

# A User Interface for Freeform Modeling Based on Convolution Surfaces from Sketched Silhouette Curves

Zuzana Kúkelová

Faculty of Mathematics, Physics and Informatics, Comenius University  
Mlynská dolina, 842 48 Bratislava, Slovak Republic  
kukelova@salamon.sk

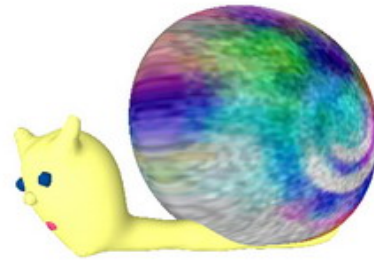
## Abstract

Traditional 3D modeling systems are commonly intended for designing complicated and exact objects, but they are not always suitable for quick design of freeform shapes. It is because these systems are highly complex and user has to follow a long and tedious learning process before being able to create meaningful models. For these reasons, in recent years there have been many efforts in developing more intuitive interfaces for interactive and quick design. Because sketching is very natural process for most people and makes the design process faster and more intuitive, it become popular to design 3D freeform shapes just from 2D sketches, drawn by user. In this paper we present such a system for modeling 3D shapes from sketched silhouette curves. Our approach uses convolution surfaces with linear weight distribution as a geometrical representation. This convolution surface model allows modeling complex and smooth objects with arbitrary genus including objects with semi-sharp features or holes. With our system the user can easily create variety of shapes only by sketching silhouette curves on different projection planes. Additional modeling operations such as smooth merging of components, carving or texturing are also included. Also first-time users including small children manage this system very well and can create interesting models within minutes.

**Keywords:** Sketch-based Modeling, Interaction techniques, Convolution surfaces

## 1 Introduction

Nowadays, computer graphics and geometric modeling is widely applied in various branches of the industry. Modeling applications have become essential tools in animation, entertaining and manufacturing industries and play a main role in the process of design. Not a long time ago all the technical and graphic designs were drawn on the paper and bigger changes required a new redrawing. The new approach has brought a bigger range of computers and mainly the development of the WIMP (Window-Icon-Mouse-Pull-Down) user interface. Although WIMP interface represents significant improvement over command line applications, menu-based interactions do not map quite clearly the spontaneous process of drawing.



**Figure 1:** 3D textured object created in our system.

Existing 3D modeling systems based on this standard WIMP interaction metaphor are commonly intended for designing complicated and exact objects, but they are not always suitable for quick design of freeform shapes. This fact is caused by the enormous complexity of these classical 3D modeling systems, by the greater time demand necessary to manage them better, but especially by the need of some technical knowledge and detail knowledge about the geometric representation to work with them. User has to follow a long and tedious learning process before being able to create meaningful models. Because of this complexity these modeling systems are mostly inaccessible for children and less experienced users.

For these reasons, in recent years there have been many efforts in developing more intuitive interfaces for interactive and quick design.

Because sketching is very natural process for most people and makes the design process faster and more intuitive, it becomes popular to design 3D freeform shapes just from 2D sketches, drawn by the user. This sketch-based modeling provides not only advantages in the drawing speed but it also supports natural human abilities and offers rich expression afforded by pen movements and pressure.

In this paper, we present such a sketch-based user interface, named SketchCo. This system is intended for modeling 3D freeform shapes from sketched silhouette curves and is mainly addressed to non-expert users and children. In our approach we use convolution surfaces with linear weight distribution as a geometrical representation.

Our goal is not to provide a complete modeling tool, with huge number of modeling operations and possibilities. We rather propose a system in which also

first-time users can create interesting models only by sketching silhouette curves on different projection planes.

In the next section, we briefly survey previous work in sketch-based freeform user interfaces and compare our system to related existing modeling system. Then we present our method for creating convolution surfaces from sketched silhouette curves. In section four, we describe in detail an implementation and advantages of our system SketchCo. Then we present some results and finally the conclusions and possible future work.

## 2 Related work

Most of the sketch-based modeling systems are intended for sketching three-dimensional shapes. Although some of these systems solve also other problems, their main goal is to retrieve 3D information and to reconstruct 3D shapes from given 2D strokes.

The first interface that used sketched silhouette curves for freeform modeling was system Teddy introduced by Igarashi [6]. This system had great impact and became very popular due to its simplicity and innovative interface. Despite of Teddy's simplicity and powerful collection of tools controlled by strokes, the used polygonal representation has limitations especially on surface smoothness and fairness.

Later the authors of Teddy proposed an enhancement to this system, in which they try to eliminate disadvantages of polygonal representation and generate a visually smooth polygonal mesh [5].

