

AUTOMATIC TEXTURE RECONSTRUCTION OF 3D CITY MODEL FROM OBLIQUE IMAGES

Junhua Kang^a, Fei Deng^{a,*}, Xinwei Li^a, Fang Wan^a

^aSchool of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan430079, PR.China,-jkhkang@whu.edu.cn,
fdeng@sgg.whu.edu.cn

KEY WORDS: Texture Reconstruction, 3D City Model, Mesh Parameterization, Oblique Image

ABSTRACT:

In recent years, the photorealistic 3D city models are increasingly important in various geospatial applications related to virtual city tourism, 3D GIS, urban planning, real-estate management. Besides the acquisition of high-precision 3D geometric data, texture reconstruction is also a crucial step for generating high-quality and visually realistic 3D models. However, most of the texture reconstruction approaches are probably leading to texture fragmentation and memory inefficiency. In this paper, we introduce an automatic framework of texture reconstruction to generate textures from oblique images for photorealistic visualization. Our approach include three major steps as follows: mesh parameterization, texture atlas generation and texture blending. Firstly, mesh parameterization procedure referring to mesh segmentation and mesh unfolding is performed to reduce geometric distortion in the process of mapping 2D texture to 3D model. Secondly, in the texture atlas generation step, the texture of each segmented region in texture domain is reconstructed from all visible images with exterior orientation and interior orientation parameters. Thirdly, to avoid color discontinuities at boundaries between texture regions, the final texture map is generated by blending texture maps from several corresponding images. We evaluated our texture reconstruction framework on a dataset of a city. The resulting mesh model can get textured by created texture without resampling. Experiment results show that our method can effectively mitigate the occurrence of texture fragmentation. It is demonstrated that the proposed framework is effective and useful for automatic texture reconstruction of 3D city model.

1. INTRODUCTION

In recent years, we have witnessed the technological advancements such as faster computation speed and more accurate sensory data acquisition in the field of photogrammetry, remote sensing and computer vision (Kim and Rhee, 2014). The photorealistic 3D city models are increasingly important in various geospatial applications related to virtual city tourism, 3D GIS, urban planning, real-estate management. It represents layouts, activities and functionalities of a real-world community and can enable human users to better visualize their living spaces. To reconstruct a realistic and accurate 3D city model, there are two main issues that are involved and studied. The first problem is the accurate reconstruction of high-precision 3D geometric models. The other is the realistic texture mapping of model surfaces. Over the last few years a great amount of research has been conducted on geometric reconstruction for 3D city models. Some photogrammetric softwares such as Smart3DCapture by Acute3D and Photoscan by Agisoft can provide high quality 3D mesh models (Zhu et al., 2015). Therefore, texture reconstruction has become a crucial factor influencing the realistic 3D model. High-resolution and high detailed texture can make the visual appearance of city model more realistic. Recently, with the development of the oblique photogrammetry technology, oblique image has become a powerful source for texture reconstruction of realistic 3D digital city due to the advantages of big coverage, sufficient and accurate texture information (Deng and Kang, 2015; Fabio Remondino, 2015). However, the large deformation between different angle views makes oblique image matching more difficult. In the meantime, high requirements of computer memory are needed when processing oblique image data with large quantities. All of

these have proposed a challenge for 3D texture reconstruction using oblique images.

Many researchers have been working on texture reconstruction over the last few years. In the earlier research, a vertex shader method is quite popular, which directly extract the RGB color from the aerial images through back-projection and set the values to every polygon vertex of 3D model (Debevec et al., 1998; Soucy et al., 1996). The general drawback of this method is low resolution of texture. In this method, each surface triangle's texture is generated from the interpolation of the corresponding vertices' color, which results the less detailed representation. Then straightforward approaches are introduced to improve the texture quality. In these methods, texture is applied instead of color to each triangle by computing (u, v) coordinates on the original image for all vertices of model (Kim and Rhee, 2014; Luo et al., 2015; Tsai and Lin, 2007; Zhou et al., 2015). Due to the large coverage for oblique images, it needs to select the optimal image for texturing. In (Kim and Rhee, 2014) and (Luo et al., 2015) textures are selected based on the visibility only. In (Zhou et al., 2015) a simple quality measure is introduced for selecting suitable image based on the viewing angle. They choose the optimal texture by calculating the angle of the normal vector and center of projection direction. In (Frueh et al., 2004) and (Iwasczuk et al., 2015) geometric and matching qualities are defined based on geometric resolution, occlusion, viewing angle and projection accuracy. However, these methods are memory inefficiency since all vertices in mesh's (u, v) coordinates are computed from all original images. At the meantime, texturing

* Corresponding Author

neighboring triangles from different images can make the facade texture image fragmentary.

