行政院國家科學委員會專題研究計畫 成果報告

網格參數化技術及其於三維著色系統之應用
研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 94-2213-E-009-088-
執行期間：94年08月01日至95年07月31日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：莊榮宏

計畫參與人員：博士班研究生-兼任助理：陳治君、李汪曄、何丹期、簡民昇
碩士班研究生-兼任助理：林奕均

處理方式：本計畫可公開查詢

中 華 民 國 95年12月26日

摘要

三維表面著色是一項讓使用者直接在三維模型表面上進行繪圖的技術，畫到模型表面的筆觸必須透過表面參數化技術儲存至貼圖上。目前存在的三維表面著色系統之表面參數化在著色過程中是固定的，此類的參數化並無正確捕捉到繪圖筆觸中顏色訊號的變化。為了建立筆觸顏色訊號與貼圖取樣多寡的關聯性，我們必須根據使用者所輸入的筆觸資訊動態的重新參數化，並要達到可互動的速度。我們提出一套基於迭代式最佳化之重新參數化機制，以讓顏色變化劇烈處得到較多的貼圖取樣，且同時保持可互動的速度。

關鍵字: 表面參數化，表面著色，貼圖

ABSTRACT

Surface painting is a procedure that allows the users to paint onto a surface directly. The painting strokes are stored in a texture via surface parameterization techniques. In current surface painting systems, the underlying surface parameterization is fixed during the painting process. Such a parameterization is not sensitive to the frequency spectrum of the color signal introduced by painting strokes. To associate the regions of higher color signal variation with more texture samples, we need to do the re-parameterization according to user's strokes at interactive rates. We propose a re-parameterization scheme that is based on an iterative-optimization aiming to allocate more texture samples for regions of high signal variation and to perform at an interactive rate as well.

Keywords

surface parameterization, surface painting, texture mapping

1. INTRODUCTION

Surface painting(also called 3D painting) is a technique that allows the users to paint directly onto a 3D surface. If the discretization of the surface is fine enough, user can directly paints on the vertices of the surface. However, in general, the desired precision for the color is greater than the geometric detail of the model. Assuming that a surface is provided with a parameterization, it is convenient to store colors in the parameterized texture space. In current surface painting systems, the underlying mesh parameterization is predefined and fixed during the painting process. Such a parameterization is not sensitive to the frequency spectrum of the color signal as the result of painting strokes, and in consequence, may introduce distortion at arbitrary locations and waste texture space in areas of no stroke. Moreover, current surface painting systems parameterize the surface only based on the geometric aspects. Even though these systems provide tools allowing users to adjust the underlying parameterization, but it is not intuitive for normal users.

Most surface parameterization schemes assume no prior know-ledge of the signal, and take only surface's geometry information into account. For surface painting, we want to allocate more texture samples in regions of greater signal detail by doing the re-parameterization on the fly according to the painting strokes. Moreover, the re-parameterization should be fast enough to achieve an interactive rate.

Our proposed method first derives an initial parameterization, and then, during the painting process, analyze the color signal frequency introduced by painting strokes, and utilizes an iterative optimization to do the re-parameterization to interactively allocate more texture samples for regions with high color signal variation. The proposed method is simple to implement and works well for models with either low or large polygon count.

We describes a novel and simple framework of the re-param- eterization necessary for future surface painting systems. Along the way to achieve this goal, we present the following contributions:

- Propose a modified signal metric L^2_s that measures the geometry and signal stretch of a parameterization.
- Propose an interactive approach for the re-parameterization aiming to increase the sampling ratio in regions with high signal variation.

2. PREVIOUS WORK

Parameterization is a mapping from two dimension to higher dimension. Several schemes have been proposed that flatten a surface region and establish a parameterization over the last decade in computer graphics [3, 4, 7, 12, 13]. Sander et al. proposed a non-linear stretch that integrate the sum of squared singular values over the map [10]. We refer to this metric as geometry stretch. The parameterization is derived by a coarse-to-fine optimization scheme that minimizes the geometry stretch over the map. Note that the resulting parameterization may encounter parametric crack problem.

Sander et al. developed a signal-stretch metric that combines both surface area and surface signal bandwidth [9]. It is shown that the stretch metric is related to SAE (Signal-Approximation Error) - the difference between a signal defined on the surface and its reconstruction. Sander's signal stretch can be seen as the extension of the geometry stretch. It is, however, not suitable for surface painting since it utilizes an expensive global optimization, and cannot support the interactive re-parameterization required after each stroke is painted.

Hanrahan and Haeberli firstly proposed the concept of three dimensional surface painting, in which the color signal is stored directly in mesh vertices [6]. Based on this method, the shading result is interpolated between mesh vertices, though we could not reveal rich texture detail. Igarashi and Cosgrove stored the paint strokes image that occurred for each pose as separate charts packed into a texture atlas [8]. Mesh triangles affected by painting strokes are found and projected onto a two dimensional domain to form an atlas. Similarly, each subsequent stroke is stored in a new atlas. When the painting process complete, all the atlas are packed together to form the final texture atlas. The major disadvantage of the method is that a stroke that overlapping other strokes may appear in more than one atlas. In such cases, texture space may be wasted. Carr and Hart proposed a method aiming to derive a parameterization that is sensitive to signal distribution [2] base on their prior work, multi-resolution meshed atals (MMA) [1]. In their method, the mesh is first divided into several charts based on the method proposed by Sander et al. [11] to form the MMA tree hierarchy. During the surface painting process, all painting strokes are rendered into texture and then the stroke frequency distributed on the texture is analyzed using graphics hardware. After the analysis, an importance value computed from frequency analysis for previous strokes