Owada et al. proposed a similar interface as the Teddy system but instead of polygonal representation they used volumetric representation [10].

Both representations have their advantages and disadvantages. However for freeform modeling, implicit surfaces are more natural. It is because they are suitable for designing topologically complex objects but especially because they provide very nice smoothness characteristic, which is very important in freeform modeling. Furthermore, implicit functions allow to introduce new operations on such models. This is why in recent years several projects based on implicit surface representation have followed the steps of Igarashi's Teddy.

Karpenko et al. [8] were first, who adopted an implicit representation in this way. Like Igarashi, they present a simple interface for freeform modeling based on sketch methods. They used variational implicit surfaces [13], [14] for object modeling. This representation proposed natural modeling operations like merging of two blobs and oversketching for redefining the boundary of a blob. The main problem of this approach is the difficult reconstruction of the 3D shape thickness. For this purpose the initial stroke is not sufficient and therefore new constraint points have to be created in both sides of the silhouette curve. But as the number of constraints increases the time it takes to compute the coefficients for the variational implicit

surfaces grows as well and therefore this method has high computation cost. Another limitation is that the representation cannot support sharp edges.

Another modeling system that uses variational implicit surfaces as a geometric representation for freeform shapes is system BlobMaker [1]. This system is very similar to Karpenko's one but has some enhancements. Its great contributions are the skeletons. It is because the skeletons can be merged and modified without using the mesh information.

Skeletons have many advantages and in more recent sketch-based modeling systems play important role.

One of the recent freeform user interfaces that use skeletons is system ConvMo [12]. This system is like ours based on cylindrical convolution surfaces [7]. One of the main contributions of this system is method for creating convolution surfaces and an enhanced convolution surface model. This enhanced model allows the user to modify shape thickness and generic circular cross-section of convolution surfaces. This system includes designing objects of arbitrary genus or objects with semi-sharp features and shapes with holes. A weakness of convolution surfaces is that unwanted bulges appear around junctions of skeleton line segments. Their solution produces satisfactory surfaces but not in all situations. For example this method is not suitable for folding surfaces. In order to remove the unwanted bulging effect at the interior of the fold, the skeleton is simplified and the field contribution at joints in this zone is diminished, but this causes that the surface moves away from the stroke. Another problem lies in skeleton extraction algorithm as it may produce unstable results for certain inputs. To obtain convolution surface that interpolate the given silhouette global interpolation process is needed. But for some shapes neither this interpolation is sufficient and the fairness of resulting surface is worse than the non-interpolating one and some oscillations appear. ConvMo is very useful interface for designing prototypes for example for entertainment graphics applications like cartoons. However this system is too complex and it has not as simple interface as its predecessors and therefore is not so suitable for children.

The mentioned disadvantages of system ConvMo try to be eliminated in [1]. For the purpose of reconstruction of three-dimensional surfaces which fit the given silhouette they use blended implicit spheres, placed along the skeleton. These spheres have smaller regions of influence as convolution cylinders, so they follow more faithfully the shape of silhouette curve and therefore fit to this curve better than cylindrical convolution surfaces used in ConvMo. They also propose a new spherical kernel, which can produce high-class continuity surfaces. Authors take advantages of skeleton and enrich the modeling tools with new skeleton manipulation operators like thickness control of the 3D shape, (copy/cut)-paste operation on the skeleton or rotation around an articulation. The main drawback of this approach is that this representation cannot support sharp edges because they are smoothed.

Inspired by these systems, we have developed a system SketchCo. As we mentioned before our approach is very similar to that used in ConvMo. Also we use convolution surfaces with linearly weighted distribution to produce tree-dimensional shapes from sketched silhouette curves. But in contrast to the complicated interface of ConvMo with number of menus and sliders we create a very simple sketch-based modeling system accessible also to non-expert users in which very interesting colored and textured models can be created.

In our approach we also try to improve skeleton extraction algorithm, because that used in ConvMo produce sometimes unstable results. We decide for default use of global interpolation in order to obtain surfaces that best fit the given silhouette. With our skeleton extraction and simplification algorithm and this global interpolation process we obtain suitable surfaces also for folding strokes.

### 3 Reconstruction of convolution surface from silhouette curve

In this section we describe main steps of our method of creating convolution surfaces from sketched silhouette curves.

We first introduce some theoretical background.

#### 3.1 Convolution surfaces

For reconstruction of three-dimensional shapes we use convolution surfaces.

A convolution surface is an implicit surface defined by a skeleton consisting of three-dimensional points, and a potential function representing the contribution of each skeletal point to the scalar field.

