# A Multi-Resolution Approach for Color Correction of Textured Meshes


Mohammad Rouhani
mohammad.rouhani@technicolor.com

Matthieu Fradet
matthieu.fradet@technicolor.com

Caroline Baillard
caroline.baillard@technicolor.com

Technicolor
Rennes, France



## Abstract

*Mesh texturing is an essential part of 3D scene reconstruction that enables a more realistic perception than the geometry alone and even compensates for inaccurate geometry. In this work we present a flexible formulation for color correction of textured scenes based on color augmentation per face. It can be employed as a post-processing step after selecting the best keyframe per face to compensate for color differences between pairs of neighboring faces. We present a Markov Random Field (MRF) formulation to find the best keyframes as well as the optimal color augmentations. We use a simple model to avoid reflection and camera vignetting during the view selection. Our model for color correction finds the piecewise-linear augmentation to be added to the texture patches of faces. It encourages smoothness inside every fragment while compensating color differences along view transitions. Moreover, we speed up the optimization by breaking down the formulation into multiple binary MRFs that estimate the best augmentations from coarse to fine resolutions. The experimental results prove our method outperforming the state of the art methods.*

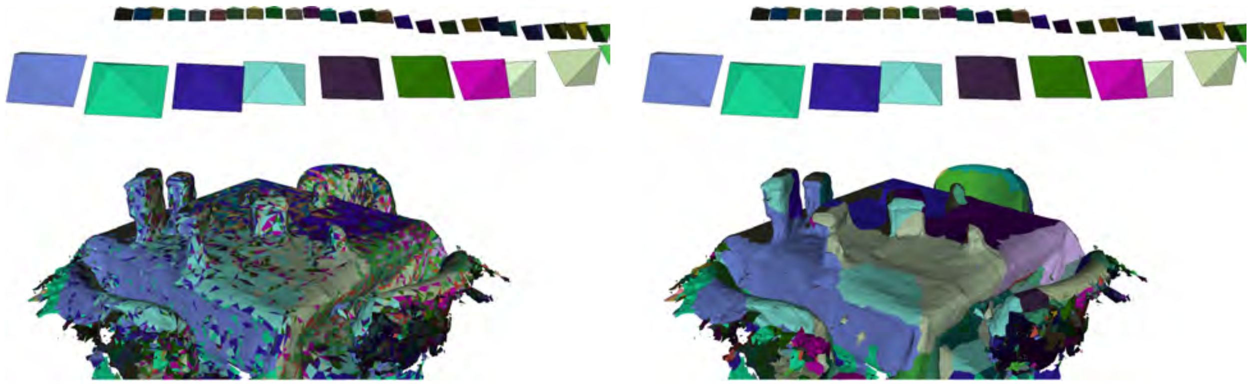

## 1. Introduction

Texture mapping is a classic problem in 3D computer vision and it usually comes as the last stage of scene reconstruction workflow [2]. The acquired textured scenes can be used for mixed reality applications such as real estates and interior design, to mention a few [21]. Lately, with the rise of commodity RGB-D sensors together with precise visual tracking techniques, both geometry and texture can be easily captured for static and even dynamic scenes [11], [7]. However, the texture information is claimed to be visually more important than the scene geometry, as a high-quality texture can compensate for inaccurate geometry [12].

Most texture reconstruction techniques are based on an optimal view selection. Usually, during acquisition, a set of views with corresponding poses and RGB keyframes are captured, as well as a 3D mesh containing the geometric information. A part of the texturing problem is then to select, for each face of the mesh, the view providing the ”best” texture information for that face. A naive approach may select the closest viewpoint for texturing a given triangular face of the mesh, but the smoothness of the final texture map can not be guaranteed due to changes in illumination and camera position and parameters (see Fig. $1(a)$). Blending several views may smooth the texture map at the price of ghosting and blurring effects. Alternatively, Markov Random Field (MRF) provides an optimization framework for selecting the closest view while preserving the texture smoothness. Figure $1(b)$ provides an example of an optimal view selection together with the associated camera poses discriminated by different colors. In this view selection, the scene is clustered into several fragments as neighboring faces are encouraged to be textured by the same camera.

The optimal view selection leads to several components, referred to as *fragments*, that are textured from the same keyframe. The texture map is smooth and without any artifact inside every fragment but color levels may change drastically along view transitions due to illumination changes (see Fig. $2(a)$). In this paper we present a novel formulation to reduce seam effects in the texture map, both within and over fragments. More precisely, we find the optimal color augmentation per face which will best smooth the texture map, as illustrated in Fig. $2(b)$. This color augmentation is smooth within every fragment, and it slightly changes towards view transitions in order to compensate color differences between fragments.