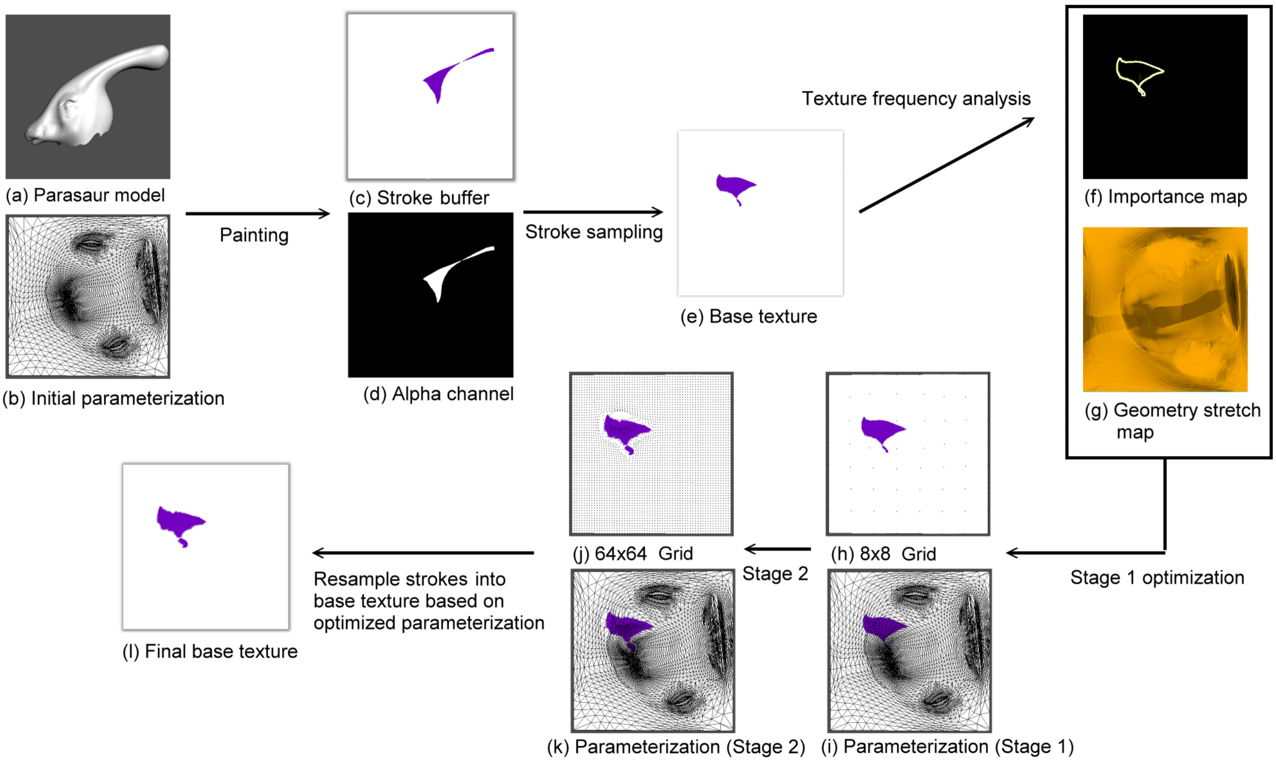


Figure 1: The overall process of our method.

is attached to each chart and the MMA hierarchy (quaternary tree) is re-balanced to generate a new parameterization. Each chart should consist of a quite large number of faces in order to reduce distortion introduced by the parameterization. Moreover, the re-balancing is more significant with more number of charts. Therefore this method is suitable for meshes with large number of triangles.

3. A RE-PARAMETERIZATION FRAMEWORK FOR SURFACE PAINTING

To optimize the sampling resolution in parametric space, our basic idea is to increase the resolution of regions with high signal variation while decreasing the resolution of others. Our parameterization optimization framework for surface painting comprises the following steps as shown in Figure 1:

1. Transform the closed-surface Ω_T into an open-surface Ω'_T using topological surgery, construct a global initial parameterization for the surface mesh, and generate a base texture based on the parameterization.

2. Resample painting strokes into the base texture. Analyze the signal frequency on base texture, generate importance map and geometry stretch map using graphics hardware.

3. Apply a uniform grid G underlying the parameterization domain, in which each point of G is assigned a L_s^2 stretch value derived from importance map and geometry stretch map, and then perform a two-stage optimization to get an optimized uniform grid G_{opt}.

4. Re-parameterize Ω'_T according to the optimized uniform grid G_{opt}, and resample the painting strokes according to the new parameterization.

3.1 Initial parameterization

To parameterize surface Ω_T onto a planar domain, Ω_T should be topologically equivalent to a disk. If Ω_T is a closed-surface, we perform the topological surgery proposed in [5] to transform Ω_T to an open-surface Ω'_T that is equivalent to a topological disk.

Although many parameterization techniques are adequate to derive a global initial parameterization, the one aiming to guarantee uniform sampling and preserve conformality structure of the input mesh is most preferable. Here, we use the method proposed by Yoshizawa et al. [13] due to its preferable properties and requires solving a simple, sparse linear system, which is usually handled in a matter of seconds using Conjugate Gradient solver with good preconditioning.

3.2 Stroke sampling

During painting process, we use the method proposed by Carr et al. [2] to sample painting stroke into texture. Each paint stroke applied in the same object pose (i.e. modelview coordinates of the model) is rendered directly into base texture map using graphics hardware. For this task, we need a stroke buffer for storing the painting data and a depth buffer for the depth of current object pose.

