Visual Comput (2007)
DOI 10.1007/s00371-007-0115-x

ORIGINAL ARTICLE

Chunxia Xiao
Wenting Zheng
Yongwei Miao
Yong Zhao
Qunsheng Peng

A unified method for appearance and geometry completion of point set surfaces

C.X. Xiao (✉) · W.T. Zheng · Y.W. Miao · Y. Zhao · Q.S. Peng
State Key Lab of CAD&CG, Zhejiang University, Hangzhou, P.R. China
{cxxiao, wtzheng, miaoyw, zhaoyong, peng}@cad.zju.edu.cn

C.X. Xiao
Computer School, Wuhan University, Wuhan, P.R. China

Abstract This paper presents a novel approach for appearance and geometry completion over point-sampled geometry. Based on the result of surface clustering and a given texture sample, we define a global texture energy function on the point set surface for direct texture synthesis. The color texture completion is performed by minimizing a constrained global energy using the existing surface texture on the surface as the input texture sample. We convert the problem of context-based geometry completion into a task of texture completion on the surface. The geometric detail is then peeled and converted into a piece of signed gray-scale texture on the base surface of the point set surface. We fill the holes on the base surface by smoothed extrapolation and the geometric details over these patches are reconstructed by a process of gray-scale texture completion. Experiments show that our method is flexible, efficient and easy to implement. It provides a practical texture synthesis and geometry completion tool for 3D point set surfaces.

Keywords Texture synthesis · Texture completion · Geometry completion · Geometric detail

1 Introduction

As numerous 3D surface scanning devices have become available in recent years, 3D scanning has become a major approach for acquiring the shape of complex 3D objects. However, obtaining a fine and usable 3D model from the acquired surface samples is still a difficult task. Due to occlusions, low reflectance, and measurement error in the scanning, the acquired geometry is frequently imperfect, that is, it contains holes. In addition, large holes may also be introduced by surface editing operations. These holes have to be filled in a manner that not only conforms to the global shape of the entire surface but also exhibits its primary geometric detail. The color texture of the defective surface should also be repaired consistently. Therefore, example-based texture synthesis on 3D point set surfaces is also in demand to facilitate downstream processing.

Compared with the problem of inpainting and texture synthesis of 2D images, geometry completion and texture synthesis on 3D point-sampled geometry is more challenging for several reasons. Three-dimensional point sampling is irregular and does not give rise to a regular parameter domain as the image. In addition, similarity measurements between the point sets are difficult to define. To our knowledge, little work has been done on color texture synthesis over point set surfaces. Alexa et al. [1] and Clarenz et al. [4] addressed the issue as an application of their algorithm. Recently, some researchers have focused on the task of appearance and geometry completion of defective point set surfaces. Nevertheless, patches generated by most of these algorithms are too smooth, conveying no geometric details [2, 3, 6, 13, 14, 17, 26]. Other methods either have to deal with the complex boundary conditions to solve a PDE equation [16, 18] or have to determine the geometry similarities in 3D [18, 22].

In this paper, we propose a novel method for texture synthesis and context-based geometry completion on a point-sampled surface, and our major contributions are as follows:

Firstly, we present a novel texture synthesis and texture completion algorithm that directly works on the irregular sampled point set surfaces. The algorithms are based on global optimization and produce smooth texture synthesis effects while keeping the intrinsic structures of the sample texture over the point-sampled geometry. We also present algorithms to enhance the roughness appearance of the texture synthesized surface in rendering. By regarding the textures on the existing surface as the input texture sample, the texture completion can be accomplished by optimizing a constrained global texture energy function.

Secondly, based on covariance analysis, we propose a mean curvature flow for filtering point-sampled geometry. We apply the filtering operation to derive the base surface of the sampled geometry. We define the geometric details as the displacement between the surface and its base surface. The geometric details are then peeled off and converted into a signed gray-scale texture attached to the base surface for downstream processing.

Thirdly, our approach reconstructs the geometric details on the smooth patch by implementing texture completion in the gray-scale texture space. Therefore, compared with Sharf et al. [22], we can avoid some complex operations such as the similarity measurement, and rigid transformation of the 3D points set are also avoided. An overview of our entire geometry completion processing pipeline is illustrated in Sect. 6.

We can achieve consistent context-based completion and can deal with more complex boundary conditions compared with the texture and geometry completion algorithms that employ PDE [16, 18]. Our method uses a simple local parameterization scheme, and it therefore avoids the elaborate task of solving large systems for finding global parameterizations, and the solution remains feasible even for complex topologies. The local parameterization is performed on relatively planar clusters; therefore, the distortion due to parameterization is minimal. Furthermore, by converting the 3D geometry completion problem into the task of 3D texture completion, we are able to further utilize a wealth of currently available surface texture synthesis and completion techniques to serve our purpose. For example, we can perform geometry structure guided geometry completion and can transfer details between geometry models. Our algorithm supports robust user control over the geometry synthesis process, and provides a practical geometry completion editing tool.

2 Related work

Example-based texture synthesis takes a given small texture sample to generate a large scale texture that is visually consistent on 2D images or 3D surfaces. Many approaches have been proposed on this hot topic, but here we only focus on the more recent work on texture synthesis on 3D surfaces that are relevant to our approach. It is found that most of the existing texture synthesis algorithms on 3D surfaces are mesh oriented [15, 23, 25, 27, 29, 32, 33]. Since texture synthesis algorithms on meshes normally make use of the topology information, they cannot be applied to point set surfaces directly. So far, few works have focused on texture synthesis over point set surfaces. Alexa et al. [1] and Clarenz et al. [4] showed some texture synthesis results on point set surfaces as applied by their algorithms.

Recently, geometry completion focused on repairing the uncompleted meshes. Also, point-sampled surfaces have received much attention in computer graphics. Many methods have been presented which can be mainly divided into two categories according to the strategies they adopt.

The first strategy, given by the following examples, is to create a smooth patch covering the hole-region and satisfying the boundary conditions. Carr et al. [2] used globally supported radial base functions (RBFs) to fit data points by solving a large dense linear system. Ohtake et al. [17] proposed a hierarchical approach for 3D scattered data interpolation with compact RBFs. Davis et al. [6] constructed a volumetric signed distance function around surface samples, and then applied an iterative Gaussian convolution to propagate adjacent distance values to patch the holes. Liepa [14] proposed a hole filling technique which consists of four steps of boundary identification, hole triangulation, refinement, and fairing to interpolate the shape and density of the surrounding mesh. Verdera et al. [26] extended a PDE-based image inpainting technique to 3D geometry meshes, and Clarenz et al. [3] patched a surface by applying an optimization process which minimizes the integral of the squared mean curvature to yield a smooth surface. Levy [13] filled holes by extrapolating geometry boundaries in the parameterization domain. Ju [10] constructed an inside/outside volume using octree grids for any model represented as a polygon soup, then the holes are repaired by contouring.

