



## Color adjustment in image-based texture maps

Rongjiang Pan<sup>a,c,\*</sup>, Gabriel Taubin<sup>b</sup>



<sup>a</sup> School of Computer Science and Technology, Shandong University, Jinan, China

<sup>b</sup> School of Engineering, Brown University, Providence, United States

<sup>c</sup> Engineering Research Center of Digital Media Technology, Ministry of Education of PRC, Jinan, China

### ARTICLE INFO

#### Article history:

Received 22 December 2014

Revised 16 April 2015

Accepted 17 April 2015

Available online 23 April 2015

#### Keywords:

Texture map

Color adjustment

Poisson equation

Visible seam

Multi-view 3D reconstruction

### ABSTRACT

We propose a color adjustment technique to eliminate the visible seams in image-based texture map of a 3D object. The process is carried out in three steps. First, texture coordinates are locally displaced to minimize the misalignment of adjacent texture patches. Second, color discontinuities between different texture patches at each corner of the mesh faces are resolved. We minimize a global energy function over the mesh to ensure continuous color transitions and fit the color gradient at each corner of the mesh faces. Finally, the color adjustment at the corners is propagated over the texture patch for each face by solving a Poisson equation with mixed boundary conditions. By means of the proposed processing techniques, the visibility of seams is minimized while fine details are preserved in image-based texture maps. This can be used as a last refinement stage in image-based 3D reconstruction pipelines. The proposed color adjustment algorithm is tested on a variety of real-world datasets and compares very favorably with known methods.

© 2015 Elsevier Inc. All rights reserved.

### 1. Introduction

During the past few years, dramatic progress has been made in the area of image-based object modeling. A typical image-based 3D reconstruction pipeline begins with the acquisition of a photo collection of an object. Then, the camera poses, i.e. the position, orientation and intrinsic parameters of each camera, are estimated using structure-from-motion (SfM) techniques on the photos. Afterwards, a 3D geometric model is recovered using multi-view stereo (MVS), silhouette and contours cues. Finally, the calibrated images are utilized to generate a texture map for the recovered geometric model. A texture map is an important component that makes the model more realistic and compensates for deficiencies in the camera registration and reconstructed geometry.

In the final step, the images are usually back-projected onto the geometric model to generate a texture patch for each surface patch. However, every surface patch is usually

visible in several input images and there is often no one-to-one correspondence between texels in the texture and pixels in the images. The measured color and intensity for a surface patch observed in different photos will not agree due to lighting, camera setting and surface reflectance variations across the views. Visible seams and color discontinuities often arise between neighboring texture patches. The problem can be alleviated by blending multiple image views per face and careful selection of views. However, blending often causes image quality degradation due to resampling and misalignment of geometric projections with inaccurate reconstructed geometry and camera parameters. Optimal view selection cannot completely eliminate color discontinuities and seams in the presence of large photometric difference between images.

In this paper, we propose a color adjustment technique to obtain a seamless texture map from multiple registered images. The process is carried out in three steps. First, texture coordinates are locally displaced to minimize the misalignment of adjacent texture patches. Second, color discontinuities between different texture patches at each corner of the mesh faces are removed through global optimization. Finally, the color adjustment at the corners is propagated over the texture patch for each face by solving the Poisson equation. The

\* Corresponding author.

E-mail addresses: [panrj@sdu.edu.cn](mailto:panrj@sdu.edu.cn) (R. Pan), [taubin@brown.edu](mailto:taubin@brown.edu) (G. Taubin).

