

# Laplacian-based Design: Sketching 3D Shapes



Bin Sheng and Enhua Wu

**Abstract**— The sketch-based shape modeling is one of the most challenging and active problems in computer graphics. In this paper, we present an interactive modeling system for generating free-form surfaces using a 2D sketch interface. Since inferring 3D shape from 2D sketches is an one to many function with no unique solution, we propose to interpret the given 2D curve to be the projection of the 3D curve that has minimum curvature among all the candidates in 3D. In this way, firstly, we present an algorithm to efficiently find a close approximation of this minimum curvature 3D space curve. In the second step, our system could identify the 3D surfaces automatically, and then we apply Delaunay triangulation on these surfaces. Finally, the shape of the triangular surface mesh that follows the 3D profile curves is computed using harmonic interpolation by solving Laplacian equations. We present experimental results on various kinds of drawings by the interactive modeler.

**Index Terms**— Curvature, Interpolation, Free-form surfaces, Sketch.

## I. INTRODUCTION

Nowadays, most designers still prefer to express their creative design idea through 2D sketches. Obviously, freehand sketching, the informal drawing of shapes using free-form lines and curves, has remained one of often the most powerful and most natural ways of conveying information about the shape, composition and relationships of an object's components. However, space curve sketching using 2D user interfaces has a challenging and active research problem of interpreting 2D sketches of curves and surfaces as a 3D entity. A sketch is a projection of a 3D object onto an arbitrary 2D plane. Reconstruction is the inverse process of such projection. Obviously, the projection from 2D to 3D is mathematically indeterminate. It is well known that the fundamental difficulty is in choosing the appropriate one.

Anyway, most of the sketch-based modeling tools try to avoid addressing this fundamental problem directly due to its complexity. These systems provide a fixed canvas to the user to draw the 2D curve, thus fixing the 3D curve to be on the canvas.

They provide tools to change the shape of the canvas to enable the user to draw non-planar curves. One of the clever uses of this canvas method was by [1], where the system provides a

planar canvas first and the user starts with a planar closed curve. Often, curves that are not drawn on the canvas are interpreted as gestural inputs for viewpoint or object manipulation. In spite of all the successes that have been achieved in the field of sketch-based modeling, the fundamental problem of space curve determination still remains to be a challenging one.

In this paper, we attempt to provide a physically sound and mathematically elegant method of 3D space curve determination from a 2D curve. We propose that the 3D curve corresponding to the given 2D curve is the one that minimizes the maximum normal curvature in 3D. In addition, we propose an FEM-based sketch modeling method for general unstructured free-form hand-drawings.

We sketch the network of planar and non-planar input curves and specify the patch connectivity. Our system automatically generates a free-form surface that fits the network of curves. Once an initial 3D model, based on the input sketch is presented, the user could modify it to get the desired shape. The 3D surface shape is generated in the form of a 3D parametric surface, which is obtained by transforming the 2D plane into 3D surfaces and automatically interpolating the mesh vertices of the 2D triangular mesh into 3D. (See Fig. 1).

Primarily, there are four main contributions in this paper as follows:

- 1) We prove the existence of the projection from 2D curves to 3D curves using curvature minimizing, and developed the technique to account for a profile curve to infer a 3D curve.
- 2) We generate smooth free-form surfaces with minimal user input, and adopt a spanning tree algorithm for fast surface patch identification.
- 3) Given the articulated sketches, we present an effective algorithm to generated shape of the surface mesh using a finite element method (FEM). We use harmonic interpolation by solving Laplacian equation.
- 4) We developed an interactive modeling system to test various drawings. It is proved that the system can create many reasonable shapes from the profile and silhouette sketches

The paper is organized as follows. After a review to the related research in Section II, we provide an intuitive and mathematically elegant solution to 3D space curve determination from a 2D curve in Section 3. The proposed free-form surface generation is presented together with both boundary identification and internal interpolation in Section 4. Then some models created with the new method are detailed in Section 5. Finally, we draw the conclusions and discuss

