EUROGRAPHICS Symposium on Sketch-Based Interfaces and Modeling (2010), pp. 1–8
M. Alexa and E. Do (Editors)

Automated Surface Extraction from Curve Networks for Sketch-based 3D Modeling Systems

paper1006

Abstract

This paper addresses conceptual sketch-based three-dimensional shape design as a special case of 3D modeling during early design stages. We will discuss the development and implementation of a sketch-based modeling prototype, which is built upon prior experiences made with earlier systems. Thereby, our focus is on developing an intuitive, consistent and expressive modeling metaphor. This combines techniques for interactive 3D curve creation from sketch-based input, automatic topology extraction and subsequent surface creation. In order to achieve these goals general aspects of conceptual and sketch-based modeling approaches will be discussed. The resulting system will be validated by creating a set of 3D shapes of varying complexity and with respect to different modeling scenarios. Creating those models will also allow for conclusions about the effectiveness and applicability of the proposed shape creation workflow.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Line and Curve Generation

1. Introduction

The creation of digital three-dimensional (3D) content and its representation have become an increasingly important topic over the last decades. This includes the domain of sketch-based modeling (SBM) techniques which orient towards the traditional interaction with pen and paper. Compared to initial works by Sachs et al. [SRS91] and Zeleznik et al. [ZHH96] remarkable advances have been accomplished with respect to a better integration of designers and the consideration of human factors during early design processes. By using alternative input and interaction modalities SBM systems strive for an easy and intuitive conceptual modeling while maintaining valid and consistent results. Although such systems have been enhanced extensively over the years towards this goal, they are still not adapted sufficiently to support wide application. There are, for example, only few 3D modeling tools in the field focusing on the particular characteristics of the conceptual construction phases. In this regard, the main challenge for the design of such systems is to balance a set of fundamentally different requirements for the modeling process. On the one hand they have to provide a precise design, reliable creation and predictable behavior. On the other hand they should allow for fast and intuitive geometry creation and support explorative design processes for which only diverse and incomplete mental concepts are available.

Motivated by these currently unmet requirements, we present a conceptual 3D modeling system which extends existing approaches for SBM modeling and enhances them with advanced modeling features. Thereby, our goal is to identify a minimal set of efficient modeling and interaction metaphors to allow for the expressive, interactive and intuitive creation of conceptual 3D models. The resulting models may serve as base-meshes in subsequent 3D shape creation workflows or can directly be imported into standard Computer-Aided Design (CAD) packages. In addition, an enhanced method for automating the surface construction process using topological curve networks is discussed. The final system concept offers an improved modeling workflow for 3D object creation based on the construction and editing of topological curve networks. The description of the system is substantiated by a set of application scenarios and test models created with the presented system. A final conclusion reveals remaining questions and further challenges working with SBM metaphors.

2. Related Work

In contrast to traditional CAD modeling systems, SBM system focus on the incorporation of traditional pen and paper

interaction paradigms, which involve the acquisition and interpretation of two-dimensional (2D) data represented in a sketch-like manner (e.g. strokes and lines). Thereby, amount, geometrical structure, order and purpose of this input within the final model may vary between modeling stages and artistic styles. Most of the existing systems use perceptional features (e.g. silhouettes, creases and edges on object surfaces) as input constraint to construct the geometrical model. The survey provided by Olsen et al. [OSSJ09] gives a profound overview about existing 3D SBM systems and techniques. In the course of this paper we will focus especially on constructive approaches which are tailored towards conceptual 3D design. These approaches can be classified as follows:

Contour-based techniques focus on a technical drawing of an object defined by its edges. Based on internal rule-based heuristics those initial sketches can be used to reconstruct the underlying shape. Among other contributions such systems have been presented by Sachs et al. [SRS91], Zeleznik et al. [ZHH96], Faria et al. [FCAD03] and Ku et al. [KQW06]. The reconstruction of such objects inherently imposes the so-called *one-to-many* mapping problem [ML07]. Despite its similarity to the original drafting process the results are often limited in their complexity as perspective 2D drawings inherently include ambiguities (e.g. occlusion).

Silhouette-based approaches pioneered by Igarashi et al. [IMT06] directly use planar contour sketches as input for a subsequent inflation-like surface construction. Similar to the first category rather simple 2D silhouette sketches can impose complex reconstruction problems as shown by Karpenko et al. [KH06]. A hybrid approach is presented by Schmidt et al. [SWSJ06]. It utilizes simple inflation, extrusion or sweep shapes, which are merged interactively using an implicit volumetric description. Holes and shape hierarchies are provided by the Constructive Solid Geometry (CSG) like nature of this representation. A drawback is the construction of sharp surface features at lower levels of resolution. Thus, a combination of multiple modeling metaphors and operators is required. More general, such distinct metaphors often imply a separation of shape deformation- and control operations, as there is only an abstract relation between moving a slider and its impact on the perceived object features [OSD06].