The resampling is done by a vertex shader and a fragment shader. The vertex shader transforms the world space position into model view coordinates and then swaps each vertex's model view coordinates with its texture coordinates.

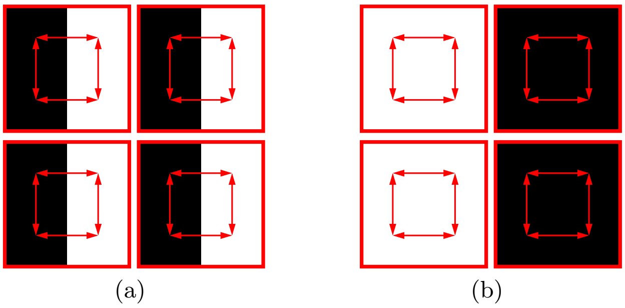


Figure 2: Problems of four-tap filter.

The fragment shader is applied to render the new base texture by taking the stroke buffer, depth buffer, and the original base texture as input. The alpha channel in stroke buffer represents the existence of paint strokes to ensure that only the strokes can overwrite the existing base texture. The depth buffer is used to prevent paint being applied to invisible portions of the model. This process is performed for the stroke painted at each pose.

3.3 Importance map and geometry stretch map

To analyze the base texture for finding regions that requires additional samples, a four-tap gradient magnitude filter is used in [2] to find undersampled regions. The four-tap filter fetches four samples from the input texture, and outputs the result in half resolution. For the four-tap gradient magnitude filter, some gradient features will be missed. For example, as shown in Figure 2, each red rectangle represents 4 pixels on the texture, and we detected the gradient in s-direction of paint (a), but not in paint (b).

Here we modify previous four-tap filter. For each pixel on the base texture, we calculated its magnitude of the gradient using fragment shader arithmetic by central difference. Actually, this is the *Sobel Filter* in the filed of image processing. The filter is applied for each pixel of the base texture, therefore the output image is the same resolution as the base texture. Figure 3 demonstrates the result of two filters which shows that our modified filter is more accurate than four-tap filter.

Besides the importance map, another map called geometry stretch map is also computed. This geometric stretch map stores L^2 stretch value for each face on parametric domain as shown in Figure 1(g). We normalize the value of geometric stretch of each face to lie between 0 and 1. Next, we render the mesh on parametric domain using the normalized geometric stretch value as the color of the face.

3.4 The L_s^2 stretch

After the generation of importance map and geometry stretch map, the L_s^2 stretch is derived from these two maps. As mentioned in [9], the signal stretch can have zero gradient since the signal may be locally constant on a region of the surface. Therefore, a tiny fraction of geometry stretch is added into the energy function to be minimized. The L_s^2 stretch is defined as follows:

$$L_s^2(s,t) = \begin{array}{ll} 1 - L^2(s,t) & \text{, if } E_h(s,t) = 0 \\ 1 - (\alpha \cdot L^2(s,t) + \beta \cdot E_h(s,t)) & \text{, otherwise.} \end{array}$$

where $E_h(s,t)$ is the signal stretch proposed by Sander et al. [9], and $L^2(s,t)$ is the geometry stretch [10]. The two values are obtained from importance map and geometry stretch map, respectively. In the region with signal variation, we use the weighted geometric stretch and signal stretch as in [9]. Otherwise, in the region without signal variation, we purely take the geometry stretch into account to prevent undersampling in regions with no signal variation. The L_s^2 stretch could be considered as the extension of signal stretch.

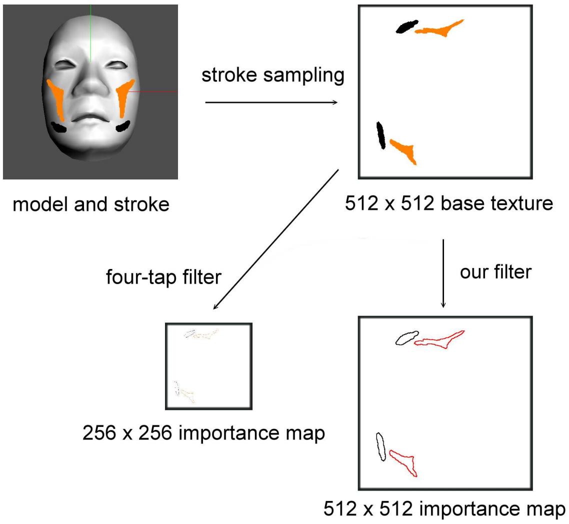


Figure 3: Four-tap filter and our filter.

3.5 Iterative optimization based on uniform grid

For interactive applications, the parameterization proposed by Sander et al. [9] has two major problems when it is applied to surface painting systems. First, since the signal introduced by painting strokes is not constant over the triangle, numerical integration is used to compute the signal stretch on each triangle. All the mesh triangles are subdivided into 64 sub-triangles and the signal stretch are evaluated at all the vertices. The second drawback is that the optimization process proposed by Sander et al. is a nonlinear, global optimization. As a result, the parameterization is expensive and therefore not suitable for interactive surface painting applications.

To reduce the cost of computing signal stretch, instead of subdividing each triangle, we derive the signal stretch on parametric domain. We apply an $N \times N$ uniform grid G to the parametric domain in which the initial parameterization lies as shown in Figure 4. These grid points, rather than the mapping of mesh vertices, are used to sample L_s^2 stretch on parametric domain, that is, we compute L_s^2 stretch for each grid point. Such an approach allows us to control the sampling resolution. Moreover, the grid is used to be the target for stretch optimization. By doing this, the computational complexity of performing optimization will be dependent on the resolution of the grid, rather than the mesh.