The second strategy is to repair the holes according to the context information so that the geometry detail can be reconstructed simultaneously. Savchenko et al. [21] warped a given shape model towards the missing region using control points followed by performing a fairing step along the boundary of the hole. Sharf et al. [22] introduced a context-based method which extended the texture synthesis techniques from 2D image to 3D point-based models for completing the defective geometry models. As in the 2D case of texture synthesis, the characteristics of the given surface are analyzed, and the hole is iteratively filled by copying patches from valid regions of the given surface. Pauly [20] presented a method using a database of 3D shapes to provide geometric priors for regions of missing data. Lai et al. [12] proposed a method of geometric detail synthesis and transferring for meshes based on the concept of geometry images. Park et al. [18] restored both shape and appearance from the incomplete point surfaces by conducting a local parameterization to align patches and then solved Poisson equations on a 2D domain for warping the patch to cover the hole region. Zhou et al. [30] presented a surface texture inpainting method for triangular meshes employing a Poisson-based interpolation technique. Nguyen et al. [16] transformed the 3D geometry synthesis problem into the 2D domain by parameterizing surfaces, and then solved the geometry completion problem with an interactive PDE solver.

3 Texture synthesis

Recently, Kwatra et al. [11] presented an approach for 2D texture synthesis based on global optimization of texture quality with respect to a similarity metric, and the similarity metric is based on Markov random field (MRF) similarity criterion. They defined a global texture energy to optimize the entire texture with an expectation maximization-like algorithm. In this section, we extend this method to point-sampled surfaces. As point surfaces are irregularly sampled in 3D, it is difficult to define such a global texture energy function on point-sampled geometry. Preprocessing of the point set surface is necessary.

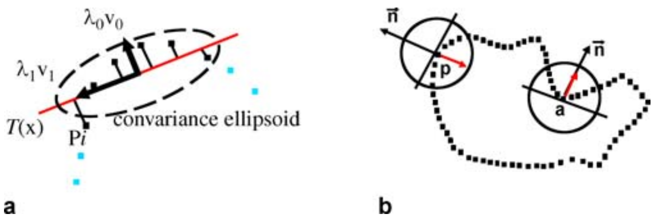

Fig. 1. a Covariance analysis. **b** Defining the sign of the curvature normal

The point-sampled geometry $M = \{p_1, p_2, \dots, p_n\}$ is clustered into uniform patches $\{C_{O,i}\}$, which are the units for further computing. The neighboring patches are overlapped to make it less computationally expensive for computing the energy. Furthermore, this avoids the synthesized texture from getting blurred in regions where there is a mismatch between the overlapping clusters. We then set up a global continuous direction field on the point set surface and conduct local parameterization for each patch. By building the correspondence between irregularly 3D sampling points and the regular 2D texture samples, we can define the global texture energy directly on the surface to be optimized. At the end of this section, we show how to enhance the roughness appearance of the geometry surface in rendering.

3.1 Analysis of local surface properties

Covariance analysis on a local point cloud can be applied to estimate various local surface properties [9, 19, 28], such as the normal vector of an approximation surface and the surface variation. Let $\bar{P}$ be the centroid of a local point set P, C be its covariance matrix. Since the eigenvalues $\lambda_i (i = 0, 1, 2)$ of C are real-valued and the eigenvectors $\upsilon_i (i = 0, 1, 2)$ form an orthogonal basis, the eigenvalues λ_i measure the variation of the local point set along the direction of the corresponding eigenvectors (see Fig. 1a).

Assuming that $\lambda_0 < \lambda_1 < \lambda_2$, the plane $(x - \bar{p})\upsilon_0 = 0$ minimizes the sum of squared distance to the neighbors. Pauly et al. [19] defined $\sigma_n(p) = \lambda_0/(\lambda_0 + \lambda_1 + \lambda_2)$ as the surface variation at point p in a neighborhood of size n, and the surface variation σ_n is closely related to curvature.

3.2 Surface clustering

We first utilize the hierarchical clustering algorithm [19] to split the point cloud using a binary space partition, and

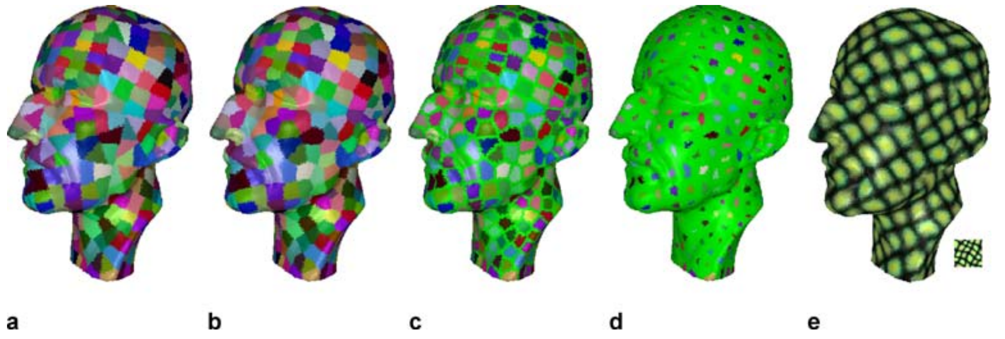

Fig. 2a–e. Surface clustering of point-sampled geometry. **a** Initial clustering. **b** Uniform clustering. **c** Overlapped cluster with small overlapping parameter. **d** Larger overlapping area. **e** Texture synthesis result

then split the point cloud M into a number of subsets $\{C'_i\}$. Note that these initial clusters may contain sharp edges and corners, as shown in Fig. 2a. Let y'_i be the centroid of the cluster C'_i, and the simplified model $Y' = (y'_1, \dots, y'_m)$ of M is obtained. To get a more even distribution of clusters, we find a neighborhood $N_i = \{j : 0 < \|y'_j - y'_i\| < r\}$ for each point $y'_i \in Y'$. For each point sample $p_i \in C'_i$, we locate the nearest $y'_j \in N_i$ to p_i, and then p_i is distributed to the new cluster C_j. Therefore, the partitioning result of the entire point set is transformed to uniform clusters $\{C_i\}$, as illustrated in Fig. 2b.

