.

Self-Organization of Surface Shapes

S. Z. Li
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 2263

Abstract A problem in surface modeling and approximation is how to sample a surface into a set of significant points. It is desirable that the sampling is done in such a way that best preserves the original shape. A principle is that highly curved area should be sampled densely and *vice versa*. This paper presents a self-organization method for automated surface sampling in this principle. Given a scale shape function of local curvedness of the surface and a number of samples, the set of optimal locations of sample points is defined as the solution to a system of nonlinear equations. The solution can be found using a simple iterative algorithm involving no free parameters. The algorithm forms topology-preserving meshes from random initialization. Mesh spacing vs. 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 with synthetic data.

1 Introduction

Object representation is one of major issues in computer vision. There are a variety of representation schemes. Curves and surfaces are most common basis to represent objects. Often, these physical entities need to be sampled or discretized into a set of significant points. The purpose of the sampling may be to obtain some control points or to reduce computational complexity and achieve data compression. 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 [4, 5, 6]. They can be formulated in a more concrete way. For example, Terzopoulos and Vasilescu [6] introduces a dynamic model of nodes and springs for adaptive mesh generation. The model is described by a system of equations of motion. Kosters [4] 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.* [1] use the Kohonen self-organizing neural network [3] to generate nonuniform meshes. Vemuri *et al.* [7] 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.

This paper presents a self-organization method for the shape-based surface sampling. Given a scale *shape function*, which describes local "curvedness" of the surface to be sampled, 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 algorithm produces orderly mesh points from random initialization, being a feature of self-organizing. The spacing between generated meshes decreases as the local curvedness of the surface increases and vice versa. Degree of the spacing vs. 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. Experimental results are presented.

2 Formulation

Let $S = \{\mathbf{p}(u_1, u_2) = (x(u_1, u_2), y(u_1, u_2), z(u_1, u_2)) \mid (u_1, u_2) \in \mathbf{D}^0\}$ be a 3D surface where $\mathbf{D}^0$ is the (continuous) domain of parameters u_1 and u_2. A special case is the graph surface, or Monge patch, where $\mathbf{p}(u_1, u_2) = (u_1, u_2, z(u_1, u_2))$. This is the case considered in this paper. Assume and the surface is to be sampled into a number of M (discrete) points. Denote the set of subscripts of the points by

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

Let

$$\mathbf{v} = (u_1, u_2)^T \tag{2}$$

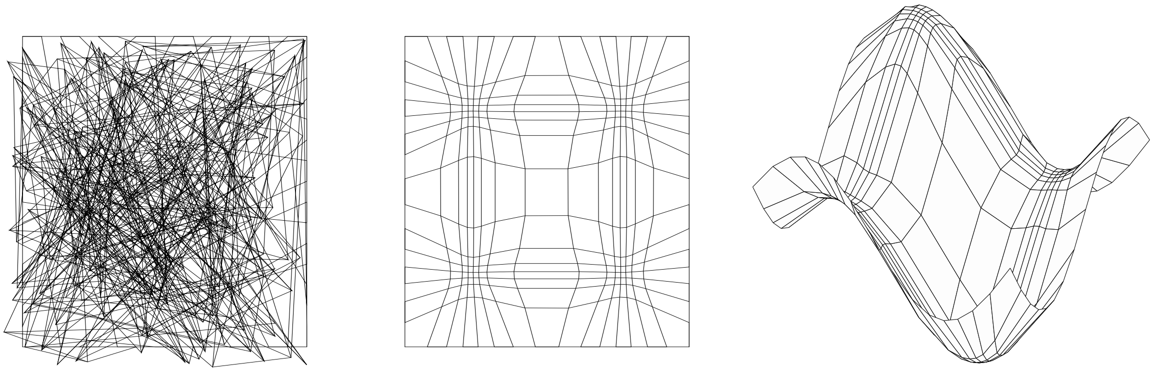

Figure 1: Orderly meshes (middle) are formed from a randomly initialized meshes (left). The plot shows the bi-sine surface $\mathbf{p}(u_1, u_2) = (u_1, u_2, 0.35[\sin(2\pi u_1) + \sin(2\pi u_2)])$ with the formed meshes. Parameter $q = 0.3$.