We then optimize G by the following steps:

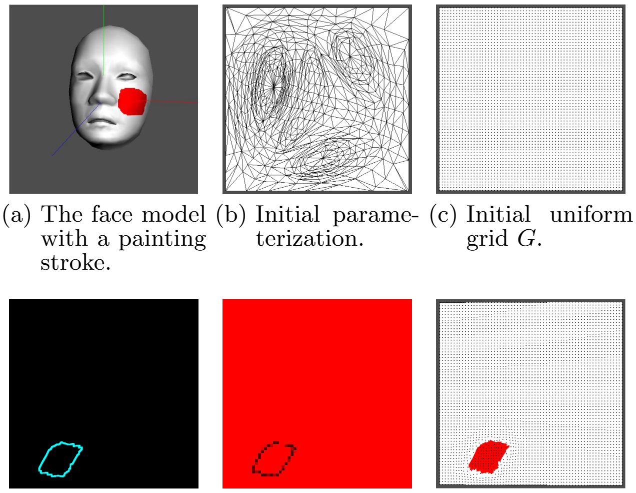
1. For each point $N \in G$, derive $L_s^2(N)$ from importance map and geometry stretch map using graphics hardware.

2. For each interior point $N_i \in G$ in turn,

$$\text{compute} \quad \widetilde{N_i} = \frac{\sum_{N' \in \text{1-ring of } N_i} L_s^2(N') \cdot N'}{\sum_{N' \in \text{1-ring of } N_i} L_s^2(N')},$$

$$\text{set} \quad N_i = \widetilde{N_i}.$$

3. Repeat 1 and 2 until $\|\widetilde{N_i} - N_i\| < \epsilon$ for every i.

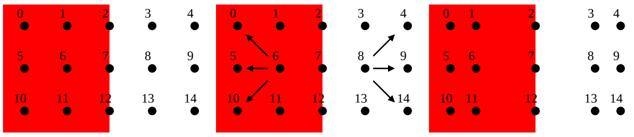


(a) The face model with a painting stroke. (b) Initial parameterization. (c) Initial uniform grid G.

(d) Importance map. (e) Stretch on sample points (64×64). (f) Optimized sample points G_{opt}.

Figure 4: The optimization result base on one iteration.

Figure 4(a) illustrates the face model with a red painting stroke and the resulting importance map is shown in Figure 4(d). A 64×64 uniform grid G is applied to the parametric domain, where each sample point is assigned a L_s^2 stretch value as shown in Figure 4(e) (we take only signal stretch into account in this case). Figure 4(f) shows the optimized grid G_{opt}, where the sample points are more sparse in the regions with signal variation. The optimization procedure on the grid points is illustrated in Figure 5. Figure 5(a) depicts the grid points and the corresponding parametric domain with signal distributed. Since the L_s^2 stretch values of p_2, p_7 and p_{12} are smaller than that of p_0, p_5 and p_{10}, p_1, p_6 and p_{11} are moved toward p_0, p_5 and p_{10}. Similarly, p_3, p_8 and p_{13} are moved toward p_4, p_9 and p_{14}; as shown in Figure 5(b)(c).



(a) Initial grid. (b) The optimization process. (c) The optimized grid.

Figure 5: Chart diagram of the optimization process.

The optimization procedure is an iterative optimization process, in which the local optimization optimizes a grid point in one iteration. After the optimization, we will get an optimized uniform grid G_{opt}. On G_{opt}, grid points will become dense in the regions with high L_s^2 stretch (lower signal variation), and sparse otherwise. After the optimization process, the underlying parameterization will be recomputed by barycentric interpolation according to the optimized grid points as described in next section.

3.6 Re-parameterization

After optimizing the initial uniform sample points G, we re-parameterize the parameterization by the barycentric interpolation based on the optimized uniform sample points G_{opt}. For each vertex $v_j \in V_I$, let N^{j_0}, N^{j_1}, N^{j_2} and N^{j_3} be the sample points of the cell that contains v_j. Barycentric coordinates w_0, w_1, w_2 and w_3 are derived such that

$$v_j = \sum_{i=0}^{3} w_i \cdot N^{j_i}.$$

The new position of v_j will be

$$v_j^{opt} = \sum_{i=0}^{3} w_i \cdot N_{opt}^{j_i},$$

where $N_{opt}^{j_0}$, $N_{opt}^{j_1}$, $N_{opt}^{j_2}$ and $N_{opt}^{j_3}$ are the homologous points of N^{j_0}, N^{j_1}, N^{j_2} and N^{j_3} in G_{opt}.

3.7 Stroke resampling over optimized parameterization

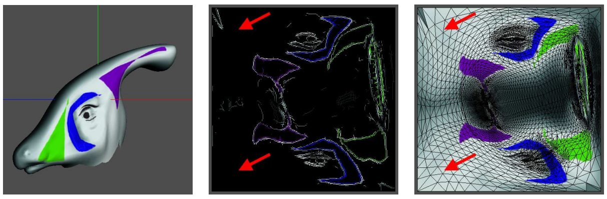
Finally, we resample the base texture based on the optimized parameterization. The sampling process is similar to the method mentioned in section 3.2. The only difference is that now we have two texture coordinates, i.e. parameter values t_{prev} and t_{opt} for each vertex, which are derived from initial parameterization ϕ and optimized parameterization ϕ_{opt}, respectively. As described in section 3.2, we first swap each vertex's model view coordinates with its current texture coordinates t_{opt} in vertex shader, and then we resample painting strokes to form a new base texture. The resampling procedure here consists of two step. The first step resamples current painting strokes stored in stroke buffer; step two resamples previous painting strokes stored in the previous base texture. Therefore current model view coordinates and t_{opt} are used to sample current stroke from stroke buffer and t_{prev} is used to sample previous stroke form previous base texture.