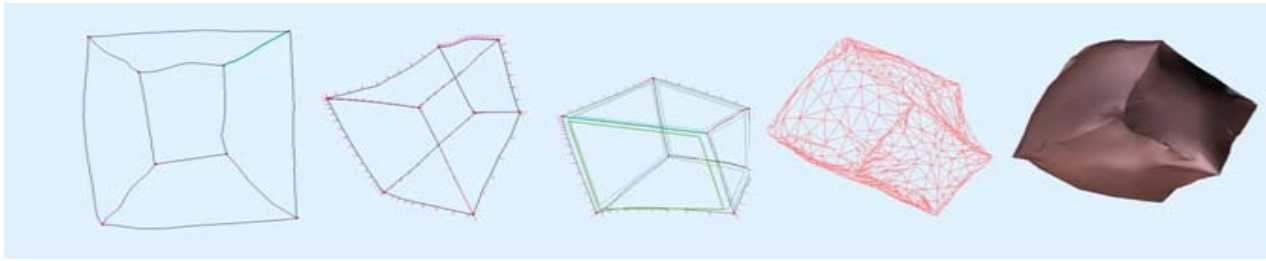


Fig. 1. The user draws the 3D shape from 2D sketches (from left to right): (a) User draws the profiles on the 2D interface. (b) Our program infers the 3D curve from the 2D sketches, including the normal information. (c) The surface patches are identified. (d) The triangular mesh is generated by solving Laplacian equations. (e) The created model is texture-mapped.

possible future work.

## II. RELATED WORK

Many techniques are used by artists in order to suggest the object's shape such as characteristic lines or shaded areas. Since drawing is a familiar task for a lot of people, 'sketching' has been introduced as a natural alternative for 3D modeling tasks. We can find that the existing sketching approaches can be divided into three categories: the line-and-stroke approach, the painting-or-shading approach, and the curve-and-gesture approach. The approach proposed here belongs to the first category.

The principle of the line-and-stroke approach is to infer a 3D volume from characteristic lines drawn by a user. When ambiguities occur, they can be removed by user's selection of the correct model into a list of possible reconstructions. In these cases, spline curves are usually used as described in [2, 3]. Modifying existing curves by over-sketching was initially presented by [4]. This approach has been adopted for curve modification and manipulation by [5, 6]. [5] used the correlation between the curve and its shadow as a visual cue to help the user compute and modify the shape of the 3D curve. [7] take a multi-view sketching approach using epipolar lines as cues to deduce shape information. The epipolar geometry approach works for simple cases but fails to give desirable results for more complex objects.

In the recent past, there has been some research devoted to extract depth information from 2D curves and strokes directly. Most of the existing literature on space curve generation relies on the domain specific knowledge. The edge-vertex graph method has been utilized to extract the depth information in systems best suited for architectural designs and CAD type applications [8,9]. Furthermore, [10] use an interesting domain specific learning based approach using geometric correlation to deduce the 3D model that is again more suitable for CAD tools. The underlying assumption behind this learning technique is that the input sketch comprises of straight lines and thus does not account for free-form curves. It also follows from the paper that the human ability to perceive the 3D information from 2D sketch is based on prior knowledge about the drawn object or scene. However, learning based methods are computationally expensive.

One of the earliest examples of constructing free-form space

curves is the 3-Draw system [11] that extracts the 3D coordinate information by using a tracker-based system. A curvature minimizing approach was followed by [12]. [13] utilize the input sketch to generate a curved canvas where the drawn curve essentially becomes an extruded surface from a given viewpoint providing the user the capability of drawing non-planar curves on that surface. [14] use the approach of oriented projective representations of 2D points, but they do not generate actual 3D values from 2D points.

Recently, some papers have been presented on fitting surfaces to a network of curves. Teddy [1] is one of the exemplary works in the field of free-form surface generation using polygonal representation. [15] on the other hand used a parametric representation of surface using a constraint based design. Volumetric representation is also used effectively by [16]. Furthermore, recent researches on surface generation techniques use implicit surface representation. Notably, [17] obtain an analytical convolution surface from sketched silhouettes. And [18] make clever use of lofting techniques to create subdivision surfaces from curve network.

The harmonic interpolation [19] technique is different from the many previous methods for generating the 3D surface mesh. The technique is automated by solving Laplacian equations. Since the surfaces are assumed free-form and the network of silhouettes is assumed very steady, forming the skeleton. Once the skeleton is determined to have a specific shape, the surface is stretched or shrunk to conform to the skeleton shape. So such shape modeling technique can give a physically plausible 3D shape on the sketches. Our system generates free-form surfaces from sketched network of curves by using a Delaunay triangulation [20] followed by Laplacian interpolation of mesh vertices and surface fairing.

## III. CURVE SKETCHING

For modeling users' drawings, it seems not to be appropriate to generate realistic profiles and extract them without referring to actual topology analysis, because 3D models usually have complicate profiles. To the best of our knowledge, there is no literature on research that performs domain independent mathematical treatise to generate intuitive free-form space curve. Here we present an interpolation method to address this issue by finding the depth of the sketched 2D curve points that approximates the minimum curvature 3D curve. The method discussed here tries to empower the user by simplifying the user

interface, and shortening the system learning time of the user as well as the and overall time to sketch the object.