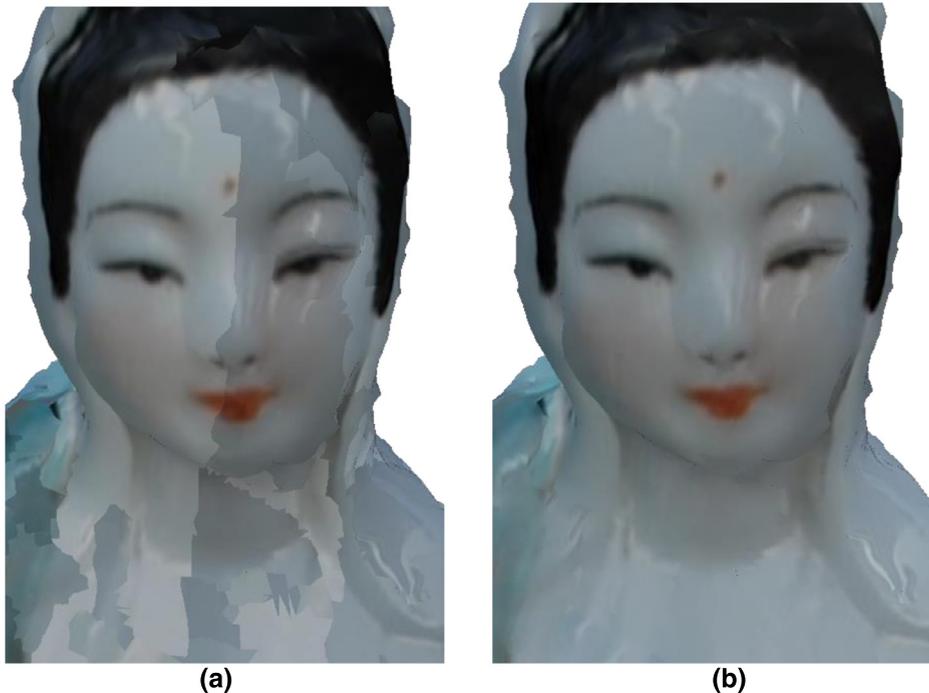


Fig. 1. The reconstructed model with texture from registered images (a) before color adjustment and (b) after color adjustment.

proposed color adjustment algorithm has the following advantages compared to previous approaches:

- The misalignment of adjacent texture patches is solved by local displacement of texture patches to minimize the photometric discrepancy function at each vertex.
- A global energy function is designed to minimize the color discontinuities between texture patches and match the desired gradients at each corner.
- Solving the Poisson equation with mixed boundary conditions (gradients on the boundary and specified values at corners) for each texture patch produces smooth color transition across the boundary of texture patches and keeps the local change within each texture patch.
- By combining global color adjustment for corners of the mesh faces and local color adjustment within each texture patch, it is computationally efficient to obtain high-quality textures for large real-world datasets.
- The proposed color adjustment algorithm is independent of the particular choice of view selection and texture representation.

An example is shown in Fig. 1. With our color adjustment algorithm, the visibility of seams in Fig. 1(a) is minimized and fine details are maintained in the final texture map (Fig. 1(b)).

## 2. Related work

Texturing a 3D model using images is a fundamental and extensively studied problem in computer graphics. For casual images, user-defined constraints are usually required to establish a mapping from the surface to the image. Constrained

parameterization, photogrammetric approach with estimated camera parameters for the entire model and local camera parameters for each vertex of the mesh are proposed [1].

In the context of image-based modelling, the images and the geometric model are registered within the global coordinate system. The model can be reprojected onto the source images using the estimated camera parameters. Each surface patch is usually visible in several image views. To create a single texture from multiple images for 3D reconstruction, one faces two problems: how to select views to texture each face and how to use these views to generate the texture patch.

Some approaches select multiple views for each face based on visibility determination. Information from multiple views is blended with heuristic weighting functions [2–5], such as, viewing angle, distance between the surface and the camera, and projection area. Blending colors from different image views often causes image quality degradation, either in image space [2,5] or in frequency space [3,4]. In case of inaccurate reconstructed geometry and camera parameters, the image patches from different views may be misaligned, and ghosting artifacts often appear in the texture patch. Furthermore, details and sharp features can be smoothed out when combining images at different scales. Recently, a super-resolution framework [6] was proposed to recover higher level-of-detail texture maps. A normal displacement map on the surface to improve the geometry estimates and the camera calibration parameters are also optimized. However, it requires a tremendous amount of computation for optimization in a very high dimensional space.

