Front. Comput. Sci.
DOI 10.1007/s11704-016-5422-9

REVIEW ARTICLE

A survey of sketch based modeling systems

Chao DING, Ligang LIU (✉)

School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China

© Higher Education Press and Springer-Verlag Berlin Heidelberg 2016

Abstract As 3D technology, including computer graphics, virtual reality and 3D printing, has been rapidly developed in the past years, 3D models are gaining an increasingly huge demand. Traditional 3D modeling platforms such as Maya and ZBrush, utilize "windows, icons, menus, pointers" (WIMP) interface paradigms for fine-grained control to construct detailed models. However, the modeling progress can be tedious and frustrating and thus too hard for a novice user or even a well trained artist. Therefore, a more intuitive interface is needed. Sketch, an intuitive communication and modeling tool for human beings, becomes the first choice of modeling community. So far, various sketch-based modeling systems have been created and studied. In this paper, we attempt to show how these systems work and give a comprehensive survey. We review and categorize the systems in four aspects: the input, the knowledge they use, the modeling approach and the output. We also discuss about inherent challenges and open problems for researchers in the future.

Keywords sketch-based modeling, interactive design, sketch comprehension

1 Introduction

3D modeling is a basic and core problem in computer graphics and has matured significantly over the last decades. Quantities of methods have been proposed, such as polygonal modeling [1], digital sculpting [2,3], scanning [4–6] and image-based modeling techniques [7]. Quite a few of these technologies have been successfully integrated into commercial 3D modeling softwares, such as 3D MAX and MAYA. However, among these techniques, some require that the target modeling object should already exist, like scanning and image-based methods. Meanwhile, other methods which need manual operations are suitable for creative work when target models do not exist or even are not real. It is also worth noting that manual techniques are useful for detail preserving and complex object modeling even target models exist.

However, there is no free lunch. To guarantee the quality of produced models, these softwares sacrifice usability and require highly trained people to manipulate them. This makes it very hard for using easily or for amateurs to easily create some 3D models which might not be that professional but just usable or maybe interesting. This is the original intention that drives researchers to create easy-to-use modeling systems.

Nevertheless, creating such a system is not an easy problem, and for a very long time, scientists had been looking for a more natural and convenient way for people to present 3D stuff in 2D canvas. Sketch, a natural way to express rough ideas, has been widely used in design such as cloth and architecture. Imagine that you have a very cool idea to solve a difficult problem, is it your first thought to make a draft and draw sketches? When you want to tell others about your idea, will you draw sketches to express yourself? Figure 1 shows how a chair sketch is drawn [8]. Because of the popularity of sketch, plenty of interactive methods based on sketch are created, such as sketch-based segmentation [9,10] and sketch-based animation [11,12]. Figure 2 shows a well-known sketch-based modeling system which is called teddy [13].

However, using 2D sketches to express 3D objects is ambiguous and computers might misunderstand it. Figure 3 shows a typical problem. Different 3D objects z1, z2, z3 share

Received October 11, 2015; accepted April 12, 2016

E-mail: lgliu@ustc.edu.cn

the same 2D sketch projected onto a plane. Thus it is impossible for computers to distinguish them without any prior knowledge. This important prior knowledge is where the main difficulty lies, and is the key to understanding sketches as well.

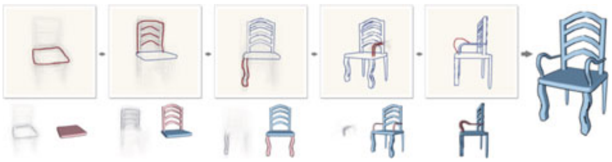


Fig. 1 A chair which is designed by sketches [8]

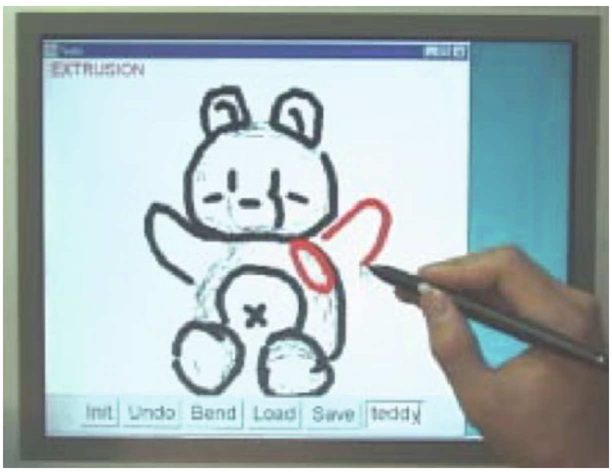


Fig. 2 Teddy: a well-known sketch-based modeling system [13]

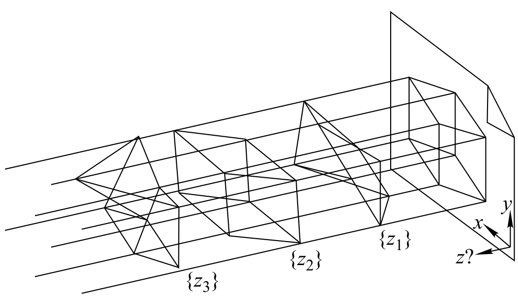


Fig. 3 Ambiguous interpretation of a 2D sketch in 3D: one 2D sketch corresponds to numerous 3D objects [14]

On the other hand, sketch is a drawing done quickly without a lot of details. How to extract a real object from the sketch is also a challenge. The robustness to the variety of sketches by different people is another big concern.

Fortunately, over the past 20 years, the sketch modeling community have made great efforts in sketch-based modeling (SBM) and have achieved large amounts of astonishing and outstanding work. Refs. [14,15] are two well-known survey papers which summerize those work comprehensively. However, compared to them, this paper is still different and novel in two aspects. For the contents, we includes the data-driven methods for sketch-based modeling. As for the recent decade, series of excellent methods in machine learning have been applied to SBM, the paper thus offers a detailed analysis for those up-to-date work. On the other hand, for the categorization, previous surveys have put great efforts on specific techniques of SBM. Here we realize that the interpretation of sketch is based on not only the techniques they used, but also the knowledge behind. Therefore, this paper additionally makes a discussion on the knowledge part, providing a brand new point of view for analysis.

In this survey, we use the notion of pipeline and put forward our own analysis in a methodology perspective. Figure 4 demonstrates our ideas. We initially need the input, and with certain knowledge assisted, we use different approaches to achieve user-defined output such as scenes and models. This pipeline is more suitable for recent SBM work while agreeing with traditional approaches. At first, we discuss the input which contains sketch filtering and sketch acquisition, instead of discussing the details of filtering separately. Then we show how different techniques are used in these systems and what prior knowledge they choose. We categorize the approaches into different classes such as sketch-based editing and sketch-based retrieval methods. Interested readers can quickly choose useful approaches. Subsequently, we mention the different kinds of output. At last, we make a review and discussion on the current work and show possible future directions for researchers.

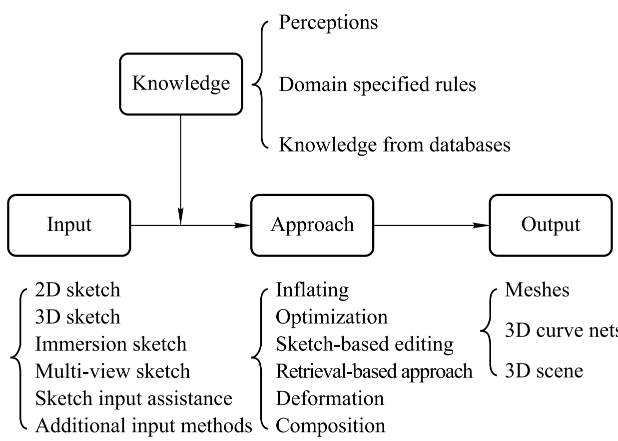


Fig. 4 The SBM pipeline: with proper knowledge supported, SBM systems use different approaches to generate needed output

2 Input

The most fundamental step in SBM systems is obtaining a sketch from users. According to representing formation, we divide the input into several categories. We also put the common additional input assistance into this classification. In this section, we will first show the common input device and then discuss each category of input in detail.