In this paper, we introduce an automatic framework of texture reconstruction to generate blended textures from oblique images for photorealistic visualization. In our method, a final texture map is generated to be used for rendering the whole reconstructed model, which can effectively mitigate the occurrence of texture fragmentation. The core technique of our methodology involves: Mesh segmentation, mesh unfolding, texture atlas generation and texture blending. The detailed description of our texture reconstruction framework will be presented in the follow section.

2. METHODOLOGY

In this section, the proposed scheme for automatic texture reconstruction will be presented. Our focus is texture reconstruction of 3D mesh surfaces for attaining photo-realistic 3D models. We already have an approximate surface model which is represented by a standard triangle mesh. Oblique images have been precisely calibrated and accurate exterior and interior parameters are obtained. Figure 1 illustrates the flowchart of our texture reconstruction approach. In which, several procedures are performed as follows: 1) mesh segmentation; 2) mesh unfolding; 3) texture atlas generation; 4) texture blending. First, we employ a variational shape approximation approach considering the local surface consistency to segment the 3D mesh into nearly planar regions. Then these regions are unfolded onto a 2D texture plane without any overlap by using a topology-preserving mapping method called Least Squares Conformal Maps (LSCM). During this process, the (u, v) coordinates for each mesh triangle in texture domain are calculated. Texture information of each region on texture atlas is acquired from all visible images. Occlusion detection based on visibility analysis is taken into account in this step to solve self-occlusion problem. After that, the final texture is generated by combining texture from several corresponding image regions using a blending scheme.

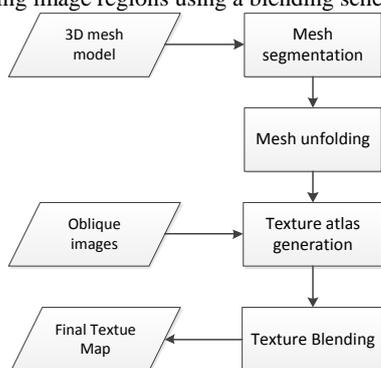


Fig.1. Whole flow of texture reconstruction from oblique images

2.1 Mesh parameterization

2.1.1 Mesh segmentation: Our triangulated surface mesh is generated from the Smart3DCapture software and is an important shape representation of complex 3D city model. The first sub problem for our texture reconstruction method is to divide the mesh surface into a collection of patches that can be unfolded with little stretch. To guarantee the distortion when flattening each region onto a plane is sufficiently small, we employ a variational shape approximation approach to segment the 3D mesh into nearly planar regions by analysis of planarity of 3D surface (Cohen-Steiner et al., 2004). This approach

optimally approximates a mesh surface by a specified number of planar faces using the Lloyd algorithm (Lloyd, 1982), which is commonly used for solving the k-mean problem in data clustering (Yan et al., 2006).

We assume M is the mesh model, $R = \{R_i\}_{i=1}^n$ represents the non-overlapping surface regions of M . All triangles in every region can be represented with $T_i = \{t_m^i\}_{m=1}^{n_i}$, and $\bigcup_{i=1}^n T_i = T$.

We use $P_i = (X_i, N_i)$ to represent the fit plane which fitted from the surface region R_i . X_i and N_i represent the average vertices and average normal respectively. We define an objective function $F(R, P)$ that evaluates the error between region R and fit plane P . The optimal partition can then be found by minimizing $F(R, P)$ for a certain partition size n . The square error of the piecewise planar approximation of the input mesh is chosen here as objective function $F(R, P)$. For a segment S_i the error between the region R_i and fit plane P_i is given by:

$$E(S_i) = E(R_i, P_i) = \sum_{m=1}^{n_i} d(t_m^i, P_i) = \sum_{m=1}^{n_i} \iint_{x \in R_i} \|n(x) - n_i\|^2 dx \quad (1)$$

where $d(t_m^i, P_i)$ represent the error between the triangle t_m^i and fit plane P_i , $n(x)$ is the normal of vertex x which belong to region R_i , n_i is the average normal of fit plane P_i . Then the objective function is simply defined as:

$$F(R, P) = \sum_{i=1}^n E(S_i) = \sum_{i=1}^n E(R_i, P_i) \quad (2)$$

By minimizing the objective function, several best segmented regions can be obtained.