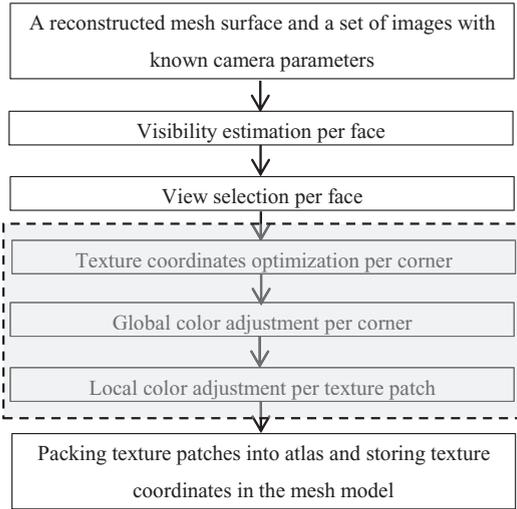


Fig. 2. Workflow of image-based texture map.

Other approaches assign exactly one view to each face [7–9]. The view is usually selected to make the seams between neighboring texture patches less visible by optimization. Velho and Sossai [8] considered view selection in distortion-based atlas generation process. They optimized an energy functional to reduce texture stretch and minimize the number of charts. Lempitsky and Ivanov [9] selected a single view per face using pairwise Markov random field optimization that favors smoothness in the texture assignment and penalizes the introduction of sharp seams. To overcome inaccuracies in camera registration and geometry reconstruction, local image transformations are introduced into the selection of views [7]. Waechter et al. [11] improved the pairwise Markov random field energy formulation by augmenting the data term with a photo-consistency check.

However, optimal view selection cannot completely eliminate color discontinuities and seams. Colors of adjacent texture patches need to be adjusted so that their seams become less visible. Luminance changes are often noticeable at texture patch borders by averaging the colors at a seam and using heat diffusion to achieve a smooth color transition [8]. Poisson blending [12] was applied to each texture patch in [7], where a one-pixel wide boundary around each patch is

setting to the average of the pixel values on two sides of the boundary. Multi-band image blending is applied to the vicinity of patch boundaries to achieve color continuity in [10]. Instead, Lempitsky and Ivanov [9] removed the mosaic seams via a global seam levelling procedure. After finding the optimal color adjustment on each corners, the color correction for each texel is interpolated using barycentric coordinates. An improved approach for seam-leveling is proposed in [13]. Waechter et al. [11] improved the global adjustment by introducing color lookup support region along the seams. A local color adjustment is subsequently performed with Poisson blending, where the mean of pixel colors from images assigned to the patch and the neighboring patch is used as Poisson equation boundary conditions. However, the leveling function in [9,11] is defined up to an additive constant which can be chosen arbitrarily. When the corrections are added to the input images to yield the final texture, some pixel values go out of range and unnatural spots result in the texture. In addition, averaging colors in boundary conditions of Poisson equation can also smooth out details in local color adjustment.

Our work focuses on color adjustment for texture patches to both make seams invisible and keep fine details. Based on previous work [9,11], we propose a novel energy functional in the global color adjustment and a local Poisson image editing with mixed boundary conditions. Details are described in Section 3.

### 3. Color adjustment for texture patches

Our method requires a 3D surface mesh model of the target object and a set of images with known camera parameters. The images and the geometric model are already registered within a global coordinate system. We assume that the reconstructed geometric model is given in a form of triangular mesh. After preprocessing, each face is assigned to exactly one view for creating its texture patch. The workflow is illustrated in Fig. 2. We highlight the key technical contributions of the proposed method. Note that our proposed method of color adjustment for texture patches does not rely on the visibility estimation and view selection described below, and will work with results generated by other methods. For color textures, we consider each channel separately and perform the global and local color adjustment steps independently for each channel.

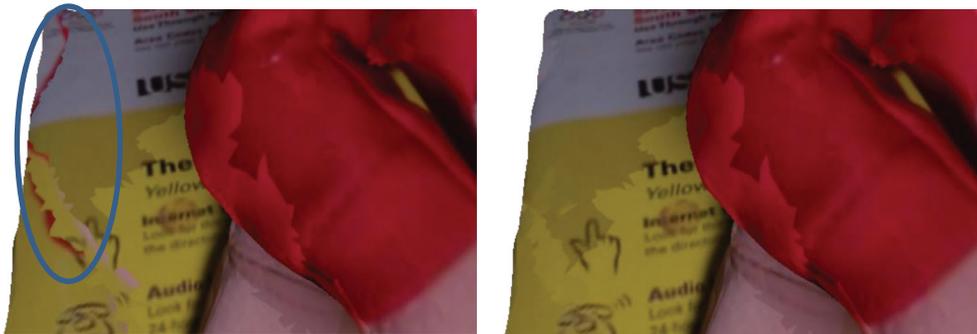


Fig. 3. Incorrect visibility detection in the vicinity of occluded area (marked in left) is solved through a displaced occlusion testing.



