

Lujin Wang · Xianfeng Gu · Klaus Mueller · Shing-Tung Yau

# Uniform Texture Synthesis and Texture Mapping Using Global Parameterization

**Abstract** Texture mapping and texture synthesis are two popular methods for the decoration of surfaces with visual detail. Here, an existing challenge is to preserve, or at least balance, two competing metrics: scale and angle. In this paper we present two methods for this, both based on global conformal parameterization. First, we describe a texture synthesis algorithm for surfaces with arbitrary topology. By using the conformal parameterization, the 3D surface texture synthesis problem can be converted to a 2D image synthesis problem, which is more intuitive, easier, and conceptually simpler. While the conformality of the parameterization naturally preserves the angles of the texture, in this paper we provide a multi-scale technique to also maintain a more uniform area scaling factor. A second novel contribution is to employ the global parameterization to simultaneously preserve orthogonality and size in texture mapping. For this, we show that a conformal factor-driven mass-spring method offers a convenient way to trade off these two qualitative metrics. Our algorithms are simple, efficient and automatic, and they are theoretically sound and universal to general surfaces as well.

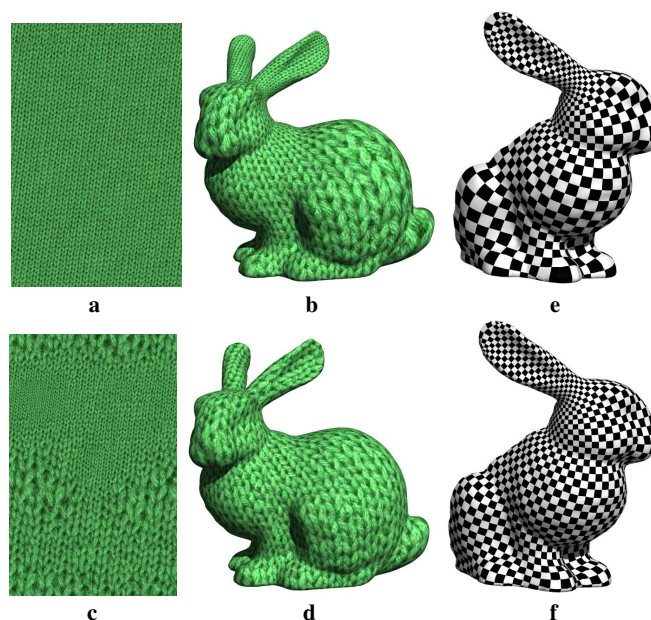
**Keywords** Texture synthesis · Texture mapping · Riemann surfaces · Global conformal parameterization

## 1 Introduction

Texture mapping is a technique that is used to map 2D images to both planar and curved surfaces in order to enhance their visual effect. Texture synthesis has similar goals, but instead of using the explicit tiling approach of texture mapping, it aims to generate the surface decor from a relatively

Lujin Wang, Xianfeng Gu, Klaus Mueller  
Department of Computer Science, Stony Brook University, NY 11794-4400, USA  
E-mail: {lujin|gu|mueller}@cs.sunysb.edu

Shing-Tung Yau  
Department of Mathematics, Harvard University, Cambridge, MA 02138, USA  
E-mail: yau@math.harvard.edu



**Fig. 1** Texture synthesis and texture mapping using global conformal parameterization. (a) uniform texture synthesized on geometry images without considering area stretching factor; (b) nonuniform texture by mapping (a) on the 3D surface; (c) nonuniform texture synthesis considering area stretching factor; (d) uniform texture by mapping (c) on the 3D surface; (e) conformal texture mapping based on global parameterization; (f) more uniform texture mapping using our quasi-isometric parameterization.

small texture sample. Both approaches are useful in their own right. Texture mapping is the technique to use when the goal is an exact preservation of the detail in the provided texture sample, but it tends to suffer from repetition and tiling artifacts, as well as seams, and it also requires overhead for texture storage. The detail generated in texture synthesis, on the other hand, is not an exact match, but only similar to the provided texture sample. It, however, is seamless and non-repetitive. Texture synthesis replaces the memory overhead of texture mapping with computational overhead incurred by the synthesis process. Both methods have similar demands in terms of avoiding local and global distortion, which, as we

will show, can be achieved by preserving the texture samples local angles and global scale during the mapping or the synthesis process, using a conformal parametrization approach.

*Global conformal parameterization* was introduced in [13]. It guarantees that the shapes embodied in the textures are preserved on the surface, and it is global, which means there are no seams or cracks. Furthermore, the parameterization can segment the surface into patches, where each patch is mapped to a planar rectangle. This is valuable for real applications. The existence of the global conformal parameterization is equivalent to the fact that all oriented surfaces are Riemann surfaces [15]. Therefore, global conformal parameterization discovers more profound geometric structures on surfaces. For example, it induces the so-called affine structure, which is the foundation for generalizing splines with planar domains to be defined on surfaces [12]. It applies the concept of a differential form from Riemann surface theory [15], which can be interpreted as a pair of smooth vector fields orthogonal to each other.

