Adaptive Sampling and Mesh Generation

S. Z. Li
School of Electrical and Electronic Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 2263
szli@ntu.ac.sg

ABSTRACT

A principle in surface sampling and mesh generation is that highly curved area should be sampled densely and *vice versa*. This paper presents an approach for automated surface sampling and adaptive mesh generation in accordance with the principle. The approach is self-organizing, forming topology-preserving mesh from random initialization. Mesh spacing versus surface curvedness can be easily controlled by a single parameter in the shape function. Key locations can be prescribed by imposing additional boundary conditions. Experiments are presented.

Keywords: Adaptive mesh, system of nonlinear equations, self-organization, shapes, surface sampling.

Introduction

Representation is one of major issues in CAD/CAM, geometric modeling, computer vision and graphics. There are a variety of representation schemes. Curves and surfaces are frequently used to represent objects. In many applications, these physical entities need be approximated by or sampled into a set of significant points. As the result, a curve is approximated by curve segments and a surface by patches such as polygons. Orderly samples which are related by neighboring relationship form a mesh. Mesh generation is also a problem common in fields which involve finite element computation [1].

It is desirable that given a number of sample points, the shape of the original curve or surface is preserved as much as possible. An intuitive principle is that highly curved area should be sampled densely while flat area sparsely. The spacing between generated mesh points decreases as the local "curvedness" of the surface increases and vice versa, This can be formulated in a more concrete way. For example, Terzopoulos and Vasilescu [2] introduces a dynamic model of nodes and springs for adaptive mesh generation. The model is described by a system of equations of motion. Kosters [3] defines what he calls angle parameterizations for reparameterization of curves and surfaces. His criterion is that the surface normal vary with constant speed along parameter curves. Ahn *et al.* [4] use the Kohonen self-organizing neural network [5] to generate nonuniform mesh. Vemuri *et al.* [6] construct surface reparameterization in terms of arc length along the lines of curvature and asymptotic lines, which is invariant with respect to rigid motion in 3D.

The contribution of this paper is an approach for shape-based surface sampling and mesh generation. Given a scale *shape function*, which describes local "curvedness" of a surface to be processed, and *a number* of sample points, the problem is formulated as solving a system of nonlinear equations. The *solution* to the system represents a set of generated *adaptive mesh* points. The solution can be found using a simple iterative algorithm in which no free parameters are involved. The approach is self-organizing. It produces orderly or topology preserving meshes out from random initialization. It is flexible. Degree of the spacing versus the curvedness can be controlled by a single parameter in the shape function. Important point locations can be prescribed by imposing appropriate boundary conditions. Although the formulation to be introduced is for surfaces, it is general and applicable to curves and data of higher dimensions. Another important feature of the approach is that the corresponding algorithm can be easily implemented using parallel distributed architecture and this may be done in hardware.

In comparison with other work, the present approach has several advantages. The computation is obviously the simplest of all and, unlike [2] and others, involves no free parameters. Therefore it is the most suitable for hardware implementation. It has the same feature of self-organization as the Kohonen network [5] used in [4]. But it has avoided using the stochastic network which is known to be slow in convergence. The deterministic approach here is several order faster. The approach in [3] gives delicate angle parameterizations but may have problems converging (refer to Table 1 of [3]); whereas the present approach empirically always converges with different surface parameters.

The paper is organized as follows: Section 2 formulates the problem of adaptive

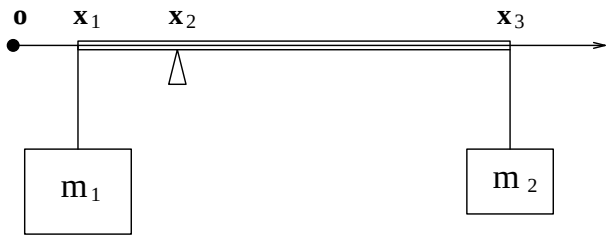

Figure 1: Balance of lever system.

of sampling and mesh generation as solving a system of nonlinear equations. Section 3 presents a simple algorithm for finding the solution. Section 4 demonstrates the approach using experiments. Section 5 concludes the paper.