Firstly, we assume that the depth values of the control points are equally spaced and the closest point on the curve from any control point has the same depth value as the control point. Hence if we identify these key points on the curve that are closest to the control points and space their depth values equally, we will have the first approximation of the space curve. Since we do not have these 3D Bezier curve control points and hence of its 2D projection, identifying even the correct number of key points is difficult. However we can find that the number of critical points in the curve is the lower bound on the number of control points of the hypothetical Bezier representation of the same curve. Hence the fundamental source of approximation in our algorithm for curvature minimizing depth interpolation comes from the fact that we assume that these critical points are key points.

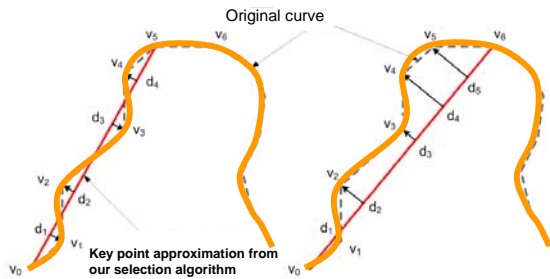


Fig.2 Illustration of our approximation algorithm to select the key points on original curve.

The second source of approximation comes from the fact that, there need not be a critical point corresponding to every control point. Hence all the required control points cannot be found by just one sequence of critical points. Differing from the previous approaches which hierarchically subdivide the curves, we solve this issue of selecting the key points, by simplifying the original 2D curve from the first point location. In practice, the hieratical method usually obtains the slow convergence, because that it takes much iteration to capture the key points, due to the local smoothness of the 2D curve. On the other hand, if we assume the starting point could be the key point, the alterative solution comes naturally and effectively. As shown in Fig. 2, the basic idea of simplification is eliminating intermediate nodes in a straight boundary segment except two end nodes. Based on this idea, the algorithm inherently includes a routine for simplifying a curved boundary exactly made of many small segments. This is an artifact caused by the digitalizing the user's drawing.

In this case, the key points are extracted by the curve simplification. And the specific algorithm proposed here is based on the concept that newly simplified boundary segments are within a preset tolerance from original curve segments from 2D user's drawing. We chose one deviation tolerance scheme that limiting a maximum error between newly generated simplifying stroke segments and original stroke segments in the simplifying process. For example, as illustrated in Fig. 2,

through the boundary simplification, a simplified boundary segment from  $V_0$  to  $V_5$  because the difference between the intermediate points and the new line segment is limited to the preset tolerance. The next new line segment from  $V_0$  to  $V_6$ , on the other hand, does not satisfy this requirement. Now  $V_0 V_5$  becomes a new simplified boundary segment and the algorithm selects the  $V_0$  and  $V_5$  to be the key points.

In our implementation, the resulting 3D space curve, though a very close approximation of the curvature minimizing curve,

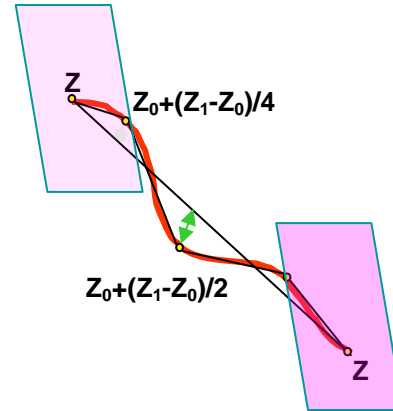


Fig.3 Reconstruction of 3D space curves using our approximate (interpolation) method for different depth values.

has non-smooth, high frequency transitions of the depth values across the critical points. We use the Laplacian filter to obtain a final smooth space curve as it is extremely fast and simple as required by interactive sketching applications. However, it is a known fact that this filter produces shrinkage effect for geometric primitives. In our algorithm, the shrinking of curves is prevented by keeping the depth at the curve end points constant. We apply a few iterations of this smoothening filter only to the interpolated depth values of all the interior points on the curve, as follows:

$$z_i' = z_i + \lambda(\Delta z)$$

where

$$\Delta z = ((z_{i-1} + z_{i+1}) / 2) - z_i, \lambda < 1.0 \quad (1)$$

In this way, the system produces an excellent approximation of the curvature minimizing curve. (See Fig. 3).

From our test, this simplification routine can smooth an outline as well as reduce the number of the points lying on the curve. According to our experimental results, the algorithm can simplify the curve quantitatively but remain the shape well. Hence in this process of selecting the key points, more than 95% of the original points are reduced, the remaining can be considered as the key points safely. Since we assume that all these critical points are representatives of key points, we assign equally spaced depth values within the range to the critical points. And the depth of all other points on the 2D curve is computed by linearly interpolating the depth between the critical points based on the curve length parametrization.