2.1.2 Mesh unfolding: As discussed previously, several surface regions are obtained with the mesh segmentation procedure. Considering the surface-to-texture mapping problem, these disjoint regions need to be unfolded to the texture domain. That means, a mapping relationship need to be constructed that maps each point on the surface of a region onto the corresponding texture space. In this section, a mesh parameterization method least-squares conformal map (LSCM) (LEVY et al., 2002) is employed to establish a one-to-one mapping relationship between object space and texture space (LEVY et al., 2002; Li et al., 2011).

LSCM was first proposed by (LEVY et al., 2002). Based on a least-square approximation of the Cauchy-Riemann equations, LSCM treats the unfolding problem as finding the minimum of an objective function which was defined minimizing the local angle deformations. Here, we give a brief description of LSCM. The LSCM parameterization algorithm generates a discrete approximant of a conformal map. This conformal map preserves the local isotropy and keeps the line element unchanged. Given a discrete 3D surface region S and a smooth target mapping, this mapping can be viewed as a function $t = \psi(s)$, where $t = u + iv$ and $s = x + iy$. According to the properties of conformal mapping ψ is conformal on S if and only if the Cauchy-Riemann equation satisfies the following condition:

$$\frac{\partial \psi}{\partial x} + i \frac{\partial \psi}{\partial y} = 0 \quad (3)$$

However, since this conformal condition cannot in general be strictly enforced on whole triangulated mesh, a least squares sense is adopted to minimize the violation of the conformality condition. Therefore, the objective function is defined as:

$$E(S) = \sum_{T \in S} \int_T \left| \frac{\partial \psi}{\partial x} + i \frac{\partial \psi}{\partial y} \right|^2 dA = \sum_{T \in S} \left| \frac{\partial \psi}{\partial x} + i \frac{\partial \psi}{\partial y} \right|^2 A_T \quad (4)$$

Where T is a triangle on the surface region S , A_T is the area of triangle T and notation $|z|$ stands for the modulus of the complex number z (LEVY et al., 2002). The objective function in (4) can be minimized using least-squares minimization method. Thus, the 3D surface mesh can mapped to a 2D parametric plane with multiple correspondences as constraints by using the LSCM technique. The planar coordinates in parametric space (u, v) of 3D triangular meshes are obtained.

2.2 Texture atlas generation

After having unfolded each surface region to a 2D parametric plane using the LSCM technique, the next step is to reconstruct a diffuse texture for each region. In this section, texture atlas is generated for the reconstructed mesh model by using sufficient texture information of oblique images from different views.

2.2.1 Photo selection and visibility check: Any 3D point in mesh model can be projected into 2D original images to retrieve the color by knowing the exterior and interior orientation parameters. However, since not each original oblique image carries all texture information for the whole mesh model, a photo selection method first be used to filter out the images that have enough information about a particular 3D model. To find out if a given 3D point is visible in a certain image, we first transform the 3D point from world coordinate system into camera coordinate system using the known exterior orientation parameters. Then pixel coordinate of the point can be obtained using the interior orientation parameters. If the pixel coordinates are in the range of the image dimensions, this 3D point is visible in this camera's view. However, in an urban context, there is always complex occlusion between buildings, which makes it difficult to recognize if a 3D point was truly observable by the camera. Thus, in our paper, a visibility analysis method is used to test if a 3D point is occluded or not.