Formulation

To illustrate the idea of adaptation based on shapes, let us first consider a lever system as in Fig.1. Some weights m_1 and m_3 are hung at the endpoints at coordinates x_1 and x_3 of the lever, respectively. To achieve the balance, the fulcrum point x_2 should be such that the total moment is zero. In the one dimensional case shown in the figure, the following must be satisfied

$$m_1(x_2 - x_1) + m_3(x_2 - x_3) = 0 \tag{1}$$

Obviously, the balanced fulcrum point x_2 is closer to the endpoint at which heavier weight is hung. This principle is applied to our surface sampling.

A 3D surface can be expressed as

$$S = \{\mathbf{p}(u_1, u_2) = \ \mid (u_1, u_2) \in \mathbf{D}^0\} \tag{2}$$

where $\mathbf{p}(u_1, u_2)$ is a point in 3D and $\mathbf{D}^0$ is the (continuous) domain of parameters u_1 and u_2. Generally, a 3D point is $\mathbf{p}(u_1, u_2) = (x(u_1, u_2), y(u_1, u_2), z(u_1, u_2))$. For clarity, the analysis and discussion in this paper is made on a special form where $\mathbf{p}(u_1, u_2) = (u_1, u_2, z(u_1, u_2))$, that is, the graph surface or Monge patch [7]. However, the ideas apply to the general form as well.

Assume that the surface is to be sampled into a number of M (discrete) points. Denote the set of subscripts of the points by

$$\mathbf{D} = \{1, ..., M\} \tag{3}$$

Let

$$\mathbf{v} = (u_1, u_2)^T \tag{4}$$

be a vector of the parameters such that $\mathbf{p}(u_1, u_2) = \mathbf{p}(\mathbf{v})$. Let $r(\mathbf{v}) > 0$ be a scale *shape function* reflecting "curvedness" of the surface. The function is generally defined upon the derivatives or curvatures derived from the surface function (see later an example of definition). It is a measure of local surface shape and provides

quantitative information on which the sampling is based. Now the problem is to determine a set of discrete points

$$\mathbf{V} = \{\mathbf{v}_i \mid i \in \mathbf{D}\} \tag{5}$$

more exactly, their locations in $\mathbf{D}^0$, according to $r(\mathbf{v})$, such that the *given number* of M sample points $\mathbf{p}(\mathbf{v}_i)$ best represent the surface S. Define a neighborhood system $\mathbf{N} = \{N_i \mid i \in \mathbf{D}\}$, with N_i being the set of indices of neighbors of point $\mathbf{v}_i$, where $i \notin N_i$ and $j \in N_i$ iff $i \in N_j$. For instance, N_i can be defined as the set of its four or eight neighboring points in a grid. However, inhomogeneous neighborhood systems in which the number of neighbors may vary from points to points are allowed.

In a way analogous to (1), the optimal locations $\mathbf{v}_i$ of the sample points satisfies the following system of nonlinear equations

$$\sum_{j \in N_i} r(\mathbf{v}_j)(\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \qquad \forall i \tag{6}$$

being nonlinear since the shape function r is nonlinear in general. As the result, sample points are placed densely where the value of $r(\cdot)$ is high, meaning the surface is highly curved, and vice versa. Solution $\mathbf{V}$ that satisfies the above system is a equilibrium. The solution can be found using a very simple algorithm (see later). Yet, the algorithm yields nice results as will be seen in experiments.

