

Sketch Based 3D Modeling with Curvature Classification

Abstract

In this paper, we introduce a simple approach for sketching 3D models in arbitrary topology. Using this approach, we have developed a system to convert silhouette sketches to 3D meshes that mostly consists of quadrilaterals and 4-valent vertices. Because of their regular structures, these 3D meshes can effectively be smoothed using Catmull-Clark subdivision. Our approach is based on the identification of corresponding points on a set of curves. Using the structure of correspondences on the curves, we partition curves into junction, cap and tubular regions and construct mostly quadrilateral meshes using these partitions.

1. Introduction

Sketch based interfaces for modeling started almost 20 years ago with the pioneering work by Tanaka et al. [1] and it gained significant attention with revolutionary work by Igarashi et al. [2]. Since then a wide variety of methods have been introduced [3, 4]. In this paper, we present a curvature based curve partitioning approach for sketch based modeling of free-form 3D shapes. For partitioning curves we have developed 2D-correspondence functions as an extension of shape diameter functions [5]. 2D-correspondence functions are used to identify correspondence points on curves and to robustly classify junction, cap and tubular regions of curves (see Figure 2). Based on this classification, it is easy to construct 3D meshes with mostly quadrilateral faces and 4-valent vertices. These meshes can effectively be smoothed by Catmull-Clark subdivision [6] since they have only a few extraordinary vertices (see Figure 1). We have also implemented the method to demonstrate the effectiveness of the new approach. Our current implementation construct 3D models only from sketched silhouettes. We do not allow T-junctions (occluded curves), which can provide additional information about the shapes [7, 8].

Our process for 3D sketching consists of 3 steps: curve construction, curve partitioning and mesh construction.

(1) *Curve Construction*: This is a 1-manifold curve construction from a given set of arbitrary strokes. Section 3 briefly describes this process.

(2) *Curve partitioning*: This is the process of partitioning the curves into positive, negative and zero curvature

regions using 2D-correspondence function. This is main contribution of our paper and it is described in Section 4 in detail.

(3) *Mesh construction*. This is the final process that converts curves to 3D models. Section 5 provides a description of this process.

In the next section, we briefly discuss previous work on sketch based shape modeling.

2. Related Work

Sketch based modeling has been influential in creating many exciting products and software. For instance, the ideas developed in papers [9] and [10] provided the conceptual basis of Sketchup software [11], which is widely used for modeling simple architectural geometrical shapes. There also exists methods for modeling trees [12], modeling garments [13], modeling using symmetries [14], modeling with 3D curve networks [15, 16], designing highways [17] and terrains [18].

In this paper, we are interested in sketching free form smooth 3D shapes like Teddy [2], which is probably the one of the most influential work in sketch based modeling. Teddy is based on centroidal distance transform which is related to medial axis transform [19]. The distance from a point on the boundary to the medial axis is the radius of the maximal ball, whose center lies on the medial axis, touches the boundary at the point, and is completely contained in the object. This ball is called the medial-ball, and its radius can be seen as a form of local shape-radius connecting the boundary to the medial axis. The problem with medial axis transform is that definition and extraction of the medial axis or even

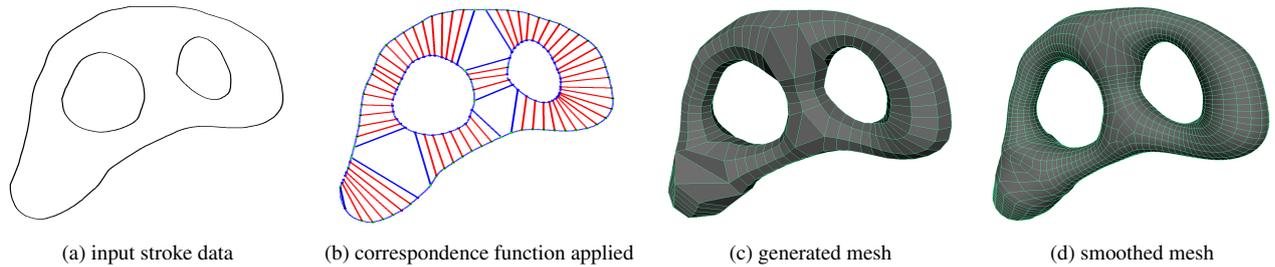


Figure 1: A Genus-2 Mesh created by our system. Our method can partition user drawn sketches shown in (a) into a set of junctions, tubes and caps regions using a correspondence shown in (b). Using this partitioning it is easy to construct 3D models as shown in (c). These models can effectively be smoothed since they consists mostly quadrilaterals as shown in (d).

of discrete approximations using skeletons are complex and often error prone [5]. Another problem with medial axis transform is that the resulting meshes are usually triangulated, which may require a beautification step [20]. Our corresponding function approach, which is inspired by 3D shape diameter function [5], is also related to medial axis transform but it is robust and guarantees to construct meshes with mostly quadrilaterals and 4-valent vertices .