Recent approaches utilize sketched *3D space curves* as a natural extension of planar 2D contours. Those curves are directly constructed in 3D space without an intermediate retrieval step. Although this makes the construction process more complex with respect to the input methods, it also imposes alternative solutions to the classical mapping problem. Curve-based 3D surface construction is a convenient feature among many CAD modeling packages. However, their intuitive and determined construction using sketch-like input is still a challenging task. A CAD oriented SBM approach is presented by Michalik et al. [MKB02]. In this work, curve projection techniques and on-surface registration allow for a seamless transition between the creation of new objects and the boundary-based modification of the resulting shape. Another system tailored towards sketch modeling in a CAD environment is presented by Kara et al. [KS07]. Using predefined templates of low complexity 3D curves are created by projecting the input strokes. Further editing steps via constrained over-sketching from single or multiple perspectives allow for subsequent editing and exact placement. The resulting surface topology finally can be derived from the template. Further conceptual modeling approaches exploit expressive modeling metaphors as extrusion (e.g. SESAME [OSD06, IMT06, ZNA07], Google SketchUP). Those systems usually have to balance resulting object and interface complexity versus maintaining the reduced simplicity using a consistent modeling metaphor.

Systems as proposed by Bae et al. [BBS08] and Schmidt et al. [SKSK09] utilize sketch-based command strokes to steer the construction process and perform mode switches during modeling. In order to interpret the input those systems require heuristics to construct the given curves in the right context towards the existing geometry. Further, the system by Bae et al. represents an advanced and ergonomically optimized 3D drawing program, while its input processing provides an innovative and promising alternative to classical CAD oriented approaches. Rose et al. [RSW*07] have presented a system which constructs developable surfaces out of such 3D boundary constrains by utilizing a local enveloping technique. The system detects multiple surface configurations for ambiguous networks, while the overall runtime depends on the network complexity.

Das et al. [DDGG05] use a 2D projection technique to create simple space curves and subsequently surfaces from boundaries described by the resulting network. Due to the projection paradigm and lower curve complexity, the overall shape variety is limited to simpler objects. An interactive strategy to extract surfaces from networks is presented by Sheng et al. [SWS08]. Using spatial correspondences between curves, the system automatically identifies surfaces by detecting closed cycles within the resulting graph structure. One downside is the lower mesh quality and non-interactive computation times. Using an analyze-and-edit approach instead of direct construction Gal et al. [GSMCO09] have shown how to efficiently use such networks to topologically describe and modify complex 3D objects in an intuitive and consistent way.

3. Automated Conceptual 3D Surface Extraction

To provide an expressive and intuitive modeling metaphor the presented system utilizes a 3D curve concept to construct a topological *curve network* structure. This network holds geometric as well as topological information and provides the transition from unstructured 2D sketch input towards a 3D surface representation of the desired shape. To extract the resulting geometry, edge sequences have to be found and are used to construct a *quad mesh* structure that allows

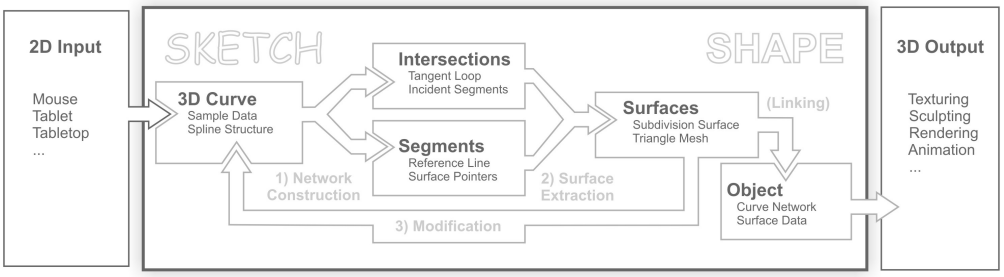


Figure 1: *Schematic overview of the conceptual modeling workflow within the system.*

for compact and flexible surface representation. For final export and rendering steps the quad mesh is converted into a triangular mesh description. This workflow is illustrated in Figure 1.

3.1. 3D Space Curve Construction

In order to handle sketch-like data the input is converted into parametric freeform curves. Those are interpreted as discontinuities on a later surface, while shading or on-surface details are not part of this primary sketch input. Based on a 3D point classification (see Figure 2) three different surface point types can occur: Points $\mathbf{p}^0$ which lie directly on a surface are characterized by exactly one position and a corresponding normal direction. Object surfaces are limited by sharp edges that are characterized by a set of $\mathbf{p}^1$ points, which introduce a singularity on the object surface. All points on this curve are characterized by two normal directions. If both are equal the curve can be considered being smooth in this point.