We use a well-known z-buffer technique for visibility analysis, which was first proposed by (Greene et al., 1993). This method is a 3D-based method and is commonly used in computer graphics for visibility computations. In this method, each selected original image is analyzed individually. An image matrix with the same resolution as the image is created and used to store the distance (Z) of each pixel between the object surface and the projection center. For two 3D points, if their texture coordinates are located in the same pixel of original image, we compute the distance from point to perspective

center. if the distance of one point is lesser than the other, the same location of the corresponding image matrix is stored the lesser distance. Therefore, only pixels whose rays don't intersect any other feature along the way from point to perspective center are considered to be visible.

2.2.2 High-resolution texture atlas generation: As previously discussed, we perform the visibility analysis procedure for all vertices in triangle mesh and get the color from the original images, and then resulting texture map can be generated using these color information. However, the texture resolution is very poor and it varies over the model. The reason is that this process only obtains a color for each vertex of the triangle mesh. The color between vertices is interpolated from the surrounding vertex colors by the rendering system. In order to generate a high-resolution texture atlas, a mesh resample method is proposed to obtain a densely sampled mesh.

In mesh parameterization step, 3D triangle mesh has been mapped to a texture plane by using the LSCM technique. This texture mapping method is bijective, which means a triangle in 3D surface mesh uniquely maps to a 2D triangle in the texture space and the order of triangle vertices will be preserved. So any 2D point in texture space can either uniquely be mapped into one triangle in 3D surface or not part of the surface at all. For the mesh resample, we use an equidistant point grid in texture space as shown in figure 2, and each grid point represents a texture pixel. In order to determine the location of a grid point $p_i = (u_i, v_i)$ in the corresponding 3D triangle, we calculate the barycentric coordinate of p_i in texture plane. As Figure 2(b) shows, p_1, p_2 and p_3 are the vertices of a triangle t in object space, the barycentric coordinates of p_i with respect to the vertices of the triangle t can be represented as: $\mathbf{p}_i = (b_1, b_2, b_3)$, and that $\mathbf{p}_i = b_1 \mathbf{p}_1 + b_2 \mathbf{p}_2 + b_3 \mathbf{p}_3$, with $b_1 + b_2 + b_3 = 1, b_i = Area(t_i) / Area(t)$. $Area(t_i)$ denotes the signed area of the triangle. Since the barycentric coordinates are proportional to ratios of signed triangle areas, the location of p_i in object space can be obtained using the corresponding barycentric coordinates in texture space. Lastly, as figure b(c) shows, (x_i, y_i) is the texture coordinates of point p_i in image space, which calculated using the known exterior and interior parameters. Therefore, the color of grid point p_i in texture space can be given with the corresponding pixel of (x_i, y_i) in image space. Grid points that are not part of the triangle mesh and not visible are given black color. Then resulting texture atlas is generated with several selected oblique images.

