.: Image-based non-photorealistic rendering for realtime virtual sculpting. *Multimedia tools and applications 74*, 21 (2015), 9697–9714. 19

[LZC11] LEE Y. J., ZITNICK C. L., COHEN M. F.: Shadowdraw: real-time user guidance for freehand drawing. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 27. 8, 17, 18

[Mas19] MASTERPIECEVR: MasterpieceVR, 2019. https://www.masterpiecevr.com Last accessed on 22-10-2019. 2

[MNZ*15] MAGNENAT S., NGO D. T., ZÜND F., RYFFEL M., NORIS G., ROTHLIN G., MARRA A., NITTI M., FUA P., GROSS M., ET AL.: Live texturing of augmented reality characters from colored drawings. *IEEE transactions on visualization and computer graphics 21*, 11 (2015), 1201–1210. 8, 14, 15

[MQW01] MCDONNELL K. T., QIN H., WLODARCZYK R. A.: Virtual clay: A real-time sculpting system with haptic toolkits. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), pp. 179–190. 19

[MSK00] MITANI J., SUZUKI H., KIMURA F.: 3d sketch: sketch-based model reconstruction and rendering. In *International Workshop on Geometric Modelling* (2000), Springer, pp. 85–98. 11

[MSR09] MURUGAPPAN S., SELLAMANI S., RAMANI K.: Towards beautification of freehand sketches using suggestions. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling* (2009), pp. 69–76. 8, 18

[MVN13] MATTHEWS T., VOGTS D., NAUDÉ K.: Sketch-based interfaces: Drawings to data. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (New York, NY, USA, 2013), SAICSIT âĂŹ13, Association for Computing Machinery, p. 359âĂŞ368. URL: https://doi.org/10.1145/2513456.2513482, doi:10.1145/2513456.2513482. 19

[NISA07] NEALEN A., IGARASHI T., SORKINE O., ALEXA M.: Fibermesh: designing freeform surfaces with 3d curves. In *ACM SIGGRAPH 2007 papers*. ACM, 2007, pp. 41–50. 7, 8, 10, 11

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: a vector representation for smooth-shaded images. *ACM Transactions on Graphics (TOG) 27*, 3 (2008), 1–8. 11

[OSJ11] OLSEN L., SAMAVATI F., JORGE J.: Naturasketch: Modeling from images and natural sketches. *IEEE Computer Graphics and Applications 31*, 6 (2011), 24–34. 1, 3, 5, 7, 8, 9, 10, 11, 14

[OSSJ08] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: A taxonomy of modeling techniques using sketch-based interfaces. In *Eurographics (STARs)* (2008), Citeseer, pp. 39–57. 4, 6

[OSSJ09] OLSEN L., SAMAVATI F. F., SOUSA M. C., JORGE J. A.: Sketch-based modeling: A survey. *Computers & Graphics 33*, 1 (2009), 85–103. 6

[PGC16] PATEL P., GUPTA H., CHAUDHURI P.: Tracemove: A data-assisted interface for sketching 2d character animation. In *VISIGRAPP (1: GRAPP)* (2016), pp. 191–199. 8, 17, 18

[Pix19] PIXOLOGIC: ZBrush, 2019. https://pixologic.com/ Last accessed on 15-10-2019. 20

[RRS19] ROSALES E., RODRIGUEZ J., SHEFFER A.: Surfacebrush: from virtual reality drawings to manifold surfaces. *ACM Trans. Graph. 38*, 4 (2019), 96:1–96:15. 8, 13

[SAG*13] SHTOF A., AGATHOS A., GINGOLD Y., SHAMIR A., COHEN-OR D.: Geosemantic snapping for sketch-based modeling. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 245–253. 8, 10, 11

[SBS19] SMIRNOV D., BESSMELTSEV M., SOLOMON J.: Deep sketch-based modeling of man-made shapes. *arXiv preprint arXiv:1906.12337* (2019). 8, 10, 12

[SCC11] STANCULESCU L., CHAINE R., CANI M.-P.: Freestyle: Sculpting meshes with self-adaptive topology. *Computers & Graphics 35*, 3 (2011), 614–622. 19

[SH16] SCHMALSTIEG D., HOLLERER T.: *Augmented reality: principles and practice*. Addison-Wesley Professional, 2016. 2

[SI07] SHIN H., IGARASHI T.: Magic canvas: interactive design of a 3-d scene prototype from freehand sketches. In *Proceedings of Graphics Interface 2007* (2007), ACM, pp. 63–70. 8, 10, 11