Graph cut provides a powerful tool for discrete optimization and it has been widely used in computer vision [4]. We employ this tool for two subsequent stages, namely view selection and color augmentation. For view selection we avoid complex models by adopting the Potts model for the



(a) (b)

Figure 1. Optimal view selection : $(a)$ a greedy algorithm selects the closest view point per face; $(b)$ graph-cut encourages neighboring faces to be captured with the same camera.

regularization. We use Graph cut for finding the optimal leveling function as well, which shows the color augmentation per face. In addition, we propose a coarse to fine approach for breaking down this multi-label problem to several binary MRFs.

Our hierarchical approach for finding the optimal leveling function is quite fast and more robust to noise and outliers, compared to other models such as least squares. The rest of the paper is organized as follows: in Section 2 we present the related work on texture reconstruction; our proposed approach for view selection and color augmentation is explained in Section 3 and compared with state of the art methods in Section 4. Finally, the main points of the paper are summarized in Section 5.

## 2. Related Work

Given the reconstructed geometry, the camera trajectory, as well as a set of RGB keyframes captured along this trajectory, a static 3D scene can be thoroughly textured. A naive approach selects the *closest camera* to texture a given triangular face. This greedy technique does not guarantee a seamless texture map at the end. Several views can be blended by weighted averaging along the seams, but this local approach may cause blurring or ghosting effects [2].

(a) (b)

Figure 2. $(a)$ The optimal view selection in Fig. $1(b)$ results in seams between fragments; $(b)$ our method removes these seams.

Energy minimization, on the contrary, provides a powerful approach for texture reconstruction. Goldluecke et al. find the optimal texture using an elaborated convex optimization method that is solved using PDEs [10]. A Bayesian generative model has been used in [19] to reconstruct the texture in dynamic scenes.

Markov Random Field (MRF) has been widely employed for optimal view selection. The data term in MRF measures the quality of different views per face while the regularization term guarantees smoothness over the final texture. In [13] a score combining the distance from the camera to the considered face and the angle between the camera vector and the face normal is computed for each view in which the face is visible. It can be equivalently measured through the area of projection of the 3D face into the keyframe [1] as a larger projection implies that the camera is closer and more fronto parallel to the face. In [9] the data term integrates the total color variation mapped onto the triangle in order to encourage a sharper image with better resolution.

The regularization term in MRF encourages smoothness in the texture or the label domain. A simple Potts model encourages neighboring faces to be textured by the same keyframe [20]. It leads to several fragments that are smoothly textured inside fragments but might not accord along borders (see Fig. $2(a)$). A more complex regularization term in [1] and [8] measures the color consistency between neighboring faces. Precisely speaking, the common edge is projected into the given views and color differences are integrated along the edge.

The final texture map might be corrupted due to inaccuracies in both geometry and camera poses; the misregistrations leads to texture discontinuities along view transitions. *Gal et al.* in [9] allow translating the texture patch of every face by extending the *view* labels to *view-translation* labels that considers 9 possible translations. *Zhou et al.* [22] propose an alternating optimization to find the optimal color and the camera poses simultaneously. Then, the optimal non-rigid deformation is applied on every keyframe to align the geometric and photometric information.

Optical flow between overlapping images is used in [6] to warp input images and correct the local misalignments. Then, blending colors results in a sharper texture map with reduced ghosting effects. A patch-based method is proposed in [3] to compensate for large misalignments of the keyframes. They propose a two-step approach that involves search and vote for alignment and texture reconstruction. *Maier et al.* present a coarse-to-fine approach in [14] to reconstruct a high-quality geometry and texture. However, a high-quality texture map is usually claimed to be more important than a detailed geometry. The authors in [12] present a framework to obtain a low-polygonal, lightweight textured mesh based on the primitive abstraction of the scene.

In addition to errors in geometry and camera poses, the final texture map may suffer from illumination changes [21]. Gain and bias compensation can reduce the global illumination differences between the images to be stitched [5]. Multi-band blending is employed in [1] for further color correction. The authors represent the keyframes through a Laplacian pyramid and design a proper weighting function to average different levels for computing the texture elements. Poisson image editing provides a smooth color changes for stitching images in the gradient domain [16]. This tool has been used in [20] for blending keyframes in the final texture map.