#### IV. SURFACE CONSTRUCTION

After transforming the 2D curves into 3D, our system also has the ability to form these curves into a network (graph). In this process, we call the end points of a curve to be the linkage points. And every point on the drawn curves could also be the linkage point, assigned by the users. Hence user could interactively specify the connectivity between the linkage points on the 2D interface. In this way, the curve network could be generated. Afterwards, we use an automatic process of patch identification, where the resulting network is the input. Once the patch boundary connectivity is known, the topological triangular mesh can be constructed, and then we can generate 3D surfaces by using Laplacian interpolation.

##### A. Automatic Surface Recognition and Normal Smoothing

Our goal is to generate the 3D free-form surfaces from a network of curves. In our current implementation, once the user sketches the desired shape of the object in the form of outline curves, the system should have some "intelligence" to help the user specify the connectivity of the face forming the surface.

Theoretically speaking, the patch recognition is a cycle-search work in the undirected graph represented by the network of curves. However, in practice, only some of the cycles in the graph are proper to form the boundary of the surfaces. In order to find the selected surface quickly, we apply the breadth-first spanning tree algorithm starting at the vertex selected by the user. In this way, all the simple cycles can be determined based on the spanning tree. And if there are multiple simple cycles found in the graph, the user can further extend the path starting from the root of the spanning tree. And the surface can be recognized eventually when the system can find a single simple circle.

In addition, normal computation and interpolation are required on the boundary vertices, in order to estimate detailed shape information. Usually, the corner vertices in the input closed boundary curve sequence have at least degree three. Hence a normal vector with a stable local coordinate system can be computed at these corner vertices. For the vertices whose degrees are one or two, we can set the normal manually for them. Then these coordinate systems are interpolated using quaternions at other intermediate vertices on the boundary curve. We use the publicly available spherical linear interpolation [21] for this purpose. In order to reduce the noise in normal estimation, the input segments which suffer from sharp transitions were smoothed.

##### B. Triangulation and Harmonic Interpolation

After we obtain all the surface patches and their boundary curves, the generation of the meshes for the 3D profiles has the following two steps:

- 1) **Topological Triangulation**— The input to this step is a closed sequence of connected curves. The output is a topology of the triangulation (with unrefined geometric coordinates for the vertices) that consists of input vertices on the curve and the newly introduced vertices in the interior of the patch.

- 2) **3D Interpolation of Surface Mesh**— The triangular mesh is automatically extended to the shape that represents the freeform surface by harmonic interpolation. Afterwards, mesh refining is processed to reduce the geometrical noise in generated mesh.

Specifically, in step 1, the boundary curves of the surface patches that are created in previous process are projected onto a plane that best approximates the curve points, which is a PCA method. Hence our triangulation method is applied on 2D domain, generally based on the *Delaunay triangulation*, where a triangle is created by joining nearest neighboring nodes as its edges. Since all the vertices having been projected onto the plane, such triangulation actually is a 2D Delaunay triangulation, called *topological triangulation*. Here, the input for mesh generation is a Planar Straight Line Graph (PSLG), including the vertices of the boundary curves projected onto the plane. A quality mesh generation program computes a constrained Delaunay triangulation that is a refinement of the input PSLG. You can see [22] for descriptions of this computation. The modeler uses the freely distributed program Triangle by [23] for this task. As a result, the topological triangular mesh that is suitably fine for modeling the surface is obtained.

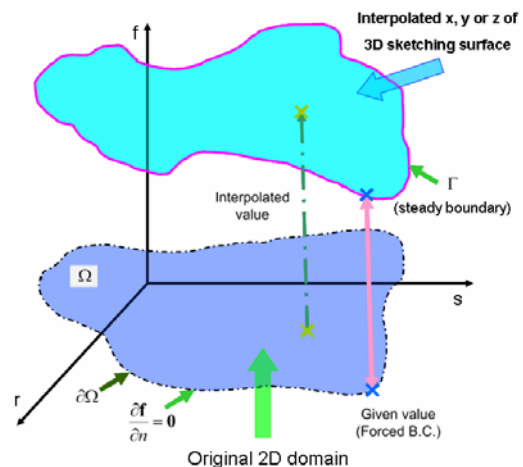


Fig.4. Interpolation of 3D surface mesh

On the other hand, the nodes in the meshes which are obtained by triangulating the surface domain with the boundary curve constraints in 2D are interpolated to give a 3D shape by using the transformed profiles in 3D. The surface mesh vertices are interpolated to conform to the profiles transformed in 3D using Laplace interpolation. We seek to define the parametric surface by the function