3.8 Two-stage re-parameterization framework

Compared to Sander's signal-specialized parameterization [9], the proposed framework tends to be a local optimization process. Figure 6 shows the re-parameterization result using a 256x256 uniform sample points. We can see that the relaxation of sample points is bounded inside the cell it lies. As shown by the red arrow in Figure 6, there should be less sample space in these regions with lower signal gradient. However, the movement of the sample points in these regions is not much due to the fact that the L_s^2 stretch of these points are almost the same. Therefore the relaxation works well in the regions with high gradient, but may not work well in the other regions. To solve this problem, a two stage optimization framework is used instead of the single stage optimization.

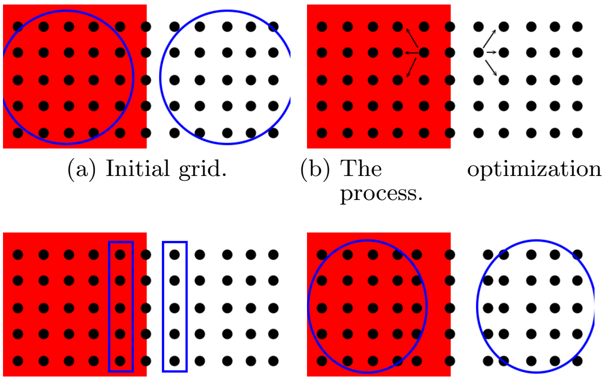
In the two-stage optimization, we expect that the first stage diminishes the texture sample space in region with lower signal gradient and the second stage magnifies the texture space in regions with high signal gradient. To achieve this goal, a lower resolution uniform grid is used in the first stage and a high resolution grid in the second stage. Figure 7 illustrates the optimization result using a high resolution grid. As shown in Figure 7(c), only these sample points near the regions of low L_s^2 stretch value (high signal variation) will be moved after the iterative optimization. Other sample points will remain fixed in other regions where neighboring points have the same L_s^2 stretch value. Figure 7(d) shows the final result of the optimization. The regions with lower signal stretch are expected to obtain less texture space. Apparently, optimization using a high resolution grid does not work well for this purpose, see the comparison highlighted by the blue circle in Figure 7(a) and Figure 7(d).

The optimization resulting from using a lower resolution grid will have more convergence effect in the regions of high L_s^2 stretch (lower signal variation), and allocate less texture space in these regions. See the comparison shown in Figure 8(a) and Figure 8(d).



(a) Parasaur model (b) Optimized uniform sample points (c) Base texture

Figure 6: Parasaur model : single stage optimization using 256x256 uniform sample points

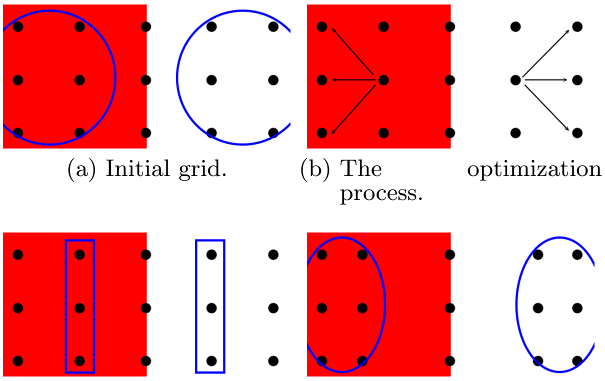


(a) Initial grid. (b) The optimization process.

(c) The relaxed grid points. (d) The optimization result.

Figure 7: High resolution uniform grid points.

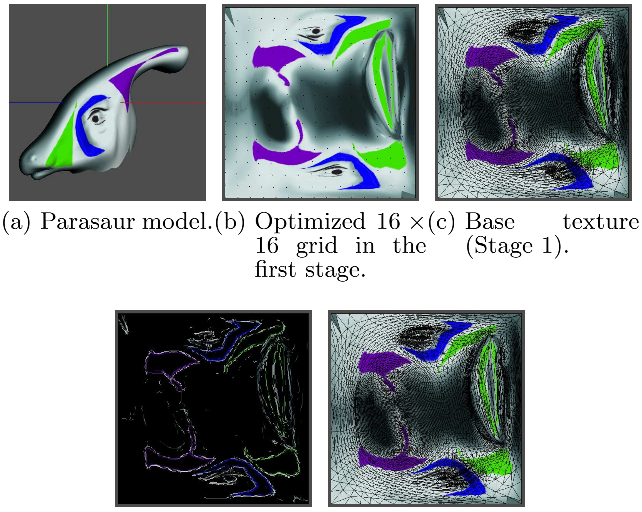
Figure 9(b) shows that the sample points of 16 × 16 resolution int the first stage and Figure 9(d) is the result of using the sample points of 256 × 256 resolution in the second stage. We see that the texture space in regions of lower signal gradient is diminished in stage one; as shown in Figure 9(c), while in stage two, more texture space in the regions of high signal gradient are allocated; see Figure 9(e). Figure 10 shows the result of single stage optimization and two stage optimization for comparison. Obviously, the texture space is used more efficiently using the two stage optimization method, especially in the regions of lower signal gradient, see the comparison highlighted by the red arrows in Figure 10.



(a) Initial grid. (b) The optimization process.

(c) The relaxed grid points. (d) The optimization result.

Figure 8: Low resolution uniform grid points.



(a) Parasaur model. (b) Optimized 16 × 16 grid in the first stage. (c) Base texture (Stage 1).