Given a surface of the form $S = \{\mathbf{p}(u_1, u_2) = (u_1, u_2, z(u_1, u_2)\}$, a curvedness measure of the surface can be defined, for example, as follows

$$\kappa(u_1, u_2) = \frac{\sqrt{z_{u_1u_1}^2 + z_{u_2u_2}^2}}{(1 + z_{u_1}^2 + z_{u_2}^2)^{3/2}} \tag{7}$$

where z_u and z_{uu} are, respectively, the first and second partial derivatives of z. It is an analogy to the curvature magnitude for planar curves. We normalize the measure so that its value fall into the range $[0, 1]$ and add to it a parameter $q > 0$, yielding a shape function

$$r(\mathbf{v}) = r(u_1, u_2) = q + \frac{\kappa(u_1, u_2) - \min_{u_1,u_2} \kappa(u_1, u_2)}{\max_{u_1,u_2} \kappa(u_1, u_2) - \min_{u_1,u_2} \kappa(u_1, u_2)} \tag{8}$$

The parameter q controls the *contrast* of $r(u_1, u_2)$ on which the result of shape-based sampling is dependent (see experiments).

The measure defined in (7) is intuitive, simple to compute and yet producing satisfactory results in our experiments. However, it is not a precise characterization of the curvedness, especially when invariance is concerned. Nonetheless, the algorithm to be presented for computing the optimal mesh remains the same with different choices of shape measures. Therefore, one may want to use other alternatives which are less simple but have better analytic properties. For example, there are two interesting surface shape functions, namely shape index and curvedness [8]. The latter is defined as $\sqrt{(\kappa_1^2 + \kappa_2^2)/2}$ where κ_1 and κ_2 are the two principal curvatures. We believe this definition of curvedness will also yield good result though we have not experimented with it. A disadvantage of that definition is the need to compute the principal curvatures.

In the discussions hitherto, the problem is formulated in the parameter domain. It may be more reasonable to formulate the problem in the space S of the orginal surface. In this way, one may seek $\{\mathbf{p}_i\}$ over the surface such that the following system is satisfied

$$\sum_{j\in N_i} r(\mathbf{p}_j)\mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j) = 0 \tag{9}$$

where $\mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j)$ is the projection of $\mathbf{p}_i - \mathbf{p}_j$ onto the plane tangent to the surface point $\mathbf{p}_i$. However, solving the above, one may need an algorithm of more complexity than the previous one because the projection is involved.

Equation (6) approximately minimizes

$$E(\mathbf{V}) = \frac{1}{2}\sum_i \sum_{j\in N_i} r(\mathbf{v}_j)\|\mathbf{v}_i - \mathbf{v}_j\|^2 \tag{10}$$

where $\|\mathbf{v}_i - \mathbf{v}_j\|^2 = (u_{1,i} - u_{1,j})^2 + (u_{2,i} - u_{2,j})^2$ (refer to (4)). To minimize the above, the following has to be satisfied

$$\frac{\partial E(u_1, u_2)}{\partial u_{k,i}} = 0 \qquad k = 1, 2 \quad \forall i \in \mathbf{D} \tag{11}$$

or

$$\sum_{j\in N_i} [(r(\mathbf{v}_i)(\mathbf{v}_i - \mathbf{v}_j) + \mathbf{s}(\mathbf{v}_i)\|\mathbf{v}_i - \mathbf{v}_j\|^2] = \mathbf{0} \tag{12}$$

where $\mathbf{s}(\mathbf{v}_i) = [r'(u_{1,i}), r'(u_{2,i})]^T$ is a vector. The above equation is reduced to (6) by assuming $r'(u_{k,i}) \approx 0$ $(k = 1, 2)$. The assumption implies the shape function is nearly constant in the neighborhood. It holds if the surface is flat enough or if the number M of sample points is large enough.

If the term $\|\mathbf{v}_i - \mathbf{v}_j\|^2$ in (10) is substituted by $\|\mathbf{p}_i - \mathbf{p}_j\|^2 = \|\mathbf{v}_i - \mathbf{v}_j\|^2 + \|z(\mathbf{v}_i) - z(\mathbf{v}_j)\|^2$, where $\mathbf{p}_i = \mathbf{p}(\mathbf{v}_i) = (\mathbf{v}_i, z(\mathbf{v}_i))$, a modified criterion is defined

$$E_1(\mathbf{V}) = \sum_i \sum_{j\in N_i} r(\mathbf{v}_j)\|\mathbf{p}_i - \mathbf{p}_j\|^2 \tag{13}$$