Let  $f: R^3 \rightarrow R$  be the indicator function representing the geometry of a modeling primitive  $P$ :

$$f(\mathbf{r}) = \begin{cases} 1, & \mathbf{r} \in P; \\ 0, & \text{otherwise.} \end{cases}$$

Let  $h: R^3 \rightarrow R$  be a potential function that describes the field generated by a single point of that primitive. Let  $\mathbf{p}$  be a point in the skeleton, then the total amount of field at point  $\mathbf{r}$ , generated by the whole primitive is

$$F(\mathbf{r}) = \int_{R^3} f(\mathbf{p})h(\mathbf{r}-\mathbf{p})d\mathbf{p} = (f \otimes h)(\mathbf{r}),$$

which is a convolution of two functions  $f$  and  $h$ .

Convolution surfaces were introduced by Bloomenthal and Shoemake [3]. They used a Gaussian kernel, approximated by a cubic spline. McCormack and Sherstyuk [9] define a new kernel that allows direct analytical solution of field equation for a wide family of primitives. The kernel is:

$$h(r) = \frac{1}{(1+s^2r^2)^2},$$

where  $r = \|\mathbf{r}-\mathbf{p}\|$  and coefficient  $s$  controls the width of the kernel. The field equation can than be simplify as

$$F(\mathbf{r}) = \int_V \frac{d\mathbf{p}}{(1+s^2r^2(\mathbf{p}))^2}.$$

This model has uniform distribution and the resulting surface has a constant radius. Jin et al. [7] proposed a polynomial weighted model and derived analytical solution for line segments, arcs, and quadratic curves. With this model also generalized cylindrical convolution surfaces can be conveniently modeled.

In our work we use linearly weighted model and as skeletal elements line segments. This model produces cylindrical convolution surfaces.

General convolution surface with polynomial weight distribution  $w(t)$  for curved skeleton segment  $\mathbf{c}(t)$ ,  $0 \leq t \leq l$  can be defined

$$F(\mathbf{r}; \mathbf{c}(\cdot), w(\cdot)) = \int_0^l w(t)h(\mathbf{r}-\mathbf{c}(t))dt = \int_0^l \frac{w(t)}{(1+s^2r^2(t))^2}dt.$$

For a line segment, with start point  $\mathbf{b}$  and a unit direction  $\mathbf{a}$ , and the parametric representation  $\mathbf{L}(t) = \mathbf{b} + t\mathbf{a}$ ,  $0 \leq t \leq l$ , the squared distance from a point  $\mathbf{r}$  to the line  $\mathbf{L}(t)$  is given by

$$r^2(t) = \|\mathbf{r} - (\mathbf{b} + t\vec{\mathbf{a}})\|^2 = \|\mathbf{b} + t\vec{\mathbf{a}} - \mathbf{r}\|^2 = d^2 + t^2 - 2ht,$$

where  $\vec{\mathbf{d}} = \mathbf{r} - \mathbf{b}$ ,  $d = \|\vec{\mathbf{d}}\|$ , and  $h = \vec{\mathbf{d}} \cdot \vec{\mathbf{a}}$ . Thus equation for field function becomes

$$F(\mathbf{r}; \mathbf{L}, w) = \int_0^l \frac{w(t)}{(1+s^2(d^2 + t^2 - 2ht))^2}dt.$$

As we mentioned we use only linear weight distribution. In this case the field function, varying from  $w_0$  to  $w_1$  along the line segment is

$$F(\mathbf{r}) = w_0 F_1(\mathbf{r}) + \frac{w_1 - w_0}{l} F_l(\mathbf{r}),$$

where

$$F_1(\mathbf{r}) := F(\mathbf{r}; \mathbf{L}, 1) = \frac{1}{2p^2} \left[ \frac{h}{s^2h^2 + p^2} + \frac{l-h}{s^2(l-h)^2 + p^2} \right] + \frac{1}{2sp^3} \left[ \tan^{-1} \left( \frac{sh}{p} \right) + \tan^{-1} \left( \frac{s(l-h)}{p} \right) \right],$$

and

$$F_i(\mathbf{r}) := F(\mathbf{r}; \mathbf{L}, t) = \frac{1}{2s^2} \left[ \frac{1}{s^2 h^2 + p^2} - \frac{1}{s^2 (l-h)^2 + p^2} \right] + hF_1(\mathbf{r}),$$

with  $p$  being a distance term  $p^2 = 1 + s^2 (d^2 - h^2)$ .