With the increasing elaboration and enlargement of techniques, the input devices vary from native mouses to digital sketch pads. Modern sketch pads may have pressure sensors to transform the force of input into needed information. Like real pencil-and-paper systems, sketch input devices want to transfer more information than merely the pressure information. With the help of modern input devices, SBM systems can help people express their modeling intention. Figure 5 shows a shadow guided user interface which uses shadow to help users draw sketch.

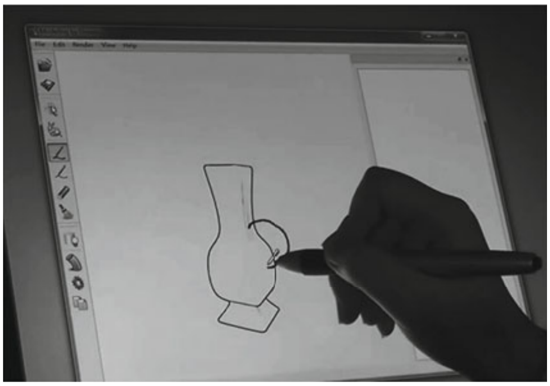


Fig. 5 Shadow guided user interface [8]

2.1 2D sketch

2D sketch is the most natural and easiest kind of input. With mouses or pen-based input devices, one can provide a sketch with positional information in some 2D coordinate systems, mostly in window coordinates [16–19]. This can be accomplished in not only the interactive mode but also the off-line mode.

2D sketch drawn by novice users may have some noisy or erroneous samples, so it must be filtered suitably before interpreting. Poor drawing skills and invariable device errors are the main reasons. The device errors are unavoidable due to device inner noise. Meanwhile, it is hard for users to draw straight line as well as strict sphere. Therefore, suitable filter should be applied before interpreting. Olsen et al. [14] have well categorized the filter into three procedures: 1) resampling and smoothing, 2) fitting, and 3) oversketching. These filter steps can also be used in 3D sketch which we will talk later.

More generally, 2D sketch input contains a special kind of image input, saying 2-value image. This kind of images only contains the information of image's silhouette, and can be easily transformed into sketches, so we suppose this kind of images as input for SBM system as well.

2.2 3D sketch

Sometimes, the input sketch drawn in a window coordinates should be transformed to that in another coordinates. Due to the pre-built coordinates, we call it 3D sketch input. We call the window plate associated with the view point as drawing canvas.

The simplest situation is that the drawing canvas is a plain plate [20–22]. When drawing a sketch on a drawing canvas, it means mapping an input sketch in a 2D coordinate which is a window coordinate, to a user-specified plane in model coordinates which is in 3D space. The mapping procedure usually follows the rule of projection. One can change the view point and direction to add more information to a sketch while 2D sketch cannot do. With further assumptions such as plane-symmetry, one can infer a unique symmetric 3D curve.

When targeting coordinate is not a plane, but an arbitrary surface, which commonly appears in sketch-based editing system [16,23–25] or editing steps in SBM system [13,26], we also regard this kind of sketch input as 3D sketch input. The input sketch also follows the rule of projection, just similar to that in the plane situation, but is more like drawing directly on a model. With further sketch gestures, one can operate on models.

2.3 Immersion sketch

Virtual reality has been gained its popularity in these days. With immersion devices, people can directly show modeling intentions and do 3D modeling. SPACESKETCH [27] is one of these work which is a unique application that performs shape modeling by using two space wands in a 3D environment. By using a 3D TV display as shown in Fig. 6, sketching an object in a stereoscopic space is thus possible.

Noting that a real SBM system can employ more than one kind of input according to its design intention. Many SBM work takes 3D input as editing gestures, while they use 2D sketch to build a prototype model [13,26]. On the other hand, 3D sketch input system can benefit from 2D input such as over sketch, filtering, etc.

Apart from different kinds of input, there are some ways of assistance that can reinforce the input. One can use multi-view information to help modeling interpretations or use special kind of sketches such as hatching to provide extra information. Considering not everyone is a good painter of sketches, some work attempts to help users to sketch better.

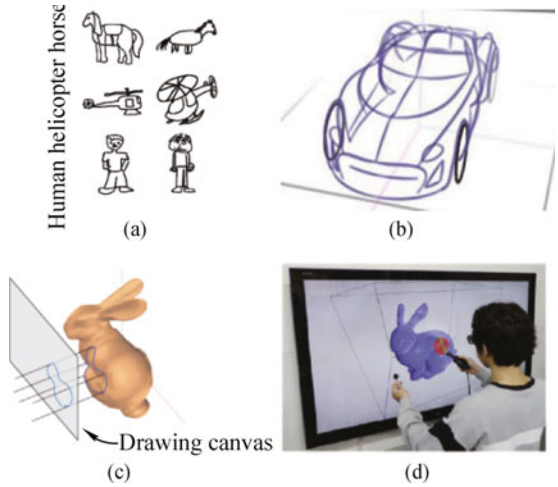


Fig. 6 Different forms of sketch inputs. (a) 2D sketch; (b) 3D sketch in curve net drawn from different views [20]; (c) sketches are embedded into 3D by projecting onto a drawing canvas, or perhaps onto existing geometry [14]; (d) sketch-based modeling system which can create models in 3D space [27]

2.4 Multi-view sketch

Multi-view method is well known as an image-based reconstruction method. Borrowing from that, by taking image silhouette or sketches as input, one can build 3D models from multi-view information.

Date back to earlier days, CAD modeling system is first applied to the so called multi-view techniques, as they acquire three orthogonal view points with each view consisting of few strokes. Also, with a 3-view coordinates, one can infer a unique CAD model. As shown in Fig. 7, this method is only suitable for man-made objects where their projective silhouettes are generally perpendicular or parallel to each other [28].

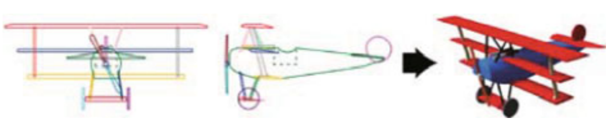


Fig. 7 Multi-view techniques which are used in sketch-based modeling system [28]

Multi-view systems can also be used for organic models. For more complicated models, one can apply multi-view techniques to each part of the final model. SBM work using sketch-based editing steps implicitly employs multi-view techniques [29].

2.5 Sketch input assistance

Although sketch is simple and natural, it is too abstract and hard to draw for those who are not well trained. Therefore, some work tries to help people express the original sketch they want to pass on to.

Lee et al. [30] use the database which contains different kinds of sketch to help users improve their sketch. It uses shadow to suggest the following strokes they may want to draw according to the database. The probability is represented by the gray scale. Inspired by this work, Fan et al. [8] further adopt the silhouette of existing 3D model from a database lo suggest the following strokes.

2.6 Additional input methods

Silhouette and feature lines are the common styles of input sketch. However, in some sense, it is too sparse to express an appropriate surface. Some work employs other kinds of input to add more information to surface. Hatching [31] is used for representing the normal vector to surface borrowing ideas from literary sketch. Annotation [32] is used for structures such as symmetry and others.

3 Knowledge

As mentioned before, constructing a concrete model from sketches is an ill-posed problem. So we need extra information and constraints to exclude the bad ones. We call the information as knowledge. We classify the knowledge into three categories corresponding to their methodology: perceptions, domain special rules, knowledge learning from database.

As we will mention below, perceptions are a set of common visual rules which people shares on sketch recognition. Interpreting sketch not following these rules may lead to an unnatural results, so almost every SBM work takes the perception as constraints directly or implicitly. When modeling from scratch, it is the only knowledge you can use. However, merely with the perception information, it is often not enough to infer a plausible model witch complicated topology. Particularly for a sophisticated model, you will need further information such as domain special rules and knowledge learning from database. The distinction between the above two is that domain special knowledge is a set of already summarized rules. Meanwhile, knowledge learning from database is hard to summarize and need to learn.