Obviously, the quantity $\|\mathbf{p}_i - \mathbf{p}_j\|^2$ is better than $\|\mathbf{v}_i - \mathbf{v}_j\|^2$ in achieving invariance in the case of graph surfaces. If $r(\mathbf{v}_j)$ is replaced by $r(\mathbf{p}_j)$, the following results

$$E_2(\{\mathbf{p}_i\}) = \sum_i \sum_{j\in N_i} r(\mathbf{p}_j)\|\mathbf{p}_i - \mathbf{p}_j\|^2 \tag{14}$$

Minimizing the above over $\mathbf{p}_i \in S$ relates to solving Equ.(9).

Algorithm

The system of nonlinear equations (6) can be solved using a very simple algorithm. Rearranging the equations yields

$$\mathbf{v}_i = \sum_{j\in N_i} r(\mathbf{v}_j)\mathbf{v}_j / \sum_{j\in N_i} r(\mathbf{v}_j) \qquad \forall i \tag{15}$$

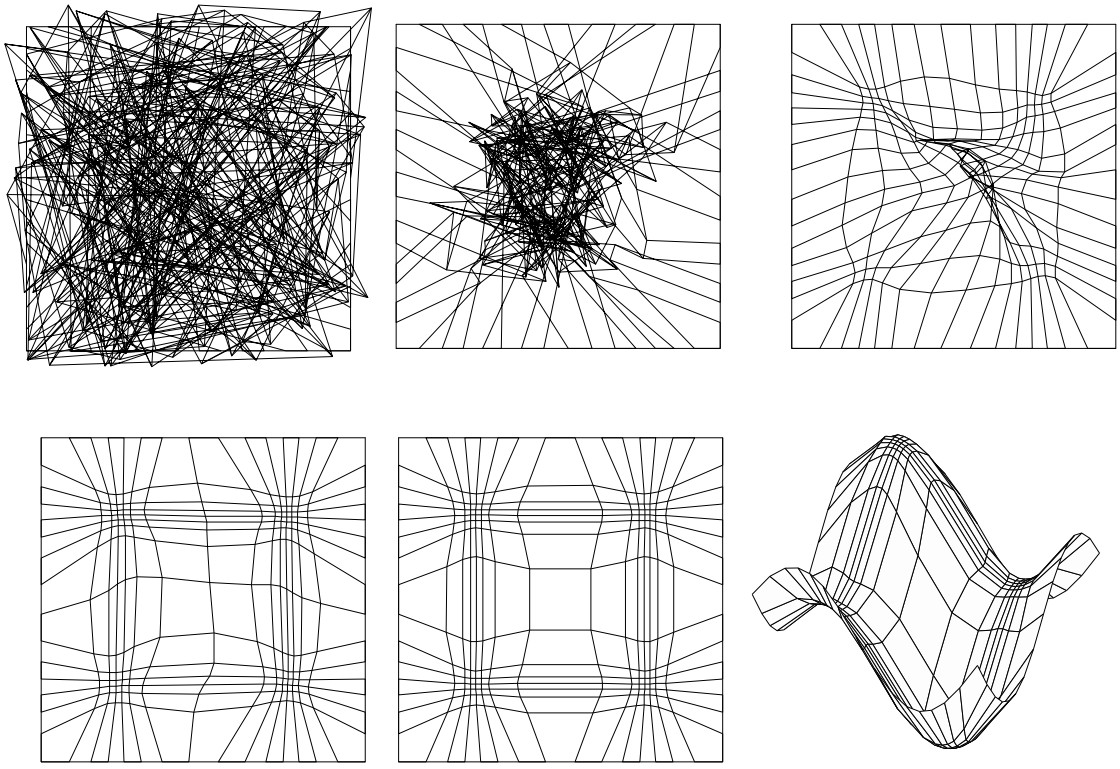

Figure 2: Orderly mesh is formed from randomly initialization. The surface is of bi-sine: $\mathbf{p}(u_1, u_2) = (u_1, u_2, 0.35[\sin(2\pi u_1) + \sin(2\pi u_2)])$. From left to right, top to bottom are the initial random mesh and the results after 2, 5, 20 and 200 iterations, respectively. The last is the plot of the surface with the resulting mesh after 200 iterations. Parameter $q = 0.3$.

