Appearance and Geometry Completion with Constrained Texture Synthesis

Chunxia Xiao, Wenting Zheng, Yongwei Miao, Yong Zhao, and Qunsheng Peng

State Key Lab of CAD&CG, Zhejiang University, 310027, Hanghou, P.R. China

Abstract. A novel approach for appearance and geometry completion over point-sampled geometry is presented in this paper. Based on the result of surface clustering and the 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 texture on the surface as the input texture sample. We convert the issue 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.

1 Introduction

As numerous 3D surface scanning devices are 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, measure error in the scanning, the acquired geometry is frequently imperfect, that is, it contains holes. In addition, large holes may also be introduced by some surface editing operations. These holes have to be filled in a manner not only conforming to the global shape of the entire surface but also exhibiting its primary geometric detail. At the same time the color texture of the defective surface should also be repaired consistently.

Compared with the problem of inpainting and texture synthesis of 2D image, the geometry completion and texture synthesis on 3D point-sampled geometry is more challenging for many reasons. The point sampling is irregular and does not give rise to a regular parameter domain as the image. In addition, the similarity measurements between the point sets are difficult to define. In this paper, we propose a novel method for appearance and geometry completion on point-sampled surface, our major contributions are as follows:

Firstly, Based on global optimization we present a novel texture synthesis producing smooth texture synthesis effects while keeping the intrinsic structures

of the sample texture over the point-sampled geometry. 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, By performing a mean curvature flow we 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 and converted into a signed gray scale texture attached to the base surface for the downstream processing. Thirdly, our approach reconstructs the geometric details on the smooth patch by implementing texture completion in the gray scale texture space, therefore some troublesome operations such as similarity measurement, rigid transformation of the 3D points set are avoided.

By applying the above algorithms we can achieve consistent context-based completion and can deal with more complex boundary condition compared with the texture and geometry completion algorithms employing PDE. Furthermore, by converting the 3D geometry completion into the task of 3D texture completion, we are able to further utilize a wealth of currently available surface texture synthesis and completion technique to serve our purpose.

2   Related Works

There has been a number of works focused on Example-based texture synthesis for 3D surface. A comprehensive survey is outside the scope of this paper. It is found that most of existing texture synthesis algorithms on 3D surface are mesh oriented. Since texture synthesis algorithms on meshes normally make use of the topology information, they cannot be applied to the point set surface directly. By now, few works are focused on texture synthesis over point set surface. In our scope, Alexa et al. [1] and Clarenz et al. [2] showed some texture synthesis results on point set surface as the application of their algorithm. Recently, geometry completion focused on repairing the uncompleted meshes or point-sampled surface has received much attention in computer graphics. Lots of works have been presented, and these methods can be mainly divided into two categories according to the strategies they adopted.

One strategy is to create a smooth patch covering the hole-region and satisfying the boundary conditions. [3] used globally supported radial base functions (RBFs) to fit data points by solving a large dense linear system, [4] proposed a hierarchical approach to 3D scattered data interpolation with compact RBFs. Davis et al. [5] constructed a volumetric signed distance function around the surface samples, and then applied an iterative Gaussian convolution to propagate adjacent distance values to patch the holes. Liepa [6] proposed a hole filling technique to interpolate the shape and density of the surrounding mesh. Verdera et al. [7] extended a PDE-based image inpainting technique to 3D geometry meshes. Ju [8] constructed an inside/outside volume using octree grids for any model represented as a polygon soup, then the holes were repaired by contouring.

The other strategy is to repair the holes according to the context information so that the geometry detail can be reconstructed at the same time. Sharf et al. [9]

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. Pauly [10] presented a method using a database of 3D shapes to provide geometric priors for regions of missing data. Lai et al. [11] proposed a method of geometric detail synthesis and transferring for meshes based on the concept of geometry images. Park et al. [12] restored both shape and appearance from the incomplete point surfaces by conducting a local parameterization to align patches and then solved a Poisson equations on 2D domain for warping the patch to cover the hole region. Minh et al. [13] 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 [14] presented an approach for 2D texture synthesis based on global optimization of texture quality with respect to a similarity metric based on Markov Random Field (MRF) similarity criterion. In this section, we extend this method to point-sampled surface. As the point surface is irregularly sampled in 3D, it is difficult to define such a kind of global texture energy on point-sampled geometry. Preprocessing of the point set surface is necessary as the preparation.