The intrinsic difficulties for texture synthesis are due to two main aspects. The first originates from the local geometric properties of the surface. The texture image is defined on a flat planar region, and once it is mapped to the curved surface region, there must be distortion, which is challenging to mediate. The second is caused by topology. Since the surface and plane are topologically different, there is no global one-to-one smooth mapping between them. Thus the existence of seams and singularities is unavoidable, as well as it is challenging to remove these seams and minimize the number of singular points. While texture synthesis applied directly on surfaces (see Section 2) can do a reasonably good job, it requires a surface flow analysis which can be complicated. It is more intuitive to do texture synthesis on a 2D plane, where the sample itself is also defined. Conformal parameterization offers a convenient way to do this, and this is topic of the first half of the paper.

For texture mapping, while the angle is already preserved using the conformal mapping, a related application is to also seek the preservation of scale for texture mapping. Although it is mathematically impossible to make the parameterization preserve both the angle and the area, we can try to describe a convenient paradigm with which the two can be traded off. This is the focus of the second half of the paper.

Compared to traditional methods, our texture synthesis method based on global conformal parameterization has the following advantages:

- *Global structure.* Traditional texture synthesis methods are unable to produce textures with strong global structures, because they generate the textures locally and extend to cover the surface without coherent global consideration. In practice, it is highly desirable to generate textures with global structures. Since our method is based on global parameterization, it is easy to synthesize globally structured textures.
  - *Purely 2D operation.* Traditional methods need to march on the 3D surfaces, and the data structure and the operations for this are complicated. In our method, all operations are entirely performed in the 2D parameter domain, which is much simpler and more efficient.
- Further advantages of our conformal parameterization method, both for texture mapping and synthesis, are:
- *Angle preservation.* Conformal parameterization preserves the angles from the surface to the parameter plane. Therefore, the local mapping from the texture to the surface is just a scaling without angular distortion. The synthesis method then needs to only focus on the scaling factor, without having to pay attention to angle changes.
  - *Regularity.* Global conformal parameterization induces a canonical way to segment the surface, such that each segment is conformally parameterized by a rectangle (Fig. 2). The regular pattern of this global parameterization is helpful to simplify the algorithms.
  - *Rigor.* Global conformal parameterization is based on solid geometric theories, and based on the parameterization, the distortion of textures on surfaces can be quantitatively measured. This makes it convenient for quality control.

---

## 2 Related Work

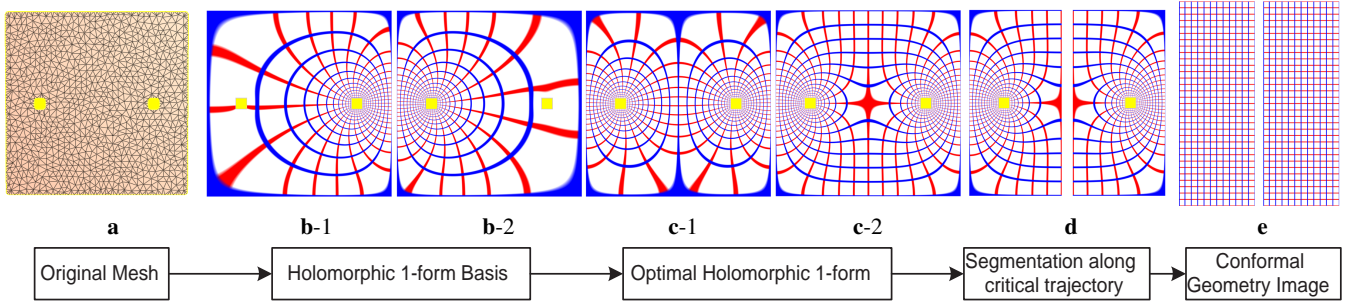
Before presenting our new method, we shall discuss related work in two areas: texture synthesis and global conformal parameterization.

**Texture synthesis** 2D Texture synthesis methods are either pixel-based [9, 30, 1], or patch-based [32, 8, 19, 16, 2]. Pixel-based synthesis algorithms are more flexible since textures are synthesized pixel by pixel by finding a matching neighborhood, while patch-based methods tile matched patches together and tend to be faster and more stable. Hybrid methods [20] lay out patches and use pixel-based algorithm to hide the seams.

Surface texture synthesis extends these 2D texture synthesis methods to synthesize texture directly on the surface, including pixel-based methods [31, 29, 33, 27] and pattern mapping methods [21, 23, 25, 5]. With these methods, the discontinuities and cracks almost disappear, while the distortion problem is greatly reduced.

Another way to put textures on the surface is a texture mapping based on parameterization. But texture periodicity is obvious and often spoils the visual quality of the results.

Traditional surface texture synthesis methods are generally performed on the 3D surfaces to be decorated. In contrast, our paper proposes a novel method which synthesizes the texture in the 2D parameter space, which is easier, more intuitive, and conceptually simpler. By using global conformal parameterization, the synthesized textures will not suffer from angular distortion. But we also have to deal with area stretching problems during the mapping. We will achieve this by using variable-size textures with local control. While most texture synthesis methods generate textures with uniform size features, textures with variant feature size are generated in [28, 16]. Their methods synthesize textures variant



**Fig. 2 Process of Global Conformal Parameterization.** The input surface is represented as a triangle mesh (a). The holomorphic 1-form basis is computed, (b-1) and (b-2) are the 2 base 1-forms [13]. By linear combining the basis, different holomorphic 1-forms can be constructed as shown in (c-1) and (c-2), then the optimal 1-form with most uniform 1-form is selected (c-2) [14]. The red and blue curves are the horizontal and vertical trajectories. Horizontal trajectories intersect at the zero point, the center of (c-2). The mesh is then segmented along the horizontal trajectory through the zero point as shown in (d), each segment is conformally parameterized to a planar rectangle illustrated in (e). The trajectories are mapped to the iso-parametric curves.

with respect to certain directions, and local control changes slightly compared to our technique.