which is in the form $\mathbf{V} = g(\mathbf{V})$. In this form, $\mathbf{V}$ is a fixed point of the map $g(\mathbf{V})$ and the solution for $\mathbf{V}$ can be computed using the following simple iterative algorithm (see p.688 of [9])

$$\mathbf{v}_i^{(t+1)} = \sum_{j \in N_i} r(\mathbf{v}_j^{(t)}) \mathbf{v}_j^{(t)} / \sum_{j \in N_i} r(\mathbf{v}_j^{(t)}) \qquad \forall i \tag{16}$$

together with some imposed conditions on the positions of boundary points. The criterion for convergence is

$$|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}| < \epsilon \tag{17}$$

where $|\cdot|$ is a norm and ϵ is a small number. Internal points are allowed to move freely. Boundary conditions are subscribed, for example, as follows: Firstly, consider the boundaries of the parameter domain $\mathbf{D}^0$. The four corner points are fixed all time; other boundary points are allowed to move along the boundaries but not in- or outward. Secondly, locations of feature points such as some discontinuities, extrema and zero-crossings in the interior of $\mathbf{D}^0$ can be fixed as interior boundaries. The algorithm can easily cope with such prescriptions (see an experiment).

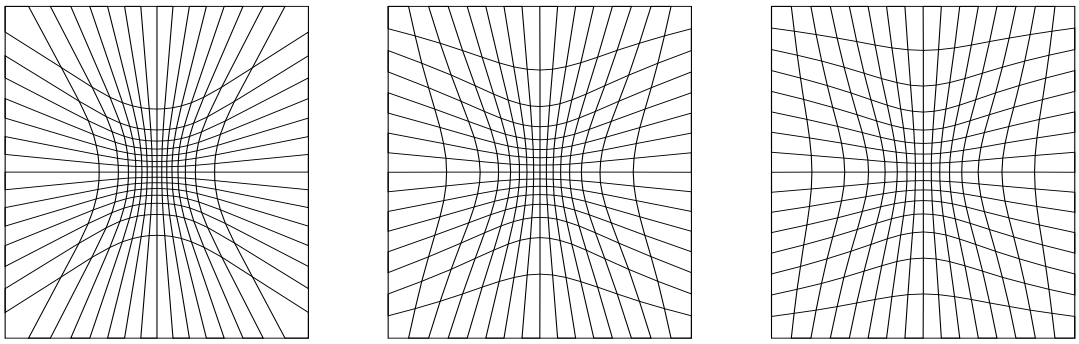

Figure 3: Effect of the shape parameter q on resulting mesh. The surface is parabolic: $\mathbf{p}(u_1, u_2) = (u_1, u_2, -3[(u_1 - 0.5)^2 + (u_2 - 0.5)^2])$. The parameter is set to $q = 0.1, 0.2$ and 0.3 (from left to right), respectively.

The algorithm has a feature of self-organization [5]. Starting with randomly initialized point positions $\mathbf{v}_i^{(0)}$, it automatically develops topology-preserving, orderly mesh (see an experiment). On the other hand, regarding shape-based sampling, the mesh points are spaced according to shape information described by $r(\mathbf{v})$. A further advantage of the algorithm is that there involves no free parameters that need to be adjusted in the above iterative algorithm to yield a meaningful solution (the parameter q is in the given shape function).

Equation (16) consists of operations within neighboring points. Looking more closely, one can find that in each interaction, the value of $\mathbf{v}_i$ is replaced by the weighted sum of its neighboring values

$$\mathbf{v}_i^{(t+1)} = \sum_{j \in N_i} w_j^{(t)} \mathbf{v}_j^{(t)} \tag{18}$$

where $w_j^{(t)} = r(\mathbf{v}_j^{(t)}) / \sum_{j \in N_i} r(\mathbf{v}_j^{(t)})$. This is obviously parallelizable.