### 3.2 Filtering the stroke and skeleton extraction

First our system converts the silhouette curve drawn by the user into a simple polygon by sampling the input device movement. This set of sampled points is then simplified by filtering. It is because points on this sampled input stroke are not evenly spaced and has too much resolution for our application.

There are several different algorithms for reducing points in a simple polygon. Filtering algorithms used in previous approaches can be divided into two main groups. The first group of filtering algorithms are algorithms dependent on screen resolution in which the edges have uniform length [6], [8]. The second group are algorithms where the edge length is adaptively established [12][1].

We have also decided to use adaptive filtering and have implemented classical Douglas-Peucker approximation algorithm. This algorithm produces a highly nonuniform distribution of sampled points along the silhouette. It has undesirable effects on the reconstruction, because for our purpose, namely for the triangulation we need uniformly spaced points on simplified polygon. Therefore after applying Douglas-Peucker algorithm we add intermediate points between successive points of simplified polygon that are too far. This produces stable result suitable for reconstruction, see Figure 2a).

After filtering we analyze the filtered stroke and reject it if it has self-intersection parts. In our approach if the stroke is self-intersecting, the system will request user to sketch another curve without self-intersections.

To generate a convolution surface from this polygon we must first find a suitable skeleton. For our purpose the accuracy of the skeleton is very important. It is because skeleton and parameters determined from it determine the result shape of generated convolution surface. And this convolution surface must fit the given silhouette curve as close as it is possible.

We considered two algorithms for finding skeleton, Discrete Chordal Axis Transform (CAT) presented in [11] and Medial Axis Transform (MAT) with finding approximate medial axis comprising of line segments similar to the approach presented in [12].

In both approaches skeletonization receives the filtered polygon and in the first step applies a Constrained Delaunay triangulation (CDT), using as constraints the edges of this polygon, see Figure 2b). The triangulation has three types of triangles:

*Terminal triangle* or a T-triangle: a triangle with two external edges (blue triangles in Figure 2 c))

*Sleeve triangle* or an S-triangle: a triangle with one external edge (green triangles in Figure 2c))

*Junction triangle* or a J-triangle: a triangle that has no external edges (red triangles in Figure 2c))

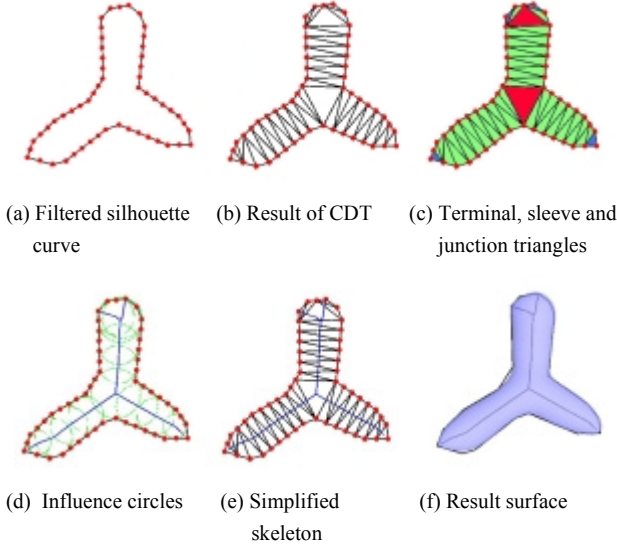
CAT is very simple method for finding the rough skeleton used also in [2], [1]. The chordal axes are determined by connecting the midpoints of the internal edges of triangles according to their connectivity. But as we say, for us is accuracy of the skeleton very important and therefore in some cases chordal axes are not sufficient and additional computations are needed to produce a good-fit convolution surface.

This was reason why we have decided to use another approach, namely finding approximate medial axis comprising of line segments similar to algorithm used in [12]. In this algorithm, after applying CDT to filtered simple polygon, they link up the circumcenters of all triangles according to their connectivity. Then they prune redundant circumcenters to obtain a simplified skeleton with fewer line segments. To do this Tai et al. use three criteria: overlapping, ordering and proportionality criterion. But for certain inputs may this skeleton extraction algorithm produce unstable results.

In our approach we do not use only circumscribed circles of triangles because their circumcenters are for some triangles too far from exact medial axis. For each type of triangle T-triangle, S-triangle and J-triangle we decide to use different circle. For J-triangles we made circumscribed circles with their circumcenters, for T-triangles inscribed circles with their incenters and for S-triangles circumscribed circles or circles with external edge of polygon as tangent and third vertex of triangle as point of this circle. For S-triangle we choose that circle from this two types of circles for which the area of intersection of circle with polygon is greater, see Figure 2 d).