The point-sampled geometry $M = \{p_1, p_2, ...p_n\}$ is firstly clustered into uniform patches $\{C_{o,i}\}$ that are the units for further computing. The neighboring patches are overlapped to make it less computationally expensive for computing the energy, furthermore, avoid the synthesized texture 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 to conduct local parameterization for each patch. By building up the correspondence between irregularly 3D sampling points and the regular 2D texture samples, the global texture energy can be defined directly on the surface and to be optimized.

3.1   Surface Clustering

We first utilize the hierarchical clustering algorithm [15] to split the point cloud. The point cloud M is then divided into a number of subsets $\{C_i'\}$. Nevertheless, these initial clusters contain sharp edges and corners, as shown in Fig.1(a). Let y_i' be the centroid of the cluster C_i'. To get a more even distribution of clusters, we find neighbor $N_i = \{j : 0 <= \|y_j' - y_i'\| < r\}$ for each point y_i'. For each point sample $p_i \in C_i'$, we locate $y_j' \in N_i$ that is the nearest to p_i, and then p_i is adjusted to the new cluster C_j. Therefore, the partitioning result of the entire point set is reformed to uniform clusters $\{C_i\}$, as illustrated in Fig 1(b).We then grow each $\{C_i\}$ and form 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 $h = 0.5 \cdot d$, where d is the average radius of the clusters $\{C_i\}$. In Fig1. (c), the green color indicates the overlapped area of adjacent clusters, in Fig1. (d), the parameter h takes a larger value.
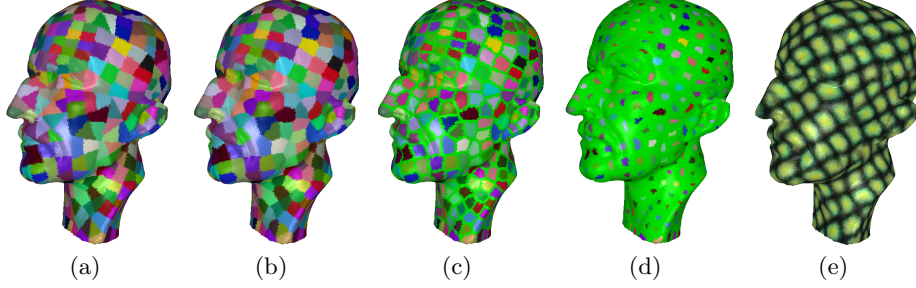
Fig. 1. Surface clustering of point-sampled geometry. (a) Initial clustering, (b) uniform clustering, (c) overlapped cluster with small overlapping parameter, (d) Larger overlapping area, (e) The texture synthesis result.

3.2   Global Optimized Surface Texture Synthesis

We establish the 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 $C_{O,i}$. By applying the method presented in [1], we can establish a global direction field on the point set surface $Y = \{y_1, ...y_m\}$. 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 ν_i be the direction of y_i, and n_i be the normal of y_i. We define ν_i the up direction and $\mu_i = \nu_i \times n_i$ the right direction for the texture, as shown in Fig. 2(a). A local frame $\{\mu_i, \nu_i, n_i\}$ is then established. We project the vectors from y_i to all surrounding points x_i in the cluster $\in C_{o,i}$ onto the tangent plane of y_i. The resulting vectors are normalized and multiplied by the distance between y_i and p_i(Fig. 2(a)). In this way we preserve the distance information between the points on the surface to some extent.
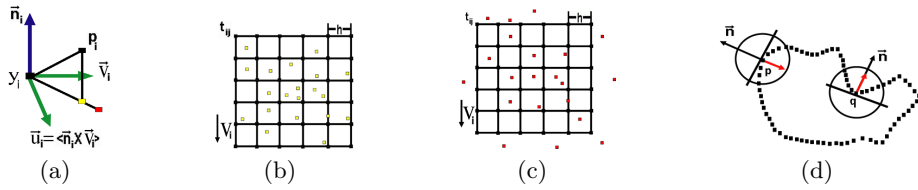

Fig. 2. the construction of the grid for each patch (cluster). (a) Local parameterization by projection, (b) regular grid G generated on the parameterization domain, (c) to generate the sample texture from user specified region. (d) mean curvature flow.