Our work follows an approach similar to [13] where a leveling function is added to the texture map to smooth color levels over fragments. This function is piecewise-smooth, and it compensates for color differences along view transitions. *Lempitsky and Ivanov* in [13] find the optimal leveling function by solving a least squares problem. In our approach, we employ a MRF model, instead, as it provides a more flexible framework for energy minimization. Our MRF model smooths color levels along view transitions: it compensates color differences along borders while guaranteeing smoothness inside each region. Moreover, we employ more robust function (such as $l_1$ norm) to measure the energy of MRF, which makes our method more robust than $l_2$ norm used in [13].

## 3. Proposed Approach

In this section we present our framework for texturing 3D meshes. We employ MRF for both phases, namely, view selection and color augmentation. In the first phase we find the optimal view per triangular face among the set of existing keyframes. In the second phase the best color augmentation is found among the permitted values. Moreover, we propose a multi-resolution approach to break down the problem to several binary MRFs.

### 3.1. View Selection

The reconstructed manifold can be encoded as a graph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ with nodes representing the triangular faces and edges representing the pairs of neighboring faces. Given a set of $K$ keyframes, the view selection problem consists in finding the set of keyframe indices per face $\mathbf{l} = \{l_f\}_{f\in\mathcal{F}}$ so that the following total energy is minimized:

$$E(\mathbf{l}) = \sum_{f\in\mathcal{F}} \psi_f(l_f) + \gamma \sum_{(f,g)\in\mathcal{E}} \varphi_{fg}(l_f, l_g) \qquad (1)$$

where the data term $\psi_f$ measures the quality of the keyframe for texturing the face $f \in \mathcal{F}$ and the regularization term $\varphi_{fg}$ controls the interactions between neighboring faces $(f, g) \in \mathcal{E}$ by measuring the label or color consistencies.

In our implementation we use the area of projection as the data term to encourage closer and more fronto-parallel keyframes to be picked up for texturing the face:

$$\psi_f(l_f) = -area(\Pi_{l_f}(f)) \qquad (2)$$

where $\Pi_{l_f}(f)$ is the projection of the triangular face $f$ into the $l_f$-th keyframe. Vignetting effects can be reduced by penalizing those keyframes where face $f$ is projected away from the image center. In order to avoid the use of reflecting surfaces for texturing a face, we compute the average color of the face in different views. Then, those views with colors highly different from the rest are strongly penalized so that they are not picked up during the optimization. Figure 3 illustrates how this data term discourages reflections.

For the regularization term we measure color compatibility between every two neighboring faces. More precisely, we consider the common edge between two faces $f$ and $g$ and project it into views $l_f$ and $l_g$ to measure the average color differences [20]. Alternatively, we can simply use the Potts model $\varphi_{fg}(l_f, l_g) = 1_{\{l_f \neq l_g\}}$ that vanishes only if both faces obtain the same label. Our experiments show that the final result does not change for these two different smoothness terms after optimizing the color augmentations in the second phase.

### 3.2. Color Augmentation

The main contribution of this paper is to use the same frameworks for both view selection and color augmentation. The previous phase consisted in finding the best view per face, but eventually, color levels may change between fragments textured by different keyframes. Pre-processing the keyframes using the gain compensation over the shared region may partially improve the consistency between fragments (see Figure 4). In the second phase, we aim at finding the optimal color values $\mathbf{c}$ to be added to the texture patches

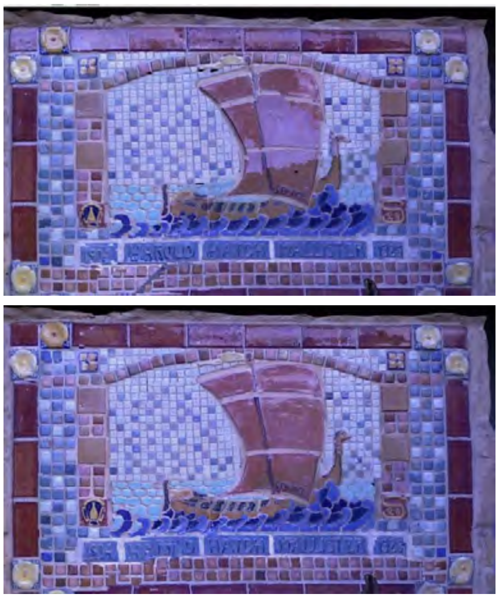


Figure 3. The reflecting surfaces are discouraged to be used in texturing by tweaking the data term.