(d) Optimized 256 × 256 grid in the second stage. (e) Base texture (Stage 2).

Figure 9: Parasaur model : Two-stage optimization using 16 × 16 and 256 × 256 grids.

4. RESULTS AND PERFORMANCE ANALYSIS

All results are performed with a AMD Athlon64 3000+ PC, 512 MB RAM and an NVIDIA GeForce 6800 graphics card. It is running Windows XP with NVIDIA Cg 1.3 compiler, vp40 vertex shader profile and fp40 fragment shader profile. We use the pBuffer extension for efficient texture rendering.

We compare our result with that based on static parameterization in current surface painting systems. Figure 11 and Figure 12 show the painting result of our surface paint-

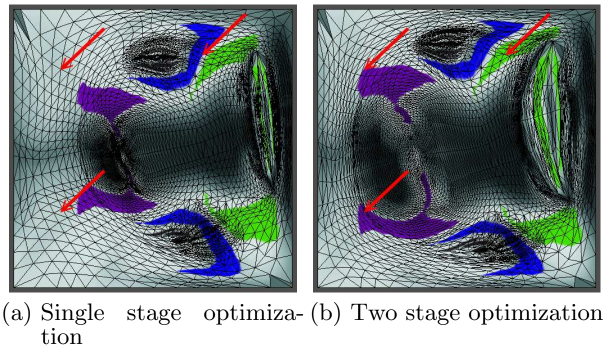

(a) Single stage optimization (b) Two stage optimization

Figure 10: Comparison of single and two stage optimization

ing system on the venus model. The left columns show the result of current surface painting systems, i.e. with fixed underlying parameterization. The right columns show the result of our two-stage optimization process. Our method depicts better texturing quality than that for current surface painting systems.

Figure 13 and Figure 14 demonstrate the painting results of other models. Aliasing occurs in undersampling regions and our method alleviate this problem efficiently.

The strokes of all the results shown in Figure 11 to Figure 14 are painted manually. Figure 15 shows the result where four images is texture mapped to simulate painting strokes. The artifact, blur, occurs due to the fact that texture is undersampled using fixed parameterization as shown in the left column of Figure 15. The right column shows that the result of two-stage optimization is much more better.

Table 1 lists the computation time for initial parameterization, two-stage optimization and re-parameterization occurs during surface painting process. Because the optimization procedure is done on parametric domain, the computation cost of two-stage optimization is independent on the face number of input model. The timing required by the two-stage optimization is reasonable for the interactive application of surface painting systems. Figure 16 shows the optimization time after each stroke is applied on the triceratops model. Since the geometry stretch is minimized in the first optimization process, the timing is higher than succeeding optimizations.

The signal-specialized parameterization proposed by Sander et al.[9] is thought to be the state-of-art work in mesh parameterization which is sensitive to surface signal. We compare the parameterization performance between our two-stage optimization framework and signal-specialized parameterization.

Figure 17(a)(b) show the result of signal-specialized parameterization using 2048×2048 and 128×128 texture maps, respectively. Figure 17(c)(d) shows the result of our two-stage optimization under two different texture map resolutions. Our result is pretty good under resolution of 256x256 and still fine under resolution of 128x128.

5. CONCLUSION

We have proposed a rapidly re-parameterization framework for surface painting which redistributes texture sample space according to the surface signal variation. A two stage uniform grid optimization framework is proposed which diminished sample space in lower gradient regions in stage one and magnifies sample space for high gradient regions in stage two. In addition, this two stage optimization framework is suitable for interactive use required by surface painting. For the optimization process, we derived the modified L^2 metric denoted as L_s^2. The L_s^2 metric takes signal stretch into account in the regions of signal variation and combines geometry stretch in the regions without signal variation.

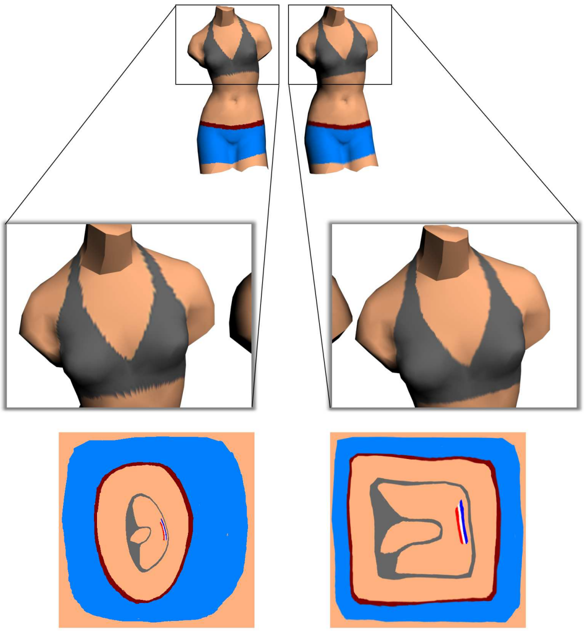


Figure 11: Painting results of the venus model. Result of a fixed-parameterization (left column) and the result of our two-stage optimized parameterization (right column).