**Conformal parameterization** Several recent advances in surface parameterization [10] have been based on solving a discrete Laplace system [22, 6]. Lévy et al. [18] describe a technique for finding conformal mappings by least squares minimization of *conformal energy*, and Desbrun et al. [4] formulate a theoretically equivalent method of discrete conformal parameterization. Sheffer et al. [24] give an angle-based flattening method for conformal parameterization. Gu and Yau [13] considered construction of a global conformal structure for a manifold of arbitrary topology by finding a basis for holomorphic differential forms. Gortler and Gotsman proposed one forms on meshes in [11] and applied for surface parameterization. Degerner et al. [3] provided user control to trade off between angle and area preservation.

Lai et al. [17] synthesize geometric textures based on geometry images. Their synthesized geometric textures will have more distortions than ours, because we use an improved global conformal parameterization with segmentation.

### 3 Global Conformal Parameterization

We assume that the reader is familiar with the basics of complex analysis and differential geometry. A detailed explanation for these concepts can be found in [15].

Suppose  $M$  is a surface with handles, either open or closed. A *global conformal parameterization* is a map  $\phi : M \rightarrow R^2$ , such that each point  $p$  on  $M$  is mapped to a point on the planar parameter domain  $\phi(p) = (u(p), v(p))$ . Furthermore,  $\phi$  is *angle preserving*, this is equivalent to the following fact: suppose we arbitrarily draw two intersecting curves  $\gamma_1, \gamma_2$  on  $M$ , the intersection angle is  $\alpha$ , then their images  $\phi(\gamma_1)$  and  $\phi(\gamma_2)$  are planar curves, the intersection angle is also  $\alpha$ . Mathematically, the conformality of the parameterization is formulated in the following way: the first fundamental form of  $M$  under conformal parameterization  $(u, v)$  is represented as

$$ds^2 = \lambda^2(u, v)(du^2 + dv^2), \quad (1)$$

where  $\lambda$  is called the *conformal factor*, it indicates the area ratio between the area on  $M$  and that on the plane.

In practice, it is more convenient to compute the gradient fields of  $\phi$ , namely  $(\nabla u, \nabla v)$ . If  $\phi$  is conformal, then they satisfy the following criteria:

$$\nabla v(p) = \mathbf{n}(p) \times \nabla u(p),$$

where  $\mathbf{n}(p)$  is the normal at the point  $p$ , also

$$\nabla \times \nabla u = \nabla \times \nabla v = 0,$$

because the gradient fields are curl-free. Formally, a pair of vector fields satisfying the above conditions is a *holomorphic 1-form*. There is an infinite number of this kind of vector fields, they form a  $2g$  dimensional real linear space, where  $g$  is the number of handles (genus) of  $M$ . The method of computing holomorphic 1-form basis is introduced in [13].

The concept of holomorphic 1-form and the computational procedure are demonstrated in Fig. 2. For simplicity, we only illustrate a naive example: a planar 2-hole square. In practice, the pipeline works for all 3D surfaces with arbitrary topologies. The red curves are the integration curves  $\nabla u$  and called horizontal trajectories, the blue curves are the integration curves of  $\nabla v$  and are called vertical trajectories. These trajectories are the preimages of the iso- $u$  and iso- $v$  curves. Fig. 2(c-1) and (c-2) show different holomorphic 1-forms, (b-1) and (b-2) are the bases for the linear space of all holomorphic 1-forms.

From the infinite set of holomorphic 1-forms, we need to pick the best one for our texture synthesis. We choose the one with the most uniform conformal factor using the method introduced in [14], as shown in Fig. 2(c-2).

The global behavior of the trajectories are very complicated. From Fig. 2, it is obvious that the vertical and horizontal trajectories are orthogonal everywhere and two horizontal trajectories do not intersect each other in general. There are special points on  $M$ , where two horizontal trajectories intersect (two vertical trajectories also intersect). It can be proven that, at those points, the conformal factors are zero, therefore, such kind of points are called *zero points* of the holomorphic 1-form. In general, for a genus  $g$  closed surface,

there are  $2g - 2$  zero points. In Fig. 2(c-2), the intersection points of the red curves is the zero point. The trajectories through zero points are called *critical trajectories*.

A trajectory can be a finite circle, a finite curve segment terminating at the boundaries, or an infinite spiral dense on the surface. If the horizontal critical trajectories are finite, then the whole family of horizontal trajectories are finite due to [26]. In practice, for simplicity, we choose a holomorphic 1-form with finite horizontal trajectories.

The critical horizontal trajectories segment the surface  $M$  into several connected components, each component is either a topological disk or a topological cylinder and can be parameterized by  $\phi$  to a planar rectangle. Fig. 2(d) illustrates this fact, the critical horizontal trajectory segments the surface into 2 patches, and each is conformally mapped to a rectangle. The horizontal trajectories are mapped to the iso- $v$  curves (red), while the vertical trajectories are mapped to the iso- $u$  curves (blue).