Experiments

The purposes of the following three experiments are aimed to demonstrate, respectively: (1) the self-organizing, topology-preserving aspect of the algorithm as well as shape-based sampling; (2) how degree of mesh spacing vs. the curvedness can be controlled by the parameter q in Equation (8) and (3) examples with prescribed locations of interior boundary points. Surfaces[1] in these experiments are in the form $S = \{\mathbf{p}(u_1, u_2) = (u_1, u_2, z(u_1, u_2) \mid (u_1, u_2) \in \mathbf{D}^0\}$ where $\mathbf{D}^0 = \{u_1, u_2\}$ is a square. They are to be sampled into a number of $M = m \times m$ points, so in this case $\mathbf{D} = \{(1, 1), (1, 2), ..., (m, m)\}$. The definition of the shape function in Equations (7) and (8) is used.

[1]The three surfaces for the first three experiments here are those used in [3]

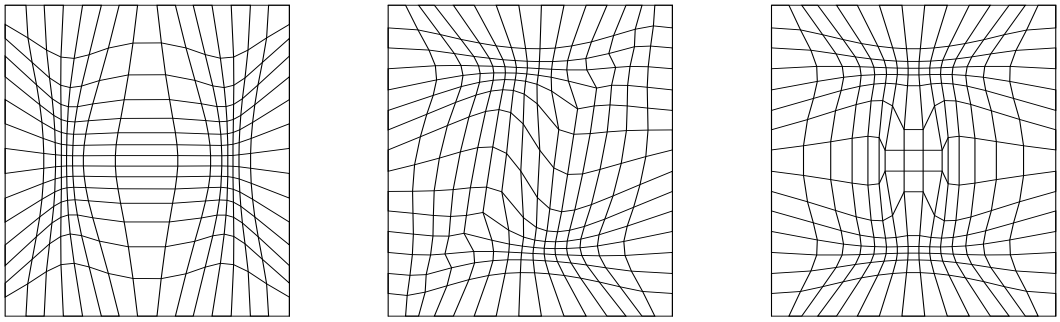

Figure 4: Setting interior boundary conditions. The surface is cubic: $\mathbf{p}(u_1, u_2) = (u_1, u_2, 32(u_1 - u_1^2)(u_2^3 - 3u_2^2/2 - u_2/2)$. From left to right are formed mesh without prescribed conditions on the interior boundary points, with all points in a diagonal direction fixed and with the 12 interior points in the center fixed.

The first experiment is illustrated in Figure 2. The surface function here is $\mathbf{p}(u_1, u_2) = (u_1, u_2, 0.35[\sin(2\pi u_1) + \sin(2\pi u_2)])$. Initially, $\mathbf{V}^{(0)} = \{\mathbf{v}_i^{(0)} \mid i \in \mathbf{D}\}$ is set at random. Orderly mesh is formed as the iteration of Equation (16) goes. The surface with the formed mesh is plotted in the lower-right. Here, the parameter in the shape function is $q = 0.3$.

The second experiment is illustrated in Figure 3. The surface function is $\mathbf{p}(u_1, u_2) = (u_1, u_2, -3[(u_1 - 0.5)^2 + (u_2 - 0.5)^2])$. The figure illustrates results with parameters $q = 0.1$, $q = 0.3$ and $q = 0.5$, respectively.

The third experiment is shown in Figure 4. The surface function is $\mathbf{p}(u_1, u_2) = (u_1, u_2, 32(u_1 - u_1^2)(u_2^3 - 3u_2^2/2 - u_2/2))$ and the shape parameter is $q = 0.3$. On the left is the resulting mesh without interior boundary points, in the middle is that with all points in a diagonal direction fixed and on the right is that with 12 interior points in the center fixed.