3.1 Perceptions

Despite creating a 3D model from scratch is a tough task for computers, with the well developed human visual system, it is far more easier. One can effortlessly infer a credible 3D model from sketch. The scientists from relevant areas such as cognitive science have fully studied this question and have some useful conclusions. They think humans do share a series of common visual rules on sketch recognition for sketch drawing and communication. They call this set of rules as perceptions.

What are these rules? Actually Hoffman [33] concludes the basic ten rules of inferring 3D models from sketch.

- Always interpret a straight line in an image as a straight line in 3D.
- If the tips of two lines coincide in an image, then always interpret them as coinciding in 3D.
- Always interpret lines collinear in an image as collinear in 3D.
- Interpret elements nearby in an image as nearby in 3D.
- Always interpret a curve that is smooth in an image as smooth in 3D.
- Where possible, interpret a curve in an image as the contour of a surface in 3D.
- Where possible, interpret a T-junction in an image as a point where the full contour conceals itself.
- Interpret each convex point on a bound as a convex point on a contour.
- Interpret each concave point on a bound as a saddle point on a contour.
- Construct surfaces in 3D that are as smooth as possible.

Xu et al. [34] also point out some other sketch inferring rules, like 1) less distortion, 2) small projection, and 3) planarity.

Curve Net system [21,22,35] is a special form of 3D model expression and is commonly used in design regions. Due to its lacking of support of other information, the only rule it can rely on is perception. Bae et al. [22] infer being perpendicular by cross-shape and Lee et al. [21] infer being parallel in 3D by that in 2D. All these offer a perfect demonstration on how perceptions play an important role.

Symmetry is another important property of many objects, as symmetry in 2D may properly imply the important symmetric structure information in 3D. In the majority of known work, symmetry is considered to be hold automatically for 3D model inferring, especially in freeform modeling. Like in the literature, Teddy [13] and Fibermesh [13,26] make assumptions that the model you create is symmetric in the sketch plane, and as well ILoveSketch [21] interprets a symmetric 3D curve pair by assuming that the input sketch is symmetric. Further more, not only symmetry, Zou et al. [36] use topology constraints to constrain the interpretation of models.

3.2 Domain specified rules

It is a common rule that the smaller your problem is, the much more information you can utilize. When your target model is a special kind of staff, you can utilize the properties of this special kind of objects, such as the geometry primitives of architecture and a smooth surface of an organic model whose normal vector changes in a gentle way. We mention the four frequently occurred domains in sketch modeling tasks: 1) man-made objects, 2) architecture, 3) tree, and 4) human body. Afterwards, we also mention some interesting approaches and categorize them in subsection 5) other domains.

3.2.1 Man-made objects

The man-made objects share common properties, such as self-symmetry, along with the geometry primitives such as cubes and spheres. Using these properties can substantially eliminate the ambiguity of sketch.

Applying the properties that man-made objects are consisted of geometry primitives, Shtof et al. [37] transform the sketch modeling task into inferring the parameters of certain primitives and combining them to fit the input sketch. With further assumptions, one can improve the quality of results. Chen et al. [38] assume that target objects are cylinder-like objects, so the modeling procedures can be much simpler. We first snapped out the bottom surface of the cylinder which is a disk, and then drag the mouse along the height direction of the cylinder-like objects to achieve a sketch fitness of 3D model.

Even 3D curve net systems can get benefits from this kind of knowledge. When modeling a man-made object, Schmidt et al. [35] use the assumptions that the surface of man-made objects constrained by surface that is parallel to the coordinates, helps eliminating the ambiguity of 3D curves.

3.2.2 Architecture

Architecture is often consisted of geometry primitives, and the position between these geometry primitives are perpendicular. Architecture may be self-symmetry. Due to its regularity, one can model a building in few strokes. Lee et al. [21] use the properties that buildings are cubic-like models, inferring that linked surface is perpendicular to each other. Chen et al. [39] use different geometry primitives to fit sketch silhouette. Just like man-made staffs, with further information such as symmetry, Jiang et al. [40] infer the sketch more properly.

3.2.3 Tree

Tree modeling is also a classic modeling task. Chen et al. [41] can infer a real 3D tree model, which only need the sketch to represent the direction of branches. It utilizes the huge tree modeling knowledge to reduce the need of complicated sketch input. Tan et al. [42] reconstruct the occluded branches with the assumption that the tree branch structure is locally self-similar.

3.2.4 Human body

Human body modeling benefits from the structure of human body which is self-symmetric and with restrict pose. Levi et al. [29] exploit skeleton as the guidelines and inflate it to fit sketches and to construct 3D models. Gingold et al. [32] make use of the properties that limbs are of ellipse-like geometry, and thus reduce the freedom of sketch inferring. Davis et al. [43] use the joint-angle constraints which mean human body bones can be in limited angle to remove impossible pose. Kazmi et al. [44] use the human body constraints to conduct a well drawn contours, find good correspondence between strokes, and perform plausible human body deformation.

3.2.5 Other domains

There are few works hard to classify but worth to mention. Flower modeling system [45] takes advantage of the properties that petals are in truncated conic shapes. Pan et al. [46] use fluid model to restrict sketch guided animation. For other methods utilizing domain knowledge, we will skip the details here.

3.3 Knowledge from databases

Adopting the special domain knowledge, one can infer objects without ambiguity, but usually there is no special domain knowledge about the target models. In this case, we may have a database of target models. With those rapidly developed data-driven methods, one can learn knowledge from databases and apply it to modeling procedures.

Data-driven methods serve an important role in discovering geometric, structural, and semantic relationships between shapes. The results of SBM can be greatly improved with introducing the data-driven methods. Nevertheless, detailed background of data-driven methods has gone beyond our paper. The work [47] recently give a thorough survey in this area. The key step is to decide what kind of knowledge to learn, how to extract knowledge from databases and then use it to direct sketch-based modeling. We classify the data-driven methods into two categories: 1) geometry knowledge and 2) structure knowledge. We will give a more exhaustive discussion for knowledge learning and application in Section 4.

3.3.1 Geometry knowledge

Geometry information describes the shape and style of objects, but it is hard to directly learn knowledge from 3D shapes and apply it into sketches. Sketch-based approaches follow another way. They first find the silhouette of the 3D objects and use them as the representation of the 3D candidates. After transforming 3D databases into 2D databases, many data-driven methods for 2D can be introduced to accomplish according tasks, such as retrieval and classification. Finally, based on the corresponding relationship between 3D and 2D, the learning results can be recovered back to 3D.

The most simple task for learning geometry knowledge is to find out and output the best match for objects from the database. This occurs mostly in scene modeling. Refs. [18,48] directly pick the models from the database to be the modeling results. Apart from learning from the database, it is more like a retrieval system.

Database is always too small to find a proper candidate in database. Deformation is a natural way to enlarge the database to create a perfect match. Xu et al. [49] use the retrieval results as input and deform it under a silhouette constraint.

Knowledge learned can be applicable not to 3D modeling merely. Fan et al. [8] further consider how to use the knowledge from database to assist users to sketch. This work supports the user with shadow guidance dynamically which is different from previous work. This system shows the results of interpretation of users' strokes by generating shadow image of the queries and uses the shadow to guide the type and position of the subsequent strokes drawing.

3.3.2 Structure knowledge

Though, none but geometry information is far from enough. Some objects which share similar shapes may be of different functions. When it comes to the highly abstract property of sketches, two different objects may share similar sketches much more probably. Thus, to take the structure knowledge into consideration, it may reduce the repetitiveness and irrationality during the modeling process to a great extent.