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

$$\mathbf{V} = \{\mathbf{v}_i \mid i \in \mathbf{D}\} \tag{3}$$

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 consists of its four or eight neighboring points in a grid.

The optimal locations $\mathbf{v}_i$ of the sample points satisfies the following system of nonlinear equations

$$\sum_{j \in N_i} r(\mathbf{v}_j)(\mathbf{v}_i - \mathbf{v}_j) = \mathbf{0} \qquad \forall i \tag{4}$$

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. 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 $S = \{\mathbf{p}(u_1, u_2) = (u_1, u_2, z(u_1, u_2)\}$, the shape function can be defined, for example, as follows

$$\kappa(u_1, u_2) = \frac{\sqrt{z_{u_1 u_1}^2 + z_{u_2 u_2}^2}}{(1 + z_{u_1}^2 + z_{u_2}^2)^{3/2}} \tag{5}$$

where z_u and z_{uu} are, respectively, the first and second partial derivatives of z. The measure is an analogy to the curvature magnitude for planar curves. Then 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

$$r(\mathbf{v}) = r(u_1, u_2) = q + \frac{\kappa(u_1, u_2) - \min_{u_1, u_2} \kappa(u_1, u_2)}{\max_{u_1, u_2} \kappa(u_1, u_2) - \min_{u_1, u_2} \kappa(u_1, u_2)} \tag{6}$$

The parameter q controls the *contrast* of $r(u_1, u_2)$ on which the result of shape-based sampling is dependent (see an experiment). Recently, two interesting surface shape functions, namely shape index and curvedness, are proposed [2]. 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.

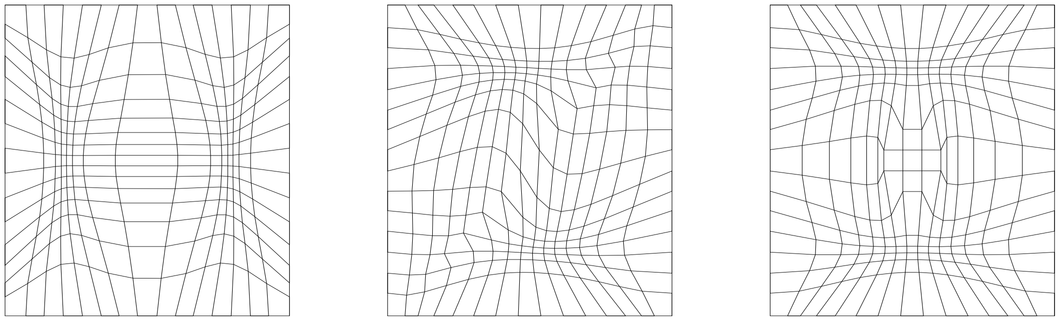

Figure 2: Setting interior boundary points with surface $\mathbf{p}(u_1, u_2) = (u_1, u_2, 32(u_1 - u_1^2)(u_2^3 - 3u_2^2/2 - u_2/2)$. From left to right are formed meshes without interior boundary points, with all points in a diagonal direction fixed and with 12 interior points in the center fixed.

3 Algorithm

The system of nonlinear equations (4) can be solved using a very simple algorithm. Rearranging the equations yields

$$\mathbf{v}_i = \sum_{j \in N_i} r(\mathbf{v}_j)\mathbf{v}_j / \sum_{j \in N_i} r(\mathbf{v}_j) \qquad \forall i \tag{7}$$

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

$$\mathbf{v}_i^{(t+1)} = \sum_{j \in N_i} r(\mathbf{v}_j^{(t)})\mathbf{v}_j^{(t)} / \sum_{j \in N_i} r(\mathbf{v}_j^{(t)}) \qquad \forall i \tag{8}$$

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

$$|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}| < \epsilon \tag{9}$$

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).

The algorithm has a feature of self-organizing [3]. Starting with randomly initialized point positions $\mathbf{v}_i^{(0)}$, it automatically develops topology-preserving, orderly meshes (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).

4 Experiments

Experiments were conducted 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 (6) and (3) Examples with prescribed locations of interior boundary points. Some results are shown in Figs 1, 2 and 3. Initially, mesh points $\mathbf{V}^{(0)} = \{\mathbf{v}_i^{(0)} \mid i \in \mathbf{D}\}$ were set at random. As the result of iteration using (8), orderly meshes were formed (middle) with rather rapid convergence.