After finding and connecting these centers of circles for all triangles we also prune redundant centers. We do this in order to remove unwanted bulging effect and to simplify next computations. We have implemented several criteria with different parameters for skeleton simplification. We found out that if we prune too much centers of circles the resulting surface will move after global interpolation step away from the stroke in some parts, especially for folding strokes. Therefore we have decided to use less strictly simplification criterion. In this criterion we process the triangles in order, starting from a terminal until a junction triangle or another terminal is reached. Then we prune circle and its center if it lies mostly inside the next obtained circle in mentioned order. This criterion is very similar to overlapping criterion used in [12].

Although obtained skeleton has more line segments, this skeleton extraction and simplification algorithm produces stable result and after global interpolation the resulting surface fit the given silhouette very well (Figure 2)



**Figure 2:** Process of generating convolution surface from silhouette curve.

### 3.3 Constructing and fitting the convolution surfaces

After computing the skeleton we create for each skeleton line segment convolution surface. To obtain convolution surface that fits the given silhouette curve, we need to determine parameters of its field function  $F(\mathbf{r})$ .

The first approximation of  $F(\mathbf{r})$  is computed by considering each convolution surface as being isolated. With this assumption we infer for each skeleton line segment with endpoints  $\mathbf{p}$  and  $\mathbf{q}$  and length  $l$  the weights  $w_0$  and  $w_1$ . We compute these weights like in [13]. We set the field contribution at the point  $\mathbf{r}_p$  at the distance  $R_p$  above  $\mathbf{p}$  and at the point  $\mathbf{r}_q$  at the distance  $R_q$  above  $\mathbf{q}$ , where  $R_x$  denotes the radius at center  $\mathbf{x}$ , to be equal to the implicit surface threshold  $T$  (Figure 3). So we obtain:

$$IT = IF(\mathbf{r}_p) = w_0 (IF_1(\mathbf{r}_p) - F_i(\mathbf{r}_p)) + w_1 F_i(\mathbf{r}_p),$$

$$IT = IF(\mathbf{r}_q) = w_0 (IF_1(\mathbf{r}_q) - F_i(\mathbf{r}_q)) + w_1 F_i(\mathbf{r}_q).$$

Because  $F_i(\mathbf{r}_p)$  and  $IF_1(\mathbf{r}_q) - F_i(\mathbf{r}_q)$  are relatively small, we can approximate the weights  $w_0$  and  $w_1$  as follows:

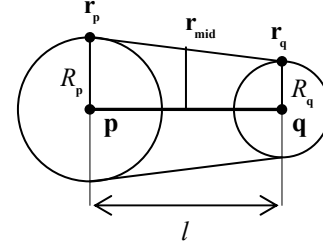
$$w_0 = IT / (IF_1(\mathbf{r}_p) - F_i(\mathbf{r}_p)),$$

$$w_1 = IT / F_i(\mathbf{r}_q).$$

The resulting surface, however, bulge in the middle. We solve this problem like in [13]. To ensure that the surface passes through the point  $\mathbf{r}_{mid}$ , which is at the distance  $(R_p + R_q)/2$  above  $(\mathbf{p} + \mathbf{q})/2$  we scale the weights  $w_0$  and  $w_1$  by factor of  $\kappa$ :

$$\kappa = \frac{IT}{w_0 (IF_1(\mathbf{r}_{mid}) - F_i(\mathbf{r}_{mid})) + w_1 F_i(\mathbf{r}_{mid})}$$

Finally we diminish the field contribution at skeletal joints because here the bulging effect is very strong. We do this by dividing the weights at the skeletal endpoints by the number of line segments ending or starting at these points. This cancels out multiple field contribution. Our experiments also show that we obtain better first approximation by multiplying the radius  $R_x$  at each center  $\mathbf{x}$  by a factor of 0.8.



**Figure 3:** Constructing of a convolution surface for skeleton line segment with endpoints  $\mathbf{p}$  and  $\mathbf{q}$  and length  $l$ .

The important characteristic of implicit surfaces is that they automatically blend when their field functions are summed. So the field function  $F(\mathbf{r})$  of convolution surface that fits the silhouette curve can be obtained by summing the field functions for all skeletal line segments  $L_i$ ,  $i = 1, 2, \dots, n$ :

$$F(\mathbf{r}) = \sum_{i=1}^n \lambda_i F_i(\mathbf{r}, L_i),$$

where  $\lambda_i$  is control scale. In the first approximation  $\lambda_i$  is equal to 1.