Structure knowledge can provide a good guideline on how the deformation and composition take place. Su et al. [50] exploit bounding box as proxy and explore its deformation space to guide the deformation of a model in the database. Xie et al. [51] apply the context information under the database. Taking chair modeling as an example, chair arm should be connected to the chair back, and chair leg should be connected to the chair pad, not the arm. Xu et al. [18] adopt the ideas to the sketch-based scene modeling. It explores the relationship between objects, that lamp should be on the desk and sofa should be right beside the tea table.

4 Approach

The most important stage of SBM is to interpret 3D models. With the suitable assistance of knowledge, man can eliminate the ambiguity of sketches and construct reasonable models. Nowadays, there are many classical and promising approaches. Some of them are general to all the input and knowledge, while some are specified to certain ones.

We put these approaches into categories, includes using 2D sketches to inflate a promising 3D model, using sketches as a gesture to guide surface editing, and using sketches which directly retrieve models from database, or deform models from database to fit the input sketches, or even compose parts of models from database.

This is a subjective categorization, so there may be examples which can not be classified into any suitable category. As Teddy [13] uses inflating and editing approaches, the two categorized methods, to construct a 3D model, we will present a table to show the main methods used in each of those famous paper.

4.1 Inflating

In the early approaches of SBM, often without database assistance and domain specified knowledge supported, one can only use hand-craft methods to construct models. The same situation happens for tasks of freeform surface construction, which also have less knowledge people can refer to.

The hand-craft methods may be task-specified, but they do share some common within. Firstly, their inputs are always 2D sketches, mainly close sketches. Secondly, these methods are suitable for organic models or prototype models, but not suitable for man-made models or precise models. Thirdly, they often get fine-tuning models with sketch editing tools.

Teddy [13] is the first approach to build inflated surface to create simple organic toys. They take single 2D closed sketch as input, and find the spine of sketch by using the chordal axes. The system then wraps the spine with the polygonal mesh and use constant Delaunay triangulation of the polygon. Some refined steps are performed as well.

When faced with unclosed sketches, such as T-junction and cusp, teddy systems do not work. Karpenko et al. [17] improve the method from William [52] for inferring shapes with hidden contours, which can handle T-junction and cusp perfectly. The interpretation goes through three steps: 1) inferring the hidden part of contours, 2) building certain maps between complete contours and abstract topological surface and embedding them to R^2, 3) lifting the map to smooth surface in R^3.

Nealen et al. [53] use the famous shape descriptor [54] to guide the inflating. Figure 8 shows the sketch based operations used in the editing process including cut, extrusion and tunnel. After creating an initial model by inflating the closed area, the user can use different operations to edit the model. Drawing a stroke crossing the model is the way to cut the model. With a silhouette stroke following a closed stroke on the object's surface, the user can show extrusion modeling intention. Meanwhile, tunnel operation can be done by drawing closed strokes on both sides of the model.

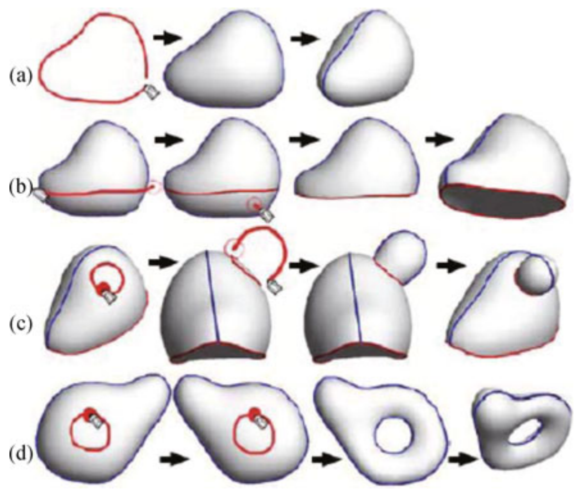


Fig. 8 Sketch-based editing techniques used in [53]. These operations include (a) creation, (b) cut, (c) extrusion, and (d) tunnel

4.2 Optimization

We have exploited the different types of knowledge in previous sections including perception, knowledge, domain special knowledge, and database-based knowledge. We may use these knowledge in the modeling step, and one proper way to use these knowledge is to put certain constraints to represent the implied knowledge. It is commonly used in perception and special domain knowledge modeling systems.

Taking architecture modeling systems as an example. Lee et al. [21] construct 3D architectural models quickly from drawing through an interactive sketching system. The key step is using perceptual constraints to fit user's outlined surface polygons. More specifically speaking, the constructing procedure can be divided into two parts, saying connecting strokes to form polygons, and determining the position relationship. When constructing polygons, heuristic methodology is applied. While determining the positions, it considers the characteristics of architectures. The lines on the surface commonly bear some properties, like axis being align, parallel or even the whole surface is symmetric. The system encodes these information to energy function and the energy would be quite low if those relationships are satisfied. Subsequently, by optimization tools, situation with minimal energy can be solved, which is what we need.

Xu et al. [34] note that the designers maximally reveal 3D information from a proper viewpoint, which achieves minimal variation and foreshortening while keeping the projection accuracy. Besides those sketch fidelity mentioned above, shape regularity shares the same importance to sketch curve understanding. Similarly to the architecture paper, many objects, especially the man-made objects, possess regular shapes, and thus regular sketch strokes accordingly. The regularity lies in being perpendicular, parallel or symmetric, etc. Similar to the sketch fidelity, shape regularity is also represented by some mathematical energy. As a result, the shapes and positions of these strokes can be determined by energy optimization. This optimization methodology has been widely applicable to curve net modeling systems [20,22,55].

4.3 Sketch-based editing

Mesh deformation tools make complicated geometry transformation possible through controlling 3D handle vertices. Large range of applications and problems can be effectively solved by these tools, such as generating a continuous sequence of animations, creating a novel model from an existing one. However, using 3D handle vertices to control the deformation is a less intuitive way to transmit users' intentions. Considering the intuition of sketch-based interface and the power of mesh deformation tools, several approaches have been combined with those two.

Assuming that the target model is smooth, inflating methods [13,17,53] can overcome the ambiguous properties of sketches, but fail to generate complicated models. Therefore, sketch-based editing tool is an important component of these approaches. Teddy [13] directly draws lines on the prototype to operate on the models such as cutting, growing and blending. Fibermesh [53] uses drawn sketches as a proxy to deform the model while preserving the Laplacian of the surface. Work without a sketch-based editing component as SmoothSketch [17] is not a complete SBM system.

These works [16,23,26] pay great attention to sketch-based deformation, where individual drawn strokes can be treated as a deformation clue. The strokes indicate the source vertices on the mesh and the corresponding target vertices' positions. Plenty of mesh deformation techniques can be used then to meet the corresponding deformation. Kho and Garland [16] use drawn reference curve to define a region of interest which will be deformed. By building a correspondence between drawn reference curve and target position for drawn reference curve or by manipulating a deformation parameter, users can achieve the desired deformation. As shown in Fig. 9, Nealen et al. [26] also use corresponding reference strokes to deform models while preserving the global and local geometry. This idea is realized by the discrete Laplace and Poisson models [56]. Zimmermann et al. [23,24] improve the ability of automatically inferring the corresponding reference curve to the drawn stroke which is near the original model. The region of interest is also automatically determined. Furher more, Paoli et al. [57] build layered, shape interdependent 3D volumes near the original model by directly sketching near the original model instead of on the original model.

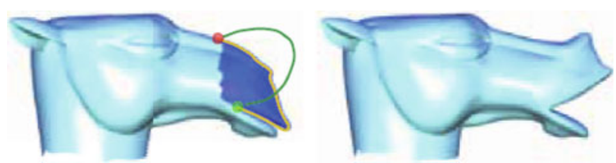


Fig. 9 Sketching an approximate CAMEL lip by deformation method [27]

Kraevoy et al. [25] deform a 3D template model with multi-stroke contour drawings while previous work takes sequential strokes as input. The multi-stroke contours provide general shape information such as scale and pose while the template model contains detailed information not provided by the strokes. The correspondence between 2D contour drawings and 3D template vertices is built by using a hidden Markov model. An iterative correspond-and-deform framework is also proposed, which is crucial in making this SBM work.