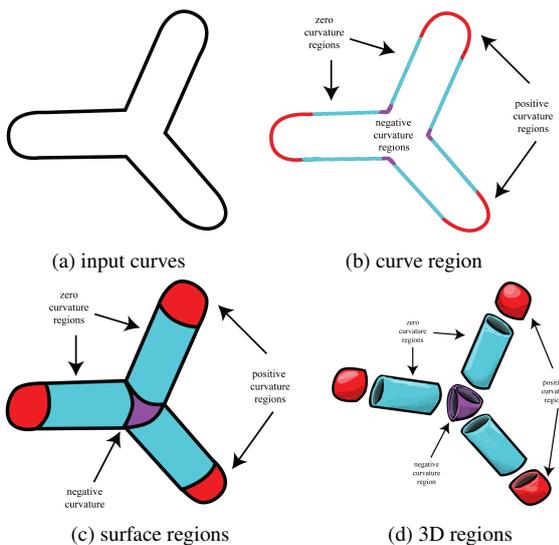


Figure 2: Our correspondence function allows us to robustly classify the portions of curves as junctions, caps and tubes. These regions correspond zero, negative and positive total Gaussian curvature regions. After this classification, it is easy construct 3D mesh surfaces that consist of mostly quadrilateral faces.

Our work is also related to Sketch-based subdivision method [21], which presents a sketch-based interface to design subdivision models. By taking profile curves of the surface as an input, it generates a coarse and quad dominant control mesh with few extraordinary vertices or faces. The corresponding limit surface interpolates

the profile curves with the capability of local control across these curves and of the model in general. However, this approach creates models that are extrusions of the profile curves in z direction by resulting a flat looking model. Our approach can be seen as a combination of the organic look of [2] and smoothness achieved by subdivision surfaces suggested by Nasri et al. [21].

Implicit surfaces provide an alternative for sketching free form smooth surfaces which includes convolution surfaces [22], Variational Hermite-RBF implicits [23], Shapeshop [24] and others [25, 26, 27, 28, 29]. Since implicit surfaces can provide exact and approximate set operations, they are particularly attractive to model complicated surfaces. In this work, we do not use an implicit based approach.

3. Curve Construction

We assume that the initial silhouette is drawn as a set of unorganized strokes. The directions of strokes may not be consistent with how artists would draw. We assume that there is no self-intersections and/or T-junctions. Each stroke is represented as a poly-line. Curve construction process combine, resample and reorder these strokes in such a way that at the end of the process we obtain 1-manifold with well-defined inside and outside. In practice, we allow 1-manifold with boundaries and we can still obtain consistent normals.

Curve construction starts with combining individual strokes into longer curves. This process converts the strokes to a set of closed or open parameterized poly-lines. We then re-sample the curves in equal distances such that the resulting curves are independent of the speed of the sketching. We then turn the curves into 1-manifolds by giving a rotation order to each curve such that always right side indicates inside of the curve. This is done by using classical inside-outside test. This process connects the points by single-links that provide a

consistent rotation direction. We call the resulting set of curves 1-manifold since they can define a closed shape in 2D. For each sample point i , we record and calculate the position C_i , normal N_i and tangent T_i .

4. Curve Partitioning with Point Classification

We developed a 2D correspondence function to partition 1-manifold curves. The corresponding function uses the mesh partitioning property of the shape diameter function to identify parts of the curve as tubular, junction and cap

regions of 3D shapes to be constructed. These regions are categorized based on their total Gaussian curvature. A tubular region is any region with zero TGC. An obvious example of tubular region is a cylinder. A donut is also classified as a tubular region since its total Gaussian curvature is zero, although local Gaussian curvatures are not zero [30]. Junctions are regions that connect tubular regions. Their total Gaussian curvature is negative. Caps are the regions with positive total Gaussian curvature. This classification is the key for obtaining a good quality quad-mesh since each region can easily be converted to a 3D shape as shown in Figure 2(d).