(a) Surface without global interpolation (b) Surface with global interpolation

**Figure 4:** Convolution surface without (resp. with) global interpolation.

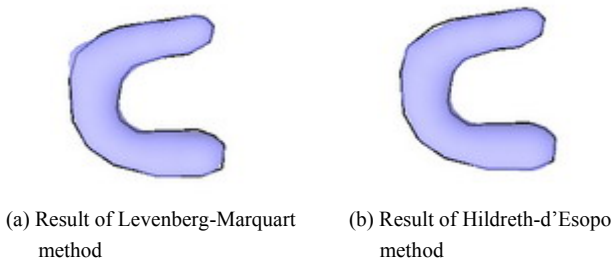
With  $\lambda_i$  equal to 1 and weights  $w_0$  and  $w_1$  computed as described above, one could notice that the surface slightly moves away from the silhouette curve and does not interpolate it, see Figure 4a). It is because the weights for each skeletal segment were computed without considering the influence of the neighboring convolution segments. Therefore if the field contribution of all segments are summed the resulting surface begins to inflate. Sometimes also circles of influence at the endpoints of skeletal line segments overlap too much. Then the inflating effect in these parts of surface is significant and therefore undesirable.

We solve this problem partially by mentioned simplifying of skeleton and diminishing of field contribution at skeletal joints. But this inflating effect is

fully suppressed in global interpolation step. In this step we try to minimize a function that characterizes the distance between the convolution surface and the silhouette points. In other words we solve the following constrained least-squares problem for  $\Lambda = [\lambda_1, \dots, \lambda_n]^T$ :

$$\min_{\Lambda > 0} (\mathbf{F}\Lambda - \mathbf{T})^T (\mathbf{F}\Lambda - \mathbf{T})$$

where  $\mathbf{F}_i = [F(\mathbf{r}_1, L_i), \dots, F(\mathbf{r}_m, L_i)]$  is vector of values of the field function  $F$  for skeleton line segment  $L_i$  at silhouette points  $\mathbf{r}_i$ ,  $i = 1, \dots, m$ ,  $\mathbf{F} = [\mathbf{F}_1, \dots, \mathbf{F}_n]$  and  $\mathbf{T} = [T, \dots, T]$  is vector of thresholds. With this minimization we achieve that the field function of resulting convolution surface will have at silhouette points value  $T$  and therefore will interpolate them. We use the condition  $\Lambda > 0$  to obtain topologically correct result objects.



**Figure 5:** Results of tested non-linear least squares minimization methods.

We have tested several non-linear least squares minimization methods. The first method was modified version of Levenberg-Marquart method, which returns only positive values. The result of this method is shown in Figure 5a). The second tested method was Hildreth-d'Esopo method, see Figure 5b), also used in [12]. These two methods give in many cases very similar results. However Levenberg-Marquart is better in speed, it gives for some silhouette curves not as good fitting surfaces as the Hildreth-d'Esopo. This was the reason, why we have decided to use Hildreth-d'Esopo method. For problem size  $n$  which is in our case usually between 10 and 100 this method is also relatively fast. Like in [12], also in our case interpolation method affects the fairness of resulting surface and sometimes unwanted oscillations appear on it.

## 4 Implementation

### 4.1 SketchCo

Based on reconstruction algorithm described in section 3, we create a sketch-based modeling system, named SketchCo. SketchCo is an application for freeform modeling based on convolution surfaces. It allows user to create surfaces from silhouette curves. The user can sketch these silhouette curves from different viewing directions and in this way add new components to

designed object. The system enables user to modify generated object and its components by performing further editing operations, such as merging or carving and create more interesting shapes. SketchCo provides also other different tools to support creating and editing surfaces. The user can translate, rotate, copy or change the color of components. A special feature of our system is also texturing. Instead of non-photorealistic rendering, which is more common for this kind of systems we have decided for colored and textured objects. It is more attractive for users especially for children. Our system provides three kinds of texture mappings: planar, cubic and spherical. All mentioned operations are supported with Undo and Redo.

It is possible to save models to a simple script file .sc. This file stores only silhouette and transformation information and some additional information like information about merging, carving or color and texture of component. This kind of file has its advantages. For example after loading and performing reconstruction procedure, the resulting surface can be polygonized with another resolution as it was saved. Also some improvements on reconstruction procedure can be made and then performed on such saved model. After loading the file user can modify not only the whole object but also its components.

The system produces an output in form of a polygonal mesh using marching tetrahedra polygonization algorithm [4]. For visualization of polygonized convolution surfaces we use OpenGL. All modeling operations are designed to keep them as easy to use as possible. Users can create a variety of models by only drawing a silhouette curve and can incrementally learn other editing operations when necessary. Moreover SketchCo's interface is very simple and intuitive. Therefore also first-time users including small children manage this system very well and can create interesting models within minutes.