In practice, it is convenient to synthesize the textures on these rectangular parameter domains. Therefore, in our algorithm, we locate the zero point first by finding a vertex with minimal conformal factor, then trace the horizontal trajectory to segment the surface.

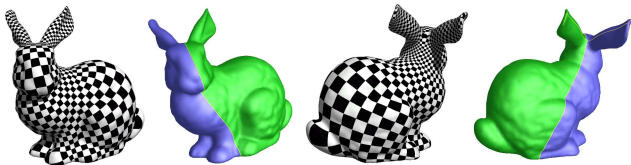


Fig. 3 Global conformal parameterization.

Fig. 3 illustrates a global conformal parameterization of the Stanford bunny surface. The bunny surface has 3 boundaries, two are at the tips of ears, one is at the bottom, therefore, it is topologically equivalent to the 2-hole disk in Fig. 2. The double covered surface is of genus 2. A zero point is between the roots of the two ears. The horizontal trajectories through it are illustrated as yellow curves. The whole surface is partitioned into 2 topological disks, each segment is color-encoded. Fig. 5(a) and (b) demonstrates that each segment can be conformally mapped to a rectangle on the plane.

Textures can be easily synthesized on those rectangles directly. For convenience, in the following discussion, we call each surface component with its conformal parameters a *conformal geometry image*.

## 4 Uniform Texture Synthesis

Global conformal parameterization on a 3D surface (see Fig. 3) induces conformal geometry images (see Fig. 5(a)(b)), which allow textures to be easily mapped to the surface without angular distortion. Unfortunately, the area stretching of textures is unavoidable, as is shown in Fig. 4. Ideally, we want to preserve both the angle and the area of the texture

on the surface, that is, we want to find an *isometric* parameterization. Although in theory this is definitely impossible, in practice, we are able to improve the texture synthesis method to make it as isometric as possible.

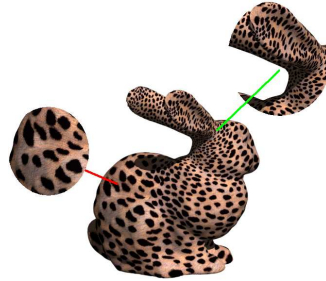


Fig. 4 Nonuniform texture on a surface. It is generated by global conformal parameterization, uniform texture synthesis on 2D geometry images and texture mapping.

We propose a multi-scale texture synthesis method to generate uniform textures on the surface. This method synthesizes nonuniform textures on a 2D geometry image by considering the area stretching factor (the inverse of the conformal factor in Eq. 1) in order to obtain the uniform 3D textures. The estimation of the area stretching factor on the conformal geometry images will be introduced first, and then the details of our multi-scale synthesis algorithm will be described.

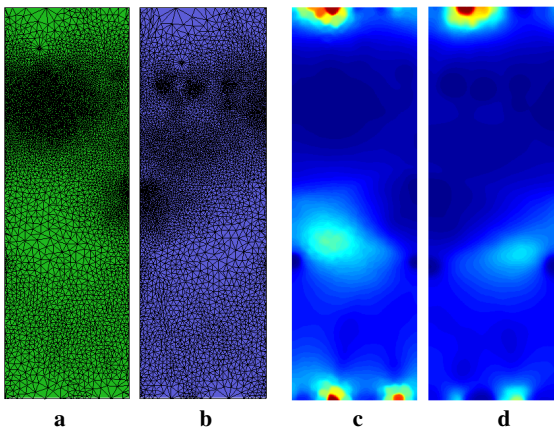
### 4.1 Estimation of the conformal factor

The conformal factor indicates the amount of area stretching from the 3D surface to the 2D parameter domain. Our goal is to calculate the inverse conformal factor field on the geometry image. The inverse conformal factor is  $\tau$ , and  $\tau = \frac{1}{\lambda}$ , where  $\lambda$  is the conformal factor in Eq. 1. If the area shrinks from the 3D mesh to the 2D plane,  $\tau$  is smaller than 1, otherwise,  $\tau$  is larger than 1. This field will be used to choose the appropriate scale level of the sample texture when we synthesize textures on certain regions of the geometry image.

First, we normalize the parameters of each conformal geometry image. Then we choose the maximal size  $C$  for each dimension of the synthesized textures. The size of the output texture is simply the product of  $C$  and the normalized parameter for each geometry image. The size of the output texture affects the speed and the quality of the synthesis, and also the texture feature size on the surface. For all results shown in this paper, we set  $C$  to be 1024.

By using Eq. 1, the values of  $\tau$  on the vertices are easily calculated directly from the geometry image with the original mesh connectivity. The  $\tau$  values of the other texels are then interpolated using a Gaussian radial basis function (RBF). The calculated inverse conformal factor fields of two geometry images are illustrated in Fig. 5(c)(d). Here, whenever the color changes from deep blue to red, the inverse conformal factor value increases gradually.





**Fig. 5** Conformal geometry images (a) and (b), and corresponding inverse conformal factor fields (c) and (d).

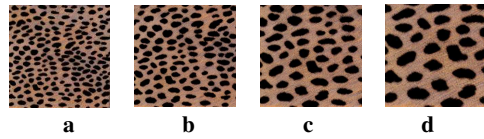
#### 4.2 Multi-scale synthesis algorithm

Most texture synthesis methods synthesize textures without or with quite simple size variations of texture features [28, 16]. In contrast, we use the conformal factor to control the local scale of the texture. Therefore, the output texture is still similar to the sample texture, but with different feature sizes in different regions.