Some potential future work are listed as follows:

- **Better stroke sampling method** The stroke sampling method[2] based on graphics hardware is simple and fast for interactive use. However the result is bad when the resampling was done under either magnification or minification. Perhaps some filter on image space could alleviate this problem.
- **Better parameterization metric** Since the same value of geometry and signal stretch does not imply the equal significance, our proposed L_s^2 stretch actially works in a heuristic manner. A good study on the weighted relationship between geometry and signal stretch may enhance the theoretical background of our method. Furthermore, a better metric, especially the one which is more sensitive for the anisotropical distribution of surface signal on parametric domain is a chance to improve the overall quality.
- **Hierarchical optimization** Optimization based on adaptive sample points can be utilized to improve the performance. In our two-stage optimization framework, the sample points are uniformly distributed on parametric domain at each step. To use the sample points more efficiently, we distribute more sample

Model	face	Init-param.	Two-stage Optimization		Re-param.
			range	avg.	
venus	1396	0.625	0.718 - 3.843	1.784	0.016
triceratops	5660	3.234	0.625 - 3.156	1.739	0.063
face	1162	0.5	1.531 - 3.828	1.690	0.016
horse	7500	4.906	1.031 - 3.125	1.375	0.078

Table 1: Statistics of initial parameterization, two-stage optimization and re-parameterization time (sec.) for four different models.

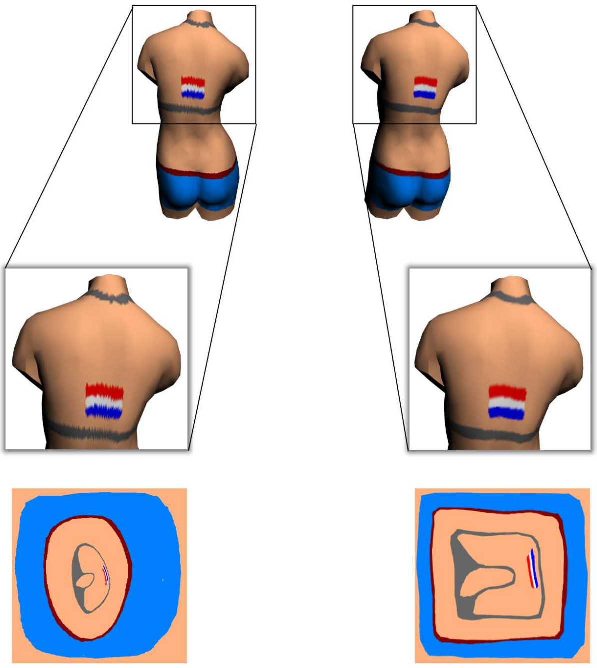


Figure 12: Back-view of the painting results of the venus model. Result of a fixed-parameterization (left column) and the result of our two-stage optimized parameterization (right column).

points on the regions of high signal gradient to accurately grab the signal variation. Less sample points are distributed on the regions of lower signal gradient, thus these regions will be converged more quickly. To achieve the goal, a hierarchy architecture of uniform grid is required to maintain the different resolution of grid points. For sampling, there are two major problems of the hierarchical method. The first one is the determination of high gradient region and lower gradient region. A two-pass method will be practical to accomplish this. The second problem is that a theoretical and efficient method to propagate the L_s^2 stretch from high resolution grid pints to lower resolution grid points is required. In addition to the problems of sampling, an efficient optimization algorithm for the hierarchical gird architecture is also required.

6. REFERENCES

[1] N. A. Carr and J. C. Hart. Meshed atlases for real-time procedural solid texturing. *ACM Transactions on Graphics*, 21(2):106–131, 2002.

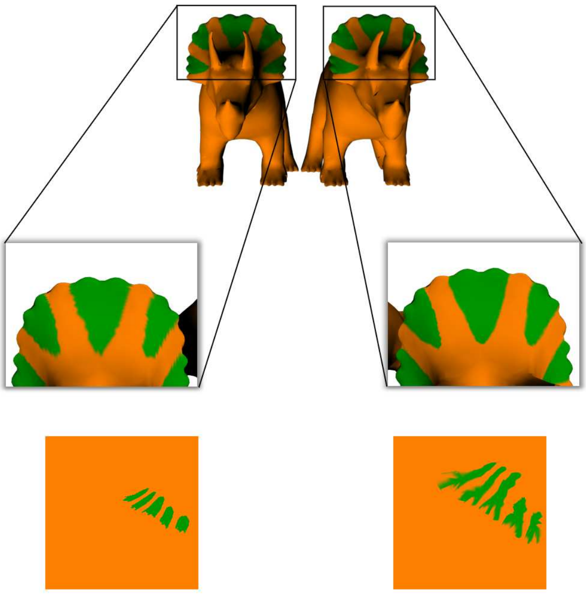


Figure 13: Painting results of the triceratops model. Result of a fixed-parameterization (left column) and the result of our two-stage optimized parameterization (right column).

[2] N. A. Carr and J. C. Hart. Painting detail. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 23, 3:845–852, 2004.

[3] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH 95 Conference Proceedings*, pages 173–182, Aug. 1995.

[4] M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[5] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. In *SIGGRAPH 2002 Conference Proceedings*, pages 335–361, 2002.

[6] P. Hanrahan and P. Haeberli. Direct wysiwyg painting and texturing on 3d shapes. In *International Conference on Computer Graphics and Interactive Techniques*, pages 215–223, 1990.

[7] K. Hormann and G. Greiner. MIPS: An efficient global parametrization method. In *Curve and Surface Design: Saint-Malo 1999*, pages 153–162. Vanderbilt University Press, 2000.