The process of curve partitioning using 2D-correspondence function is simple and efficient. Let C be a 1-manifold, which includes all C_i s, and $C \subset C$. We define 2D correspondence function (2DCF) as a one-to-one function $f : C \rightarrow C$ that corresponds every point C to another point C using the neighborhood-diameter at each point $C_i \in C$. The computation of 2D correspondence consists of two steps: Computation of initial corresponding and correspondence processing.

4.1. Computation of Initial Corresponding

Initial corresponded points are computed using a variant of the procedure that is used shape diameter function. In our computation instead of diameter, we look at minimum circles that can be enclosed between two curve pieces (See Figure 4). For every curve point

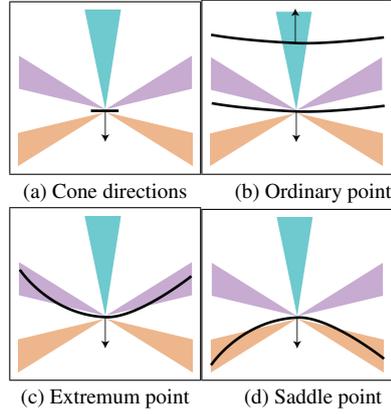


Figure 3: Classification of points.

$C_i \in C$, we define five 2D-cones (field-of-views) as shown in Figure 3(a). We search for corresponding points C_j s checking all curve points that lies inside of these 5 2D-cones. A point C_j is labeled as a corresponding point C_i if the approximate radius of the circle that passes from C_i and C_j with normal vectors N_i and N_j is the smallest among all points that that lies inside of these 5 2D-cones. Figure 4 illustrates how we identify enclosed circle for any two given points C_i and C_j . Note that if we do not find any enclosed shape that is reasonably close to circle, the we chose C_j and we compute the estimation of the 2D circle's radius as

$$r = \frac{|(C_j - C_i) \times (C_j - C_i)|}{2(C_j - C_i) \cdot N_i}$$

where \times is the cross product and \cdot is dot product. Note that the C_j s may not be from initial set C although all C_i s are in C .

The estimated values of radii of the 2D circles can be used to estimate one of the principal components of Gaussian curvature as $\kappa_1 = 1/r_i$,

which is always smaller than 1 since r_i is always larger than 1 in our units. As it is well-known from differential geometry, the Gaussian curvature of a point on a surface is the product of the principal curvatures, κ_1 and κ_2 , of the given point [31]. Although, we do not have direct information for the other principal curvature, based on in which 2D-cone its correspondence point is found, we can classify estimate κ_2 and classify C_i as saddle, extremum (maximum/minimum) or ordinary point and classification is recorded in point C_i .

Ordinary points: If its corresponding point falls in the 2D-cone centered at C_i and directed inward-normal direction $-N_i$, then C_i is most likely to be an ordinary point as shown in Figure 3(b). In this case, $\kappa_2 < \kappa_1$ and it is most likely that $\kappa_2 \ll \kappa_1$, which can be very small number. Since, $\kappa_1 < 1$, the Gaussian curvature, which will be computed as $\kappa_1 \times \kappa_2$ will even be smaller number, which can be considered as 0.

Maximum/Minimum points: If its corresponding point lies in the 2D-cones centered at C_i and directed inward-side direction $-N_i + T_i$ and $-N_i - T_i$, then C_i is most likely to be an extremum as shown in Figure 3(c). Since $\kappa_1 \approx \kappa_2$ and both are positive, Gaussian curvature will be estimated as κ_1^2 . Although this number is smaller than

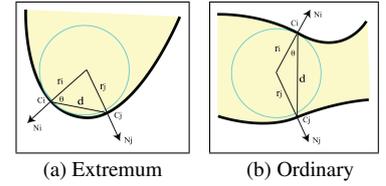


Figure 4: Enclosed circles for extremum (maximum or minimum) and ordinary points.

κ_1 , it is still a relatively large number comparing Gaussian curvatures of ordinary points.

Saddle points: If its corresponding point lies in 2D-cones centered at C_i and directed outward-side direction $N_i + T_i$ and $N_i - T_i$, then C_i is most likely to be a saddle as shown in Figures 3(c) and 5(b). In this case, κ_1 will be a negative number since the enclosed circle is found outside. There is another way to identify saddle points. If more than one ordinary point correspond to the same point C_j as shown in Figure 5(a), they are also considered saddle points. Using this additional information, we actually estimate κ_2 , which is a positive number computed as curvature of ordinary points that chooses C_j as their correspondence.