Our multi-scale synthesis algorithm is based on the patch-based texture synthesis method. Although pixel-based synthesis methods or hybrid methods should also work, better quality can be obtained using patch-based method according to our experiments. We put equal sized texture patches in the order of image quilting [8], and use the graph cut algorithm [16] to hide the seams of neighboring patches. The patch size is chosen according to texture features depending on the input texture, we use 30 to 50 pixels as patch width for the results in this paper. In the synthesis process, we choose a patch, not just from a single sample texture, but from multiple scale levels of the sample texture. First, we calculate the average value of the inverse conformal factor in the region covered by a patch; Then we decide an appropriate scale level based on this average value. From the sample texture of that scale level, we find the best matched patch to fit the neighboring patches and put it on the output texture.

**Multi-scale sample textures** In order to preserve memory and improve speed, we store a certain number of discrete scale levels of sample textures (see Fig. 6). We call the enlarged sample texture the *high-scale sample texture*, and the minified texture the *low-scale sample texture*. For better quality, the scale between neighboring levels is not a power of 2. The parameters in our algorithm include the highest scale, the lowest scale, and the desired levels, which can be specified by the user and affect the size and the quality of the synthesized textures. Different scale level textures are then generated by cubic interpolation.

For regions with higher inverse conformal factors, higher level sample textures should be chosen, because the texture

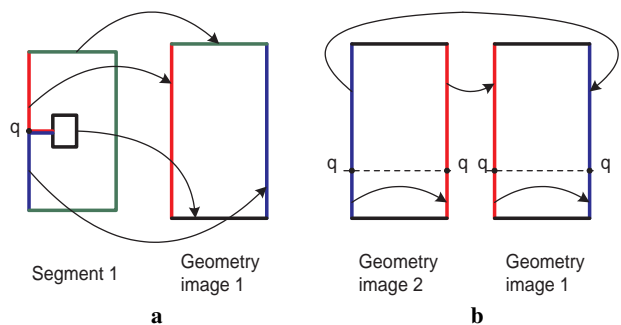


**Fig. 6** Multi-scale sample textures. From (a) to (d), scales of sample textures increase gradually.

mapping will shrink the texture. Similarly, for regions with lower inverse conformal factors, lower level textures should be selected, because texture mapping will enlarge the texture.

The lowest level needs to be determined with caution, because the sampling rate is reduced when the texture is minified. Depending on the feature size of the texture, important features may be lost irrecoverably if the sampling rate is set below a certain level. We place our original sample texture close to the lowest level to lower the risk associated with downsampling. In contrast, high scale level texture does not suffer from this problem, and can be used safely.

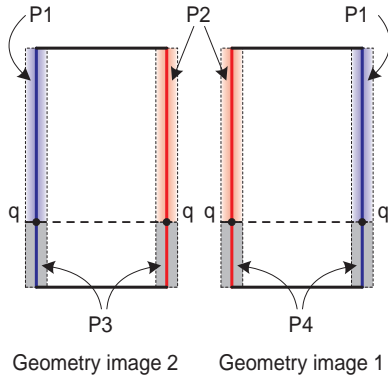
**Preserving boundary consistency** Since the surface is segmented and mapped to more than one geometry image, the boundary consistency problem needs to be addressed carefully. When mapping a segment (see Fig. 2(d)) to its conformal geometry image, boundaries on the segment are mapped to boundaries on the geometry image, respectively. Fig. 7(a) shows the mapping of a segment (segment 1) to its geometry image (geometry image 1). Fig. 7(b) shows the boundary correspondences of this geometry image to another geometry image (geometry image 2), due to an adjoining segment. Corresponding boundary parts are neighbors in 3D space, and therefore must have consistent textures.



**Fig. 7** Boundary problem. (a) corresponding boundaries on a segment and its conformal geometry image; (b) corresponding boundaries on two geometry images.  $q$  is the zero point.

Our solution to synthesize textures consistently across the corresponding boundaries is as follows. First, we add margins to geometry images which have boundary parts corresponding to other boundary parts, as shown in Fig. 8. Here,  $P1$  to  $P4$  are margins we added. During synthesis, after textures on  $P1$  of geometry image 1 is synthesized, the textures are copied to fill  $P1$  of geometry image 2. Then, when synthesizing textures on geometry image 2, for patches overlapped with margins, the matched patch will be chosen with

additional constraints, treating the overlapped parts as already synthesized pixels.



**Fig. 8** Consistent boundary synthesis, by adding margins and copying boundary texture patches.  $q$  is the zero point.

Therefore, the patch-based synthesis algorithm is slightly modified to cope with different patch-overlapping situations, which solves the boundary problem. This way, our patch-based synthesis algorithm can easily generate periodical textures, which is quite useful in texture mapping as well.

#### 4.3 Texture synthesis results

Our texture synthesis results are demonstrated in Fig. 1(a)-(d) and Fig. 9. When we synthesize uniform textures on geometry images without considering area stretching, the texture feature sizes on different regions on the surface are highly non-uniform. In contrast, by using our multi-scale synthesis method, textures on the surface are quite uniform.