$$f(r, s) = (x(r, s), y(r, s), z(r, s))$$

representing the 3D surface mesh shape, where  $r$  and  $s$  are the parameters for a position in the 2D plane domain  $\Omega$ . We now specify  $\partial\Omega$  be the outline of the patches projected onto the 2D plane, and let  $\Gamma$  be the original 3D surface value profiles which is interactively modeled and automatically captured before. There are two kinds of boundary conditions applied to the domain  $\Omega$ , the one is a *Neumann* boundary condition defined at  $\partial\Omega$



because  $\partial\Omega$  is considered as free, and the other is a *Dirichlet* boundary condition defined at  $\Gamma$ , which are illustrated in Fig.4. Let  $g(r,s)$  be the forced boundary condition at  $\Gamma$ , which is given by the 3D curves generated interactively as discussed previously. The interpolation problem is mathematically defined as follows.

$$\nabla^2 f = \frac{\partial^2 f}{\partial r^2} + \frac{\partial^2 f}{\partial s^2} = 0 \quad \text{for } (r,s) \in \Omega \quad (2-1)$$

$$f(r,s) = g(r,s) \quad \text{for } (r,s) \in \Gamma \quad (2-2)$$

$$\frac{\partial f}{\partial n} = 0 \quad \text{for } (r,s) \in \partial\Omega \quad (2-3)$$

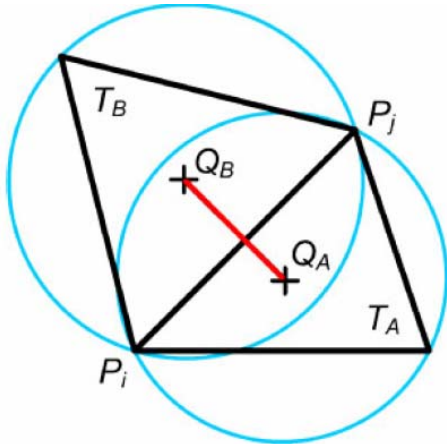


Fig. 5 Neighboring triangles of a Delaunay edge

Functions that satisfy Laplace's equation,  $\nabla F = 0$ , are called *harmonic functions*, which is usually interpreted as partial differential equations. The function  $f$  specified by Equations (2) can be viewed as 3 harmonic functions defined in  $\Omega$  that interpolates  $g(r,s)$  on  $\Gamma$ . The application of this mathematical concept to the surfaces with complicated boundaries starts from discretizing the domain into simple geometrical shapes like triangles. The parameters  $r$  and  $s$  represents the coordinates of the vertices in the discretized 2D domain. The coordinates of the transformed profiles in 3D space are treated as the discretized values of  $g$  at  $\Gamma$ . Now the Laplace interpolation problem of the surface mesh is solved by a *finite element method* (FEM) defined for the discretized domain. However, we do not need much of the apparatus of the FEM, because in this simple case, the linear equations for the coordinates,  $x_k$ ,  $y_k$ , and  $z_k$ , can be formed from a single parameter,  $\gamma_{ij}$ , associated with the mesh edge between vertices  $P_i$  and  $P_j$ , as shown in Fig.5. Because we are using a Delaunay triangulation,  $\gamma_{ij}$  has a simple geometric description. For an edge internal to the mesh, let  $T_A$  and  $T_B$  be the mesh triangles that share the common edge between  $P_i$  and  $P_j$ . Let  $Q_A$  and  $Q_B$  be the circumcenters of  $T_A$  and  $T_B$ , then  $\gamma_{ij} = \|Q_A - Q_B\| / \|P_i - P_j\|$ ; which is discussed in detail by [24].

If the edge  $P_i$  to  $P_j$  is on the boundary of the surface, then

$T_A$  has no neighbor  $T_B$ . In this case,  $Q_B$  is the midpoint of edge  $P_i$  to  $P_j$  and  $T_A$  must be acute so that  $Q_A$  stays in  $T_A$ . This restriction on the triangles at the 3D surface boundary basically requires the mesh to be adequately fine and well shaped near the boundary. To compute the discretized coordinate function,  $x(r,s)$ , the FEM method specifies a system of linear equations of the form  $Ax = b$ , where  $x_k$  is the value of the coordinate function at mesh vertex  $P_k$ . In Equation (3), we give the form of the equation associated with a mesh vertex,  $P_i$ , not on the surface skeleton, that is connect to  $m$  neighboring mesh vertices,  $P_k$ ,  $k = 1$  to  $m$  by mesh edges.

$$\left(\sum_m^{k=1} \gamma_{i,j_k}\right)x_i - \sum_m^{k=1} \gamma_{i,j_k} x_{j_k} = 0 \quad (3)$$

If  $P_i = (r_i, s_i)$  lies on the surface profiles, then the equation is simply  $x_i = x_{sk}(r_i, s_i)$ . This is a system of linear equations with sparse matrix,  $A$ , which can be solved by most sparse matrix techniques or software. A system of equations with the same matrix, but different right hand side vectors is also specified for the discretized coordinate functions  $y(r,s)$  and  $z(r,s)$ . Efficiencies in computing the harmonic interpolating mesh can be gained by observing that these three systems of equations all have the same matrix of coefficients.