Fig. 4. Misalignment of texture patches from different camera views (marked in left) is alleviated through texture coordinates optimization.

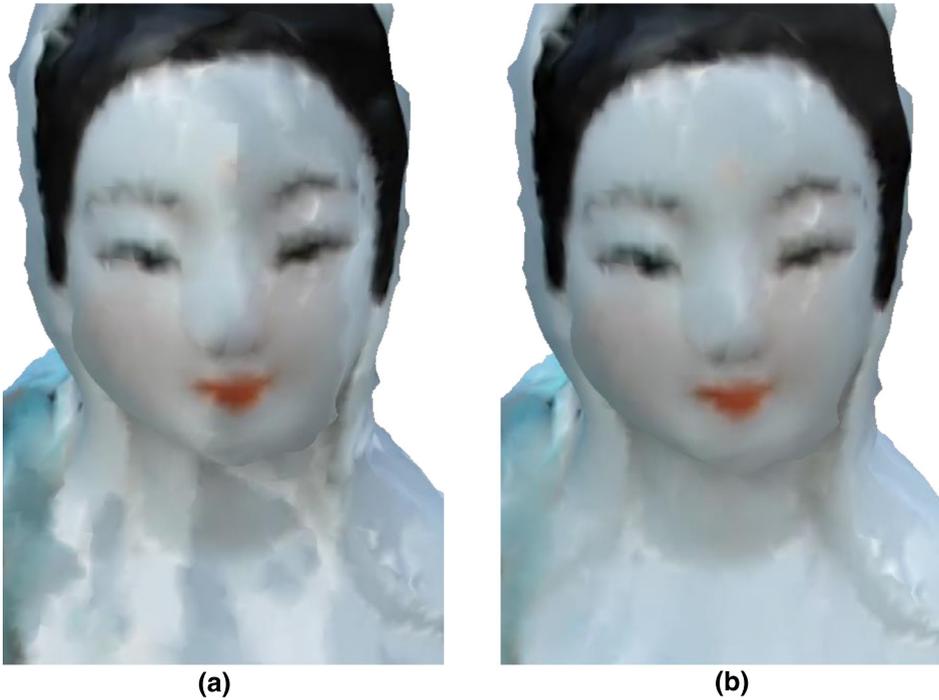


Fig. 5. The reconstructed model with per-vertex color by averaging colors of incident corners (a) before color adjustment and (b) after color adjustment.

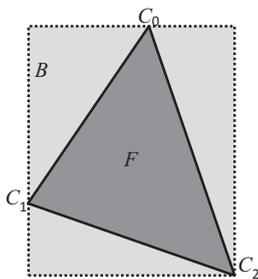


Fig. 6. The Poisson equation with mixed boundary conditions on a triangle face.

### 3.1. Visibility detection

To compute the visibility of face  $F$  from camera  $C$  with optical center  $\mathbf{o}$  and unit viewing ray  $\mathbf{v}$ , we first perform back

face culling. Then, we check whether the projection of face  $F$  into camera  $C$  lies within the camera image  $I$ . Afterwards, occlusions are tested by performing a ray cast from  $\mathbf{o}$  to each vertex  $\mathbf{p}$  of face  $F$  and checking the intersections with the input mesh.

Due to imperfect camera calibration and surface reconstruction, errors in the computation of visibility often occur in the vicinity of occluded areas (Fig. 3 left). We have developed a heuristic approach that removes the visibility of faces that incorrectly project into the vicinity of occluding boundaries in the camera view. We perform an additional occlusion testing by displacing  $\mathbf{o}$  along viewing ray  $\mathbf{v}$ . Concretely,  $\mathbf{o}$  is moved to  $\mathbf{o}' = \mathbf{o} + 0.3\|\mathbf{p} - \mathbf{o}\|\mathbf{v}$  and a ray is cast from  $\mathbf{o}'$  to each vertex  $\mathbf{p}$  of face  $F$  to check the intersections with the input mesh. A face is classified as visible in camera  $C$  if and only if it is visible in both the original and displaced occlusion testing. If a face is only visible from  $\mathbf{o}$ , information from other camera images is used to provide a better texture.