We then grow each C_i to obtain the new generated clusters $\{C_{O,i}\}$ such that each $\{C_{O,i}\}$ overlaps with its neighboring cluster $\{C_{O,j}\}$ within a band of width h. In our experimentation we set $h = 0.5d$, where d is the average radius of the clusters $\{C_i\}$. In Fig. 2c, the green color indicates the overlapped area of adjacent clusters. In Fig. 2d, the parameter h takes a larger value.

3.3 Globally optimized surface texture synthesis

We establish a texture energy function directly on the point set surface in several steps. We first define the global energy on the point-sampled surface.

Let y_i be the centroid of the cluster $C_{O,i}$. We apply the method of Alexa et al. [1] to establish a global direction field on the point set surface $Y = (y_1, \dots, y_m)$. Based on the direction field and the tangent plane computed by covariance analysis, we set up a local frame for each cluster $C_{O,i}$ to facilitate the local parameterization.

Let v_i be the direction of y_i, and n_i be the normal of y_i. We denote v_j as the up direction and $u_j = v_j \times n_j$ as the right direction for the texture, as shown in Fig. 3a. A local frame $\{u_i, v_i, n_i\}$ is then established. We project the vectors from y_i to all surrounding points p_i in the cluster $C_{O,i}$ onto the tangent plane of y_i. The resulting vectors are normalized and multiplied by the distance between y_i and p_i (Fig. 3a). In this way we preserve the distance information between the points on the surface to some extent. In most cases, as the point set is densely sampled, the patches are quite flat and so the distortion of the local parameterization is quite small. In bad conditions, we employ the multidimensional scaling (MDS) algorithm presented in [31], which does not require boundary conditions to perform better parameterization.

Now a regular parameterization grid G_i of $n \times n$ centered on y_i with interval of h is generated on the tangent plane for each $C_{O,i}$. G_i aligns with the direction v_j as the right vector and the u_i as the up vector (Fig. 3b). The parameter at each grid point t_{ij} is obtained by interpolating the parameters of points in the surrounding cells. If the surrounding cells contain no point, the parameter at the current grid point is marked as unassigned. So with the regular grid G_i, the correspondence between irregularly 3D sampling points and the regular 2D texture samples can be built for defining the global texture energy. Let $X = \{G_i\}$ denote the surface parameter space over which we want to compute the texture energy, and let Z denote the input texture sample. Let Z_i be the vectorized pixel neighborhood in Z whose appearance is most similar to G_i under the Euclidean norm. Then, we define the texture energy over X to be

$$E_t(x, \{Z_i\}) = \sum_{p \in Y} \|G_i - Z_i\|^2. \tag{1}$$

Similar to the algorithm presented in [11], we use the EM-like algorithm to optimize the texture energy over X. We modify the E and M steps to account for the specified feature of the discrete point set.

The M-step of our algorithm minimizes Eq. 1 with respect to the set of input neighborhoods $\{Z_i\}$, keeping G_i fixed and for each G_i, we find its nearest neighbor Z_i from Z by applying the mixture tree structure [7] to accelerate this search.

In the E-step, we need to minimize Eq. 1 with respect to G_i. We resolve it in a way different from [11]. Once we find the closest input neighborhood Z_i for each G_i, the texture intensity at point p_i enclosed in G_i can be obtained using bilinear interpolation. Suppose clus-

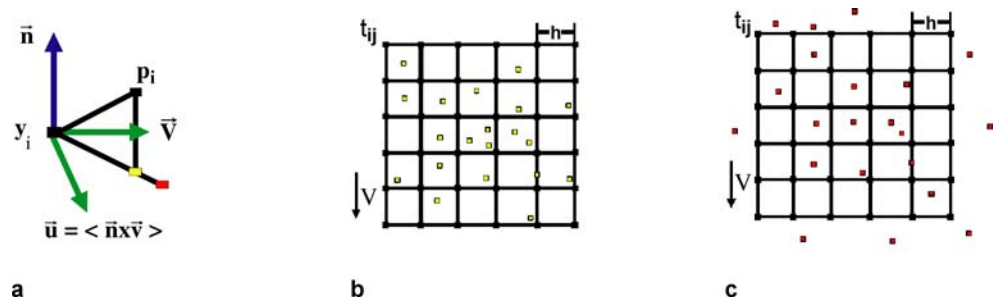

Fig. 3a–c. The construction of the grid for each patch (cluster). **a** Local parameterization by projection. **b** Regular grid G generated on the parameterization domain. **c** To generate the sample texture from user-specified region

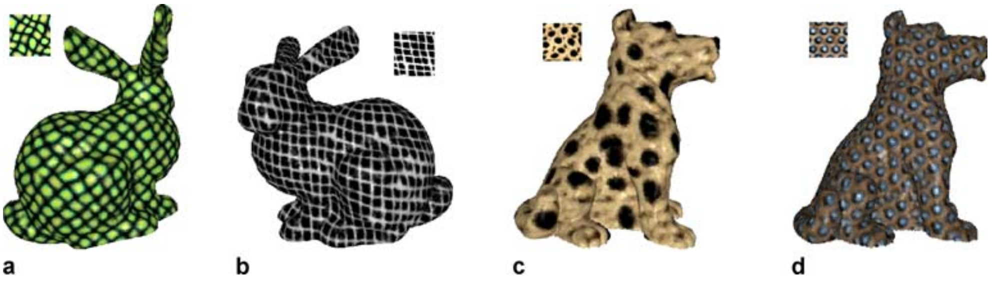

Fig. 4. Texture synthesis results based on the global optimization

ter $C_{O,i}$ and $C_{O,j}$ overlap each other, some points will be contained in both $C_{O,i}$ and $C_{O,j}$. Each of the common points may possibly take different intensity values from G_i and G_j. The minimization procedure assigns each common point an intensity value that is equal to the average of the original values in $C_{O,i}$ and $C_{O,j}$. Since the intensity at the common point has changed, we also update the intensity at t_{ij} in G_i, which is used for the next M-step.

The energy of the synthesized texture will converge after a number of iterations. Figure 2e shows some results using our method, and Fig. 4 shows more results. Multiscale synthesis can also be performed with our approach by adjusting the grid $G_i(n \times n)$.