4.4 Retrieval-based approach

With the help of database, one can infer 3D models from sketch by knowledge learning from the database. Among all the learning methods, the most naive and natural one is to directly find a good match from the database as modeling results. Taking advantages from well developed areas of example-based retrieval techniques and image retrieval techniques, sketch-based retrieval uses single or multi-view sketches as input, and query from the database to find the perfect match. As it is abstract, using just partial information about the projection of the shape leverages the difficulty of the task. Great varieties between different people within a same category even further improve the hardness.

Sketch-based shape retrieval system has been proposed in many approaches, and it is the base of data-driven SBM systems. Eitz et al. [58] have summarized the system into the following pipeline: generate sketches for each model in database from multi-viewpoint; use appropriate image descriptor to code the generated sketches and input sketches; use certain image retrieval techniques to find the perfect match for the input sketches.

Sketch-based shape retrieval system takes different techniques in each step in the pipeline. Some of the steps have a great influence on the retrieval results.

The first step is to generate 2D sketches for each 3D model in the database from different directions. Normally, we have no priori of which direction is the user's choice to sketch an object, so we should render the 3D model in database in all potential directions. In order to take every view direction equally, we put the model in the origin, sample uniformly on the unit sphere, and take sketch from each point following the direction to the origin. Larger amounts of sample directions do not lead to better results. First of all, the retrieval time increases correspondingly, and sketch from impossible directions such as seeing a cow from bottom may create ambiguous sketch. Eitz et al. [58] try to solve this problem by finding perceptually best views for objects. Su et al. [50] use the coordination between models from the same category. It first needs the models in category to be well aligned, and also the generation of 2D sketch are from uniformly sampled views, and then uses the weighted average directions as the final view direction.

The hardest part of sketch-based shape retrieval system is how to represent a sketch using a proper local descriptor. As sketch only contains several sparse strokes which is sparse and has no color, one should adjust the common image local descriptor to represent sketch patches. Gradient based descriptor such as hog is preferred giving rise to that the sketch is lacking of color and only contains lines. However, the patch size should be enlarged due to the sparsity of sketch. Current work uses bag of feature models to organize feature vectors. With defining a proper norm, one can retrieval the sketch. Figure 10 shows a typical sketch-based retrieval example. The numbers and colors represent the rank of the querying results. One should note that, due to the big variation of sketch, the feature vectors extracted from the local descriptor may have a significant difference within the same category, which brings the retrieval system a huge problem.

Directly using retrieval techniques is not enough to fit the users' modeling intentions. However, when we have less requirement for the precision of a model, retrieval-based technique is a good way for modeling, like the scene modeling system.

Shin et al. [48] use the sketch to determine what objects it represents and the projection information it involves as well. Xu et al. [18] adopt the same idea and help improve the results with the additional context information.

4.5 Deformation

As we mentioned before, directly applying retrieval results as the final interpretation model may fail to catch the model intent of input sketch. Because it is not that case, database has the exact model fitting the input sketch. Therefore, based on the supporting 3D model database, one can create a new

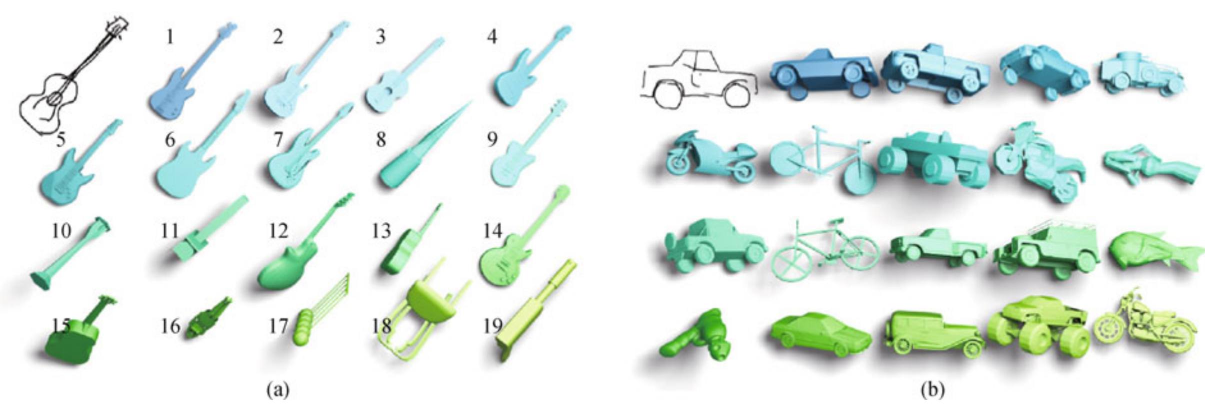


Fig. 10 Sketch-based retrieval examples: query results of (a) guitar, (b) car [58] (The numbers represent the rank of the querying results; 1 represents the highest ranked object)

model which matches the sketch perfectly but not retrieves a perfect match. Deformation is a proper way to create new models not only fitting the sketch with its silhouette but also a certain kind of staff. Deformation is a basic and classical problem in graphics field. This topic is well studied and has achieved many amazing work. Here we only focus on the deformation techniques which are involved in SBM system.

Based on the target deformation shapes, we classify the deformation work into two categories: primitives deformation and data-driven deformation.

4.5.1 Primitives deformation

The geometry primitives are usually used to build complicated models. Man-made objects are often composed by kinds of geometry primitives with little deformation. Even organic objects like human bodies can be modeled using cylinders in prototype stage. The reason why it has been widely used lies in its capability of expression and convenience of deformation. Geometry primitives can be defined by only a few parameters. For example, a cube is defined by three parameters and a sphere is defined by only one parameter, i.e., its radius. But combined with boolean operators, it can be applied to really complicated models. When coming to the SBM field, its low dimension of deformation space can lead to the ambiguity of sketch interpretation. With few strokes, certain geometry primitives can be uniquely defined which are hard for the construction from scratch.

Shtof et al. [37] utilize users' assistance to recognize and segment by choosing and snapping specific geometric primitives on the relevant strokes. Then the system automatically snaps the primitive to fitting the primitive's projection to the relevant strokes which can be easily defined by few parameters. As shown in Fig. 11, a global optimization is adopted which infers geosemantic and semantic constraints. In this way, the work combines the intelligence of human recognition and computing ability of machine.

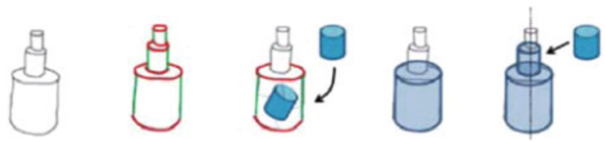


Fig. 11 Primitives deformation techniques used in [37]. (a) Input sketch; (b) semantic classification to determine the geometry primitives; (c) interactive matching; (d) and (e) snapping to build the model

Chen et al. [38] also notice that extracting primitives from sketch is a cognitive task which is easy for humans but particularly difficult for machines. As in [37], the work utilizes human to recognize and segment the input sketch and uses machine to fit the geometry constraints. However, it figures out this task in a more interactive way, called sweeping, which focuses on special primitives such as generalized cylinders, cuboid or similar primitives. The first two sweeps define the first and second dimension of a 2D profile, while the third, which is longer, is used to define the main curved axis of the primitive.

Chen et al. [39] apply this idea to architecture modeling. After a camera viewing, parameters are implicitly defined by three user drawn lines. The system automatically fits primitive geometry to the input sketch. Detailed geometries such as windows and columns are further added to form a plausible building.

Gingold et al. [32] adopt this idea to organic objects modeling task. It uses generalized cylinders and ellipsoids as geometric primitives to fit the sketch contours. With the help of annotations such as same-lengths, same-angles, alignment and mirror symmetry, the system can construct a consistent model with ambiguous input.