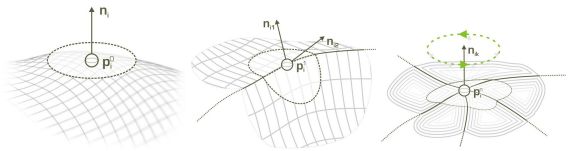


Figure 2: *Different point types that occur during surface construction.*

Positions where two or more edges intersect are defined as $\mathbf{p}^{n-1}$, where n denotes the number of surface normals, adjacent edges and surfaces within this point. Based on this definition a set of 3D curve creation techniques is proposed to acquire $\mathbf{p}^1$ and $\mathbf{p}^k$ points with $k > 1$. Those points are used as input constraints for the surface construction process. Higher order points $\mathbf{p}^k$ are not directly generated but inferred from the current structure of the topological network, which will be explained in section 3.2. Thus, the overall surface construction process can be reformulated as finding a suitable number of $\mathbf{p}^0$ points to describe the intended shape. In order to acquire an initial set of $\mathbf{p}^1$ points, the 2D input data is projected directly into the 3D scene. Using the temporal drawing order they can be used as control points for a parametric spline. This implies the necessity of effective 3D curve control operations and the analysis of additional reference geometry to aid the construction process. As in the beginning of the design process no reference geometry is available an additional *support geometry* is proposed to guide the drawing process (see section 3.5).

3.2. Topological Curve Network Construction

The set of created curves is analyzed towards their spatial properties which are encoded in the topology of the network structure. As main advantage curve networks offer structural similarities to the SBM input and support complex shape models [GSMCO09]. Note that the complexity of the curve network significantly increases with the amount of details. Under-determined networks might offer various ambiguities which can be resolved by inserting additional constrains. Thus, it is assumed that conceptual objects can be characterized by networks of reasonable complexity. Once the model is beyond the conceptual stage further processing can be done using established modeling techniques.

To support the analysis of the geometric structure, the network is extended by the definition of *curve intersections*. Such intersections are characterized by a number of adjacent lines $\mathbf{c}_k$ and a parameter $t \in [0,1]$ describing the local position of the intersection on the parameterized curves. If there is more than one adjacent line this corresponds to the definition of higher order $\mathbf{p}^k$ point. This leads to $\mathbf{i}_j = (\mathbf{c}_0, t_0, \mathbf{c}_1, t_1, \cdots, \mathbf{c}_r, t_r)$ with $t_i \in \mathbf{R}$ describing the parameter location of the intersection on the curve and $\mathbf{I} = (\mathbf{i}_0, \mathbf{i}_1, \ldots \mathbf{i}_m)$ denoting all intersections. The index thereby denotes a local sorting parameter and r the number of adjacent lines. Due to numerical limitations the determination of spatial coherences is based on thresholds. Note that these thresholds are defined in dependence of the viewing distance and therefore drawing scale in screen space. Intersections split up every curve into a set of *curve segments* where every segment starts and ends at one intersection $\mathbf{cs}_j = (\mathbf{i}_{min}, \mathbf{i}_{max})$ and $\mathbf{CS} = \mathbf{cs}_0, \mathbf{cs}_1, ..., \mathbf{cs}_k$. By this definition segments are guaranteed to be intersection free and do not cross each other.

A shape surface is given by a set of subsequent segments $\mathbf{s}_j = (\mathbf{cs}_0, \mathbf{cs}_1, \ldots \mathbf{cs}_v)$ with v defining the number of boundary curve segments of the surface and $\mathbf{S} = (\mathbf{s}_0, \mathbf{s}_1, \ldots \mathbf{s}_g)$ the amount of all surfaces.

Finally, a 3D shape network is given by $\mathbf{CN} = (\mathbf{C}, \mathbf{I}, \mathbf{CS}, \mathbf{S}, \mathbf{M}_{4x4})$. Thus, **CN** stores the geometric input information in C, constructed topology information in I and CS, and inferred new geometric information in S. Additionally every object is associated with a matrix $\mathbf{M}_{4x4}$ that describes general transformations of this shape. This allows the definition and control of patterns of the same object in the network. Patterns are, for instance, planar or rotational symmetry. Multiple scene objects and patterns can be merged as the correct temporal drawing order and topology is implicitly given by the associated network.

3.3. Surface Extraction from Topology Networks

The surface reconstruction from topology networks is a well known problem, which has been subject in several application areas. For instance, the shape reconstruction from planar closed contour lines, such as segmented medical data sets (e.g. [BV07, LLB*08]). Also in geometric modeling this issue has been investigated (e.g. [SWZ04, Lev99, RSW*07]). It has been shown that the extraction is a non-trivial problem and requires substantial amount of computing time depending on the given input information and the curve network complexity. In order to provide an interactive extraction technique a faster and more flexible method is required. Thus, we propose two different extraction strategies: A so called *batch processing* and a *progressive extraction* approach.

In *batch processing* the complete topology evaluation and surface construction is only done once. This automatic process requires no further interaction. Given the constructed network **CN** a corresponding set of surface-patches **S** has to be found. Thus, we assume that those surfaces are a subset of minimal cycles in the undirected graph structure represented by **CN**. A minimal cycle within this graph does not contain any other closed path sequences. Intuitive surfaces usually represent parts of the outer hull of the final object and do not cross interior regions. This mental concept of an *inside* can only be expressed by further geometric assumptions: First, the number of possible surfaces is limited by the number of edges in the network. For every edge there can be only two surfaces adjacent to it: $|\mathbf{S}| \leq 2 \cdot n$. Second, we introduce the concept of the *tangent-loop*. This structure denotes an ordering of edges adjacent to one vertex. All neighboring segments in a patch sequence are required to be neighbors within the associated tangent loop. This ordering is expressed by the notion of an angle $\alpha_k, k \in [1, |I|]$ around an average normal direction in i_k. If the average cross product will become exactly zero, the algorithm will identify multiple valid surface configurations. However, in conceptual modeling with freehand curves, this is rather unlikely to occur. Using this approach, the user does not have to be concerned about surface-construction issues while designing the intended shape. One downside is that the user has no feedback or direct influence on the surface-creation process and might be confronted with unintended results. In addition, the complete construction process will consume non-interactive computation times if triggered for larger networks.