3.4 Roughness enhancement for rendering

The roughness appearance of the surface is essential to attain visual consistency with the synthesized bump texture. Under different lighting conditions, the highlight of the surface may not align with the synthesized texture, leading to confusing visual effect. We adopt a method similar to the normal-mapping to resolve this problem. Assuming that the surface patch under the texture sample is continuous, we reconstruct the normal of each pixel for the sample texture with shape-from-shading technique. As each point on the surface X corresponds to several pixels on the sample texture during the texture synthesis process, the normal at the underlying points of these corresponding pixels are averaged and then transformed into the local reference frame of the surface point. It is used as the normal for shading so that the lighting of the surface will be consistent with the synthesized texture. Note that this method is only for rendering; the real geometry of the point set surface is not modified. Figure 5 shows an example of the roughness enhancement of the point set surface.

4 Color texture completion

In this section, we present a constrained texture synthesis algorithm to complete the color texture of the point-sampled geometry, so that the completed color texture is consistent with the surrounding existing texture.

The user selects a region D on the existing surface to serve as the input sample texture. For each point p_i in D, similar to that presented in Sect. 3.3, we find its neighborhood N_i and build a colored regular grid Z_i of $n \times n$, as shown in (Fig. 3c). The set $\{Z_i\}$ is used as the input texture for color completion.

To make the boundary between the completed and original regions imperceptible, similar to the controllable image synthesis [11], we add the following additional term

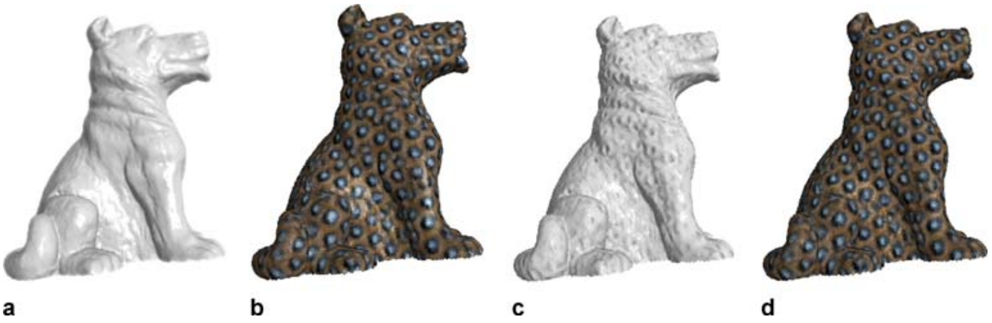

Fig. 5a–d. Enhancing the roughness appearance of the geometry surface. We reconstruct the normal of each point for rendering according to the texture synthesis result. **a** The dog model without texture. **b** Texture synthesis on the point set surface. **c** Roughness enhancement according to the synthesized texture but without drawing the textures. **d** Roughness enhancement according to the synthesized texture

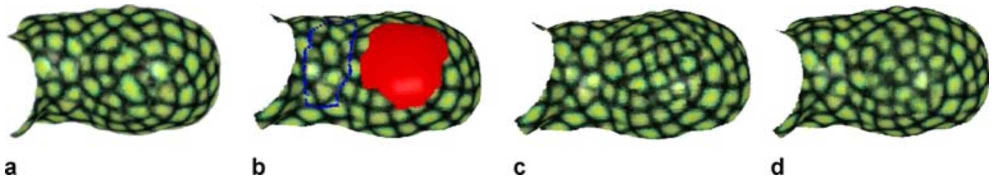

Fig. 6a–d. Texture completion. **a** Original model. **b** Defective texture model. **c** The defective region completed. **d** The defective region is completed using different sizes of input texture

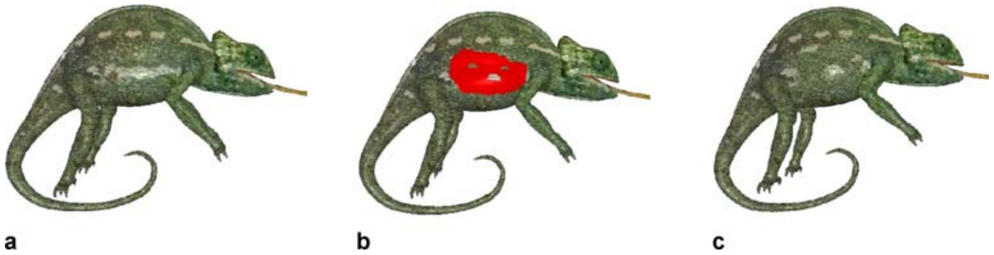

Fig. 7. a The original Chameleon model. **b** Model with corrupted texture. **c** Model with completed texture

to Eq. 1 to achieve constrained texture completion:

$$E_c(x; x^c) = \sum_{k\in\varphi}(x(k) - x^c(k))^2. \qquad (2)$$

In our approach, φ is the set of boundary points, or the set of boundary clusters which contain some boundary points, and x^c is a vector containing the current color values at the boundary points. We present a more general constrained energy function for texture completion expressed as:

$$E_t(x, \{Z_i\}) = \sum_{p\in Y}\|G_i - Z_i\|^2 + \lambda\sum_{k\in\varphi}(x(k) - x^c(k))^2. \qquad (3)$$

Note that to generate a seamless boundary texture, the cluster set $\{C_{O,j}\}$ should be chosen to cover the color defective regions and contain the boundary points, as shown in Fig. 6c.

As shown in Fig. 6b, the region enclosed by the blue curve is used for generating the sample texture, the red patch is the region to be completed, and the result is shown in Fig. 6d. In Fig. 7, we complete a model with isolated islands left on the flawed region. With our constrained texture filling technique, the result is consistent with the islands as well as the boundary of the existing surface color. This situation is difficult to handle if using the Poisson equation interpolation method [18].

5 Geometry detail encoding

The geometric detail is an important attribute of a surface. It is defined as the difference between the original point-sampled surface and its base surface. In our method, the base surface is built by smoothing the point set surface.

5.1 Surface smoothing

Using the normal v_0 and the curvature σ_n defined at each point p_i on the point set surface, we can define a curvature flow equation. The basic idea of such a diffusion flow is to allow each point to move along the normal with a speed equal to the curvature σ_n as shown in Fig. 1b.