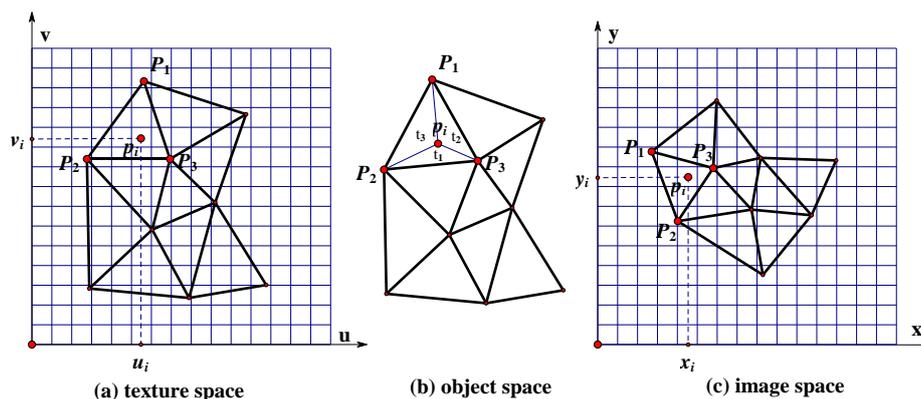


Fig.2. Mesh resample for high-resolution texture generation

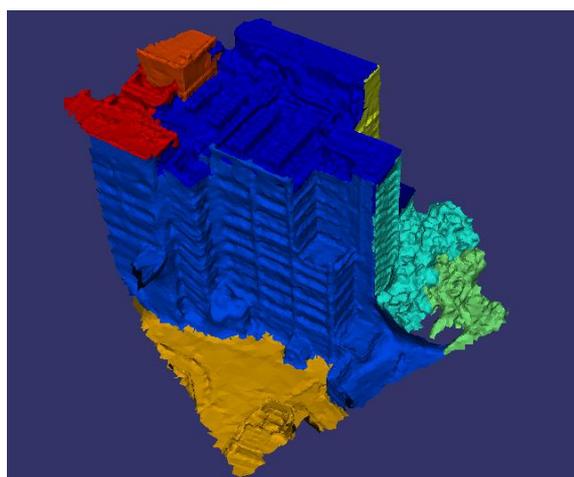
2.3 Texture blending

As mentioned above, for every selected original image, a texture is generated. Then several texture maps for each image are produced, which called texture atlas. Due to varied illumination conditions and different perspective angle view of original images, discontinuity artifacts usually become visible. For a consistent texturing we need to blend these texture maps in the overlapping regions to obtain a final texture map. We use a smart blend algorithm to blend texture in the overlapping regions, which is proposed by Michael Norel for seamless image blending (Norel, 2012). In this method, an image can be viewed as a collection of different frequency signals from the viewpoint of the frequency theory. The images are first decomposed into multi band-pass filtered component images. Then these component images are blended in different frequency domains. In this blending procedure, the width of the transition zone plays a crucial role and need be calculated. A seams detection algorithm is used to find where to cut pictures. It finds the visual error on overlap region and searches for a seam line with minimal visual error, then the transition can be done around the seam line. The final texture map can be obtained by using this smart blend technique, and the discontinuities at boundaries between texture regions are removed.

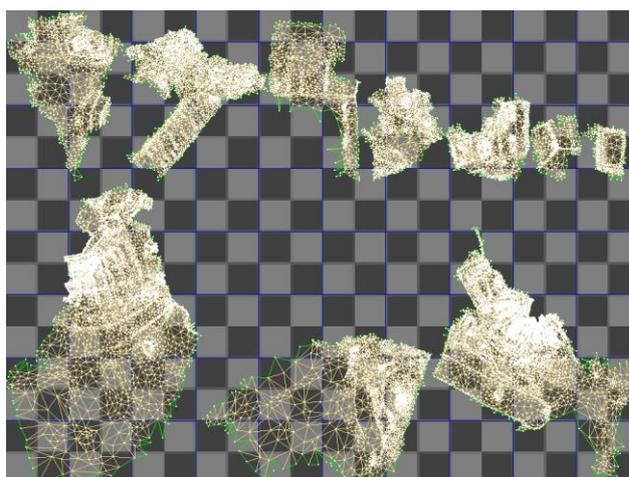
3. EXPERIMENT RESULTS

To test the performance of our texture reconstruction method, comprehensive experiment using real data were conducted, and the test area is located in the city of Hong Kong. In our experiment, 87 oblique images are acquired with a resolution of 4000 x 3000 pixels at an average flying height of 250 m above the mean ground elevation of the imaged area. The focal length of the camera is 5.35 mm, and the pixel resolution of the oblique images is 1.86 μ m. Since our method only focus on texture mapping of mesh model, before our experiment, the 3D triangle mesh model is generated using the Smart3DCapture software. The oblique images have been precisely calibrated and accurate exterior and interior parameters are known.

Based on the 3D triangle mesh, the first step of our texture reconstruction procedure is surface segmentation. Figure 3(a) depicts the results of previously described segmentation method for a building mesh model, and each color represents a region. We can see that the building has been broken into 10 regions and each region is a nearly planar region, which is useful for reducing the distortion of mesh unfolding step. The sharp feature of 3D building model is preserved to easily reconstruct a texture for each piece individually.



(a) 10 regions by our segmentation method



(b) Mesh unfolding result using LSCM technique

Fig.3. Results of segmentation and unfolding for a building model

Second, we use the LSCM technique to unfold the disjoint regions to 2D plane using the proposed mesh unfolding method.

Figure 3(b) is the result of mesh unfolding for previous mesh model which have been segmented to 10 regions. Most of them

are corresponding to geometric details of the models. Every region has been mapped to the texture domain. The mesh model consists of 24923 vertices and 49150 triangles, and has 4 holes. The result shows that our mesh unfolding method can also handle mesh with holes efficiently, and there are no overlapping triangles in 2D texture domain.