The *progressive surface extraction* approach continuously tracks the surface creation process and allows for immediate intervention. This process will be constantly triggered if new sketch curves are added. The only additional information required is the constructional order of the input curves. To construct the surfaces their boundary is assumed to be a closed curve segment sequence at all time which is referred to as *active boundary*. If two existing lines within one network are connected, the existing boundaries will be extended by the newly created segments and introduce double-edges in the active boundary sequence. This approach extends similar concepts presented by Sheng et al. [SWS08] and is illustrated in Figure 3. If an edge connects two vertices of the same border the patch sequence is closed in a circular way including the new edge, which produces a new patch sequence and changes the segments within the existing one. However, there are two main limitations: First, the construction of topological holes in the object is not directly possible as all closed sequences are interpreted as surface immediately. Second, connecting two separated networks results in a number of complex cases, that require additional geometric analysis.

3.4. Surface Construction and Parametrization

The given curve network only constrains boundaries and initial tangent information, while multiple configurations for potential surfaces still remain. Using those constrains the surface representation has to be constructed of boundaries with up to $k \in \mathbf{N}, k > 0$ limiting curves. Furthermore, it has to be stable against missing or noisy input and numerical inaccuracies, as they are quite common using conceptual freehand curves. To address both challenges a subdivision-oriented *quad-mesh* structure is used and interpreted by the means of a *spring-force network*. A related definition for triangular meshes has been presented by Das et al. [DDGG05] and Kara et al. [KS07]. The quad-mesh extraction consists of two steps: Step one determines the initial surface topology and step two performs a subsequent force based iteration along the vertex positions. The quad-mesh $\mathbf{M} = (\mathbf{mp}_0, \mathbf{mp}_1, \ldots, \mathbf{mp}_l)$ consists of a set of mesh elements $\mathbf{mp} = (p, \mathbf{mp}^*_{up}, \mathbf{mp}^*_{down}, \mathbf{mp}^*_{next}, \mathbf{mp}^*_{previous})$ which store a position $p \in \mathbf{R}^3$ and four links to their neighboring elements. The interconnecting edges between those elements can be either assumed to be linear or defined as spline segments inheriting geometric properties from the reference curve segments. To extract the final triangular mesh, the resulting quads are discretized using a defined sampling resolution. This allows for subdivision procedures to increase the overall surface resolution. Note that for insertion or dele-

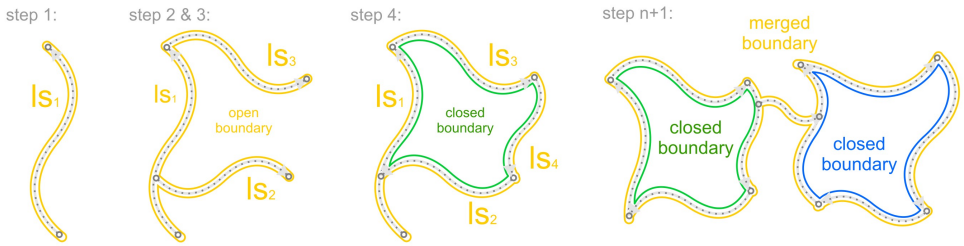


Figure 3: *Process of creating an active boundary and subsequent surface extraction.*

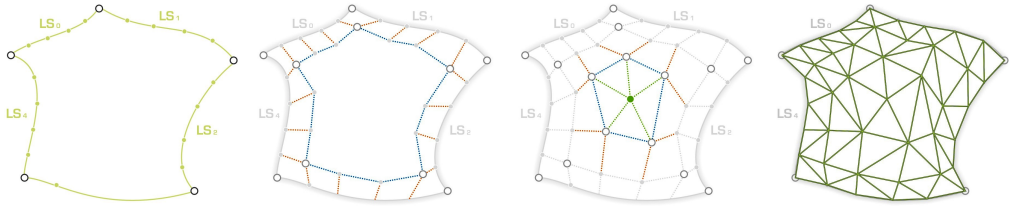


Figure 4: *Steps to create a quad-dominant surface structure from an n-sided input boundary, procedure for triangular meshes by Das et al. [DDGG05]*

tion the valence per element should be kept constant in order to maintain the advantages of the quad mesh topology [SWZ04].