Once topological structure and geometrical interpolation for surface patches is finished, final geometric positions of the new vertices are further computed to make the limit surface fair. Our approach to surface refining provides a large number of internal vertices having degrees from five to eight. It is simple, and has fast convergence. It can handle models with any arbitrary genus and works for convex patches and those with low concavity.

## V. EXPERIMENTAL RESULTS

We have tested our approach generating a number of simple models. Figs.1 and Fig.7-11 show some of the sketch-based models accursed using our interactive method. Most of the test cases are used with normally six to ten curves, producing a similar number of patches.

Table 1 summarizes our results and their performances. And Fig.6 shows the scalability of our algorithm on the size of model. It took an average of about one minute for the users to sketch simple models using our system, and the automatic generation was executed at interactive rates on a 2.4 Ghz Pentium 4 CPU with 1GB RAM.

Our system provides a simple interface where the user can draw lines, rotate the view and continue drawing until the desired shape is achieved. The identification of surface patches is automatically completed, but currently needs the user's indication. The recognition work will be improved to be further self-acting and interactive in future. In order to reduce the effect of the possible outliers in the surface fairing procedure, we damp the migration of vertices, only allowing displacements proportional to the length of surrounding edges. This way the relative movement of vertices is controlled by a

user-defined parameter. The surface generated gives a coarse representation of the object and can be further refined using subdivision surfaces.

TABLE 1: THE COMPUTATION TIMES GENERATED MODELS USED IN THIS PAPER. COMPUTATION TIMES(SECONDS) MEASURES THE TOTAL TIME ON A 2.4GHZ P4 BEGINNING WITH READING THE INPUT SKETCHES AND ENDING WITH WRITING THE TRIANGULAR OUTPUT MESH.

Model	Fig.	#Curves(input)	#Vertices	#Triangles	Time
Quadrangular Frustum	1	12	692	1419	37s
Tetrahedron	7	8	718	1432	41s
House Model	9.11	15	1304	4604	85s
Tree Model	8	20	1874	3740	88s
Leaf Model	10	10	2132	4436	136s

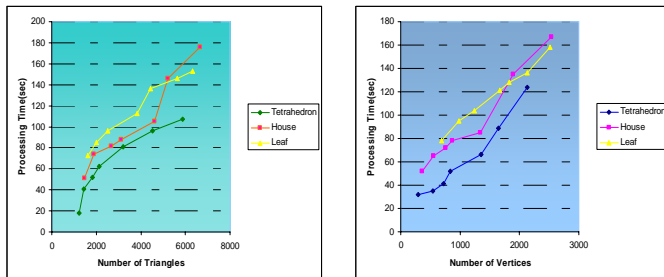


Fig. 6 (Left)The effect of the number of the vertices on the running time (seconds). (Right)The effect of the number of the triangles on the running time(seconds).

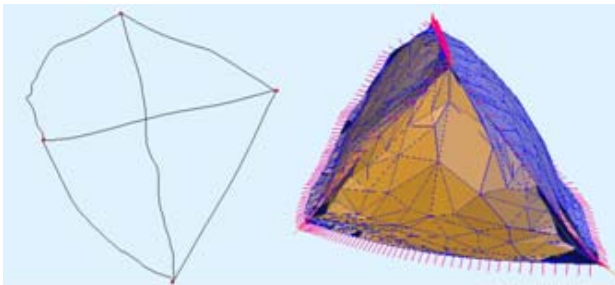


Fig. 7 One tetrahedron model generated by our system with the input sketches. See Color Plate 20.