4.5.2 Data-driven deformation

Although geometry primitives have strong abilities of expression, it is hard to express a complicated model, which may take a long step. Problems get even harder when we make freeform surface which can not be controlled by only few parameters.

But how to define the deformation of arbitrary kind of objects? With the assistance of database, one can learn the deformation constraints or deformation space containing the structure information of certain kinds of object data.

Xu et al. [49] utilize pre-analyzed database to possess useful high-level structural information, which can eliminate the ambiguity of sketch input. The correspondence is established between user-guided image-space object segmentation and pre-segmented candidate model parts. Then the system automatically deforms the candidate to input sketch while preserving the model structure.

Xie et al. [51] apply contextual cues which are extracted from pre-learned and part-based database to construct a novel model through combination. The contextual information provides not only structural cues as adjacency but also detailed information as texture and geometry style.

Recently, Su et al. [50] construct a network of shapes which implicitly defines a shape-specific deformation subspace. This work is subsequently used as the deformation constraints to SBM system.

4.6 Composition

Composition is another common way to construct models learning from database. Funkhouser et al. [59] firstly propose this method which selects each semantic part from different models and then assembles and seams them. With the well developed segmentation work of models, and many well segmented database, one can easily achieve segmentations for models with semantic labels.

Assemble approaches take sketches as the clue to find the perfect match for each stroke or user defined strokes. The match will consider the similarity not only in the part level but also in global structure level. Shin et al. [48] use the context information to decide whether a retrieval part is suitable for those already existing parts. As shown in Fig. 12, Xu et al. [18] take context information of scene as the guides for scene synthesis.

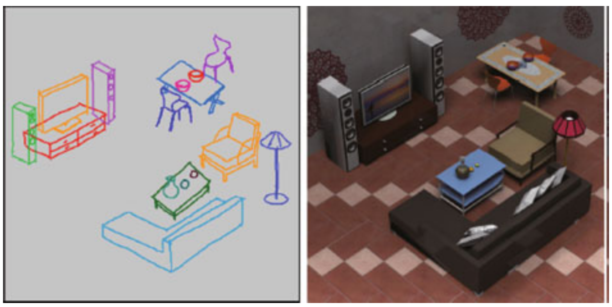


Fig. 12 Input and output of [18]: sketch-based scene production via composition

Composition often takes place with accompany of part leveled deformation. With the same reason that there is no perfect match for sketches, even part leveled models, part leveled deformation is a good improvement for assemble approaches.

5 Output

When inputs vary, we apply different methods based on the knowledge accordingly. This may leads to different types of results and we categorize them into three classes.

The main cause of various forms of results lies in the ultimate demand of application. If it needs to generate a scene, then the output would be a scene. For the case of single object, 3D mesh can be employed and the results can be directly applied to other practical applications. On the other hand, 3D curve net could keep the information of sketches, and make the further transformation much easier.

Meanwhile, the forms of results may also depend on those specific methods, as well as the inputs. For example, 3D meshes are common result for data-driven methods, and nearly all 3D curve nets are generated by 3D curves.

5.1 3D mesh

3D mesh is the most traditional way to present 3D objects in computer graphics. Thus it has become one of those outputs for sketch-based modeling. As 3D mesh has an integral mesh structure, it benefits the proceeding processing and task. For example, in the sketch-based modeling system, we can take sketch-based editing to transform those existing 3D meshes. However, it will cause the lose of former sketch information which can be used in the subsequent tasks.

5.2 3D curve net

As we discussed above, generating integral 3D models will lose the original sketch information. In order to overcome such drawbacks, some do not use 3D mesh as the ultimate form of results, so that sketch can be adopted for representation and construction. Instead, they take 3D curve net, which is composed with some 3D curves without an integral mesh information. As those 3D curves are just simply lines, the transformation becomes more efficient comparatively. At the same time, 3D curve net can be turned into 3D model easily.

5.3 3D scene

3D scene consists of many single objects and is also one of the common results for sketch-based modeling. In such applications, we usually focus on the shape and relative positions of those objects in the scene, so the objects are generally in the form of 3D meshes.

6 Discussion

6.1 Discussion of current work

In the previous sections, we have concluded the current work into a 4-step pipeline and analyzed each step in detail. Table 1 shows the representative work with respect to each category.

In the input section, we categorize different sketch inputs according to their input formats, namely 2D sketch, 3D sketch and immersion sketch. 2D and 3D sketches are the mainstream input formats which have their own main ground. Immersion sketch is a new form of input which can provide more natural and precise modeling intentions. The sketch input assistant method, which can provide extra information for sketch-based modeling, is also mentioned.

Table 1 Taxonomy of sketch-based modeling systems

Applications	Input					Knowledge			Approach						Output		
	2D	3D	I	MV	IA	P	DSR	Db	I	O	SBE	RBA	D	C	SM	S	CN
Teddy [13]	★	★				★			★		★				★		
SmoothSketch [17]	★	★				★			★						★		
FiberMesh [53]	★	★				★			★		★				★		
Sketch-based Recontruction [21]	★	★				★	★			★					★		
ILoveSketch [20]		★				★	★			★	★						★
EverybodyLovesSketch [22]		★				★	★			★	★						★
Sketching mesh deformation [16]		★				★					★		★		★		
SilSketch [23]		★				★				★	★				★		
Modeling from contour [25]		★				★	★			★	★				★		
MagicCanvas [48]	★					★						★				★	
Sketch2Scene [18]	★					★						★				★	
Sketching reality [39]	★						★						★		★		
GeosemanticSnapping [37]	★						★						★		★		
3-sweep [38]	★						★						★		★		
Structured annotations [32]	★						★						★		★		
Photo-inspired modeling [49]	★							★					★		★		
Sketch-to-Design [51]	★							★					★	★	★		
Estimating image depth [50]	★							★					★		★		
Sketching contours [24]		★				★					★		★		★		
Detail-preserving editing [26]		★				★					★		★		★		
Modeling with silhouettes [28]	★			★			★			★					★		
ArtiSketch [29]	★			★			★			★			★		★		
Modeling with shadow guide [8]	★				★			★		★			★	★	★		
Analytic drawing of scaffolds [35]		★					★			★	★						★
True2form [34]	★						★			★	★						★

Note: 2D — 2D sketch, 3D — 3D sketch, I — Immersion sketch, MV — Multi-view techniques, IA — Sketch input assistance, P — Perceptions, DSR — Domain specified rules, Db — Databases, I — Inflating, O — Optimization, SBE — Sketch-based editing, RBA — Retrieval-based approach, D — Deformation, C — Composition, SM — Single model, S — Scene, CN — Curve net

When it comes to the knowledge section, we summarize knowledge into three categories, that is, perception, domain specified rules, and database knowledge. Perception is not only the base of human interpretation but also the foundation of modeling rules which are used in almost all the work. When modeling a special kind of models, one should need more information than merely perception knowledge. One way to handle that is to use domain specified rules which is commonly used in architecture modeling, tree modeling, etc. These rules are special constraints that models should obey such as surface positions and relationships. Another way to deal with the need for more information is to learn knowledge from the database which is known as data-driven method. It is a developed domain and has achieved great success in many aspects. When utilizing these techniques in SBM systems, we mainly focus on global and partial retrieval techniques in this area.

For the third section, approach, we first introduce the inflating methods which have been usually used in early stage of SBM systems. These methods use user-defined rules to inflate a credible model. The inflating methods are leaking of surface details, so they often employ sketch-based editing steps. It takes sketch input as controller and guideline to deform a existing model. This step is gradually developed into a mature method and serval works have fully studied this method. Specified domain knowledge shares the same idea of inflating method about the rules, but eliminates the divergence of which is the rule. Using the knowledge from other areas, one can transform the knowledge into different constraints to restrict the modeling processing. With the certain constrains, we can use optimization method to solve this problem. When modeling complex ones, it is hard to define those rules and constraints that one should obey. Data-driven method deals with this crux properly. Current work is based on sketch retrieval method, which is the foundation in deformation method and composition method.