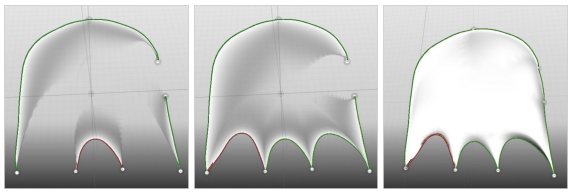


Figure 5: *The surface construction applied to an open non-convex boundary. Vertices between non-adjacent intersections are rendered transparently and can move freely during force-based iteration.*

This definition of $\mathbf{M}$ allows for interactive, flexible and force-based control of the surface under different constraints (see Figure 5). In order to find expressive and reasonable shape configurations, appropriate forces have to be defined. From the inherent definition of interconnecting links two different types can be derived directly: One force describes the attraction forces $\mathbf{f}_{int}$ between elements within one construction level, the second one between the elements of two different levels $\mathbf{f}_{ext}$. For $\mathbf{f}_{ext} = \mathbf{f}_{int}$ the iteration has a smoothing effect on the surface, and converges against a minimal twist surface [Far02]. The goal is an equilibrium state, in which the quad vertices converge to a steady geometric description. Note that current force definitions are strictly local, but can be extended towards global surface optimization techniques.

3.5. System Interaction Techniques

The system is controlled by a 2D input device (e.g. mouse, pen, tabletop) and an additional device (e.g. keyboard) to control mode switches. Command gestures are not part of the system, to avoid ambiguities with small, high frequency strokes which might be substantial part of the conceptual drawing phase. Starting with the first modeling steps the supporting geometry, mentioned in Section 3.1, consists of a simple plane which can be aligned explicitly or using the existing network. Subsequently the plane will be aided by additionally created geometry or adjusted using construction curves. With this metaphor a smooth transition between the creation, modification and augmentation of geometry is achieved. The placement and control of the supporting plane is augmented by four different methods to align the plane within space. Note that the alignment of the plane is characterized by a center position $\mathbf{p}_{plane} \in \mathbf{R}^3$, a normal direction $\mathbf{n}_{plane}$ and two local main directions $\mathbf{v}_x, \mathbf{v}_y$. The methods are illustrated within a simple example network in Figure 6. Additionally the user can define a complete curve which deforms the plane with respect to the geometry of this curve.

According to the traditional pen-and-paper metaphor two different types of curves are provided for our system: *Construction curves* which are used to create $\mathbf{p}^1$ reference points within the scene and guide the topological construction process. *Styling curves* which are used to modify existing curves, control advanced curve operators or simply can be used as spatial and visual reference during the design process. This class of curves is not included into the prepro-

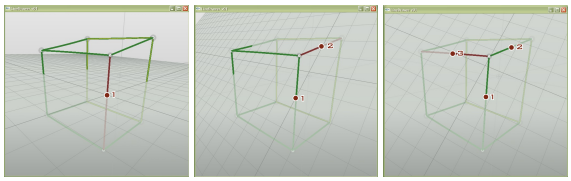


Figure 6: *Methods to align the support plane within the existing curve network using one, two or three positions.*

cessing procedure. Providing additional spatial and visual reference during the object construction can significantly improve the overall design process and understanding of the 3D scene as shown by Schmidt et al. [SIJ*07].

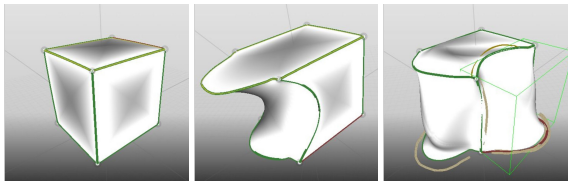


Figure 7: *Using over-sketching to modify a simple network and subsequent update of its surface.*

Furthermore, styling curves allow for *Over-sketching*, which represents a well-established feature in many SBM systems (e.g. [ZNA07, KS07, BBS08, MKB02]). This is implemented using styling curves drawn next to an existing construction curve. If the distance between styling- and target curve is below a certain threshold, the active construction curve is modified using a transition function and resampled if necessary. By updating adjacent surfaces, an iterative reshaping and styling process can be used to modify existing surfaces as shown in Figure 7.

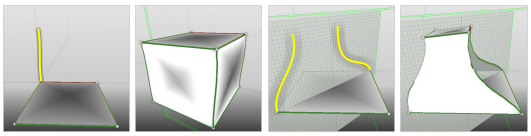


Figure 8: *Extrusion operators: Linear Extrusion using one styling stroke (yellow) and bilinear extrusion with two styling stroke and a active boundary (green).*

Extrusion has shown to be a powerful operator to speed up complex and repetitive construction tasks (e.g. [OSD06, IMT06, SWSJ06, FCAD03]). It allows for copying complete curve subsets while maintaining the topological structure. Defining one styling stroke as reference it can be used to perform a single linear extrusion operation on the current active boundary. Two reference strokes allow for an additional scaling and rotation of the target curves (see fig. 8). Both operators allow for complex extrusion of arbitrary freeform boundary curves, whereas the plausibility of results is mainly limited by the surface parametrization. As the extrusion paths are guaranteed to represent feature lines in the extruded shape, this metaphor complements the overall modeling concept.