Now a regular parameterization grid G_i of $n \times n$ centered on y_i with the interval of h on the tangent plane is generated for each $C_{O,i}$, G_i aligns with the direction ν_i as up vector and the μ_i as right vector (Fig.2(b)). The parameter at each grid

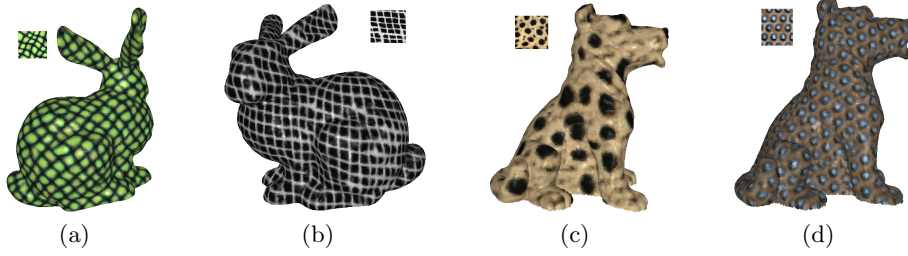<center>(a)   (b)   (c)   (d)</center>

Fig. 3. Some texture synthesis results based on the global optimization

point t_{ij} is obtained by interpolating the parameters of points in the surrounding cells. 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 parametric space over which we want to compute the texture energy and 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 [14], we use the EM-like algorithm to optimize the texture energy over X. We modify the E and M steps to account for the specific feature of the discrete point set.

The M-step of our algorithm minimizes (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.

In the E-step, we need to minimize (1) w.r.t. G_i. We resolve it in a way different from [14]. 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 cluster $C_{O,i}$ and $C_{O,j}$ overlap with each other, some points will be contained in both $C_{O,i}$ and $C_{O,j}$. Each of the common points may take possibly 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, the intensity at t_{ij} in G_i is updated which is used for the next M-step.

The energy of the synthesized texture will converge after a number of iterations. Fig.1 (e) is the result using our method, and Fig.3 shows more results. Multi-scale synthesis can also be performed in our approach by adjusting the grid $G_i(n \times n)$.

4   Color Texture Completion

In this section, we complete the color texture of the point-sampled geometry with a constrained texture synthesis algorithm, and the completed color texture

should be consistent with the surrounding existing texture. We select a user specified region D on the existing surface to serve as input sample texture. For each point p_i in D, similar to the method presented in section 3.3, we find its neighborhood N_i and build a colored regular Grid Z_i of $n \times n$, as shown in (Fig.2(c)). 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 [14], we add a additional term to Eq.(1) to achieve a general constrained energy function for texture completion.

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

In our approach, φ is the set of boundary points, or the set of boundary clusters which contain some boundary points, x^c is a vector containing the current color values at the boundary points.

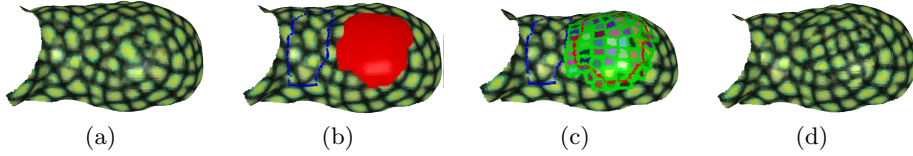(a)   (b)   (c)   (d)

Fig. 4. Texture completion. (a)Original point model,(b)defective texture model, (c)the defective regions is covered by the clusters,(d)completed texture point model.

Note that to generate a seamless boundary texture, the cluster set $\{C_{O,i}\}$ should be chosen to cover the color defective regions and contain the boundary points, as shown in Fig.4(c).As shown in Fig.4(b), the region enclosed by the blue curve is used for generating the sample texture, the red patch is the region to be completed, the result is shown in Fig.4(d). In Fig.5, we present the completion result with isolated islands left on the flawed region. With our constrained texture filling technique, the result is consistent with islands as well as the boundary of the existing surface color. This situation is difficult to handle if one applies the Poisson equation interpolation method [12].

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. Covariance analysis on local point cloud can be applied to estimate various local surface properties, for example, normal ν_0 and the curvature σ_n at each point p_i on the point set surface[15]. Based on the normal and curvature, we are able to define a curvature flow equation[16]. The basic idea of defining such a diffusion flow is to allow the point moving along the normal with a speed equal to the curvature σ_n

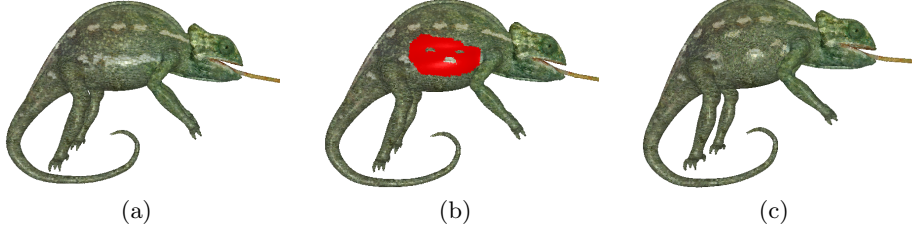(a)   (b)   (c)

Fig. 5. (a) The original Chameleon model, (b) corrupted texture model with islands, (c) the completed texture

as shown in Fig.2 (d). Since the position of 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 is an isotropic smoother and a low frequency base surface can be received applying this filtering operator.