### 4.2 Interface

Our goal was to create very simple user interface, which first-time user could manage in few minutes and without special tutorials. Figure 6 shows the outlook of our system.

SketchCo's user interface is based on traditional pointer input devices such as a standard 2-button mouse or pen based-device. It consists of one window divided into two areas. The first area, on the bottom of the window, is a toolbar. It contains these few buttons: new, load, save, palette, blending, carving, selection mode, info and close. Selection mode is special mode intended for modifying components of object. In this mode user can translate selected components. This mode has also its own toolbar with these buttons: change color of component, delete and texture component and close.

The second area of the window is working space, where the user can sketch silhouette curves and modify created objects. Users can select the best view to sketch

new silhouette or to perform the editing operations by rotating the scene. Everything is done without the use of the keyboard, only with the help of mouse.

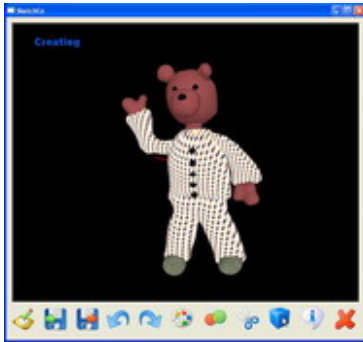


Figure 6: SketchCo’s user interface

### 4.3 Operations

We put the accent on simplicity of our system. Therefore we have proposed only 2 editing operations that are designed to keep them very easy to use.



Figure 7: Example of merging operation

The first editing operation is smooth merging of two components. As we mentioned the user can add new components to designed object by sketching silhouette curves on different projection planes. The depth of a projection plane is obtained from the first silhouette point, which can be drawn on the existing surface or “in the air”. In this way user can obtain separate surface components that are simply attached to some existing one. These components can be then smoothly merged with the use of the operation merging. In this operation user simply select two surface components, which are then automatically merged. Merging is done by summing the field functions of these two surface components and by connecting their skeletons. The result of this operation is one surface component with color of the first merged component, see Figure 7.

The second editing operation is smooth carving. In this operation the user at first selects the surface component that will be carved and then carving surface component that overlap the first selected component. After this selection the  $\lambda_i$  values for all skeletal line segments of carving component are negated. The resulting shape is then obtained by merging these two components.

Similar to previous systems our system also provides standard operations like translation, rotation, copy or changing color of surface component. Interesting and attractive operation of our system is also texturing.

With these operations, the user can create very interesting objects.

## 5 Results

With our system very interesting models can be created within few minutes. Objects such in figures 1, 6 and 9 can be created from 2 to 15 minutes. In Figure 8 are some silhouette curves with convolution surfaces generated from them

SketchCo was tested with children from 6 to 15 years and with older users. Our experiments show that all users managed our system in few minutes without any tutorial or only with the help of short demonstration video. We found out that smaller children have sometimes problems to understand, that they can create real tree-dimensional objects and add new components from different viewing directions, not only from one. This is caused by fact that they have never been creating tree-dimensional models before. After short time also these small children were able to create very interesting objects like other users and considered this system very useful, funny and interesting.

In most cases generated convolution surface reflects the user intent very well and after global interpolation step, fit the given silhouette curve. We achieve good fitting surfaces also for most of the folding strokes, which was problem in ConvMo[12]. It can be seen in Figure 8a) and 8b) and also in Figure 7. We achieve these better fitting surfaces due to our improved skeleton extraction and simplification algorithm.

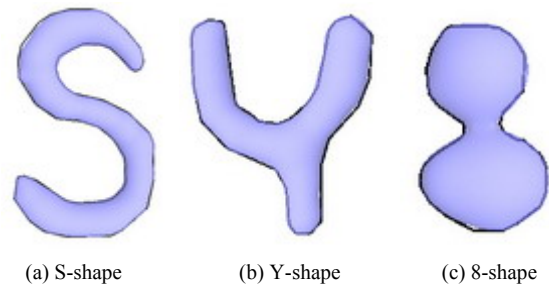


Figure 8: Silhouette curves with convolution surfaces.

## 6 Conclusion and future work

We have presented a simple system for modeling three-dimensional shapes from sketched silhouette curves based on convolution surfaces. The used convolution surface model allows modeling complex and smooth objects with arbitrary genus including objects with semi-sharp features or holes. With our system user can easily create variety of shapes only by sketching silhouette curves on different projection planes. Additional modeling operations such as smooth merging of components, carving or texturing are also included.