All of the above mentioned operators are implemented by exploiting the parametric properties of the underlying network descriptions. A main advantage is that those operators have to be only defined on the boundary curves given in the curve network. As long as the original topology and ordering is maintained, the surface extraction methods can be applied directly on the resulting curves. Thus, these operations are significantly faster than applying them directly to the complete surface. Additionally, they do not have to take the preservation of surface features into direct account, as they can be reconstructed from the boundaries.

4. Modeling Results and Discussion

In order to test the presented concepts, the described systems has been tested by constructing a set of different objects with varied complexities. The prototype has been tested on a CPU with 2.0 GHz and 1.0 GB RAM using a GeForce 7600 Go. Ensuring those minimal requirements, the prototype offers interactive modeling speed also for higher model complexities. All models have been created by a system expert and have been exported and rendered using the open source 3D modeling package Blender.

Scenario I: Car Model: The first constructed model represents a car which has been constructed using quad patches only. This minimizes the number of ambiguous configurations within the network and allows for the direct application of the batch procedure as shown in Figure 9.

Scenario II: Larva Model: The example shown in Figure 10 shows the effective application of extrusion operators for construction of more complex organic shapes. The construction of this model took 26 construction curves, 12 applied extrusion operations.

Scenario III: Bug Model: The effect of patterns proposed in section 3.2 during the modeling process is illustrated in figure 11. In the overall modeling session 32 curves have been created and additionally six extrusion operations for the body.

Scenario IV: Head Model: The example in Figure 12 reveals the inadequate reconstruction additional complex details using the current implementation, while the resulting model represents an acceptable base-mesh for further shape processing. Creating the model in its final state took 48 lines (without symmetry) and three extrusion operations.

For the test scenarios it has been shown that the batch extraction method for four sided patches works stable on quadrangular networks. In general the constructional effort for those kind of networks is too high to be effectively applied for conceptual modeling. On the other hand this procedure can be applied for both extrusion operators, while they are guaranteed to produce quadrangular surface boundaries along the extruded sequences. For patch sequences