5.1   Encoding Geometric Details as Texture

Assume that the surface M is filtered, yielding a base surface M'. Let $p \in M, p' \in M'$ is its corresponding point, and n' is the normal at point p'. Let $\delta = \parallel p - p' \parallel$ be the geometric detail of point p. Let $dire = (p - p') \cdot n'$, if $Dire \geq 0$, then $sign = 1$, else $sign = -1$. We define $c' = sign \cdot \parallel p - p' \parallel$ as the 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 information of the reconstructed points can be recomputed using the minimum spanning tree [17].With the 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.7. With different iteration times for the mean curvature flow, various frequency band 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 a hierarchical compactly supported basis functions [4], we first complete the base surface by smoothed extrapolation, as shown in Fig.6(d). 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.6 (e)). The completed
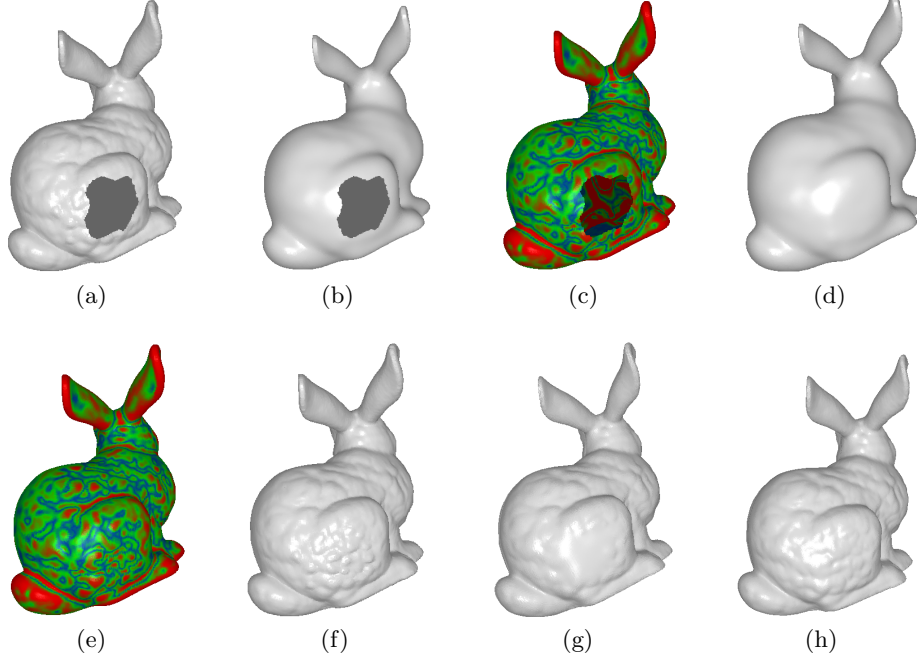
Fig. 6. Overview of our entire processing pipeline for geometry completion. (a) The defective bunny model, (b) the base surface of (a), (c) the signed gray-scale texture on the base surface. Pseudo color is used to illustrate the value clearly. Red color indicates the largest gray-scale value, while the blue indicate 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.

signed gray-scale texture is then converted back to geometric details (Fig.6 (f)), 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' to get the completed base surface N'. Suppose $\Omega' = N' - M'$ is the newly constructed 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 section 4. For each point $\nu' \in \Omega'$ with normal n', let c' be its synthesized signed gray-scale, the reconstructed location is defined as $\nu = \nu' + c' \cdot n'$, its normal can be recalculated using the technique described in [17]. Using this approach, the completed patch Ω captures the context information of the existing surface. Compared with other geometry completion approach [9] that added points by rotating, translating, and possible warped copies of points from another region, our approach is more efficient, controllable and easy to implement.