[SKČ*14] SỲKORA D., KAVAN L., ČADÍK M., JAMRIŠKA O., JACOBSON A., WHITED B., SIMMONS M., SORKINE-HORNUNG O.: Ink-and-ray: Bas-relief meshes for adding global illumination effects to hand-drawn characters. *ACM Transactions on Graphics (TOG) 33*, 2 (2014), 1–15. 7, 8, 10, 11, 14

[SPS01] SCHKOLNE S., PRUETT M., SCHRÖDER P.: Surface drawing: creating organic 3d shapes with the hand and tangible tools. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), pp. 261–268. 8, 12, 13

[SPS*18] SONG J., PANG K., SONG Y.-Z., XIANG T., HOSPEDALES T. M.: Learning to sketch with shortcut cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 801–810. 8, 18

[SSISI16] SIMO-SERRA E., IIZUKA S., SASAKI K., ISHIKAWA H.: Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 1–11. 8, 18, 19

[TBvdP04] THORNE M., BURKE D., VAN DE PANNE M.: Motion doodles: an interface for sketching character motion. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 424–431. 2, 8, 16

[WIMB10] WIESE E., ISRAEL J. H., MEYER A., BONGARTZ S.: Investigating the learnability of immersive free-hand sketching. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium* (2010), pp. 135–142. 19

[WSP05] WANG B., SUN J., PLIMMER B.: Exploring sketch beautification techniques. In *Proceedings of the 6th ACM SIGCHI New Zealand chapter's international conference on Computer-human interaction: making CHI natural* (2005), pp. 15–16. 8, 18

[XCS*14] XU B., CHANG W., SHEFFER A., BOUSSEAU A., MCCRAE J., SINGH K.: True2form: 3d curve networks from 2d sketches via selective regularization. *ACM Transactions on Graphics 33*, 4 (2014), 131:1–131:13. 3, 7, 8, 10, 11

[XFZ*18] XU P., FU H., ZHENG Y., SINGH K., HUANG H., TAI C.-L.: Model-guided 3d sketching. *IEEE Transactions on Visualization and Computer Graphics* (2018), 2927–2939. 3, 8, 12, 13, 20

[XSS08] XIN M., SHARLIN E., SOUSA M. C.: Napkin sketch: handheld mixed reality 3d sketching. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology* (2008), ACM, pp. 223–226. 1, 2, 7, 8, 12, 13, 14

[YSvdP05] YANG C., SHARON D., VAN DE PANNE M.: Sketch-based modeling of parameterized objects. In *SIGGRAPH Sketches* (2005), Citeseer, p. 89. 8, 11

[YW10] YANG R., WÜNSCHE B. C.: Life-sketch: a framework for sketch-based modelling and animation of 3d objects. In *Proceedings of the Eleventh Australasian Conference on User Interface-Volume 106* (2010), Australian Computer Society, Inc., pp. 61–70. 8, 16

[YYL*17] YU Q., YANG Y., LIU F., SONG Y.-Z., XIANG T., HOSPEDALES T. M.: Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision 122*, 3 (2017), 411–425. 3, 8, 11

[ZDG*17] ZIMMER C., DROCHTERT D., GEIGER C., BRINK M., MÜTZE R.: Mobile previsualization using augmented reality: a use case from film production. In *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications* (2017), ACM, p. 24. 20

[ZHY17] ZHAO H., HUANG P., YAO J.: Texturing of augmented reality character based on colored drawing. In *2017 IEEE Virtual Reality (VR)* (2017), IEEE, pp. 355–356. 8, 15

Author Biographies

Sukanya Bhattacharjee is a research scholar in the Department of Computer Science and Engineering at Indian Institute of Technology (IIT) Bombay. She received her M.Tech. Degree in Computer

Science and Engineering from IIT Guwahati. She currently works in the area of sketch based and interactive applications for virtual and augmented reality.

Parag Chaudhuri is currently an Associate Professor in the Department of Computer Science and Engineering at Indian Institute of Technology (IIT) Bombay. He holds a Ph.D. from IIT Delhi, and has worked as a postdoctoral researcher at MIRALab, University of Geneva. He loves teaching computer graphics to enthusiastic students.

His research interests broadly cover the areas of computer graphics, virtual and augmented reality and computer vision. His current work centers around the animation of characters and natural phenomena to populate virtual worlds, and the study of methods to create, interact with and understand visual content.