From the point of view of the statistical parameters, here we use the angle and area deformation to demonstrate the accuracy of mesh unfolding. The left histogram in Figure 4 shows the distribution of angle deformation. Vertical axis represents the number of angles in test mesh model, and horizontal axis shows the rate of change. We calculate every angle of triangles before and after parameterization, and use the rate of angle change to

present angle deformation. As left histogram shows, most of the ratios are closed to 1, which means the differences of angles before and after unfolding are near to zero. That is, the mapping technique is nearly conformal in each triangle. The right histogram of figure 4 shows the area deformations. Vertical axis represents the number of triangles in test mesh model, and horizontal axis shows the rate of change. We calculate the texture area and model area of each triangle, which have been normalized. The rate of area change is used to present area deformation. As can be seen, even though the LSCM technique is not area preserving mapping but conformal mapping, few facets are distorted and the area deformations are acceptable.

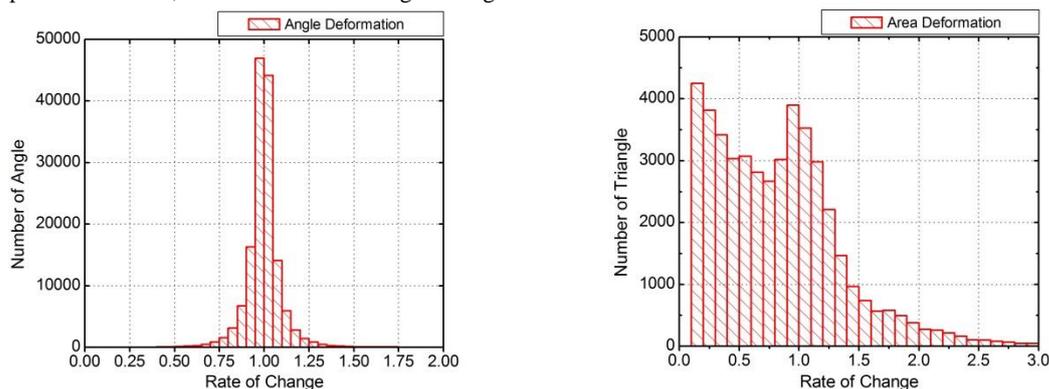


Fig.4. Angle (left) and area (right) deformations histograms

After mesh unfolding, the next step runs to texture atlas generations and texture blending. Figure 5 shows the generated texture maps based on visibility analysis. Every texture map may contain different color information for each region because of the self-occlusion. Comparing Figs. 5(a) and 5(b), the region in red rectangle of texture map 1 has little texture

information, and the corresponding part in texture map 2 has enough information (see red rectangle in figure 5(b)). However, another region in green rectangle of texture map 1 has more color information relative to the same region in texture map 2 (see green rectangle in figure 5(b)), and the texture in yellow region of this two texture map are both abundant.