Assume N be the reconstructed surface based on N' and its signed gray-scale. Ω, M and N are all continuous inside their interior region. When we adopt the
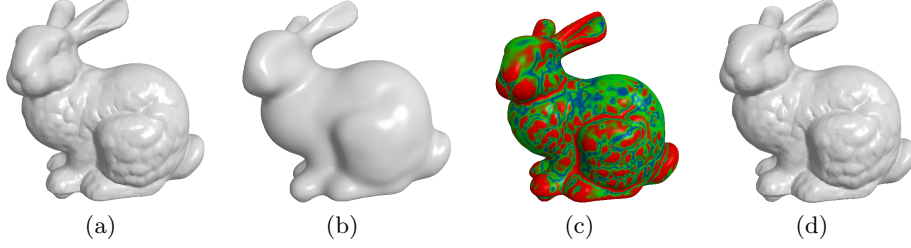
Fig. 7. Surface reconstruction from the signed gray scale texture. (a) Original Bunny model M, (b) Smoothed Bunny model M', (c) Signed gray-scale texture in pseudo color C, (d) Reconstructed surface $\overline{M}$.

points from M and the other points from Ω in the final completion result, small crack 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 δ and n' is defined as $\Delta\mu_{p'} \triangleq (\Delta_x^{p'}, \Delta_y^{p'}, \Delta_z^{p'}) \triangleq (\delta_x - n_x^{p'}, \delta_y - n_y^{p'}, \delta_z - n_z^{p'})$.

We define a band with radius d around the boundary of the smoothed surface patch Ω'. 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 $(\overline{n_x^{q'}}, \overline{n_y^{q'}}, \overline{n_z^{q'}}) \triangleq (n_x^{q'} + \omega \cdot \Delta_x^{p'}, n_y^{q'} + \omega \cdot \Delta_y^{p'}, n_z^{q'} + \omega \cdot \Delta_z^{p'})$.

By normalizing 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 consistent to the existing surface(Fig.6(f)).

6.2 Geometry Completion with Structure Propagation

The synthesis order of geometry completion is also important. By augmenting texture synthesis with some automatic guidance or interactive guidance, it can significantly improves the quality of completion by preserving some salient structures. Similar to Sun et. al. [18], 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
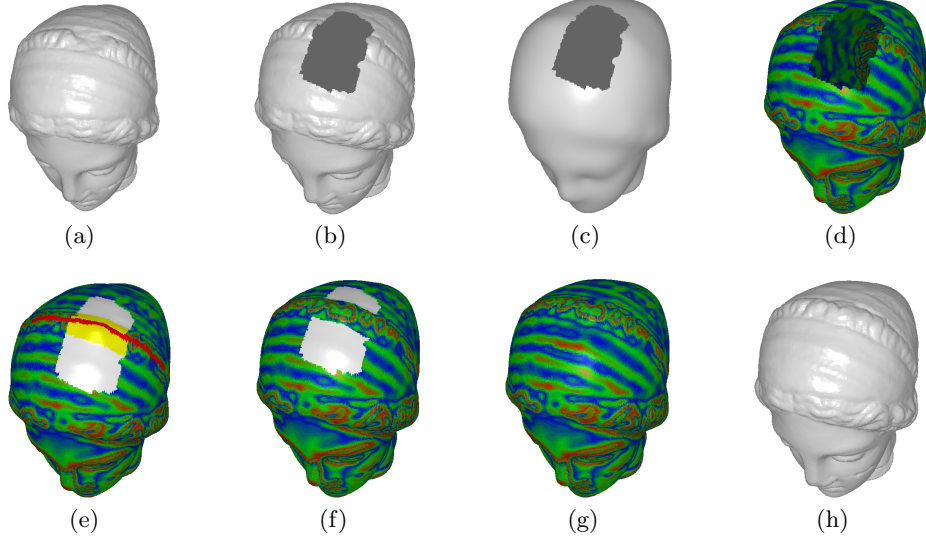
Fig. 8. Geometry completion with structure propagation. (a) Original Venus model, (b) uncompleted Venus M, (c) the base surface M' of M, (d) the geometry detail of M is converted to the signed gray-scale texture on M', (e) One red line specified by the user in completed base surface, the yellow region is the structure region needed be synthesized firstly. (f) intermediate result after synthesizing structure and texture information along the user-specified line, (g) final result after filling in the remaining unknown regions. (h) Result of geometry reconstruction.