The final experiment is performed on a real range image containing a Renault part shown in Fig.5. Smoothing is carried out to reduce noise effect in derivative estimation. The values of its shape function $r(\mathbf{v})$ at the grid points are computed from the derivative estimates at those points. The function values between the grid points are interpolated bi-linearly. Figure 6 shows the results where $M = 64 \times 64$ and $q = 0.1$ and $q = 0.3$.

The results are found to be very stable. Empirically, the convergence of the algorithm is guaranteed with continuous shape functions. A continuous shape function is either available in analytic form or obtained by linear interpolation of estimates, the estimates being computed from data, at fixed locations. To examine the uniqueness of the solution, tests are conducted over hundreds of initial settings. Little differences are observed in the final results.

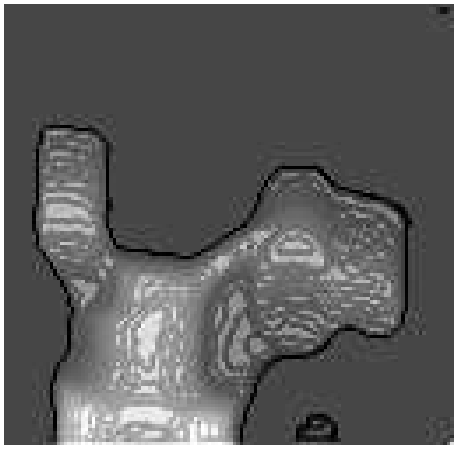

Figure 5: A range image containing a Renault part

Conclusion

The paper has presented an approach for sampling a continuous surface into discrete points while best preserving information on its shape. This is formulated as one of optimization by solving a system of nonlinear equations. The algorithm for finding the solution is simple and involves no free parameters. The algorithm is self-organizing, forming topology-preserving mesh from random initialization. Spacing between resulting mesh points can be easily controlled by a single parameter in the shape function. Important point locations can be prescribed by imposing appropriate boundary conditions. The experimental results have demonstrated the advantages.

Although the problem is discussed using graph surfaces defined on a rectangular domain, there is no limitation on the shape of the domain. The shape of the domain affects boundary conditions only. Given the boundary conditions, the solution is determined entirely by the surface shape function and the definition of neighborhood.