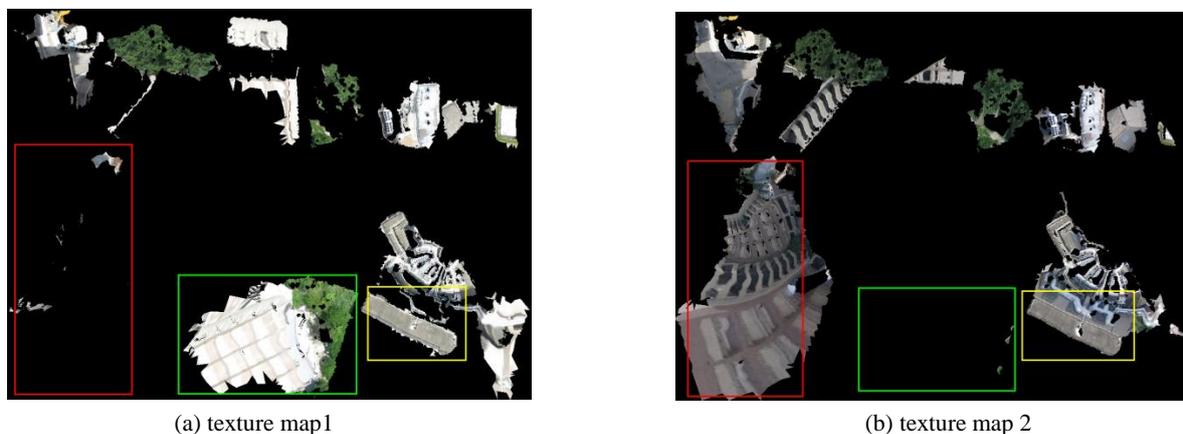
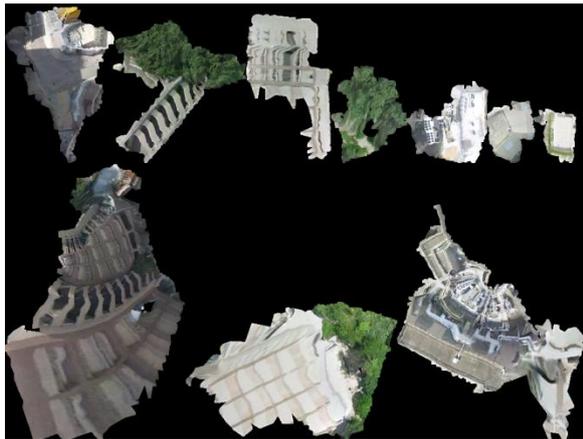


Fig.5. Results of texture atlas generation

Due to there are different color information in the same regions of different texture maps. We use the smart blend technique to blend this texture maps to generate a final texture map with complete and high-quality texture information (figure 6(a)). As shown in figure 6(a), the smart blend method can obtain seamless texture with no jumps in color and lighting appearance. Every region has enough texture information and discontinuities at boundaries between texture regions reconstructed from different images are removed. Figure 6(b)

shows the reconstructed model textured with the final texture map. The textures are complete and align correctly with the model. As we can see from the visual effect, there is no occurrence of texture fragmentation in textured model. Figure 7 shows the final output of the photo-realistic visualization for a part of city scene. These experimental results demonstrated that our texture reconstruction method is an effective and efficient automatic approach to produce near photo-realistic visualization for 3d city models.



(a) Final texture map by texture blending



(b) results of textured model

Fig.6. The final texture map and textured model

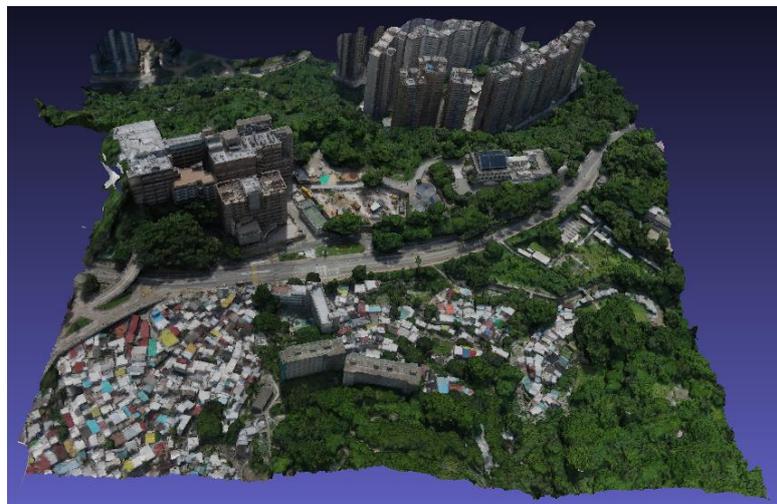


Fig.7. Photo-realistic visualization of a part of city scene

4. CONCLUSIONS

This study proposes an automatic texture reconstruction method for generating photo-realistic 3D city models using oblique images. The core technique of our methodology involves: Mesh segmentation, mesh unfolding, texture atlas generation and texture blending. The mesh segmentation is a pre-processing step used to segment a complex surface mesh in regions with less complexity, which can be unfolded with little stretch. We employed a variational shape approximation approach considering the local surface consistency to segment the 3D mesh into several regions. The results showed that this segmentation procedure results in nearly planar regions with low complexity. The unfolding step maps the disjoint regions onto a corresponding texture. We used a least-squares conformal mapping to establish the mapping relationship. The experiment results demonstrated that this mapping minimized geometric distortions, the angle and area deformations over all surface triangles are acceptable. Then the texture of each region in texture domain is reconstructed from original images. In this step, visibility analysis is taken into account to detect occlusion. This procedure results in several high resolution texture maps. In the last, the smart blending technique is used to obtain a final texture map. The blending results showed that the color discontinuities at boundaries between texture regions reconstructed from different images are removed. We used oblique images and mesh model to validate the applicability of