## 5 Quasi-isometric Texture Mapping

The multi-scale texture synthesis method just presented modifies the textures directly to improve the uniformity of the synthesized texture on the surface. In contrast, the method for texture mapping, introduced in this section, revises the parameters instead.

In theory, it is impossible to make the parameterization preserve both the angle and the area. In that case, the parameterization would be an isometry, with a surface of zero Gaussian curvature (that is, a flat surface). But what we can accomplish is to make the parameterization on the interior of one component as isometric as possible and in return sacrifice some of the angle structure along the boundaries. We apply a mass-spring method to achieve this *quasi-isometric* parameterization, which is close to the desired isometric one.

Fig. 10 illustrates the basic idea. The original conformal geometry image has a highly non-uniform density, whereas preserving the angle. After the process, the mesh (b) with quasi-isometric parameters has more uniform density, but

the boundaries are distorted. Hence, the boundary consistency is sacrificed. On the other hand, the stretch-minimizing method of Yoshizawa et al. [34] fixes the boundary vertices and therefore can keep the boundary rectangular, but the anisotropic texture stretching is considerably higher.

### 5.1 Mass-spring model

The mass-spring model is carried out on the conformal geometry images using the original mesh connectivity. The mass-spring system is modelled as follows: each vertex is treated like a node and each edge as a spring. The motion of all nodes is confined to the 2D parameter plane.

We denote the parameterization of the conformal geometry image as  $\phi : U \rightarrow R^2$ , where  $U$  is the conformal geometry image. Then, the mass-spring evolution can be formulated as  $\delta\phi(v) = \varepsilon\mathbf{F}(v)$ , where  $\varepsilon$  is a constant carefully chosen to ensure no flipping of triangles. In practice,  $\varepsilon$  is inversely proportional to the maximum magnitude of the force field. Here,  $\mathbf{F}$  is the *external force*, and calculated as

$$\mathbf{F}(v) = \sum_u \eta(u)\eta(v)(\phi(v) - \phi(u)), \eta(v) = \frac{1}{n} \frac{|\mathbf{r}(u) - \mathbf{r}(v)|}{|\phi(u) - \phi(v)|} \quad (2)$$

where  $u$  runs through all neighboring vertices of vertex  $v$ ,  $n$  is the valence of vertex  $v$ , and  $\mathbf{r}(v)$  is the 3D position of vertex  $v$ .

In Eq. 2,  $\eta^2(v)$  is a discrete approximation of the conformal factor at  $v$ . Intuitively, the external force is proportional to the conformal factor, and expands the regions with high conformal factors. The nodes on the parameter domain with higher density will be expanded gradually and make the distribution more uniform, that is, the process will improve the parameterization to be closer to an isometry.

In our implementation, we use the mass-spring model code for arbitrary nodes in [7]. Fig. 10 demonstrates the improvement of the parameterization using our mass-spring algorithm. The improved parameters are used for texture mapping. Fig. 1(e) is the texture mapping result based on global conformal parameterization, while Fig. 1(f) is the result after conformal parameterization and mass-spring relaxation, upon which the squares on the checkerboard become more isometric.

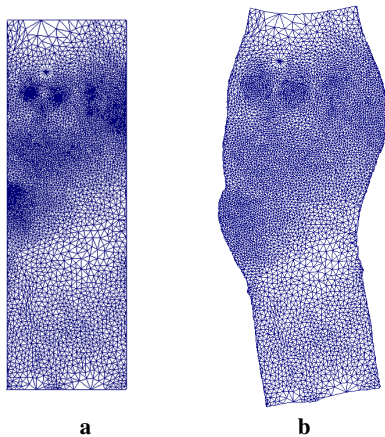
### 5.2 Texture mapping results

Fig. 11 compares the results obtained with and without our quasi-isometric parameterization method, for the task of mapping 2D textures onto 3D models. Fig. 11(a) shows the outcome of an image-to-surface mapping via standard global conformal parameterization, while Fig. 11(b) shows the result obtained when applying the mass-spring model to the conformal map first. We observe that the uniformity of the parameterization is greatly improved. And video can also be mapped to or synthesized on the surface with considerably better quality. One frame of our video (Matrix) on the surface is shown in Fig. 11(c) and (d). We should note that while