One problem is how to determinate the sign of the curvature, as illustrated in Fig. 1b. Assuming that the index set $N(p_i)$ is the k-nearest neighbor around the point p_i, the sign of the curvature can be determined by the following formulation:

$$Dire = \frac{\sum\limits_{u\in N(p_i)} G(\|p_i - u\|)(n_i,\, p_i - u)}{\sum\limits_{u\in N(p_i)} G(\|p_i - u\|)} \qquad (4)$$

where n_i is the normal of the point p_i, $G(x)$ is the standard Gaussian filter $e^{-x^2/2\sigma^2}$ and σ is a parameter controlling the individual contribution of the sampled points in the neighborhood $N(p_i)$. If $Dire \geq 0$, then $sign = 1$, otherwise $sign = -1$. The directional curvature $\overline{k_i}$ is defined as $\overline{k_i} = sign\cdot\sigma_n^i$. The formulation dynamically adapts to the local sampling density and is stable for a noisy surface.

The curvature flow equation on point-sampled geometry is then defined as follows:

$$\frac{\partial u_i}{\partial t} = -\overline{k_i}n_i \quad \overline{k_i} = sign\cdot\sigma_n^i. \qquad (5)$$

A sequence of point set surface (U_n) can be constructed by integrating the diffusion equation with a simple explicit Euler scheme.

The normal n_i can be evaluated by the following steps: First, we apply the covariance analysis to get eigenvector υ_0, and let n'_i be the corresponding normal at point p_i at the previous smoothing step. If $\upsilon_0 \cdot n'_i \geq 0$, then $n_i = n_0$, otherwise $n_i = -n_0$. Since for each diffusion iteration the variation of the normal direction at each point is small, we can derive the right result quickly, thus avoiding the time-consuming process of keeping orientation of the normal vectors consistent using the minimum spanning tree during each iteration.

Since a point is adjusted along the normal direction, the proposed curvature flow will not introduce undesirable point drifting over the surface. The curvature flow equation Eq. 5 is an isotropic smoother [28]. It is motivated by the mean curvature flow operator [8] defined on meshes, where a curvature flow operator is defined based on the 1-ring neighbors $N_i(i)$ of vertex v_i employing the vertex connectivity information.

5.2 Encoding geometric details as texture

Let M' be a base surface obtained by filtering the surface M. Let $p \in M$, and $p' \in M'$ is its corresponding point on M', and n' is the normal at point p'. Let $\delta = \|p - p'\|$ be the geometric detail of point p. Let $dire = (p - p') \cdot n'$, if $dire \geq 0$, then $sign = 1$, otherwise $sign = -1$. We define $c' = sign \cdot \|p - p'\|$ as the signed gray-scale of p'.

Once the normal n' and signed gray-scale c' of p' are obtained, its geometry information is approximately reconstructed as $\overline{p} = p' + c' \cdot n'$. Using this technique, a surface $\overline{M}$ is reconstructed by the base surface and the signed gray-scale C which approximates M. The normal of the reconstructed points can be recomputed using the minimum spanning tree [9].

With our mean curvature flow filter, the reconstructed surface from the signed geometric gray level is a good approximation to the original surface, as shown in Fig. 8. With different iteration times to integrate Eq. 5, various frequency bands of geometric detail can be extracted effectively and efficiently.

6 Geometry completion

Similar to the color texture completion, the completed geometry should keep consistent with the surrounding geometry and the boundary between the completed and existing regions should be continuous. Using hierarchical compactly supported basis functions [17], we first complete the base surface by smoothed extrapolation, as shown in Fig. 9d. Further, the geometry detail on the base surface patch should be reconstructed.

6.1 Context-based geometry completion

Using the method presented in previous sections we can complete the signed gray-scale texture on the patched smooth surface (Fig. 9e). The completed signed gray-scale texture is then converted back to geometric details (Fig. 9f), and the context-based geometry completion is achieved. This procedure is regarded as a reverse procedure of the geometric detail encoding.

Let $p \in M$ with normal n, p' is its corresponding point on M' with normal n'. We interpolate the defective base surface M' using the hierarchical compactly supported basis functions [17], and get the completed base surface N'. The method presented in [17] works well for irregularly sampled and/or incomplete point data, and is also fast enough for practical use. Suppose $\Omega' = N' - M'$ is the newly constructed smooth patch which is consistent with the boundary of the M', we then complete the texture C' of Ω' based on the existing texture on M' employing the technique presented in Sect. 5. For each point $v' \in \Omega'$ with normal n', let c' be its synthesized signed gray-scale. The reconstructed location is defined as $v = v' + c' \cdot n'$, and its normal can be recalculated using the technique described in [9]. Using this approach, the completed patch Ω captures the context information of the existing surface. Compared with other

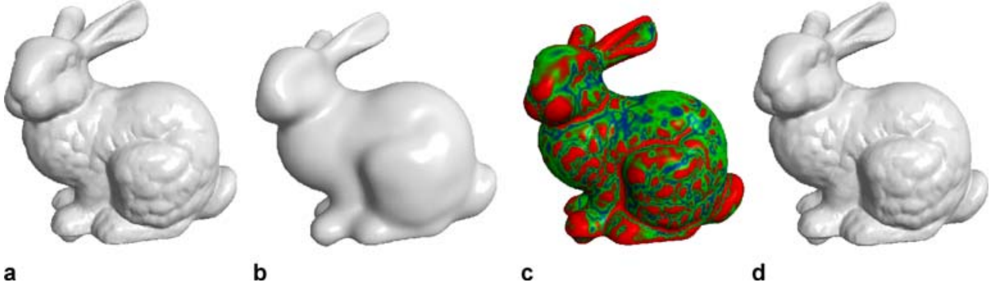

Fig. 8a–d. Surface reconstruction from the signed gray-scale texture. **a** Original bunny model M. **b** Smoothed model bunny M'. **c** Signed gray-scale texture C in pseudo-color. **d** Reconstructed surface $\overline{M}$