our proposed methodology. Experiment results show that our method can effectively mitigate the occurrence of texture fragmentation. The resulting mesh model can get textured by created texture map that provides full representation of the texture data. It is demonstrated that the proposed framework is effective and useful for automatic texture reconstruction of 3D city model.

In our paper, we only tested our method using the oblique images. Since the ground images are acquired easily and contain more facade information for building. On the account of idea's generality, In the future research, our texture reconstruction framework can be extended to deal with indoor objects.

5. ACKNOWLEDGMENTS

This study was supported by the National Natural Science Foundation of China (Grant No. 41271431).

6. REFERENCES

Cohen-Steiner, D., Alliez, P. and Desbrun, M., 2004. Variational shape approximation. *ACM Transactions on Graphics (TOG)*, 23(3): 905--914.

- Debevec, P., Yu, Y. and Borshukov, G., 1998. Efficient view-dependent image-based rendering with projective texture-mapping. Springer.
- Deng, F. and Kang, J., 2015. Automatic true orthophoto generation based on three-dimensional building model using multiview urban aerial images. *Journal of Applied Remote Sensing*, 9(1): 095087-095087.
- Fabio Remondino, T., 2015. Oblique Aerial Imagery - A Review.
- Frueh, C., Sammon, R. and Zakhor, A., 2004. Automated texture mapping of 3D city models with oblique aerial imagery. *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on. IEEE*, pp. 396--403.
- Greene, N., Kass, M. and Miller, G., 1993. Hierarchical Z-buffer visibility. *Conference on Computer Graphics and Interactive Techniques*, pp. 231-238.
- Iwaszczuk, D., Hoegner, L. and Stilla, U., 2015. Quality-based building-texture selection from different sensors, *Urban Remote Sensing Event (JURSE), 2015 Joint. Urban Remote Sensing Event (JURSE), 2015 Joint. IEEE*, pp. 1--4.
- Kim, C. and Rhee, E., 2014. Realistic Façade Texturing of Digital Building Models. *International Journal of Software Engineering and Its Applications*, 8(1): 193--202.
- LEVY, B., Petitjean, S., Ray, N. and Maillot, J.E.R., 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (TOG)*, 21(3): 362--371.
- Li, W., Yin, Z., Huang, Y., Wu, T. and Xiong, Y., 2011. Tool path generation for triangular meshes using least-squares conformal map. *International journal of production research*, 49(12): 3653--3667.
- Lloyd, S.P., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129-137.
- Luo, Q., Zhou, G., Zhang, G. and Huang, J., 2015. The texture extraction and mapping of buildings with occlusion detection, Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. IEEE, pp. 3002--3005.
- Norel, M., 2012. SmartBlend.
- Soucy, M., Godin, G. and Rioux, M., 1996. A texture-mapping approach for the compression of colored 3D triangulations. *The Visual Computer*, 12(10): 503--514.
- Tsai, F. and Lin, H., 2007. Polygon-based texture mapping for cyber city 3D building models. *International Journal of Geographical Information Science*, 21(9): 965--981.
- Yan, D., Liu, Y. and Wang, W., 2006. Quadric surface extraction by variational shape approximation. *Geometric Modeling and Processing-GMP 2006*. Springer, pp. 73--86.
- Zhou, G., Ye, S., Wang, Y., Han, C. and Wang, C., 2015. Texture mapping based on multiple aerial imageries in urban areas, International Conference on Intelligent Earth Observing and Applications. International Conference on Intelligent Earth Observing and Applications. International Society for Optics and Photonics, pp. 98080Z--98080Z.
- Zhu, Q., Jiang, W. and Zhang, J., 2015. Feature Line Based Building Detection and Reconstruction from Oblique Airborne Imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(4): 199.