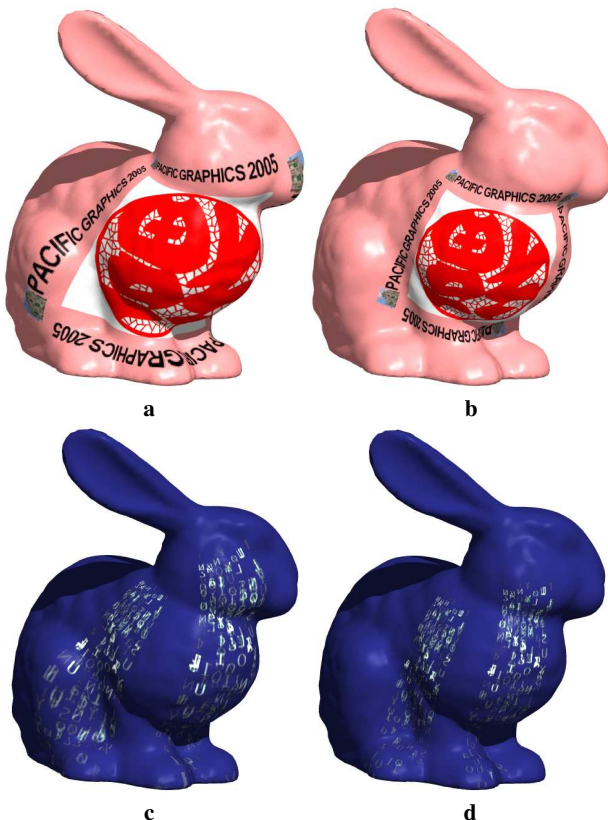


**Fig. 9** Multi-scale texture synthesis results. (a),(e) uniform texture synthesized on geometry images without considering area stretching factor; (b),(f) nonuniform texture by mapping (a) and (e) onto 3D surfaces; (c),(g) nonuniform texture synthesis considering area stretching factor; (d),(h) uniform texture by mapping (c) and (g) onto 3D surfaces; (i)-(l) more uniform texture synthesis results; (m),(o) nonuniform textures on 3D surfaces, texture features inside the handles are smaller than those outside; (n),(p) uniform textures on 3D surfaces. High resolution images as well as videos can be obtained at <http://www.cs.sunysb.edu/~lujin/paper/pg05/>.



**Fig. 10** Mesh changed on mass-spring model. (a) mesh on one geometry image; (b) modified mesh with mass-spring relaxation.

the mass-spring relaxation process is relatively slow (about 1 hour for the bunny model), it only needs to be done once for each model, and after that the improved parameterization results can be reused for various image and video mappings. The extra cost for storage is minimal.



**Fig. 11** Texture mapping results. (a),(b) image on the surface, (c),(d) video on the surface, in which (a) and (c) are based on conformal global parameterization, (b) and (d) are based on improved parameterization using mass-spring method.

## 6 Conclusions

In this paper, we have presented novel methods to accomplish distortion-minimized texture synthesis and texture mapping on 3D surfaces. For this, we have augmented the conformal mapping approach, which preserves angular fidelity, with a process that controls the distortion of scale. For texture synthesis on 3D surfaces, it allows the synthesis process to be done intuitively in 2D space and, afforded by the conformal mapping, achieves global control over the mapping result. The synthesis result is locally angle-torsion free, while globally it is continuous. Further, we also devised a method based on a mass-spring model which offers a good trade-off for angular distortion and size preservation in texture mapping. Both methods are conveniently implemented using conformal mapping, are simple and efficient, and are universal for arbitrary surfaces.

While we currently do not provide explicit controls to balance angular and size distortions, we plan to incorporate those in future work, using the updated conformal factor fields.

**Acknowledgements** This research was supported by NSF career grant ACI-0093157 and NSF career grant CCF-0448399. Models are courtesy of Stanford University, Headus Inc. and New York University. Special thanks to Miao Jin for generating the geometry images, Christopher Carner for proof reading the paper, and the anonymous reviewers for their thoughtful comments.

## References

1. Ashikhmin, M.: Synthesizing natural textures. In: ACM Symposium on Interactive 3D Graphics, pp. 217–226 (2001)
2. Cohen, M.F., Shade, J., Hiller, S., Deussen, O.: Wang tiles for image and texture generation. In: Proc. of SIGGRAPH '03, vol. 22, pp. 287–294 (2003)
3. Degener, P., J., M., Klein, R.: An adaptable surface parameterization method. In: Proc. of 12th International Meshing Roundtable, pp. 201–213 (2003)
4. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. In: Proc. of Eurographics '02, vol. 12, pp. 209–218 (2002)
5. Dischler, J.M., Maritaud, K., lévy, B., Ghazanfarpour, D.: Texture particles. Computer Graphics Forum 21(3), 401–410 (2002)
6. Duchamp, T., Certain, A., DeRose, A., Stuetzle, W.: Hierarchical computation of PL harmonic embeddings. Tech. rep., University of Washington (1997)
7. Eberly, D.H.: Game Physics. Morgan Kaufmann (2004)
8. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proc. of SIGGRAPH '01, pp. 341–346 (2001)
9. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: International Conference on Computer Vision, vol. 2, pp. 1033–1038 (1999)
10. Floater, M.S., Horman, K.: Surface parameterization: a tutorial and survey. In: Advances in Multiresolution in Geometric Modelling. Springer (2004)
11. Gortler, S.J., Gotsman, C., Thurston, D.: One-forms on meshes and applications to 3d mesh parameterization. Tech. Rep. CS TR-12-04, Harvard University (2004)
12. Gu, X., He, Y., Qin, H.: Manifold splines. In: ACM Symposium on Solid and Physical Modeling '05 (2005)
13. Gu, X., Yau, S.T.: Global conformal surface parameterization. In: Proc. Eurographics/SIGGRAPH Symposium on Geometry Processing, pp. 127–137 (2003)