6.2 Discussion of fundamental study of sketch

Obviously, the study of sketch itself can fairly improve the pace of SBM. The key to these studies is to pay great attention to sketch comprehension. However, the sketch is too abstract, and suffers from big variation of drawing. As a result, sketch comprehension is a tough task for research.

Current researchers mainly focus on the basic comprehension task—lassification. These work follows the common image-based classification pipelines, dividing sketch image into patches, transforming each patch to a certain feature space, using bag-of-feature model to represent each sketch image, and finally using certain classification model to classify the sketches. More detailed investigation can be found in [58] which demonstrates this pipeline and also in [60] which compares different feature extractions.

Yet how people comprehend sketches? Previous perception knowledge takes a glance of it. Eitz et al. [61] point out that even humans can not recognize sketch by 100 percentage and only 71 percentage of total accuracy can be achieved. Categories with similar meanings and sketches with similar appearances are the most ambiguous parts. Schneider et al. [62] try to answer this question from a new perspective. They analyze the function of each stroke in a single sketch by weighting the affection of its classification results.

7 Further work

In this section, we put forward the challenges current work may be faced with.

7.1 Finding useful new domain

Utilizing domain specified rules is the most direct way to promote the existing techniques to new domains. According to Section 3, we summarize the current domain specified rules which only impact few domains, such as tree modeling and architecture modeling, etc. Actually, there are many other domains which are fully analyzed. Borrowing the knowledge they contain, we can combine the convenience of sketch-based modeling techniques and powerful knowledge of specified domains.

7.2 Feature learning

The new trend of data-driven methods utilize the knowledge in database and guide the modeling processing. These techniques are borrowed from other topics which are not suitable for sketch-based modeling task due to the rareness of information the sketch contains. Therefore, modifying the existing techniques to adapt the sketch input is an important task for SBM system. Meanwhile, hand-craft feature can not achieve good performance against all the situation.

Recent work in machine learning shows great ability in learning feature directly from raw input from text and image, which is called deep learning. Deep learning architectures consist of multiple layers and can convert input data into multiple levels of abstraction. Utilizing deep learning in sketch learning may be a promising trend.

7.3 Using more 3D information

Current work utilizes 3D object information merely with their silhouette or feature lines. These methods transform 3D information into 2D while abandoning lots of useful 3D information.

The modeling performance will be enhanced with the help of more 3D information. There are existing methods that utilize lower-level 3D information. For example, Xie et al. [51] apply the context information under the database. With the help of certain context information, this approach can easily eliminate the ambiguity of similarity of chair leg and chair arm in chair modeling processing. With the help of other 3D information, even high-level 3D information, one may achieve better results. Take functionality as an example. Functionality describes the relationship between human activities and objects, which may be represented by certain geometry. The ambiguity of SBM can be eliminated efficiently by functionality due to the connection witch corresponding geometry feature. It is also natural to utilize functionality in SBM because functionality can be presented easily by few strokes.

Generally, modeling with more 3D information is a really promising direction in the future.

8 Conclusion

With the growing demand for 3D models and convenience of drawing, the SBM methods have been well studied. However, it has to be up against many severe challenges at the same time, including sketch comprehension and generation of realistic models from sketches. Further more, due to the abstract property the sketch inherits and the variety it contains, it is really hard to infer the strokes in sketch.

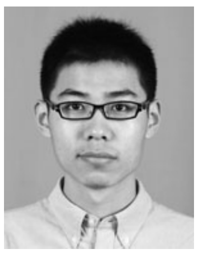
In this paper, we investigate state-of-the-art techniques in SBM. We classify them in several aspects, including input, knowledge and approach, and discuss their technical superiority as well as their significant disadvantages. Many difficulties of those representative current work are pointed out, which offers us instrumental suggestions and directions for the future research in this area.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 61222206, 11426236) and the One Hundred Talent Project of the Chinese Academy of Sciences.

References

1. Botsch M, Pauly M, Kobbelt L, Alliez P, Lévy B, Bischoff S, Röossl C. Geometric modeling based on polygonal meshes. In: Proceedings of the ACM SIGGRAPH Course Notes. 2007
2. Perry R N, Frisken S F. Kizamu: a system for sculpting digital characters. In: Proceedings of the 28th ACM Annual Conference on Computer Graphics and Interactive Techniques. 2001, 47–56
3. Zoran A, Shilkrot R, Nanyakkara S, Paradiso J. The hybrid artisans: a case study in smart tools. ACM Transactions on Computer-Human Interaction, 2014, 21(3): 15
4. Shotton J, Sharp T, Kipman A, Fitzgibbon A, Finocchio M, Blake A, Cook M, Moore R. Real-time human pose recognition in parts from single depth images. Communications of the ACM, 2013, 56(1): 116–124
5. Henry P, Krainin M, Herbst E, Ren X F, Fox D. RGB-D mapping: using Kinect-style depth cameras for dense 3D modeling of indoor environments. The International Journal of Robotics Research, 2012, 31(5): 647–663
6. Smisek J, Jancosek M, Pajdla T. 3D with kinect. In: Fossati A, Gall J, Helmut Grabner H, et al. eds. Consumer Depth Cameras for Computer Vision: Research Topics and Applications. London: Springer-Verlag, 2013, 3–25
7. Quan L. Image-based modeling. Springer Science & Business Media, 2010
8. Fan L B, Wang R M, Xu L L, Deng J S, Liu L G. Modeling by drawing with shadow guidance. Computer Graphics Forum, 2013, 23(7): 157–166
9. Yuan X R, Xu H, Nguyen M, Shesh A, Chen B Q. Sketch-based segmentation of scanned outdoor environment models. In: Proceedings of EG Workshop on Sketch-Based Interfaces and Modeling. 2005, 19–26
10. Wu H Y, Pan C H, Pan J, Yang Q, Ma S D. A sketch-based interactive framework for real-time mesh segmentation. In: Proceedings of Computer Graphics International. 2007
11. Schmidt R, Wyvill B, Sousa M C, Jorge J A. Shapeshop: sketch-based solid modeling with blobtrees. In: Proceedings of ACM SIGGRAPH 2007 courses. 2007
12. Turquin E, Wither J, Boissieux L, Cani M P, Hughes J F. A sketch-based interface for clothing virtual characters. IEEE Computer Graphics and Applications, 2007, 27(1): 72–81
13. Igarashi T, Matsuoka S, Tanaka H. Teddy: a sketching interface for 3D freeform design. In: Proceedings of the 26th ACM Annual Conference on Computer Graphics and Interactive Techniques. 1999, 409–416
14. Olsen L, Samavati F F, Sousa M C, Jorge J A. Sketch-based modeling: a survey. Computers & Graphics, 2009, 33(1): 85–103
15. Cook M T, Agah A. A survey of sketch-based 3-D modeling techniques. Interacting with Computers, 2009, 21(3): 201–211
16. Kho Y, Garland M. Sketching mesh deformations. In: Proceedings of ACM SIGGRAPH 2007 Courses. 2007
17. Karpenko O A, Hughes J F. Smoothsketch: 3D free-form shapes from complex sketches. ACM Transactions on Graphics, 2006, 25(3): 589–598
18. Xu K, Chen K, Fu H B, Sun W K L, Hu S M. Sketch2Scene: sketch-based co-retrieval and co-placement of 3D models. ACM Transactions on Graphics, 2013, 32(4): 123
19. Iarussi E, Bommes D, Bousseau A. Bendfields: regularized curvature fields from rough concept sketches. ACM Transactions on Graphics, 2015, 34(3): 24
20. 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, 151–160
21. Lee S, Feng D, Grimm C, Gooch B. A sketch-based user interface for reconstructing architectural drawings. Computer Graphics Forum, 2008, 27(1): 81–90
22. Bae S H, Balakrishnan R, Singh K. EverybodyLovesSketch: 3D sketching for a broader audience. In: Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology. 2009, 59–68
23. Zimmermann J, Nealen A, Alexa M. SilSketch: automated sketch-based editing of surface meshes. In: Proceedings of the 4th Eurographics Workshop on Sketch-based Interfaces and Modeling. 2007, 23–30
24. Zimmermann J, Nealen A, Alexa M. Sketchbased interfaces: sketching contours. Computers Graphics, 2008, 32(5): 486–499
25. Kraevoy V, Sheffer A, van de Panne M. Modeling from contour drawings. In: Proceedings of the 6th ACM Eurographics Symposium on Sketch-Based Interfaces and Modeling. 2009, 37–44
26. Nealen A, Sorkine O, Alexa M, Cohen-Or D. A sketch-based interface for detail-preserving mesh editing. ACM Transactions on Graphics, 2005, 24(3): 1142–1147
27. Nam S H, Chai Y H. SPACESKETCH: shape modeling with 3D meshes and control curves in stereoscopic environments. Computers Graphics, 2012, 36(5): 526–533
28. Rivers A, Durand F, Igarashi T. 3D modeling with silhouettes. ACM Transactions on Graphics, 2010, 29(4): 109
29. Levi Z, Gotsman C. ArtiSketch: a system for articulated sketch modeling. Computer Graphics Forum, 2013, 32: 235–244
30. Lee Y J, Zitnick C L, Cohen M F. Shadowdraw: real-time user guidance for freehand drawing. ACM Transactions on Graphics, 2011, 30(4): 27
31. Andre A, Saito S. Single-view sketch based modeling. In: Proceedings of the 8th ACM Eurographics Symposium on Sketch-Based Interfaces and Modeling. 2011, 133–140
32. Gingold Y, Igarashi T, Zorin D. Structured annotations for 2D-to-3D modeling. ACM Transactions on Graphics, 2009, 28(5): 148
33. Hoffman D D. Visual intelligence: how we create what we see. New York: W.W. Norton & Company, 2000
34. Xu B X, Chang W, Sheffer A, Bousseau A, McCrae J, Singh K. True2form: 3D curve networks from 2D sketches via selective regularization. ACM Transactions on Graphics, 2014, 33(4): 131
35. Schmidt R, Khan A, Singh K, Kurtenbach G. Analytic drawing of 3D scaffolds. ACM Transactions on Graphics, 2009, 28(5): 149