Figure 4. Pre-processing keyframes: $(left)$ without gain compensation; $(right)$ with gain compensation.

of faces within a set of permitted values $\mathcal{C} \subset \mathbb{R}^3$. Indeed, for every face $f \in \mathcal{F}$ we must find an optimal color constant $c_f \in \mathcal{C}$ so that the following energy function is minimized:

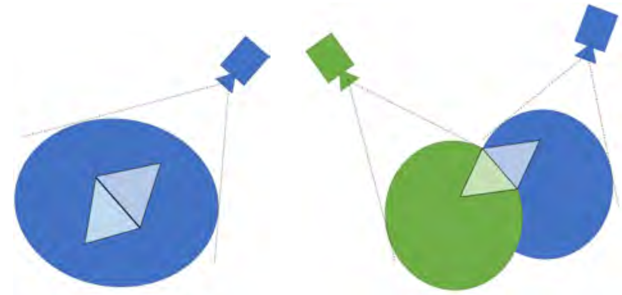
$$E(\mathbf{c}) = \sum_{f \in \mathcal{F}} \psi_f(c_f) + \gamma \sum_{(f,g) \in \mathcal{E}} \varphi_{fg}(c_f, c_g). \qquad (3)$$

This model can be either separated for different color channels or it can consider all channels simultaneously. Using all channels simultaneously does not appear to improve the final result, though the optimization in the second case is more expensive as we have a bigger solution space.

In the data term in (3) we focus on those faces along view transitions and encourage them to approach the average color value with neighboring face seen from another view. In fact, for those faces with a disagreeing neighbor $g$ (i.e, $l_f \neq l_g$) we penalize the deviation from the expected color augmentation:

$$\psi_f(c_f) = \rho(c_f - \Delta_{gf}/2), \qquad (4)$$

where $\rho$ is a robust function (such as $\rho(.) = |.|$)and the expected augmentation is computed by finding $\bar{c}_{gf}$ and $\bar{c}_{fg}$,



(a) (b)

Figure 5. Two different cases for computing the regularization term.

the average color of the common edge $e_{fg}$ projected into $l_g$ and $l_f$ keyframes, respectively:

$$\Delta_{gf} = \bar{c}_{gf} - \bar{c}_{fg}. \qquad (5)$$

The average colors are computed by uniformly sampling and integrating the values along the projected line, using a bi-linear interpolation. For faces with more than one disagreeing neighbor, we consider a mean value of $\Delta_{gf}$ in (4).

For the regularization term between neighboring faces $f$ and $g$ we consider two different cases, where the faces are seen from the same camera or two different cameras (see Fig. 5):

- *Case 1*: if $f$ and $g$ are in the same fragment, i.e. coming from the same keyframe ($l_f = l_g$), then we encourage smoothness over the region:

$$\varphi_{fg}(c_f, c_g) = \rho(c_f - c_g). \qquad (6)$$

- *Case 2*: if $f$ and $g$ come from different fragments ($l_f \neq l_g$), then the difference between the average colors should be compensated:

$$\varphi_{fg}(c_f, c_g) = \gamma\rho(c_f - c_g - \Delta_{gf}), \qquad (7)$$

  where the parameter $\gamma$ represents the importance of smoothing along view transitions between fragments.

Having defined both data and smoothness terms, the energy function of color augmentation in (3) can be optimized using the swap-move algorithm [17].

Having optimized the color augmentation per face, we reach a piece-wise constant leveling function that can be directly added to the texture map. Instead, we calculate the color augmentation per vertex, which is the average augmentation of neighboring faces, unless the vertex is along view transitions. Then, we linearly interpolate the color augmentation of the vertices of a given face to reach a piece-wise linear leveling function over that face. Finally, the black artifacts along unwrapping seams can be removed before exporting the texture map [18].

Figure 6. Different color representations ($top$) color per vertex; ($bottom$) texture per face; in the latter case, the mesh is 25 times lighter while the quality of texture is independently higher.