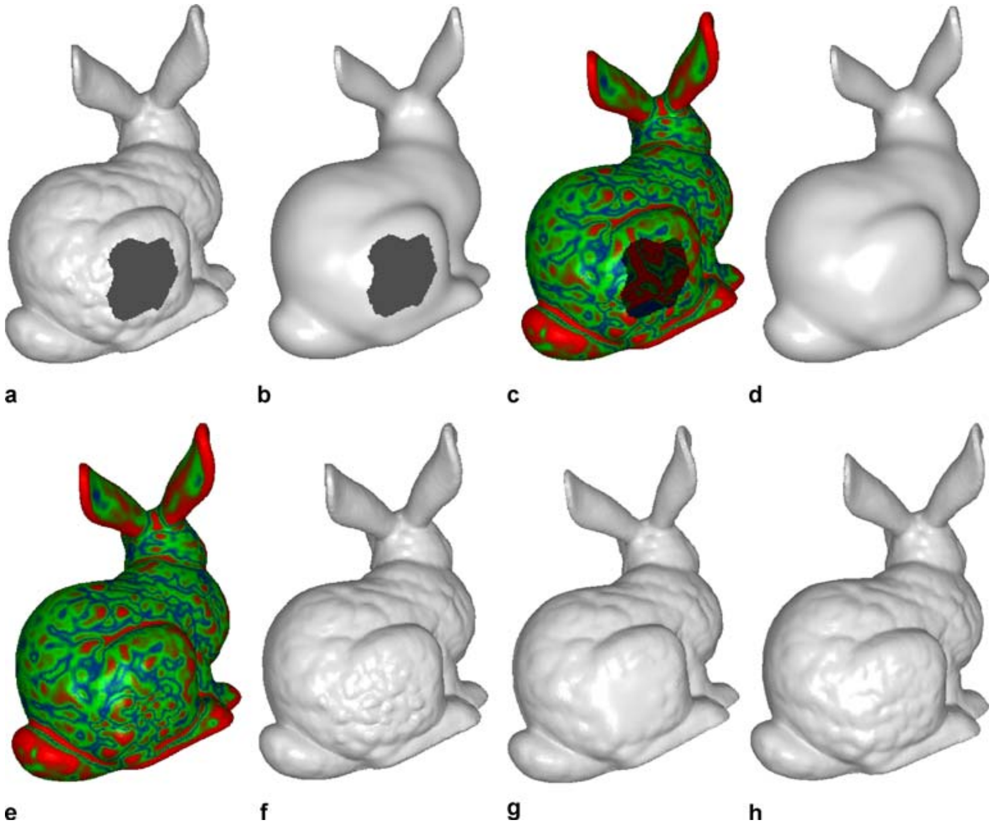

Fig. 9a–h. Overview of our entire processing pipeline for geometry completion. **a** The defective bunny model. **b** The base surface of **a**. **c** The signed gray-scale texture on the base surface. Pseudo-color is used to illustrate the value clearly. Red color indicates the largest gray-scale value, while the blue indicates the least value. **d** Completed base surface. **e** The completed signed gray-scale texture in pseudo-color. **f** The final result of geometry completion. **g** RBF interpolation of **a**. **h** The original bunny model

geometry completion approaches [22] that add points by rotating, translating, and warping copies of points from another region, our approach is more efficient, controllable and easier to implement.

Assume N to be the reconstructed surface based on N' and its signed gray-scale. The terms Ω, M and N are all continuous inside their interior region. When we adopt the points from M and the other points from Ω in the final completion result, small cracks may occur in the boundary region between Ω and M. To resolve this problem, we modify the normal of the boundary points in Ω'.

Suppose p is a boundary point on M, and p' is its corresponding point on base surface M' with normal $n' = (n_x^{p'}, n_y^{p'}, n_z^{p'})$. Let $\delta = (\delta_x, \delta_y, \delta_z)$ be the normalized vector of the difference vector $p - p'$. The vector difference $\Delta\mu_{p'}$ between n' and δ is defined as:

$$\begin{aligned}\Delta\mu_{p'} &\triangleq \left(\Delta\mu_x^{p'}, \Delta\mu_y^{p'}, \Delta\mu_z^{p'}\right)\\ &\triangleq \left(\delta_x - n_x^{p'}, \delta_y - n_y^{p'}, \delta_z - n_z^{p'}\right).\end{aligned}$$

We define a band with radius d around the boundary of the smoothed surface patch. For each point $q' = (x_{q'}, y_{q'}, z_{q'})$ with normal $n^{q'} = (n_x^{q'}, n_y^{q'}, n_z^{q'})$ in the band, we find its nearest point p' on the boundary of M'. Let the distance between p' and q' be s. We set the weight $\omega = (d - s)/d$, then the normal $n^{q'}$ of q' is adjusted as:

$$\begin{aligned}\left(\overline{n_x^{q'}}, \overline{n_y^{q'}}, \overline{n_z^{q'}}\right) \triangleq \big(n_x^{q'} + w \cdot \Delta\mu_x^{p'}, n_y^{q'}\\ + w \cdot \Delta\mu_y^{p'}, n_z^{q'} + w \cdot \Delta\mu_z^{p'}\big).\end{aligned}$$

By normalizing the above vector we get its modified normal $\overline{n}$, then the geometry position of q' can be reconstructed as $q = q' + c' \cdot \overline{n}$. As the completed base surface N' is continuous, its normal is also continuous. In addition, the completed texture C is continuous around the boundary region. By using this technique, we are able to produce a continuous surface around the boundary and the holes are filled consistently with the existing surface (Fig. 9f).

6.2 Geometry completion with structure propagation

Many recent image completion methods fill the unknown regions by augmenting texture synthesis with some auto-