Fig. 7. The results of local color adjustment per texture patch. (a) Simple interpolation using barycentric coordinates. (b) Solution of the Poisson equation with mixed boundary conditions.



Fig. 8. Results of the Goddess statue. Left: before color adjustment; middle: output from Meshlab; right: output with our method. Bottom: close-up comparison.

### 3.2. View selection

To select the best view for each face, we take into account visibility, orientation, resolution and distortion of the face in each camera. Similar weighting metrics are also presented in previous research work [2,19]. For a face  $F$  with outward normal vector  $\mathbf{n}$ , center  $\mathbf{c}$  and a camera  $C$  with optical center  $\mathbf{o}$  and unit viewing ray  $\mathbf{v}$ , we define the weights as:

$$w_{vis} = \begin{cases} 1, & \text{if } F \text{ is visible in } C \\ 0, & \text{otherwise} \end{cases},$$

$$w_{ori} = -\mathbf{n} \cdot \mathbf{v}, \quad w_{res} = 1 / \|\mathbf{c} - \mathbf{o}\|$$

$$w_{dis} = \sqrt{1.0 - \max\left(\left|2\frac{x}{w} - 1\right|, \left|2\frac{y}{h} - 1\right|\right)},$$

$$w_{total} = w_{vis} \cdot w_{ori} \cdot w_{res} \cdot w_{dis} \quad (1)$$

where  $x$  and  $y$  are the pixel coordinates of face center  $\mathbf{c}$  projected in  $C$ ,  $w$  and  $h$  are width and height of the camera image  $I$ . We select the camera with largest weight  $w_{total}$  from the input camera set, and use the corresponding image to generate the texture for  $F$ .

The orientation weight prefers views whose view directions are almost parallel with the face normal. The resolution



Fig. 9. Results of the stone relief. Left: before color adjustment; middle: output from Meshlab; right: output with our method. Bottom: close-up comparison.



Fig. 10. Comparison on the Der-Hass dataset. Left: output of [11]. Right: our method.

weight prefers views whose optical centers are close to the face since close-up images capture more surface details. The distortion weight measures how far the projection of the face center in the image is from image borders, which takes in account quality degradation towards image borders.

### 3.3. Texture coordinates optimization

After the view selection per face, imperfect view registration and reconstructed geometry may lead to misalignment of

texture patches when two adjacent faces are textured from different camera views (Fig. 4 left). It is unlikely that view assignment and color adjustment techniques alone can fully resolve the problem of misaligned texture patches. Therefore, we propose a method to minimize the misalignment of adjacent texture patches by locally displacing the texture (Fig. 4 right).

For a vertex  $\mathbf{p}$ , let  $I(\mathbf{p}) = \{I_1, \dots, I_n\}$  denote a set of images from which the texture patches of  $\mathbf{p}$ 's incident faces are generated. If the size of  $I(\mathbf{p})$  is greater than one, we perform



Fig. 11. Results of the bust. (a) and (c) are two views before color adjustment. (b) and (d) are the corresponding views after our color adjustment.

a patch-based texture optimization. Patch model was successfully applied in multiview stereo matching and reconstruction [16]. Here we extend the idea to achieve seamless texture mapping. A reference image  $I_R$  is chosen from  $I(\mathbf{p})$  according to similar weighting metrics described in Section 3.2. At vertex  $\mathbf{p}$ , we create a patch  $pat(\mathbf{p})$  which is a rectangle with center  $\mathbf{p}$  and sides parallel to the axes of  $I_R$ . The extent of the rectangle is chosen so that its projection in  $I_R$  is of size  $\mu \times \mu$  pixels. In our experiments, we set  $\mu=7$  according to the suggestion of [16]. The photometric discrepancy function  $g(pat)$  for vertex  $\mathbf{p}$  is defined as

$$g(pat) = \sum_{\substack{\forall pairs(i, j), \\ i, j \in I(\mathbf{p})}} h(i, j) \quad (2)$$