### 3.3. Multi-Resolution Framework

In our implementation we find the color augmentation per channel per face within the set $\mathcal{C} = \{-C, ..., \ 0, ..., +C\}$ with a proper step, depending on the resolution. We propose a hierarchical approach to break this multi-label formulation down to several binary MRF problems. Using a binary representation, any integer number in $[-2^n, 2^n)$ can be decomposed as:

$$c_f = (\alpha_f^{(0)} + 2\alpha_f^{(1)} + \ldots + 2^n\alpha_f^{(n)}) - 2^n \tag{8}$$

where $\alpha_f^{(i)} \in \{0, 1\}$. Therefore, the complexity of the original problem can be tremendously reduced by boiling it down to $n$ binary MRFs.

We start from the coarsest level by finding the best binary value $\alpha_f^{(n)}$ for every face $f$, and then, we switch to finer levels $\alpha_f^{(n-1)}, ..., \alpha_f^{(0)}$. Then, in every step we find the optimal binary values per face that show whether we increment the $c_f$ in this resolution:

$$\psi_f\left(\alpha_f^{(k)}\right) = \rho\left(2^k\alpha_f^{(k)} + c_f^{ac} - \Delta_{gf}/2\right) \tag{9}$$

where $c_f^{ac}$ is the accumulated values from the previous steps. Similarly, the regularizations terms for two cases in (6)-(7) should be updated by changing their variables from $\mathcal{C}$ to binary parameters. Note that once the optimal binary values are found, the accumulated colors should be updated.

## 4. Experimental Results

In this section the results of our mesh texturing and color correction framework are presented. In addition to public data sets provided by [22] several scenes have been captured using commodity sensors such as *Intel R200* and *Structure I/O*. Scene geometries are generated by Kinect Fusion [15] as well as Skanect software [1]. For texture reconstruction, the scene must be accompanied by keyframes and their associated camera poses.

In our approach, we firstly unwrap a given mesh to flatten the $uv$- coordinates of the vertices so that the mesh will be associated to a texture map where the color information can be found. This approach is in contrast to color-per-face or color-per-vertex representation as used in [22]. Figure 6 compares these color representations. As illustrated in the bottom, the mesh can be represented with less number of faces (25 times less) while the quality of texture is better and independent of the geometry.

For the optimal view selection we use the simple Potts model in (1). Measuring color compatibility between neighboring faces, as used in [13] and [20], may leads to slightly better results, but then, the MRF model for view selection would be slower. In the end, these two approaches for view selection end up with similar results when the color correction technique is applied. Figure 7($top$) illustrates the results obtain by MRF model in [8] and the simple Potts model. As depicted in the bottom, similar results are obtained after applying our color correction method.

Figure 9 illustrates the results obtained using the MRF leveling technique proposed in this paper. In the left column, the labels of the selected views are demonstrated to show the texture fragments and borders between them. Having the view index per face, the texture can be directly copied from the associated keyframe. As demonstrated in the middle column, color illuminations may change drastically along borders where view transitions happen. The right column demonstrate how the seams are efficiently removed. Figure 8 demonstrates the multi-resolution approach to find the color augmentation for a mesh containing around $50k$ faces. We used 7 levels, from coarse to fine, to cover the color domain $[-64, 64)^3$. The MRF for the multi-resolution approach is optimized in 6 seconds, while the direct MRF takes around 150 seconds.

The proposed MRF model is very flexible as different metric and energy terms can be used for measuring the data and smoothness terms. Compared to least squares error widely used in the literature (e.g., [13] and [12]) we rather use the $l_1$-norm for energy terms so that outliers have less impact on the color transfer result. Figure 10 illustrates the result obtained using the $l_2$-norm color correction proposed by [13]. The difference can be noticed in the regions where the average color of the common edge drastically changes due to the error of camera pose. This color difference will be propagated in neighboring triangles in order to smooth the texture, whereas in our result the error measurement is more robust to outliers.