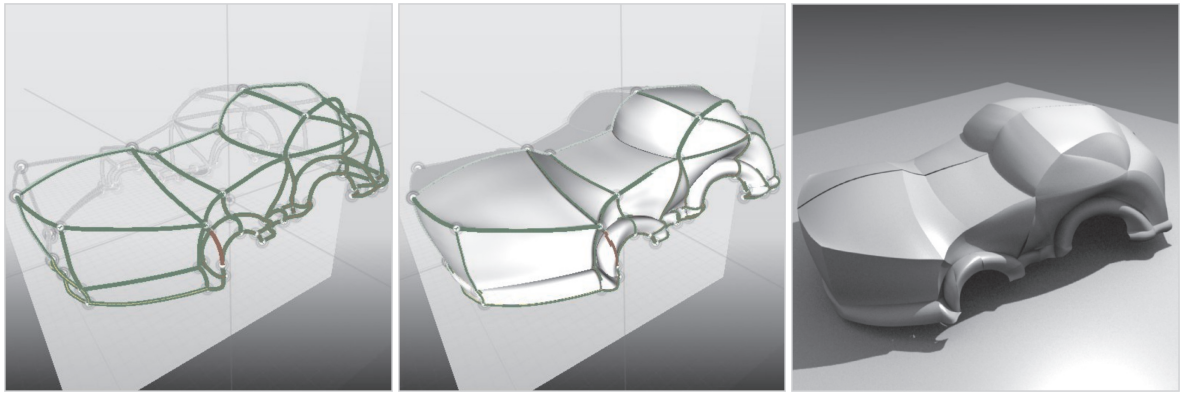


Figure 9: *Constructing a car model consisting of quad-patches only.*

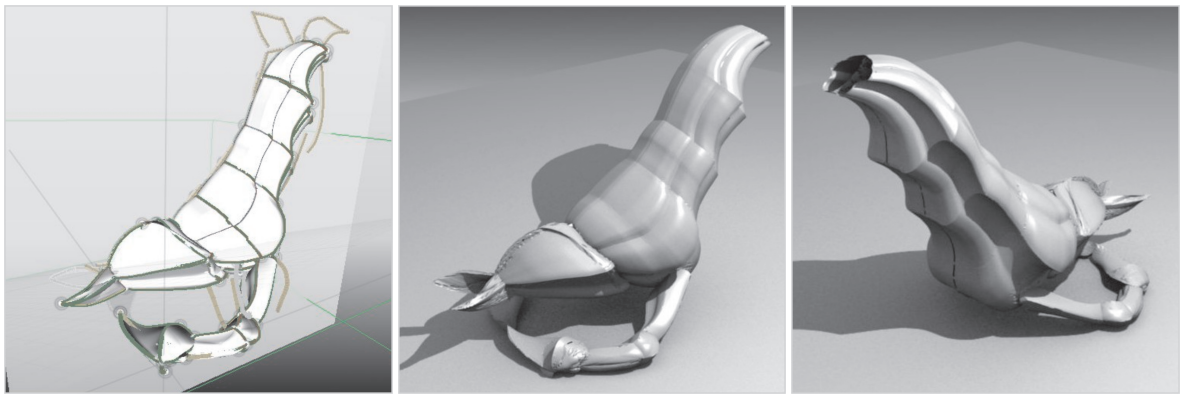


Figure 10: *Example session creating a larve model.*

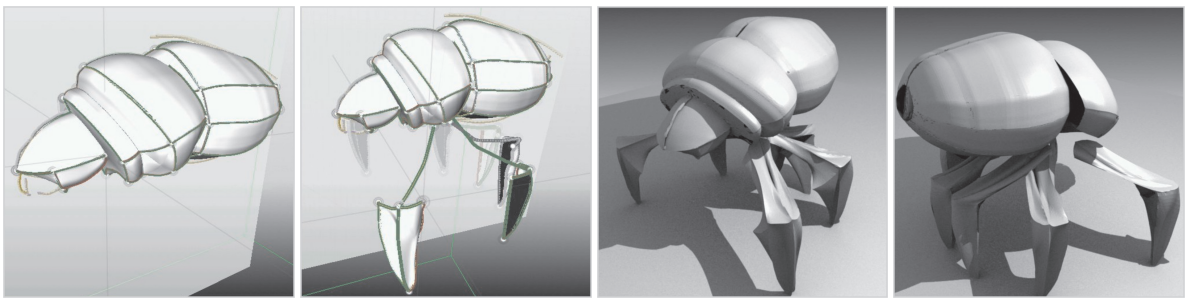


Figure 11: *Utilizing rotational symmetry features to create a insect model.*

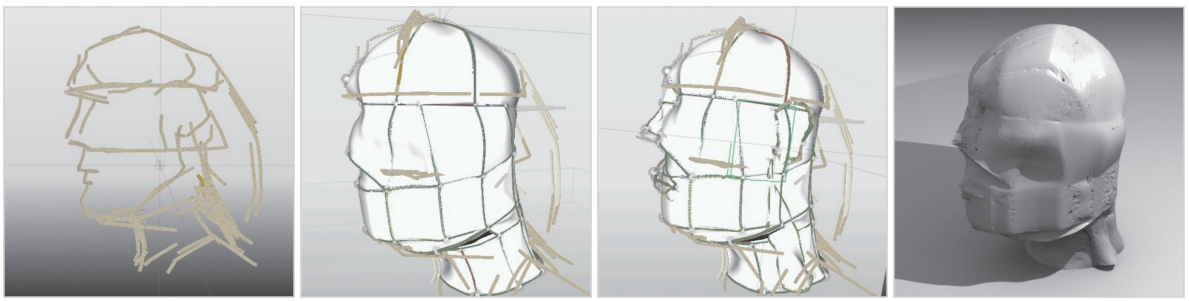


Figure 12: *Example session modeling a human head.*

with more than four boundaries, the progressive method delivers good results. Limitations concerning the surface quality are mainly due to the force based construction of minimal spanning surfaces, which tend to deliver less appropriate results for complex boundary configurations. Finally storing and representing the shape indirectly within a curve network is a compact way to represent geometry as confirmed by Gal et al. [GSMCO09].

5. Conclusions

The proposed system presents a compact set of methods for interactive, intuitive and expressive curve construction. It uses an aggregation of basic and advanced curve operators and controlled augmenting geometry to intuitively define and control the creation of curves within a 3D scene. In conjunction with those operators, parameterized curve networks allow for expressive 3D conceptual modeling. The network supports creation, deformation and augmentation of shapes, ranging from sharp distinct features as well as smooth, organic objects and hybrid configurations. Furthermore, the metaphor offers the possibility to integrate existing modeling approaches into a consistent framework. Thus, the approach is tailored towards conceptual purposes and constructing base-meshes.

Besides those advantages there are still remaining challenges: With regard to the interface, the system shares the *self disclosure* problem mentioned in [OSSJ09]. Although the basic drawing and surface extraction processes can be explored easily, the access to advanced operators requires additional learning effort and abstract understanding of the underlying concepts. The extension towards an appropriate interface and interaction metaphor for conceptual 3D SBM modeling remains an open issue. In addition to this, the current techniques for surface parametrization does not capture the variety of possible shapes, while minimal spanning configurations fail to faithfully reflect surface features encoded within the given boundaries. The topological correspondence between surfaces is also not part of the current system, but would be necessary to formulate more global surface parameterizations.

Future improvements of the system may comprehend advanced subdivision approaches to reflect inherent shape features encoded within the topology network. In addition to the proposed modeling metaphor, the overall concept is still missing a consistent interface- and interaction concept to steer control operations. Finally the practical improvement and optimization of conceptual SBM techniques can only be accomplished, by integrating them into a real practical workflow. This implies a practical assessment of the proposed prototype and its developed features. A specific user study with a larger spectrum of users with different 3D modeling backgrounds, will inspire further improvements and get the proposed techniques closer to the goal of their practical embedding.