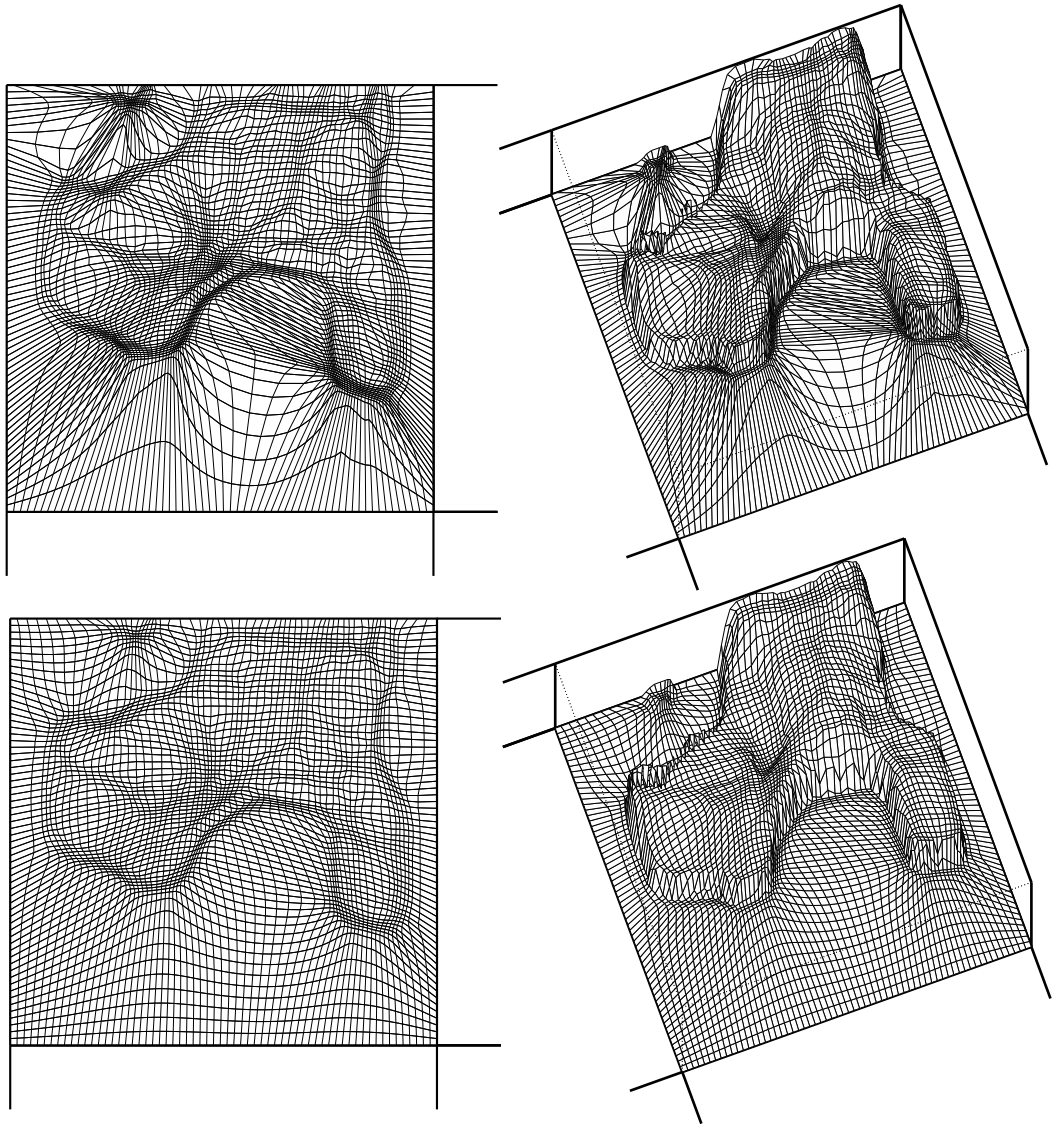

Figure 6: Results with the Renault image using $q = 0.1$ (top) and $q = 0.3$ (bottom). Left: Generated mesh in the image plane. Right: 3D plot of the result.

Acknowledgment

The author would like to thank E. Cao, Ngoi Kian Peng and H. Wang for helpful comments on the manuscript. Credits are also due to anonymous refrees for their critics and suggestions.

References

[1] J. F. Thompson, Z. U. A. Warsi, and Mastin. *Numerical Grid Generation: Foundation and Applications.* North-Holland, New York, 1985.

[2] D. T. Terzopoulos and M. Vasilescu. "Sampling and reconstruction with adaptive meshes". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 70–75, 1991.

[3] M. Kosters. "Curvature-dependent parameterization of curves and surfaces". *Computer-Aided Design*, 23(8):569–579, October 1991.

[4] C-H Ahn, S-S Lee, H-J Lee, and S-Y Lee. "A self-organizing neural network approach for automatic mesh generation". *IEEE Trans. Magnetics*, 27(5):4201–4204, 1991.

[5] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1988.

[6] B.C. Vemuri, D. T. Terzopoulos, and P.J. Lewicki. "Cananical parameters for invariant surface representation. In *Proc. SPIE Symp. on Intelligent Robots and Computer Vision VIII: Algorithms and Techniques*, volume 1192, Philadelphia, PA, November 1989. SPIE.

[7] M. Lipschutz. *Differential Geometry.* McGraw-Hill, 1969.

[8] J.J. Koenderink and A.J. van Doorn. "Surface shape and curvature scales". *Image and Vision Computing*, 10(8):557–565, October 1992.

[9] I. N. Bronshtein and K.A. Semendyayev. *Handbook of Mathematics.* translation edited by K.A. Hirsch, Van Nostrand Reinhold, New York, 1985.