[1] `http://skanect.occipital.com/`

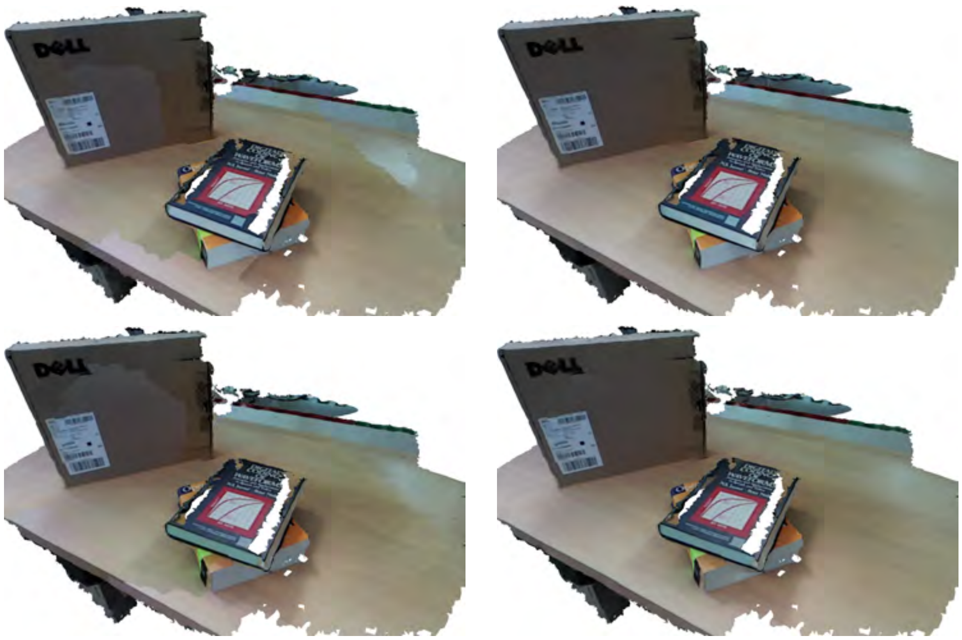
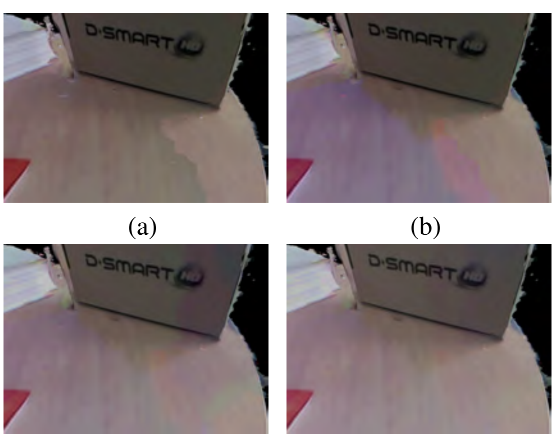


Figure 7. The impact of view selection: $(top)$ the Potts model for view selection; $(bottom)$ color compatible model [8]; $(left)$ before color correction; $(right)$ after the proposed color correction.

## 5. Conclusions

In this work we use MRF modeling to find the optimal view selection and the optimal leveling function. In the first phase we select the best keyframe per face by maintaining smoothness between neighbors, while avoiding reflection and camera vignetting. As a novel contribution we present a coarse to fine approach to find the best color augmentation by solving several binary MRFs that can be quickly optimized. The optimal color augmentation is smooth within fragments, whilst compensating for color differences between them. Moreover, it is quite robust to noise and outliers that might occur due to illumination changes and errors in the camera poses. The experimental results prove our method outperforming the state of the art methods.



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

Figure 8. Multi-resolution MRF augmentation: $(a)$ view selection before color correction; $(b)$-$(d)$ the colors per face are augmented from coarse to fine levels.

## References


[1] C. Allène, J.-P. Pons, and R. Keriven. Seamless image-based texture atlases using multi-band blending. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

[2] F. Bernardini, I. M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, 2001.