where  $h(i, j)$  is one minus the normalized cross correlation score between the pixel colors of patch  $pat(\mathbf{p})$ 's projections in images  $I_i$  and  $I_j$ . In order to align the adjacent texture patches around vertex  $\mathbf{p}$ , we allow  $\mathbf{p}$  to displace so that the photometric discrepancy function  $g(pat)$  is minimized. To make the optimization robust, we constrain  $\mathbf{p}$  to lie on the ray from the optical center  $\mathbf{o}_R$  of the camera corresponding to  $I_R$  to  $\mathbf{p}$ . Thus the projection of  $\mathbf{p}$  in image  $I_R$  does not change and the number of degrees of freedom is reduced to one. More concretely, vertex  $\mathbf{p}$  is moved to  $\mathbf{p}' = \mathbf{p} + d(\mathbf{p} - \mathbf{o}_R) / \|\mathbf{p} - \mathbf{o}_R\|$  and  $pat$  is changed correspondingly. The optimal  $d$  is obtained by minimizing Eq. (2) with simplex algorithm of Nelder and Mead in GSL [20]. We initialize  $d$  with zero and step size with a quarter of the average edge length in the mesh.

### 3.4. Global color adjustment at each corners

For a mesh, let  $C_i^j$  be the corner of face  $F_j$  in vertex  $V_i$ ,  $f_i^j$  be the value of the corner in the original texture function,  $g_i^j$  be the value of the corner in the seamless image-based texture maps we want to seek for.  $N$  denotes the set of all pairs  $(i, j)$  such that  $C_i^j$  is a corner of the mesh. We want to compute an optimized function  $g$  at each corner of each face of the mesh by minimizing the following least-squares

energy:

$$\sum_{(i,j) \in N} (\nabla g_i^j - \nabla f_i^j)^2 + \alpha \sum_{\substack{(i,j_1) \in N \\ (i,j_2) \in N}} (g_i^{j_1} - g_i^{j_2})^2 + \beta \sum_{(i,j) \in N} (g_i^j - f_i^j)^2 \quad (3)$$

The first item in the energy ensures the gradient of  $g$  fits the gradient of  $f$ , the second item tries to make  $g$  smooth on each corner between neighbouring texture patches and the last term is a regularization to make the function  $g$  a good fit to the original function  $f$ . We use the mesh structure to approximate the gradients of  $g$  and  $f$ ,

$$\sum_{\substack{(i_1, j) \in N \\ (i_2, j) \in N}} [(g_{i_1}^j - g_{i_2}^j) - (f_{i_1}^j - f_{i_2}^j)]^2 + \alpha \sum_{\substack{(i,j_1) \in N \\ (i,j_2) \in N}} (g_i^{j_1} - g_i^{j_2})^2 + \beta \sum_{(i,j) \in N} (g_i^j - f_i^j)^2 \quad (4)$$

The variables  $g_i^j$  are obtained by solving the least square Problem (4) using a sparse solver. In our experiments, we set the parameter  $\alpha$  to 100 and  $\beta$  to 0.001.

Fig. 5 shows the result of global color adjustment. The reconstructed model is assigned color to each vertex by averaging colors of incident corners, with the original color on the left, our solution on the right. Note that the color discontinuities become less noticeable after our global color adjustment.

### 3.5. Color adjustment for the texture patch of each face

For a triangular face  $F$  with corners  $C_0, C_1, C_2$ , whose corrected texture intensities are  $g_0, g_1, g_2$ , we solve the Poisson equation with mixed boundary conditions,

$$\Delta u = \Delta f \quad \text{over the texture fragment, with} \\ u_i = g_i, \quad i = 0, 1, 2, \quad \text{and the gradients of } u \text{ equals to the} \\ \text{gradients of } f \text{ on the boundary.} \quad (5)$$

where  $u$  is the texture intensities after adjustment and  $f$  is the original texture intensities assigned to the face,  $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$  is the Laplacian operator.



Fig. 12. Results of the Der-Hass. Left: before color adjustment. Right: output with our method. Bottom: close-up comparison.