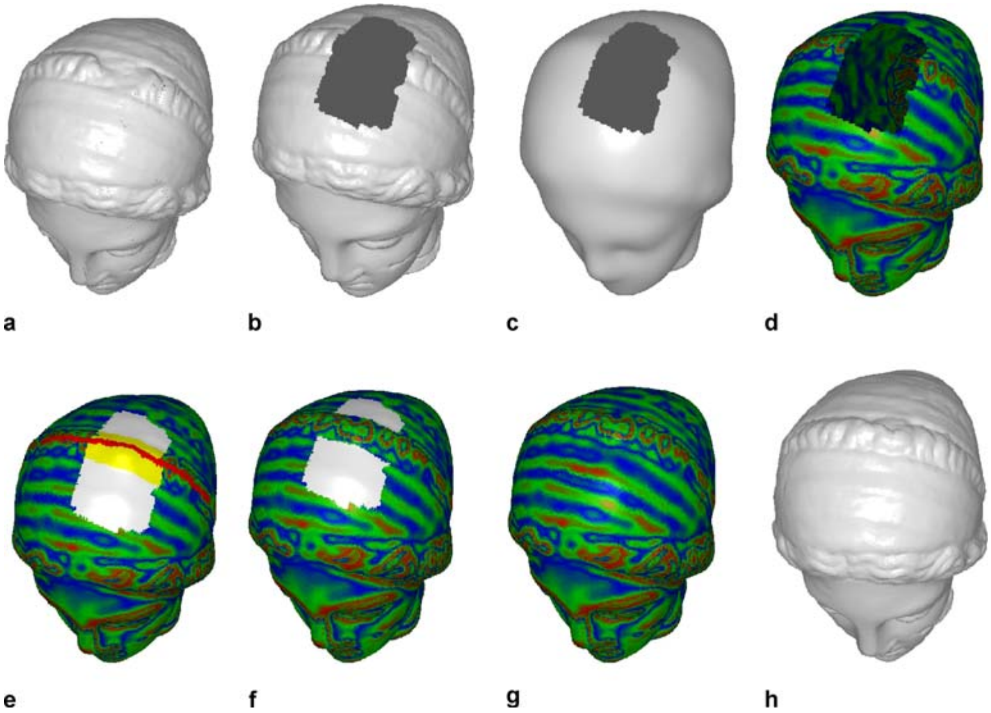

Fig. 10a–h. Geometry completion with structure propagation. The signed gray-scale textures are illustrated in pseudocolor. **a** Original Venus model. **b** Uncompleted Venus M. **c** The base surface M' of M. **d** The geometry detail of M is converted to the signed gray-scale texture on M'. **e** One red line specified by the user in completed base surface; the yellow region is the structure region needed to be synthesized first. **f** Intermediate result after synthesizing structure and texture information along the user-specified line. **g** Final result after filling in the remaining unknown regions by texture optimization. **h** Result of geometry reconstruction from the completed signed gray-scale texture

matic guidance or interactive guidance [5, 24]. This guidance determines the synthesis ordering, which significantly improves the quality of completion by preserving some salient structures.

The synthesis order of geometry completion is also important. Similar to Sun et al. [24], we propose a geometry completion method based on structure guided synthesis. The missing structure information is specified by extending a few curves or line segments from the known regions to the unknown regions. Then, the patches along these user-specified curves in the unknown region is synthesized using patches selected around the curves in the known region by using a global optimization. After the salient structure is completed, the remaining regions can be completed. In our method both the salient structures and remaining missing regions are completed based on the constrained global optimization. They are optimized using an expectation maximization algorithm.

As shown in Fig. 10, we want to fill the missing geometry on the Venus model. The user specifies the important missing texture structure by extending a few curves from the known to the unknown regions on the base surface. The signed gray-scale texture along the specified curves in the unknown region is first completed with structure guided synthesis and the other regions are then completed by texture optimization.

6.3 Geometry completion with detail cloning

Our method also provides a seamless geometry cloning tool for surface-based detail cloning. The missing region of one model can be completed with the geometry details of other models as the geometry texture. The user specifies a source region S in an arbitrary surface and the missing region D on the target surface. By smoothing the source region S, we get its signed gray-scale texture. Using the constrained optimization texture completion technique described in previous sections, the signed gray-scale texture is adopted as the input texture sample to complete the texture of the smooth patch filling hole D. After converting the gray-scale texture back to the geometric detail, we obtain the cloning result as shown in Fig. 11. The defective region of the lady model is completed by transferring the geometric details of the bob region on the model of Venus.

7 Implementation and results

The proposed algorithm was implemented on a Microsoft Windows XP PC with a Celeron 2.00 GHz CPU and 1.00 GB RAM. The computational complexity of our approach is dominated by the process of texture synthesis which is based on the nearest neighbor search. In the example shown in Fig. 4, there are 240 000 points in the Stanford bunny, the execution time per iteration takes 8–10 s, and the total execution time is 10–12 min for about 50 iterations. In the context-based geometry completion stage, the execution time is much less as the patched region is much smaller. In Fig. 9, there are 22 000 points in the patched regions, and it takes less than 1 min to accomplish the texture completion operation. In our experiments,

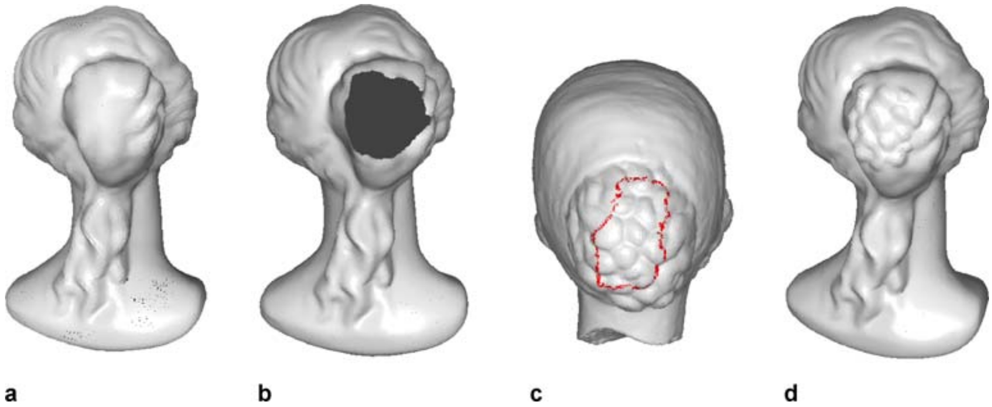

Fig. 11a–d. Geometry completion based on geometry transfer. **a** Original lady model. **b** Uncompleted lady model. **c** The sample model Venus. **d** The completion result

there are usually about 300 points in each overlapped cluster, and the scale of the synthesized texture can be controlled by adjusting the size of the regular grid G.

8 Conclusion and future work

We have presented a novel approach for surface appearance and surface content completion for the acquired 3D data set based on global optimization. The feature of our proposed approach is to transform the task of surface content completion into that of surface texture completion. The major benefit is that it is flexible and efficient to implement. Meanwhile, we have provided a practical tool for color texture synthesis and geometry completion over point set surfaces. Our system can be extended to the mesh models easily. Further research will be focused on volume texture synthesis for point data set. Another interesting future topic is to perform a user controllable non-uniform clustering so that the scale of synthesized texture can vary progressively through the point cloud.