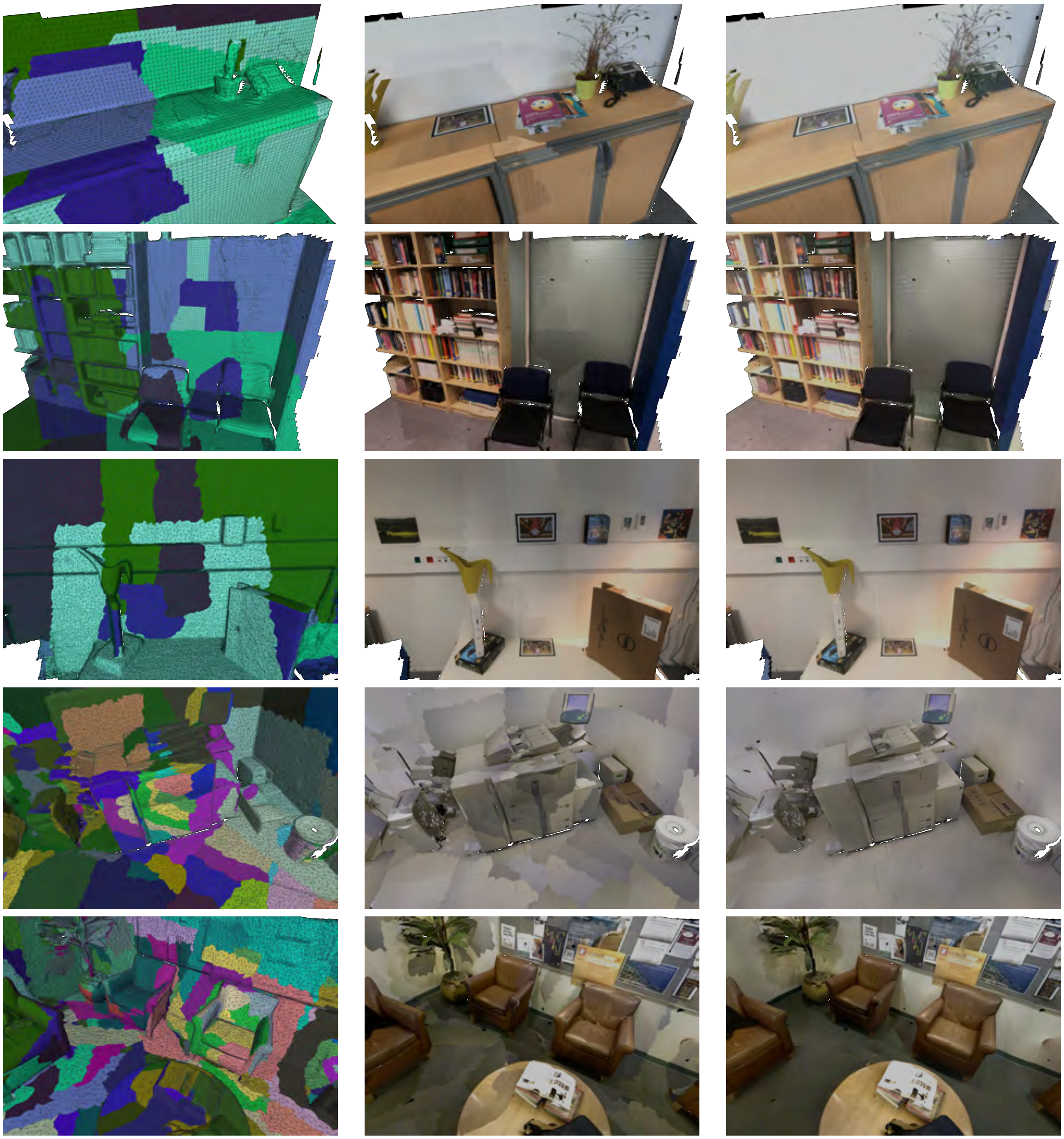
[3] S. Bi, N. K. Kalantari, and R. Ramamoorthi. Patch-based optimization for image-based texture mapping. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 36(4), 2017.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239, 2001.

[5] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007.

[6] M. Dellepiane, R. Marroquim, M. Callieri, P. Cignoni, and R. Scopigno. Flow-based local optimization for image-to-geometry projection. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):463–474, 2012.

[7] R. Du, M. Chuang, W. Chang, H. Hoppe, and A. Varshney. Montage4d: Interactive seamless fusion of multiview video



(a) (b)