In our implementation, we solve the Poisson equation in the bounding box of each triangle face. Fig. 6 shows the triangle face  $F$  and its bounding box  $B$ . For each pixel  $p$ , let  $N_p$  be the set of its 4-connected neighbors in  $B$ . For pixels interior to the bounding box,  $|N_p| = 4$ , Eq. (5) reads:

$$4u_p - \sum_{q \in N_p} u_q = 4f_p - \sum_{q \in N_p} f_q \quad (6)$$

For pixels on the border of the bounding box,  $|N_p| < 4$ , Eq. (3) reads:

$$|N_p| u_p - \sum_{q \in N_p} u_q = |N_p| f_p - \sum_{q \in N_p} f_q \quad (7)$$

For corners  $C$ , Eq. (3) reads:

$$u_C = g_C \quad (8)$$

Fig. 7 gives the comparison of results from this method and simple interpolation method. In the interpolation method,

we get the corrected color  $u_p$  of pixel  $p$  using its barycentric coordinates  $(\lambda_0, \lambda_1, \lambda_2)$  in the triangular face  $F$ ,

$$u_p = f_p + \lambda_0(g_0 - f_0) + \lambda_1(g_1 - f_1) + \lambda_2(g_2 - f_2), \quad (9)$$

$$\lambda_0, \lambda_1, \lambda_2 \geq 0, \lambda_0 + \lambda_1 + \lambda_2 = 1.$$

Note that some unnatural spots occur in the texture after interpolation using barycentric coordinates (Fig. 7(a)).

#### 4. Experiments and results

We evaluated the color adjustment method on a number of real-world data sets. In particular, we employ VisualSFM [15] to estimate camera poses and PMVS/CMVS [16] to recover a point cloud. The triangle mesh is reconstructed using SSD surface reconstruction [17]. Eqs. (4) and (6) are solved with CHOLMOD [18].

**Table 1**  
Summary of the considered data sets.

Data set	#Views	Image resolution	#Vertices	#Faces	Runtime (s)				Global adjustment	Poisson editing
					Visibility detection	View selection	Texture coordinates optimization			
Bust	25	480 × 640	27,883	55,346	2.28	0.19	4.78	8.49	1.15	
Goddess statue	45	1936 × 1296	255,665	510,408	42.95	1.77	21.47	89.62	3.41	
Der hass	79	2592 × 1728	195,252	389,528	45.15	1.57	24.61	171.59	7.58	
Stone relief	221	5184 × 3456	647,904	1,293,172	180.98	7.32	5.38	1282.70	66.22	

To compare the effect of our color adjustment method, we first use the output of visibility detection and view selection process from MeshLab [14]. Figs. 8 and 9 show the results of color adjustment with Meshlab [14] and our method. In [14], the texture to a seam's left and right side are averaged in a narrow band along the seam. While blending colors eliminates discontinuities in the vicinity of patch boundaries, significant artifacts are still exhibited far from the border of each patch in the results of Meshlab [14]. By combination of global color adjustment at each corner and local color adjustment with Poisson editing, our method yields the results that make seams unnoticeable globally.

We also compare our method with the latest state of the art color adjustment in image-based texture maps of [11]. The comparison is performed on the same images, camera parameters and geometric model of Der Hass dataset [11]. As shown in Fig. 10, our method is able to deal with most artifacts generated in [11].

Figs. 11 and 12 show results on publicly available bust dataset [21] and Der Hass dataset [11]. Even though there are large illumination and exposure changes in the input images, our method achieves high quality texture maps.

Table 1 summarizes the properties of the data sets used in our evaluation and runtime (in seconds) of the individual parts of the algorithm. The time reports are obtained on a machine with two 8-core Intel Xeon E5-2560 CPUs, 2.00GHz and 64GB of memory. So far no GPU based acceleration technique is exploited although our method is suitable to be extended to take advantage of GPU acceleration.

## 5. Conclusion

We have proposed a color adjustment approach to image based texture maps. The method starts from texture coordinates optimization, followed by global adjustment at each corner and local adjustment with Poisson editing. Experiments on several real-world models demonstrate that the resulting texture maps have less visible seams and preserve more details. The approach can be applied to post-process a preliminary texture where each face is textured by exact one image view. Our method is effective enough to process real-world datasets with hundreds of input images and millions of triangles.