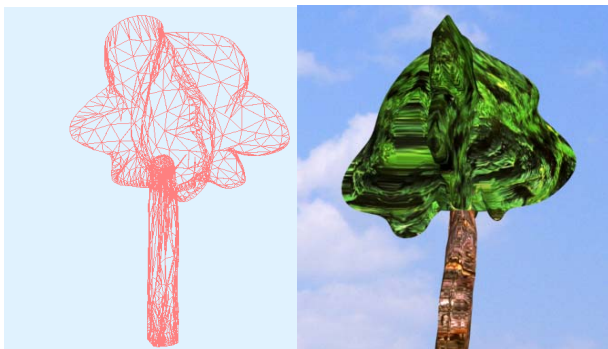


Fig. 8 Our system creates the mesh model for tree design (left). The textured tree model is rendered in the virtual environment (right). See Color Plate 21

## VI. CONCLUSIONS AND FUTURE WORKS

In this paper we have addressed the problem of generating a free-form surface from a network of curves which are in turn generated using 2D sketches. The 3D space curve generation technique uses a single view information about a 2D curve to find the curvature minimizing 3D curve. We believe that the mathematical rigor used to explain the intuitive space curve generation would find its use in perceptual studies. Compared



Fig. 9 Textured result of the house model corresponds to Fig.11. See Color Plate 22

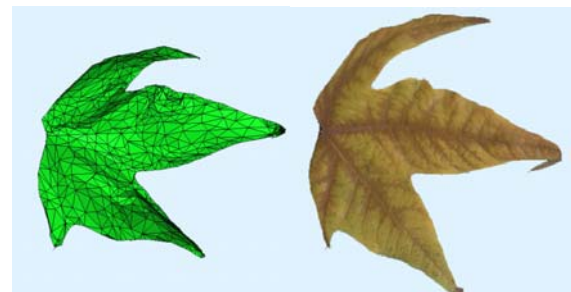


Fig.10 Triangular mesh constructed for leaf model design(left). The generated leaf surface is texture-mapped (right). See Color Plate 23.

with the previous sketch-based approaches relying on the iterative optimization strategies which frequently fell into local minima or mangled the surface mesh, our 3D surface generation using the network of curves is simple but efficient that is required for interactive applications like sketch-based modeling. In particular, we have also explored the physically-based method (FEM) to utilize the Laplacian interpolation to shape the mesh from the silhouette and profile curves. Since generated surface might not be the final surface that the user intends to keep, once the initial surface is generated, users can modify the shape of the model interactively, with locally estimating the normal and position for each mesh vertex, thus making the system analogous to a virtual sculpting tool.

Planned future work in our sketch-based free-form modeling system includes more intelligent and automatic shape identification for surface generation, e.g., currently we are trying to use heuristics to determine the more plausible shape to be generated. In addition, future investigation will attempt to improve the surface reconstruction algorithm, including extensions to sketches that cannot be described by a connected graph. And we will design for the soft curves and improve the stroke merging algorithm to increase the performance.

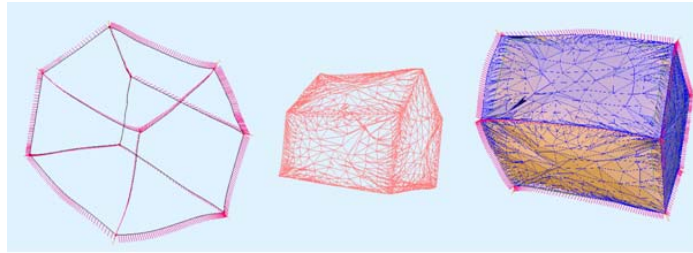


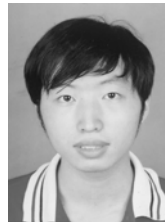
Fig.11 House Model Designing: Network of sketch curves with normal computation (left). The triangular mesh is constructed with Laplacian interpolation (center). The generated model after surface refining (right).

## ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by University of Macau.

## REFERENCES

- [1] T. Igarashi, S. Matsuoka and H. Tanaka. Teddy: A sketching interface for 3d freeform design, in *Proceedings of ACM SIGGRAPH 1999*, pp. 409-416. ACM Press / ACM SIGGRAPH, 1999.
- [2] B. Fowler and R. Bartels. Constraint-based curve manipulation, *IEEE Computer Graphics & Applications*, vol.13, no.5, pp.43-49, 1993.
- [3] C. Grimm and M. Ayers. A framework for synchronized editing of multiple curve representations, in *Proceedings of EUROGRAPHICS 1998*, vol.17, pp. C31-C40, 1998.
- [4] T. Baudel. A mark-based interaction paradigm for free-hand drawing, in *Proceedings of 7th annual ACM Symposium on User Interface Software and Technology*. ACM Press, pp. 124-125.
- [5] J. M. Cohen, L. Marosian, R. C. Zeleznik and J. F. Barzel. An interface for sketching 3d curves, in *Proceedings of the Symposium on Interactive 3D Graphics*. ACM Press, pp. 17-21, 1999.
- [6] T. Fleisch, F. Rechel, P. Santos and A. Barzel. Constraint stroke-based oversketching for 3d curves, in *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBM04)*. Eurographics Association, pp.161-166, 2004.
- [7] O. Karpenko, J. F. Hughes and R. Raskar. Epipolar methods for multi-view sketching, in *Proceedings of the Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBM04)*, Eurographics Association, pp. 167-174, 2004.
- [8] H. Lipson and M. Shpitalni. Identification of faces in a 2d line drawing projection of a wireframe object, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 18, no. 10, pp. 1000-1012, 1996.
- [9] A. Shesh and B. Chen. Smart papercan interactive and easy-to-use sketching system, in *Proceedings of Eurographics 2004*, pp. 406-416, 2004.
- [10] H. Lipson and M. Shpitalni. Correlation-based reconstructions of a 3d object from a single freehand sketch, in *AAAI Spring Symposium on Sketch Understanding (2002)*, pp. 99-104, 2002.
- [11] E. Sachs, A. Roberts and D. Stoops. 3-draw: A tool for designing 3d shapes, *IEEE Computer Graphics & Applications*, vol. 11, no. 6, pp. 18-26, 1991.
- [12] A. Pentland and J. Kuo. The artist at the interface.vision and modeling, in *Tech. Rep.*, no. 114. MIT medialab, 1989.
- [13] T. Ijiri, T. Igarashi, S. Takahashi and E. Shibayama. Sketch interface for 3d modeling of flowers, *Technical Sketch at ACM SIGGRAPH*, August 2004.
- [14] O. Tolba, J. Dorsey, and L. McMillan. Sketching with projective 2d strokes, in *Proceedings of the 12th annual ACM Symposium on User Interface Software and Technology 1999*. ACM Press, pp. 149-157, 1999.
- [15] P. Michalik, D. H. Kim and B. D. Bruderlin. Sketch- and constraint-based design of b-spline surfaces, in *Proceedings of the seventh ACM Symposium on Solid Modeling and Applications 2002*. ACM Press, pp. 297-304, 2002.
- [16] S. Owada, F. Nielsen, K. Nakazawa and T. Igarashi. A sketching interface for modeling the internal structures of 3d shapes, *Lecture Notes in Computer Science (LNCS)*, Springer-Verlag, pp. 49-57, 2003.
- [17] C.-L. Tai, H. Zhang and J. C.-K. Fong. Prototype modeling from sketched silhouettes based on convolution surfaces, *Computer Graphics Forum*, vol. 23, no. 1, pp. 71-83, 2005.
- [18] S. Schaefer, J. Warren and D. Zorin. Lofting curve networks using subdivision surfaces, in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, E. Fiume, Ed. ACM Press, pp. 103-114, 2004.
- [19] N. Sukumar. Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids, *International Journal for Numerical Methods in Engineering*, vol. 57, pp. 1-34, 2003.
- [20] J. R. Shewchuk. Lecture notes on delaunay mesh generation, University of California at Berkeley, CA, 1999.
- [21] K. Shoemaker. Wires: a geometric deformation technique, in *Proceedings of ACM SIGGRAPH 1985*, vol. 19, pp. 245-254, 1985.
- [22] P. L. George and H. Borouchaki. *Delaunay Triangulation and Meshing*, Hermes: Paris, France, 1998.
- [23] J. R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and Delaunay triangulator, in *First Workshop on Applied Computational Geometry*. ACM Press, pp. 124-133, 1996.
- [24] F. W. Letniowski. Three-dimensional delaunay triangulations for finite element approximations to a second-order diffusion operator, *SIAM Journal of Science and Statistical Computation*, vol. 13, pp. 765-772, 1992.



**Bin Sheng** graduated from Huazhong University of Science and Technology in 2004, with a B.A. in English and a B.S. in computer science. He is currently a Master student in computer science at University of Macau. He received the qualification of System Analyst recognized by the Chinese government in 2004. His research interests include image-based modeling and rendering, mesh processing, sketch-based modeling, hardware accelerated rendering and natural appearance modeling.



**Enhua Wu** graduated from Tsinghua University in 1970 with his Bachelor degree, and he received Ph.D. (CS) from University of Manchester (UK) in 1984. Since 1985, he has been the computer graphics researcher at Institute of Software, Chinese Academy of Sciences, and currently serves as the director of the CG group in the State Key Lab of Computer Science. He is also a professor in University of Macau after 1997, and now is the head of computer science department at this university. His research interests include computer graphics, computer-aided design (CAD), virtual reality and graphic hardware techniques. He is a member of IEEE and ACM.