14. Jin, M., Wang, Y., Yau, S.T., Gu, X.: Optimal global conformal surface parameterization. In: Proc. of IEEE Visualization '04, pp. 267–274 (2004)
15. Jost, J.: *Compact Riemann Surfaces*. Springer (2000)
16. Kwatra, V., Schöl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. In: Proc. of SIGGRAPH '03, vol. 22, pp. 277–286 (2003)
17. Lai, Y.K., Hu, S.M., Gu, X., Martin, R.R.: Geometric texture synthesis and transfer via geometry images. In: ACM Symposium on Solid and Physical Modeling (2005)
18. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. In: Proc. of SIGGRAPH '02, vol. 21, pp. 362–371 (2002)
19. Liang, L., Liu, C., Xu, Y., Guo, B., Shum, H.Y.: Real-time texture synthesis by patch-based sampling. *ACM Trans. on Graphics* **20**(3), 127–150 (2001)
20. Nealen, A., Alexa, M.: Hybrid texture synthesis. In: Proc. of the 14th Eurographics workshop on Rendering, pp. 97–105 (2003)
21. Neyret, F., Cani, M.P.: Pattern-based texturing revisited. In: Proc. of SIGGRAPH 1999, pp. 235–242 (1999)
22. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugate. *Experimental Mathematics* **2**(1), 15–36 (1993)
23. Praun, E., Finkelstein, A., Hoppe, H.: Lapped textures. In: Proc. of SIGGRAPH '00, pp. 465–470 (2000)
24. Sheffer, A., Sturler, E.: Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with Computers* **17**(3), 326–337 (2001)
25. Soler, C., Cani, M.P., Angelidis, A.: Hierarchical pattern mapping. In: Proc. of SIGGRAPH '02, vol. 21, pp. 673–680 (2002)
26. Strebel, K.: *Quadratic Differentials*. Springer-Verlag (1984)
27. Tong, X., Zhang, J., Liu, L., Wang, X., Guo, B., Shum, H.Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. In: Proc. of SIGGRAPH '02, vol. 21, pp. 665–672 (2002)
28. Tonietto, L., Walter, M.: Towards local control for image-based texture synthesis. In: Proc. of XV Brazilian Symposium on Computer Graphics and Image Processing, pp. 252–258 (2002)
29. Turk, G.: Texture synthesis on surfaces. In: Proc. of SIGGRAPH '01, pp. 347–354 (2001)
30. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proc. of SIGGRAPH '00, pp. 479–488 (2000)
31. Wei, L.Y., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: Proc. of SIGGRAPH '01, pp. 355–360 (2001)
32. Xu, Y., Guo, B., Shum, H.Y.: Chaos mosaic: fast and memory efficient texture synthesis. Tech. Rep. MSR-TR-2000-32, Microsoft Research (2000)
33. Ying, L., Hertzmann, A., Biermann, H., Zorin, D.: Texture and shape synthesis on surfaces. In: Proc. of 12th Eurographics Workshop on Rendering '01, pp. 301–312 (2001)
34. Yoshizawa, S., Belyaev, A., Seidel, H.P.: A fast and simple stretch-minimizing mesh parameterization. In: Proc. of Shape Modeling and Applications '04, pp. 200–208 (2004)



**Lujin Wang** is a Ph.D. candidate in Computer Science at Stony Brook University, where she is also a research assistant at the Center for Visual Computing. She received her B.E. and M.S. in Computer Science and Engineering from Beijing University of Aeronautics and Astronautics, China, in 1995 and 1998, respectively. From 1998 to 2001, she was a Ph.D. candidate in Computer Science at Tsinghua University, China. Her research interests include computer graphics and scientific visualization.



**Xianfeng David Gu** is an Assistant professor in Computer Science at Stony Brook University. He earned his Ph.D. in Computer Science from Harvard University in 2003. He won the NSF CAREER award in 2004. His research interests are computer graphics, computer vision and medical imaging. His major works include geometry images, global conformal surface parameterization, manifold splines and computational conformal geometry. For more information see <http://www.cs.sunysb.edu/~gu>.



**Klaus Mueller** is an Assistant Professor in Computer Science at Stony Brook University, where he also holds co-appointments at the Biomedical Engineering and the Radiology Departments. He earned an M.S. in Biomedical Engineering in 1991 and a Ph.D. in Computer Science in 1998, both from The Ohio State University. His current research interests are computer graphics, visualization, medical imaging, face recognition, and augmented reality. He won the NSF CAREER award in 2001 and has authored more than 70 journal and conference papers. He has served as a program co-chair at various conferences, such as the Volume Graphics Workshop, IEEE Visualization, and the Symposium on Volume Visualization and Graphics, and he is a member of IEEE and ACM. For more information see <http://www.cs.sunysb.edu/~mueller>.



**Shing-Tung Yau** is a Professor in Mathematics at Harvard University (since 1987). He received a Ph.D. in Mathematics from the University of California, Berkeley, in 1971. He held appointments at a number of institutions, including Stony Brook University (1972-73), Stanford University (1974-79), the Institute for Advanced Study, Princeton, N.J. (1979-84), the University of California, San Diego (1984-87), Chinese University of Hong Kong (1991-92), National Tsing Hua University, Taiwan (1991-92), and University of Cambridge (1994). He has received a number of prominent awards, including the Fields Medal in 1983, a MacArthur Fellowship in 1984, the Crafoord Prize in 1994, and the (U.S.) National Medal of Science in 1997. He was elected a member of the (U.S.) National Academy of Sciences in 1993, and a Foreign Member of the Russian Academy of Sciences in 2003. His research interests include differential geometry, differential equations and general relativity.