[8] T. Igarashi and D. Cosgrove. Adaptive unwrapping for interactive texture painting. In *Proceedings of the*

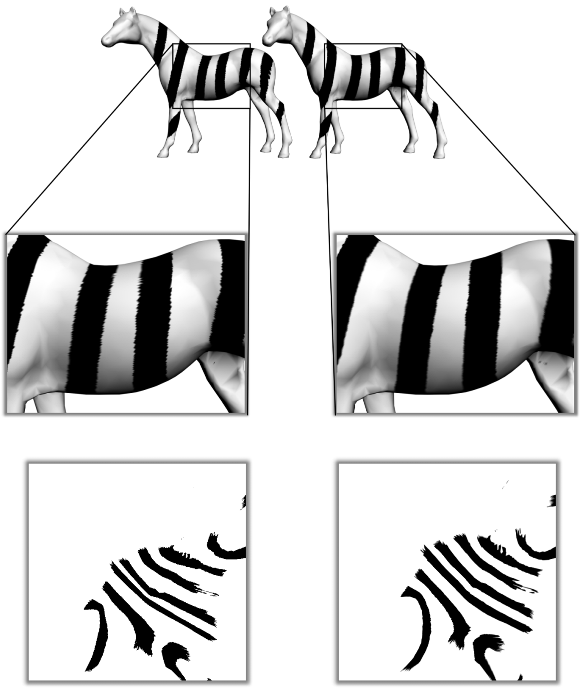


Figure 14: Painting results of the face model. Result of a fixed-parameterization (left column) and the result of our two-stage optimized parameterization (right column).

2001 symposium on Interactive 3D graphics, pages 209–216, 2001.

[9] P. V. Sander, S. J. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. In *Proceedings of the 13th Eurographics Workshop on Rendering (RENDERING TECHNIQUES-02)*, pages 87–98, June 26–28 2002.

[10] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *SIGGRAPH 2001 Conference Proceedings*, pages 409–416, 2001.

[11] P. V. Sander, Z. J. Wood, S. J. Gortler, J. Snyder, and H. Hoppe. Multi-chart geometry images. *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 146–155, 2003.

[12] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle based flattening. *Engineering with Computers*, 17, 3:326–337, 2001.

[13] S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel. A fast and simple stretch-minimizing mesh parameterization. In *International Conference on Shape Modeling and Applications (SMI 2004)*, pages 200–208, June7–11 2004.

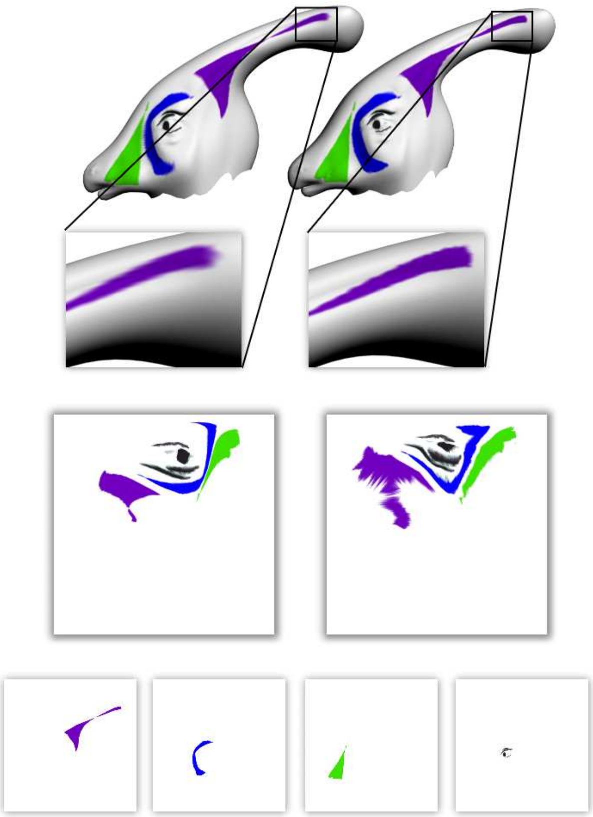


Figure 15: Four images are texture mapped to simulate painting strokes. Result of a fixed-parameterization (left column) and the result of our two-stage optimized parameterization (right column).

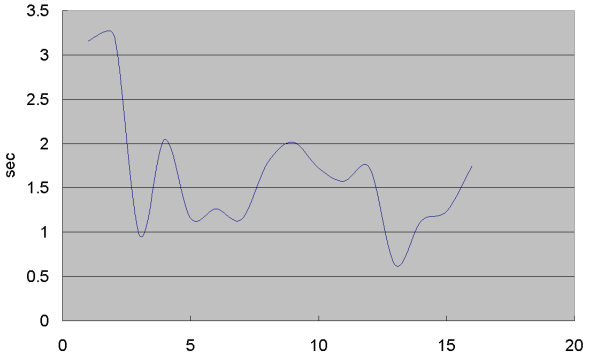
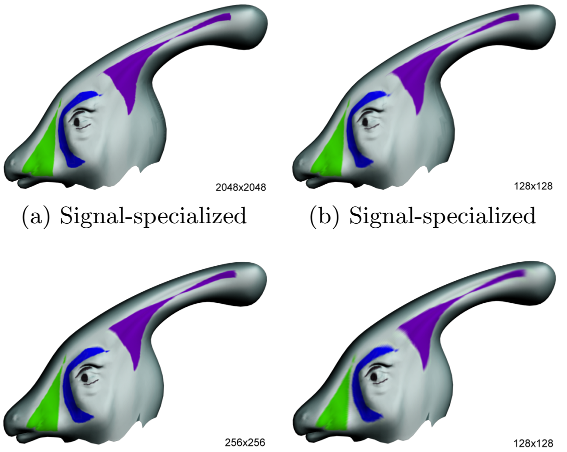


Figure 16: The optimization process of triceratops model.



(a) Signal-specialized (b) Signal-specialized

(c) Two-stage optimization (d) Two-stage optimization

Figure 17: Comparison of our result with signal specialized parameterization under different texture map resolutions.