The Gaussian curvature obtained from correspondence data is very noisy and includes some correspondence information (e.g. correspondence from saddle points) that is not needed to construct 3D model. Shape diameter function computation uses median filtering to remove similar type of noise. We, instead, process the correspondence data to distribute Gaussian curvature among the neighbors to remove the noise.

4.2. Correspondence Processing

Correspondence processing distributes the Gaussian curvature among the neighbors. This process consists of six steps:

1. Updating C : We extend C , to include all C_j s in addition to original C_i s.

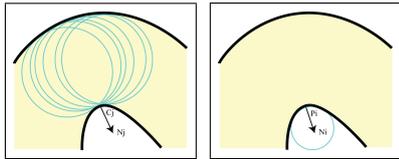


Figure 5: Correspondence to and from saddle points.

2. Splitting one-to-many correspondences: As illustrated in Figure 5(a), more than one ordinary point can correspond to a saddle point, resulting in one-to-many correspondences. When this happens, we simply move one correspondence to one of the neighborhood points until we obtain a one-to-one correspondence.

3. Removing useless correspondences: We remove correspondences of saddle points from C . As illustrated in Figure 5(b), since these correspondences go outside of the shape, they are actually useless except for classifying the point as saddle and computing an estimate of Gaussian curvature.

4. Reassign correspondences: Since correspondences are now one-to-one, both ends of the corresponding

function refer to each other.

5. Laplacian smoothing: We take the immediate neighbors of the point that we process. The parametric average of the correspondences of these neighbors

To do Laplacian smoothing we take the corresponding points of our initial point's neighbors. We then move the correspondence of our initial point to the parametric average of these corresponding points. This operation removes the Gaussian noise resulted from initial computation. Note that this operation replaces some of the elements of C with new points from the whole set C .

6. Edge Classification

Edge Classification: After Laplacian smoothing, all true saddle and extremum points are removed from C . The resulting corresponding function defines a 2D mesh M that consists of mostly quadrilaterals as shown in Figure 6.

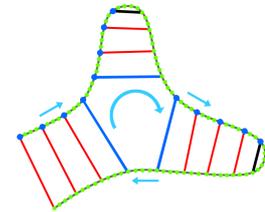


Figure 6: Edge classification using rotation system.

Each face of this mesh has a consistent rotation order. Each correspondence is an edge of this mesh and two correspondence directions, i.e. $C_i \rightarrow C_j$ and $C_j \rightarrow C_i$, are actually the two half-edges [32]. If one of the half-edge of an edge is not part of a face (see black edges in Figure 6), that particular edge is in the boundary of a tubular and cap region. On the other hand, if a half-edge of an edge belong to a non-quadrilateral face (see blue edges in Figure 6), that particular edge is in the boundary of a tubular and junction regions. The rest of the edges (i.e. red ones in Figure 6) simply correspond to tubular regions.

It is possible to further reduce the size of C by removing some edges that correspond to tubular regions. It is also possible to increase the number of edges that correspond to tubular regions. Decimation can be useful to remove the noise in the drawing. Populating, on the other hand, is useful to obtain 3D model that is closer to original drawing.

5. Mesh construction

The construction of a 3D mesh from 2D mesh M is straightforward. The mesh construction consists of three steps:

1. Constructing tubular regions: We consider tubular regions as swept by a sphere with changing radius. For

every edge of M , i.e. C_i-C_j pair, we place a sphere with a radius of the 2D circle that correspond the edge of M . We adjust the center of the sphere in such a way that the rays coming from eye are tangent to the sphere. This adjustment guarantees the correct silhouette. The centers of these spheres form skeletal 3D curves that are medial to the tubular regions. After choosing a great circle for each sphere perpendicular to the skeletal 3D curve, we can construct edge loops by connecting equidistant vertices placed around these circles. We then connect these edge loops to construct tubular meshes in 3D. This process creates not one but a set of disconnected tubular meshes.

2.

Combining tubular regions with junctions: We connect the tubular meshes using junction regions in the 2D mesh. For the border edges around each junction region, we retrieve the edge loops constructed for them. We then connect the vertices of each half of an edge loop to the adjacent half in the consecutive edge loop order one by one in the rotation .

3. *Adding caps:* We can optionally cover the open poles of the tubular mesh with caps.

We have developed a simple sketch based modeling system to show the feasibility of our method. The shapes shown in Figures 1, 8 and 7 are examples that are created using our system.