References

1. Alexa, M., Klug, T., Stoll, C.: Direction fields over point-sampled geometry. J. WSCG **11**(1), 27–32 (2003)
2. Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R.: Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of SIGGRAPH, pp. 67–76 (2001)
3. Clarenz, U., Diewald, U., Dziuk, G., Rumpf, M., Rusu, R.: A finite element method for surface restoration with smooth boundary conditions. Comput. Aided Geom. Des. **21**(5), 427–445 (2004)
4. Clarenz, U., Rumpf, M., Telea, A.: Finite elements on point based surfaces. In: Proceedings of the EG Symposium of Point Based Graphics, pp. 201–211 (2004)
5. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: IEEE Proceedings of Conference on Computer Vision and Pattern Recognition, pp. 417–424 (2003)
6. Davis, J., Davis, S., Marschner, R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: 1st International Symposium on 3D Data Processing, Visualization, Transmission, pp. 428–438 (2002)
7. Dellaert, F., Kwatra, V., Oh, S.M.: Mixture trees for modeling and fast conditional sampling with applications in vision and graphics. In: IEEE Proceedings of Conference on Computer Vision and Pattern Recognition Recognition, pp. 619–624 (2005)
8. Desbrun, M., Meyer, M., Schroer, P., Barr, A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH, pp. 317–324 (1999)
9. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: Proceedings of SIGGRAPH, pp. 71–78 (1992)
10. Ju, T.: Robust repair of polygonal models. ACM Trans. Graph. **23**(3), 888–895 (2004)
11. Kwatra, V., Essa, I.A., Bobick, A.F., Kwatra, N.: Texture optimization for example-based synthesis. ACM Trans. Graph. **24**(3), 795–802 (2005)
12. Lai, Y.-K., Hu, S.-M., Gu, D.X., Martin, R.: Geometric texture synthesis and transfer via geometry images. In: Proceedings of the ACM Symposium on Solid and Physical Modeling, pp. 15–26 (2005)
13. Levy, B.: Dual domain extrapolation. ACM Trans. Graph. **22**(3), 364–369 (2003)
14. Liepa, P.: Filling holes in meshes. In: Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 200–205 (2003)
15. Magda, S., Kriegman, D.: Fast texture synthesis on arbitrary meshes. In: Proceedings of the 14th Eurographics Workshop on Rendering, pp. 82–89 (2003)
16. Nguyen, M.X., Yuan, X., Chen, B.: Geometry completion and detail generation by texture synthesis. In: Proceeding of Pacific Graphics, pp. 23–32 (2005)
17. Ohtake, Y., Belyaev, A., Seidel, H.-P.: A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In: Proceedings of the Shape Modeling International, pp. 153–161 (2003)
18. Park, S., Guo, X., Shin, H., Qin, H.: Shape and appearance repair for incomplete point surfaces. In: Proceedings of the International Conference on Computer Vision, pp. 1260–1267 (2005)

19. Pauly, M., Gross, M., Kobbelt, L.: Efficient simplification of point-sampled surfaces. In: Proceedings of IEEE Visualization, pp. 163–170 (2002)
20. Pauly, M., Mitra, N., Giesen, J., Gross, M., Guibas, L.J.: Example-based 3D scan completion. In: Symposium on Geometry Processing, pp. 23–32 (2005)
21. Savchenko, V., Kojekine, N.: An approach to blend surfaces. In: Proceedings of Computer Graphics International, pp. 139–150 (2002)
22. Sharf, A., Alexa, M., Cohen-Or, D.: Context-based surface completion. ACM Trans. Graph. **23**(3), 878–887 (2004)
23. Soler, C., Cani, M., Angelidis, A.: Hierarchical pattern mapping. In: Proceedings of SIGGRAPH, pp. 673–680 (2002)
24. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. In: Proceedings of SIGGRAPH **24**(3), 861–868 (2005)
25. Turk, G.: Texture synthesis on surfaces. In: Proceedings of SIGGRAPH, pp. 347–354 (2001)
26. Verdera, J., Caselles, V., Bertalmio, M., Sapiro, G.: Inpainting surface holes. In: Proceedings of International Conference on Image Processing, pp. 903–906 (2003)
27. Wei, L.-Y., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: Proceedings of SIGGRAPH, pp. 355–360 (2001)
28. Xiao, C., Miao, Y., Liu, S., Peng, Q.: A dynamic balanced flow for filtering point-sampled geometry. Vis. Comput. **22**(3), 210–219 (2006)
29. Ying, L., Hertzmann, A., Biermann, H., Zorin, D.: Texture and shape synthesis on surfaces. In: Proceedings of the Eurographics Symposium on Rendering, pp. 301–312 (2001)
30. Zhou, K., Wang, X., Tong, Y., Desbrun, M., Guo, B., Shum, H.Y.: Texture montage: seamless texturing of arbitrary surfaces from multiple images. In: Proceedings of SIGGRAPH, pp. 1148–1155 (2005)
31. Zigelman, G., Kimmel, R., Kiryati, N.: Texture mapping using surface flattening via multidimensional scaling. IEEE Trans. Visual. Comput. Graph. **8**(2), 198–207 (2002)
32. Zhang, J., Zhou, K., Velho, L., Guo, B., Shum, H.: Synthesis of progressively-variant textures on arbitrary surfaces. In: Proceedings of SIGGRAPH, pp. 295–302 (2003)
33. Zelinka, S., Garland, M.: Interactive texture synthesis on surfaces using jump maps. In: Proceedings of 14th Eurographics Symposium on Rendering, pp. 673–680 (2003)

CHUNXIA XIAO was born in 1976. He received a Ph.D. form the State Key Lab. of CAD&CG, Zhejiang University, in 2006, and is currently an assistant professor at the Computer School, Wuhan University, People's Republic of China. His research interests include digital geometry processing, point-based computer graphics, image and video editing.

WENTING ZHENG was born in 1974. He received a Ph.D. from the State Key Lab. of CAD&CG, Zhejiang University in 1999, is currently an associate professor at the State Key Lab. of CAD&CG, Zhejiang University, People's Republic of China. His research interests include point-based graphics, virtual reality and programmable graphics hardware.

YONGWEI MIAO was born in 1971, and is a Ph.D. candidate at the State Key Lab. of CAD&CG, Zhejiang University, People's Republic of China. His research interests include virtual reality, digital geometry processing and computer-aided geometric design.

YONG ZHAO was born in 1982, and is a Ph.D. candidate at the State Key Lab. of CAD&CG, Zhejiang University, People's Republic of China. His research interests include point-based graphics, digital geometry processing and computer-aided geometric design.

QUNSHENG PENG was born in 1947. He received a Ph.D. from University of East Anglia in 1983 and is currently a professor at the State Key Lab. of CAD&CG, Zhejiang University, People's Republic of China. His research interests include virtual reality, realistic image synthesis, infrared image synthesis, computer animation, and scientific visualization.