36. Zou M, Holloway M, Carr N, Ju T. Topology-constrained surface reconstruction from cross-sections. ACM Transactions on Graphics, 2015, 34(4): 128
37. Shtof A, Agathos A, Gingold Y, Shamir A, Cohen-or D. Geosemantic snapping for sketch-based modeling. Computer Graphics Forum, 2013, 32: 245–253
38. Chen T, Zhu Z, Shamir A,, Hu S M, Cohen-Or D. 3-sweep: extracting editable objects from a single photo. ACM Transactions on Graphics, 2013, 32(6): 195
39. Chen X J, Kang S B, Xu Y Q, Dorsey J, Shum H Y. Sketching reality: realistic interpretation of architectural designs. ACM Transactions on Graphics, 2008, 27(2): 11
40. Jiang N J, Tan P, Cheong L F. Symmetric architecture modeling with a single image. ACM Transactions on Graphics, 2009, 28(5): 113
41. Chen C J, Neubert B, Xu Y Q, Deussen O, Kang S B. Sketch-based tree modeling using Markov random field. ACM Transactions on Graphics, 2008, 27(5): 109
42. Tan P, Fang T, Xiao J X, Zhao P, Quan L. Single image tree modeling. ACM Transactions on Graphics, 2008, 27(5): 108
43. 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, 320–328
44. Kazmi I K, You L H, Yang X S, Jin X G, Zhang J J. Efficient sketch-based creation of detailed character models through data-driven mesh deformations. Computer Animation and Virtual Worlds, 2015, 26(3–4): 469–481
45. Yan F L, Gong M L, Cohen-Or D, Deussen O, Chen N Q. Flower reconstruction from a single photo. Computer Graphics Forum. 2014, 33(2): 439–447
46. Pan Z R, Huang J, Tong Y Y, Zheng C X, Bao H J. Interactive localized liquid motion editing. ACM Transactions on Graphics, 2013, 32(6): 184
47. Xu K, Kim V G, Huang Q X, Kalogerakis E. Data-driven shape analysis and processing. 2015, arXiv preprint arXiv:1502.06686
48. Shin H J, Igarashi T. Magic canvas: interactive design of a 3-D scene prototype from freehand sketches. In: Proceedings of Graphics Interface 2007. 2007, 63–70
49. Xu K, Zheng H L, Zhang H, Cohen-Or D, Liu L G, Xiong Y S. Photo-inspired model-driven 3D object modeling. ACM Transactions on Graphics, 2011, 30(4): 80
50. Su H, Huang Q X, Mitra N J, Li Y Y, Guibas L. Estimating image depth using shape collections. ACM Transactions on Graphics, 2014, 33(4): 37
51. Xie X H, Xu K, Mitra N J, Cohen-Or D, Gong W Y, Su Q, Chen B Q. Sketch-to-design: context-based part assembly. Computer Graphics Forum, 2013, 32(8): 233–245
52. Williams L R, Jacobs D W. Stochastic completion fields: a neural model of illusory contour shape and salience. Neural computation, 1997, 9(4): 837–858
53. Nealen A, Igarashi T, Sorkine O, Alexa M. Fibermesh: designing freeform surfaces with 3D curves. ACM Transactions on Graphics, 2007, 26(3): 41
54. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rössl C, Seidel H P. Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. 2004, 175–184
55. Pan H, Liu Y, Sheffer A, Vining N, Li C J, Wang W P. Flow aligned surfacing of curve networks. ACM Transactions on Graphics, 2015, 34(4): 127
56. Alexa M. Differential coordinates for local mesh morphing and deformation. The Visual Computer, 2003, 19(2): 105–114
57. De Paoli C, Singh K. SecondSkin: Sketch-based construction of layered 3D models. ACM Transactions on Graphics, 2015, 34(4): 126
58. Eitz M, Richter R, Boubekeur T, Hildebrand K, Alexa M. Sketch-based shape retrieval. ACM Transactions on Graphics, 2012, 31(4): 31
59. Funkhouser T, Kazhdan M, Shilane P, Min P, Kiefer W, Tal A, Rusinkiewicz S, Dobkin D. Modeling by example. ACM Transactions on Graphics, 2004, 23(3): 652–663
60. Eitz M, Hildebrand K, Boubekeur T, Alexa M. Sketch-based image retrieval: benchmark and bag-of-features descriptors. IEEE Transactions on Visualization and Computer Graphics, 2011, 17(11): 1624–1636
61. Eitz M, Hays J, Alexa M. How do humans sketch objects? ACM Transactions on Graphics, 2012, 31(4): 44
62. Schneider R G, Tuytelaars T. Sketch classification and classification-driven analysis using fisher vectors. ACM Transactions on Graphics, 2014, 33(6): 174



Chao Ding is studying in the Department of Computational Mathematics from University of Science and Technology of China (USTC), China. He received a Bachelor degree in 2013, and will also receive a master degree in computational mathematics from USTC in 2016. During 2013 and 2016, he has been doing research in the Graphics and Geometric Computing Lab at USTC.

Ligang Liu is a professor at the School of Mathematical Sciences, University of Science and Technology of China, China. He received his BS (1996) and his PhD (2001) from Zhejiang University, China. Between 2001 and 2004, he worked at Microsoft Research Asia. Then he worked at Zhejiang University during 2004 and 2012. He paid an academic visit to Harvard University during 2009 and 2011. He serves as the associated editors for journals of Computer Aided Geometric Design and IEEE Computer Graphics and Applications. He is the Awardee of the NSFC Excellent Young Scholars Program in 2012. His research interests include digital geometric processing, computer graphics, and image processing.