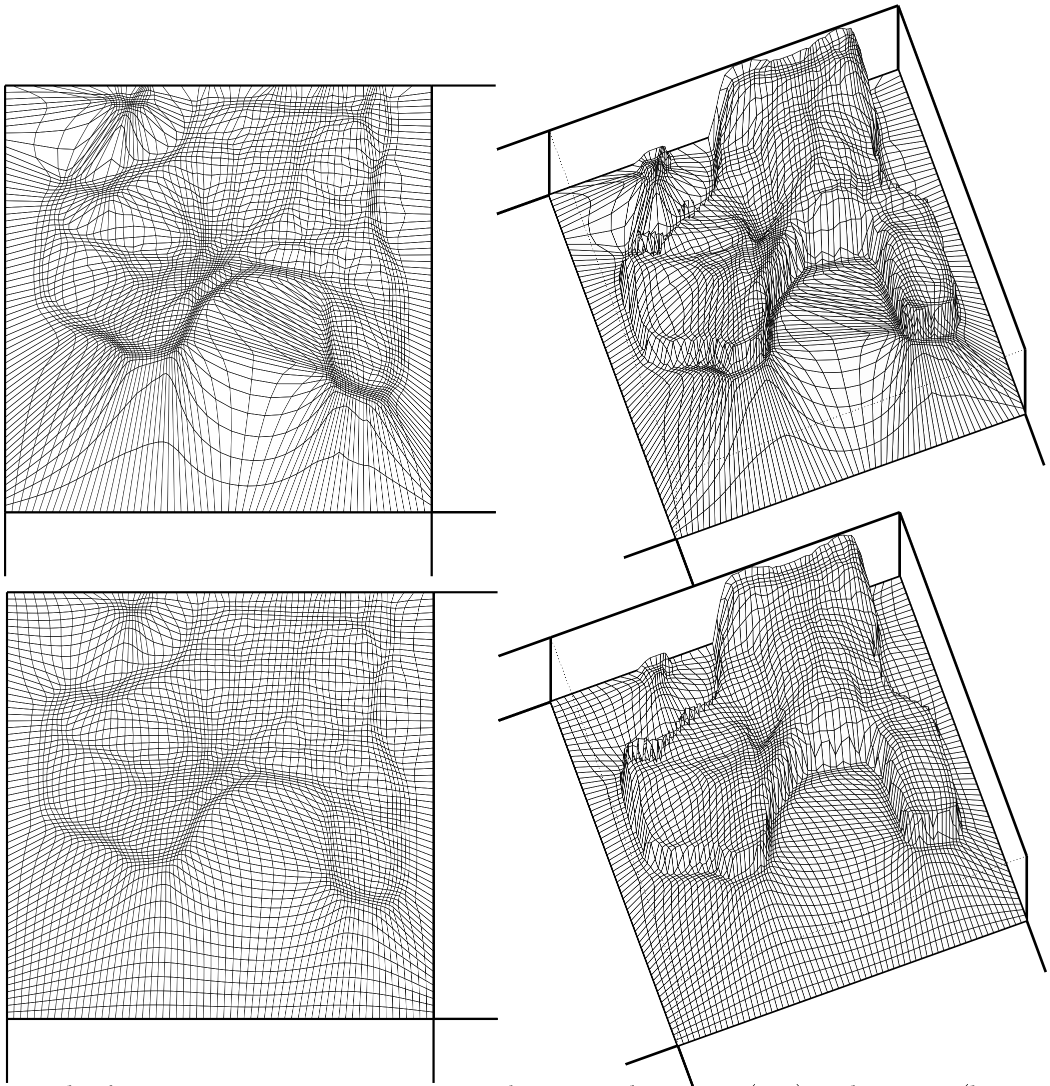

Figure 3: Results from an image containing a Renault part with $q = 0.1$ (top) and $q = 0.3$ (bottom). Left: resulting point distribution in the image plane. Right: 3D plot of the result.

References

[1] 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.

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

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

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

[5] S. Z. Li. "Object recognition from range data prior to segmentation". *Image and Vision Computing*, 10(8):566–576, October 1992.

[6] 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.

[7] B. Vemuri, D. T. Terzopoulos, and P. 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.