Our results are comparable to those obtained by our predecessors. Comparing with ConvMo [12], the closest system to ours, our sketch-based user interface is simpler and more accessible to everyone including small children. We also improved their skeleton extraction algorithm and created algorithm with stable results.

Unwanted bulging effect that appears around junctions of blended line segments is in our system partially solved by simplifying the skeleton and diminishing of field contribution at skeletal joints. As it cannot be entirely removed only by this we have decided for default use of global interpolation. Unfortunately this interpolation method affects the fairness of resulting surface. A surface after global interpolation isn't as smooth as non-interpolating one and sometimes appears unwanted oscillations on it. Therefore our future work may include finding an alternative solution to this bulging effect or improving this global interpolation by minimizing improved energy function.

Our current implementation includes only two editing operations, merging and carving. Although variety of interesting shapes can be created only by sketching silhouette curves and with the help of these operations, in future we want to add also other operations. It is possible to take the advantages of computed skeletons and to implement skeleton-based editing tools similar to that used in [1]. We also want to add tool for controlling the thickness of the 3D shape.

All models can be compactly stored using only silhouette information and after loading polygonized with arbitrary resolution. For polygonization we used marching tetrahedra algorithm [4]. Although this algorithm has satisfactory results for our current purpose, if we want to obtain meshes with better resolution in order preserve also small features, computational time increase. Therefore we want to improve this polygonization algorithm or implement marching triangles algorithm in future.



**Figure 9:** Another example of interesting 3D object created in our system.

## Acknowledgements

I would like to thank my supervisor Silvester Czanner for his useful ideas and help with this paper and Martin Bujňák for his constructive comments to the implementation of our system.

## References

- [1] ALEXE, A. - GAILDRAT, V. – BARTHE, L. 2004. *Interactive Modelling from Sketches using Spherical Implicit Functions*. AFRIGRAPH 2004, Stellenbosch (Cape Town), South Africa, November 2004.
- [2] DE ARAUJO, B. – JORGE, J. 2003. *Blobmaker: Free-form modelling with variational implicit surfaces*. Proceedings of 12° Encontro Portugues de Computacao Grafica 12° EPCG. Porto. Portugal. Oct 2003.
- [3] BLOOMENTHAL, J. – SHOEMAKE, K. 1991. *Convolution Surfaces*. Proceedings of SIGGRAPH 91. July 4. Las Vegas.
- [4] BLOOMENTHAL, J. 1994. *An implicit surface polygonizer*. In P.S. Heckbert, editor, Graphics Gems IV. Academic Press, Cambridge.
- [5] IGARASHI, T. – HUGHES, J. F. 2003. *Smooth Meshes for Sketch-based Freeform Modeling*. ACM Symposium on Interactive 3D Graphics. ACM I3D'03. Monterey, California. April 27-30, 2003.
- [6] IGARASHI, T. – MATSUOKA, S. – TANAKA, H. 1999. *Teddy: A sketching interface for 3D freeform design*. Proceedings of SIGGRAPH 99. August 1999.
- [7] JIN, X. – TAI, C. L. – FENG, J. – PENG, Q. 2001. *Convolution surfaces for line skeletons with polynomial weight distributions*. Journal of Graphics Tools. Vol.6. No.3.
- [8] KARPENKO, O. – HUGHES, J. F. – RASKAR, R. 2002. *Free-form sketching with variational implicit surfaces*. Computer Graphics Forum. Vol. 21. No. 3.
- [9] MC CORMACK, J. – SHERSYUK, A. 1998. *Creating and Rendering Convolution Surfaces*, Computer Graphics Forum. Vol. 17. No. 2.
- [10] OWADA, S. – NIELSEN, F. – NAKAZAWA, K. – IGARASHI, T. 2003. *A Sketching Interface for Modeling the Internal Structures of 3D Shapes*. Proceedings of 3rd International Symposium on Smart Graphics. Lecture Notes in Computer Science. Springer. Heidelberg, Germany. July 2-4, 2003.
- [11] PRASAD, L. 1997. *Morphological analysis of shapes*. CNLS Newsletter, No. 139, pp 1-18, July 1997.
- [12] TAI, C. L. – ZHANG, H. – FONG, C. K. 2004. *Prototype modeling from sketched silhouettes based on convolution surfaces* Computer Graphics Forum.
- [13] TURK, G. – QUYNH DINH, J. – O'BRIAN, et al. 2001. *Implicit Surfaces that Interpolate*. Shape Modelling International 2001. Genova, Italy. May 2001.
- [14] TURK, G. – O'BRIAN, J. F. 2002. *Modeling with Implicit Surfaces that Interpolate*. ACM Transactions on Graphics. Vol 21. No.4. October 2002.