optimization. As shown in Fig.8, we want to fill the missing geometry on the Venus. 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 target surface. By smoothing the source region S, we get its signed gray-scale texture. Using the constrained optimization texture completion technique described in above 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.9. The defective region of model Lady is completed by transferring the geometric details of the bob region on model Venus.
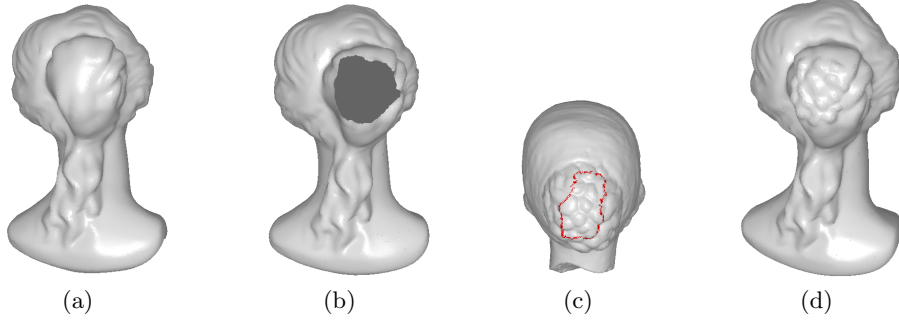
Fig. 9. Geometry completion based on geometry transfer. (a) Original lady model, (b), uncompleted lady model, (c) the sample model Venus, (d) the completion result.

7 Implementation and Results

The proposed algorithm was implemented on a Microsoft Windows XP PC with a Celeron 2.00GHz CPU and 1.00GB 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.3, there are 240,000 points in the Stanford Bunny, the execution time per iteration takes 8-10 seconds, and the total execution time is 10-12 minutes 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 6, there are 22,000 points in the patched regions, it takes less than one minute to accomplish the texture completion operation. In our experiments, there are usually about 300 points in each overlapped cluster, the scale of the synthesized texture can be controlled by adjusting size of the regular grid G.

8 Conclusion and Future Work

We have presented a novel approach for appearance and surface content completion for the acquired 3D data set based on global optimization. We 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. Our system can be extended to the mesh models easily. Further research will be focused on performing a user controllable non-uniform clustering so that the scale of synthesized texture can vary progressively through the point cloud.

Acknowledgement

References

1. M. Alexa, T. Klug, and C. Stoll. Direction fields over point-sampled geometry. *Journal of WSCG*, 11(1):27–32, 2003.
2. U. Clarenz, M. Rumpf, and A. Telea. Finite elements on point based surfaces. In *Proc. EG Symposium of Point Based Graphics*, 2004.
3. J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *ACM SIGGRAPH*, pages 67–76, 2001.
4. Y. Ohtake, A. Belyaev, and H-P Seidel. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In *Shape Modeling International*, pages 153–161, 2003.
5. J. Davis, Stephen, R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 428–438, 2002.
6. P. Liepa. Filling holes in meshes. In *Proceedings of the Eurographics/ACM SIG-GRAPH symposium on Geometry processing*, pages 200–205, 2003.
7. J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. In *Proceedings of International Conference on Image Processing*, pages 903–906, 2003.
8. T. Ju. Robust repair of polygonal models. *ACM Trans.Graph*, 23(3):888–895, 2004.
9. A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Trans.Graph*, 23(3):878–887, 2004.
10. M. Pauly, N. Mitra, J. Giesen, M. Gross, and L. J. Guibas. Example-based 3d scan completion. In *Symposium on Geometry Processing*, pages 23–32, 2005.
11. Y.-K. Lai, S.-M. Hu, D. X. Gu, and R. Martin. Geometric texture synthesis and transfer via geometry images. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 15–26, 2005.
12. S. Park, X. Guo, H. Shin, and H. Qin. Shape and appearance repair for incomplete point surfaces. In *International conference on computer graphics*, pages 1260–1267, 2005.
13. M. X. Nguyen, X. Yuan, and B. Chen. Geometry completion and detail generation by texture synthesis. In *Proceeding of Pacific Graphics*, pages 23–32, 2005.
14. V. Kwatra, I. A. Essa, A. F. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Trans.Graph*, 24(3):795–802, 2005.
15. M. Pauly, M. Gross, and L. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization*, pages 1260–1267, 2002.
16. C. Xiao, Y. Miao, S. Liu, and Q. Peng. A dynamic balanced flow for filtering point-sampled geometry. *The Visual Computer*, 2006. to appear.
17. H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Reconstruction from unorganized points. In *ACM SIGGRAPH*, pages 71–78, 1992.
18. J. Sun, L. Yuan, J. Jia, and H. Y. Shum. Image completion with structure propagation. *ACM Trans.Graph*, 24(3):861–868, 2005.