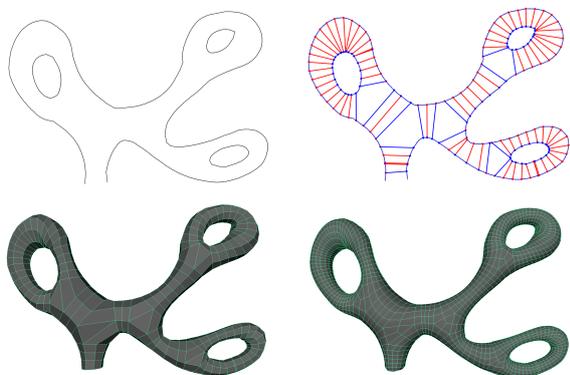


Figure 7: A genus-3 manifold mesh created by our system.

6. Conclusion and Discussion

This paper introduced an approach for sketching 3D models of arbitrary genus. Our approach provides the organic (non-flat) look obtained by Igarashi [2]. We developed a system to convert silhouette sketches to 3D meshes that mostly consists of regular quadrilaterals. Because of their regular structures, these 3D meshes can

effectively be smoothed using Catmull-Clark subdivision similar to Sketch-based subdivision method introduced by Nasri et al. [21]. We have introduced a robust correspondence function, which can be used as an alternative to medial axis transformation and shape diameter functions in applications other than sketch modeling since it can also help to partition even surfaces into extremum, saddle or ordinary regions.

Our correspondence processing is a non-linear operation that implicitly provides total Gaussian curvature in a given region. Therefore, we do not fully utilize the expected Gaussian curvature information we have obtained. However, this information can be useful to create even better meshes by directly controlling vertex and face defects by creating 3 or 5-valence vertices [30].

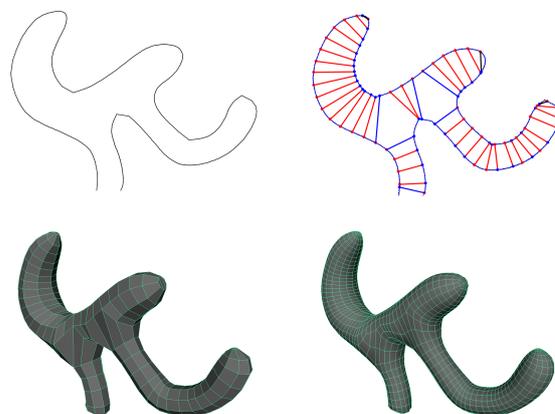


Figure 8: A genus-0 manifold mesh created by our system.

We have also implemented a simple user interface to demonstrate the effectiveness of the new approach. Our current implementation can construct 3D models only from sketched silhouettes. We do not allow T-junctions including occluded curves. Such T-junctions are essential to create more complicated shapes [7, 8]. Our method can only create shapes whose medial axis are 1-complexes, i.e. a connected set of 3D curves. Medial axes of general shapes can be 2-complexes. Our method is specifically useful for the development of interfaces for cartoonists who draw and develop characters that consists of tubular regions connected by junctions [33, 34].

References

- [1] T. Tanaka, S. Naito, T. Takahashi, Generalized symmetry and its application to 3d shape generation, *The Visual Computer* 5 (1-2) (1989) 8394.