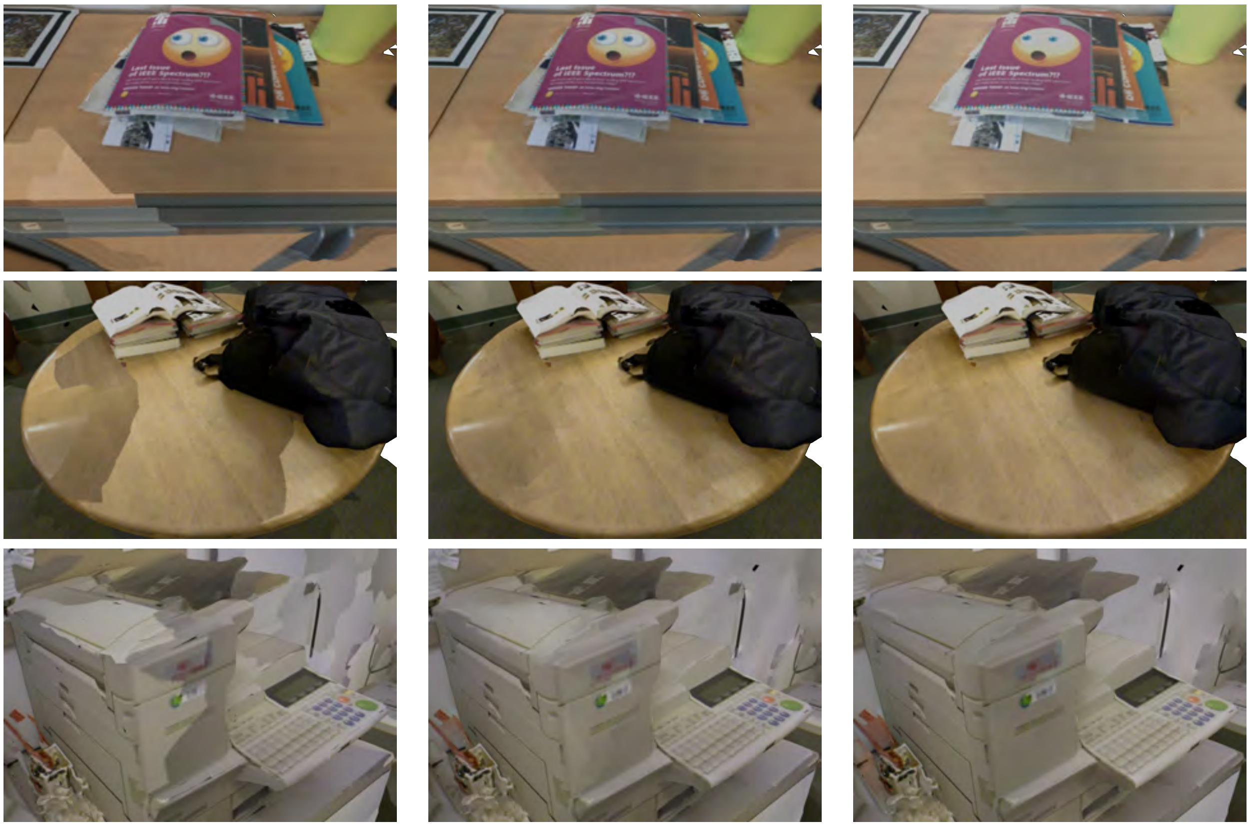
Figure 9. Optimal color augmentation : $(a)$ fragment labels after the view selection; $(b)$ textured mesh without color correction; $(c)$ textured mesh after the proposed color correction.


textures. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, pages 124–133. ACM, May 2018.

[8] Y. Fu, Q. Yan, L. Yang, J. Liao, and C. Xiao. Texture mapping for 3d reconstruction with rgb-d sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[9] R. Gal, Y. Wexler, E. Ofek, H. Hoppe, and D. Cohen-Or. Seamless montage for texturing models. In *Computer Graphics Forum*, volume 29, pages 479–486. Wiley Online Library, 2010.

[10] B. Goldlücke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. *International journal of computer vision*,



(a) (b) (c)

Figure 10. The impact of the metric: $(a)$ view selection using Potts model [20]; $(b)$ color correction by least squares ($l_2$-norm) [13]; $(c)$ color correction by our MRF model (using $l_1$-norm).


106(2):172–191, 2014.

[11] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu. Real-time geometry, albedo, and motion reconstruction using a single rgb-d camera. *ACM Transactions on Graphics (TOG)*, 36(3):32, 2017.

[12] J. Huang, A. Dai, L. Guibas, and M. Nießner. 3DLite: towards commodity 3d scanning for content creation. *ACM Transactions on Graphics*, 2017, 2017.

[13] V. Lempitsky and D. Ivanov. Seamless mosaicing of image-based texture maps. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[14] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3d: High-quality 3d reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 4, 2017.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[16] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003.

[17] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080, 2008.

[18] M. Tarini, C. Yuksel, and S. Lefebvre. Rethinking texture mapping. In *ACM SIGGRAPH 2017 Courses*, page 11. ACM, 2017.

[19] V. Tsiminaki, J.-S. Franco, and E. Boyer. High resolution 3d shape texture from multiple videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1502–1509, 2014.

[20] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! large-scale texturing of 3d reconstructions. In *European Conference on Computer Vision*, pages 836–850. Springer, 2014.

[21] E. Zhang, M. F. Cohen, and B. Curless. Emptying, refurnishing, and relighting indoor spaces. *ACM Transactions on Graphics (TOG)*, 35(6):174, 2016.

[22] Q.-Y. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG)*, 33(4):155, 2014.