However, our method cannot fully avoid out-of-focus image areas in the process of viewing selection. We have tried several weighting metrics and have not found the best one. Automatically detecting out-of-focus blur in the process of viewing selection is our future work.

## Acknowledgments

This work was supported by State Scholarship Fund of China, Program of Science and Technology Development of Shandong Province (2014GGX101016), the Fundamental Research Funds of Shandong University (2014JC003).

## References

- [1] Y. Tzur, A. Tal, FlexiStickers: photogrammetric texture mapping using casual images, *ACM Trans. Graph.* 28 (3) (July 2009) 10 (Article 45).

- [2] M. Callieri, P. Cignoni, M. Corsini, R. Scopigno, Masked photo blending: mapping dense photographic data set on high-resolution sampled 3D models, *Comput. Graph.* 32 (4) (August 2008) 464–473 ISSN 0097-8493.
- [3] A. Baumberg, Blending images for texturing 3D Models, in: *Proceedings of the British Machine Conference on Vision Association*, 2002, pp. 404–413.
- [4] Z. Chen, J. Zhou, Y. Chen, G. Wang, 3D texture mapping in multi-view reconstruction, *Adv. Vis. Comput. Lect. Notes Comput. Sci.* 7431 (2012) 359–371.
- [5] F. Bernardini, I.M. Martin, H. Rushmeier, High-quality texture reconstruction from multiple scans, *IEEE Trans. Vis. Comput. Graph.* 7 (2001) 318–332 (October (4)).
- [6] B. Goldlücke, M. Aubry, K. Kolev, D. Cremers, A super-resolution framework for high-accuracy multiview reconstruction, *Int. J. Comput. Vis.* 106 (2) (2014) 172–191.
- [7] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, D. Cohen-Or, Seamless montage for texturing models, *Comput. Graph. Forum* 29 (2) (2010) 479–485.
- [8] L. Velho, J. Sossai, Projective texture atlas construction for 3D photography, *Vis. Comput.* 23 (9–11) (2007) 621–629.
- [9] V. Lempitsky, D. Ivanov, Seamless mosaicing of image-based texture maps, in: *CVPR*, 2007.
- [10] C. Allene, J.-P. Pons, R. Keriven, Seamless image-based texture atlases using multi-band blending, in: *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, December 2008, vols. 1, 4 pp. 8–11.
- [11] M. Waechter, N. Moehrl, M. Goesele, Let there be color! Large-scale texturing of 3D reconstructions, *computer vision – ECCV 2014*, 978-3-319-10601-4, *Lecture Notes in Computer Science*, pp. 836–850.
- [12] P. Perez, M. Gangnet, A. Blake, Poisson image editing, *ACM Trans. Graph.* 22 (3) (2003) 313–318.
- [13] M. Birsak, P. Musialski, M. Arikan, M. Wimmer, Seamless texturing of archaeological data, in: *Proceedings of the Digital Heritage International Congress (DigitalHeritage)*, October 2013, 2013, pp. 265–272.
- [14] P. Cignoni, M. Corsini, and G. Ranzuglia, MeshLab, An OpenSource 3D Mesh Processing System, *ERCIM News*, No. 73, pp. 45–46, <http://meshlab.sourceforge.net>, April 2008.
- [15] C. Wu, VisualSFM. A visual structure from motion system. <http://ccwu.me/vsfm/>.
- [16] Y. Furukawa, J. Ponce, Accurate, dense, and robust multi-view stereopsis, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (8) (2010) 1362–1376.
- [17] F. Calakli and G. Taubin, SSD: smooth signed distance surface reconstruction, *Comput. Graph. Forum*, 30 (7), 1993–2002.
- [18] Y. Chen, T.A. Davis, W.W. Hager, S. Rajamanickam, Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/download, *ACM Trans. Math. Softw.* 35 (3) (2008) 22:1–22:14.
- [19] K. Pal, M. Terras, T. Weyrich, 3D reconstruction for damaged documents: imaging of the great parchment book, in: *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing (HIP '13)*, ACM, New York, NY, USA, 2013, pp. 14–21.
- [20] B. Gough, GNU Scientific Library Reference Manual, third ed., Network Theory Ltd, 2009.
- [21] M. Lhuillier, L. Quan, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (3) (2005) 418–433.