References

[BBS08] BAE S.-H., BALAKRISHNAN R., SINGH K.: Ilovesketch: as-natural-as-possible sketching system for creating 3d curve models. In *UIST '08: Proc. of User interface software and technology* (2008), pp. 151–160. 2, 6

[BV07] BAREQUET G., VAXMAN A.: Nonlinear interpolation between slices. In *SPM'07: Proc. of symposium on Solid and physical modeling* (2007), pp. 97–107. 4

[DDGG05] DAS K., DIAZ-GUTIERREZ P., GOPI M.: Sketching free-form surfaces using network of curves. In *SBIM '05: Proc. of Sketch-based interfaces and modeling* (2005). 2, 4, 5

[Far02] FARIN G.: *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., 2002. 5

[FCAD03] FARIA N., CARDOSO T., ARAÚJO B., DIAS F.: *GIDes++ - 1st Usability test at CENTIFME*. Tech. rep., INESC-ID, 2003. 2, 6

[GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iwires: an analyze-and-edit approach to shape manipulation. In *ACM SIGGRAPH'09 papers* (2009), pp. 1–10. 2, 3, 8

[IMT06] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (2006), p. 11. 2, 6

[KH06] KARPENKO O. A., HUGHES J. F.: Smoothsketch: 3d free-form shapes from complex sketches. *SIGGRAPH2006 25*, 3 (July 2006), 589–598. 2

[KQW06] KU D., QIN S., WRIGHT D.: A sketching interface for 3d modeling of polyhedrons. In *Proc. Eurographics Workshop on Sketch Based Interfaces and Modeling (SBIM06)* (2006), pp. 83–90. 2

[KS07] KARA L. B., SHIMADA K.: Sketch-based 3d-shape creation for industrial styling design. *IEEE Comput. Graph. Appl. 27*, 1 (2007), 60–71. 2, 4, 6

[Lev99] LEVIN A.: Interpolating nets of curves by smooth subdivision surfaces. In *SIGGRAPH '99: Proc. of Computer graphics and interactive techniques* (1999), pp. 57–64. 4

[LLB*08] LIU, L., BAJAJ, C., DEASY, O. J., LOW, A. D., JU, T.: Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum 27*, 2 (April 2008), 155–163. 4

[MKB02] MICHALIK P., KIM D. H., BRUDERLIN B. D.: Sketch- and constraint-based design of b-spline surfaces. In *SMA '02: Proc. of Solid modeling and applications* (2002), pp. 297–304. 2, 6

[ML07] MASRY M., LIPSON H.: A sketch-based interface for iterative design and analysis of 3d objects. In *ACM SIGGRAPH'07 courses* (2007), p. 31. 2

[OSD06] OH J.-Y., STUERZLINGER W., DANAHY J.: Sesame: towards better 3d conceptual design systems. In *DIS '06: Proc. of Designing Interactive systems* (2006), pp. 80–89. 2, 6

[OSSJ09] OLSEN L., SAMAVATI F., SOUSA M., JORGE J.: Sketch-based modeling: A survey. *Computers & Graphics 33* (2009), 85–103. 2, 8

[RSW*07] ROSE K., SHEFFER A., WITHER J., CANI M.-P., THIBERT B.: Developable surfaces from arbitrary sketched boundaries. In *SGP '07: Proc. of symposium on Geometry processing* (2007), pp. 163–172. 2, 4

[SIJ*07] SCHMIDT R., ISENBERG T., JEPP P., SINGH K., WYVILL B.: Sketching, scaffolding, and inking: a visual history for interactive 3d modeling. In *NPAR '07: Proc. of Non-photorealistic animation and rendering* (2007), pp. 23–32. 6

[SKSK09] SCHMIDT R., KHAN A., SINGH K., KURTENBACH G.: Analytic drawing of 3d scaffolds. *ACM Transactions on Graphics 28*, 5 (2009), (to appear). 2

[SRS91] SACHS E., ROBERTS A., STOOPS D.: 3-draw: A tool for designing 3d shapes. *IEEE Comput. Graph. Appl. 11*, 6 (1991), 18–26. 1, 2

[SWS08] SHENG B., WU E., SUN H.: Sketching freeform meshes using graph rotation functions. *The Visual Computer 24*, 7-9 (July 2008), 745–752. 2, 4

[SWSJ06] SCHMIDT R., WYVILL B., SOUSA M. C., JORGE J. A.: Shapeshop: sketch-based solid modeling with blobtrees. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses* (2006), ACM, p. 14. 2, 6

[SWZ04] SCHAEFER S., WARREN J., ZORIN D.: Lofting curve networks using subdivision surfaces. In *SGP '04: Proc. of symposium on Geometry processing* (2004), ACM, pp. 103–114. 4, 5

[ZHH96] ZELEZNIK R. C., HERNDON K. P., HUGHES J. F.: Sketch: an interface for sketching 3d scenes. In *SIGGRAPH '96: Proc. of Computer graphics and interactive techniques* (1996), ACM, pp. 163–170. 1, 2

[ZNA07] ZIMMERMANN J., NEALEN A., ALEXA M.: Silsketch: automated sketch-based editing of surface meshes. In *SBIM '07: Proc. of Sketch-based interfaces and modeling* (2007), pp. 23–30. 2, 6