- [2] T. Igarashi, S. Matsuoka, H. Tanaka, A sketching interface for 3d freeform design, in: Proceedings of ACM SIGGRAPH'99, 1999, pp. 409–416.
- [3] L. Olsen, F. F. Samavati, M. C. Sousa, J. A. Jorge, Sketch-based modeling: A survey, *Computers & Graphics* 33 (2) (2009) 85–103.
- [4] J. LaViola, Sketch-based interfaces: techniques and applications, *ACM SIGGRAPH 2007 courses*, 2007.
- [5] L. Shapira, A. Shamir, D. Cohen-Or, Consistent mesh partitioning and skeletonisation using the shape diameter function, *Visual Computing* 24 (4) (2008) 249259.
- [6] E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, *Computer Aided Design* (10) (1978) 350–355.
- [7] O. Karpenko, J. Hughes, R. Raskar, Smoothsketch: 3d free-form shapes from complex sketches, *Proceedings of ACM SIGGRAPH2006* 30 (3) (2006) 589597.
- [8] F. Cordier, H. Seo, Free-form sketching of self-occluding objects, *IEEE Computer Graphics and Applications* 27 (1) (2007) 5059.
- [9] J. F. H. R. C. Zeleznik, K. P. Herndon, Sketch: An interface for sketching 3d scenes, in: Proceedings SIGGRAPH 96, 1996, pp. 163–170.
- [10] T. Igarashi, J. Hughes, A suggestive interface for 3d drawing, in: Proceedings of UIST'01, 2001, pp. 173–181.
- [11] Google, Sketchup software, www.sketchup.com (2010).
- [12] M. Okabe, S. Owada, T. Igarashi, Interactive design of botanical trees using free hand sketches and example-based editing, *Computer Graphics Forum (Eurographics05)* 24 (3) (2005) 487496.
- [13] E. Turquin, M.-P. Cani, J. F. Hughes, Sketching garments for virtual characters.
- [14] A. C. Öztireli, U. Uyumaz, T. Popa, A. Sheffer, M. Gross, 3d modeling with a symmetric sketch, in: Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, 2011, pp. 23–30.
- [15] S. Morigi, M. Rucci, Reconstructing surfaces from sketched 3d irregular curve networks, in: Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, 2011, pp. 39–46.
- [16] G. Orbay, L. B. Kara, Sketch-based modeling of smooth surfaces using adaptive curve networks, in: Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, 2011, pp. 71–78.
- [17] C. S. Applegate, S. D. Laycock, A. M. Day, A sketch-based system for highway design, in: Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '11, 2011, pp. 55–62.
- [18] M.-P. Cani, Sketching terrain, Technical Report, LJK/EVASION, www-evasion.imag.fr/Positions/sujets/masterSketchTerrain2008.pdf (2008).
- [19] H. Choi, S. Choi, H. Moon, Mathematical theory of medial axis transform, *Pacific Journal of Mathematics* 181 (1) (1997) 5788.
- [20] T. Igarashi, J. Hughes, Smooth meshes for sketch-based freeform modeling, in: Proceedings of Symposium on Interactive 3D Graphics, 2003, p. 139142.
- [21] A. Nasri, W. B. Karam, F. Samavati, Sketch-based subdivision models, in: Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM '09, 2009, pp. 53–60.
- [22] A. Alexe, L. Barthe, M.-P. Cani, Shape modeling by sketching using convolution surfaces, in: Proceedings of Pacific Graphics, 2005.
- [23] E. V. Brazil, I. Macedo, M. C. Sousa, L. H. de Figueiredo, L. Velho, Sketching variational hermite-rbf implicits, in: Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium, SBIM '10, 2010, pp. 1–8.
- [24] R. Schmidt, B. Wyvill, C. M. Sousa, J. Jorge, Shapeshop: Sketch-based solid modeling with blobtrees (2005) 53–62.
- [25] M. P. C. L. B. A. Bernhardt, A. Pihuit, Matisse: Painting 2d regions for modeling free-form shapes, in: EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, (2008) C. Alvarado and M.-P. Cani (Editors), 2008.
- [26] O. Karpenko, J. Hughes, R. Raskar, Free-form sketching with variational implicit surfaces, *Proceedings of Eurographics2002* 21 (3) (2002) 585594.
- [27] R. Schmidt, Interactive modeling with implicit surfaces, Thesis, Department of Computer Science, University of Calgary, Canada (2006).
- [28] R. Schmidt, P. J. T. Isenberg, K. Singh, B. Wyvill, Sketching, scaffolding, and inking: A visual history for interactive 3d modeling, *Proceedings of NPAR 2007*.
- [29] R. Schmidt, K. Singh, R. Balakrishnan, Sketching and composing widgets for 3d manipulation, *Proceedings of EUROGRAPHICS 2008*, Editors: G. Drettakis and R. Scopigno 27 (2).
- [30] E. Akleman, J. Chen, Practical polygonal mesh modeling with discrete gauss-bonnet theorem, in: Proceedings of Geometry Modeling and Processing (GMP'06), 2006, pp. 287–298.
- [31] E. W. Weisstein, Gaussian Curvature, From MathWorld—A Wolfram Web Resource, <http://mathworld.wolfram.com/GaussianCurvature.html>, 2011.
- [32] M. Mantyla, An Introduction to Solid Modeling, Computer Science Press, Rockville, Ma., 1988.
- [33] S. Lee, J. Buscema, How to Draw Comics the Marvel Way, Simon & Schuster, Inc., New York, 1984.
- [34] S. McCloud, Understanding Comics: The Invisible Art, Collins Publishers, Inc., New York, 1994.