

A UNIFIED BLENDING FRAMEWORK FOR PANORAMA COMPLETION VIA GRAPH CUTS

Kai Chen^a, Jian Yao^{a,†}, Menghan Xia^a, Xinyuan Gui^b, Li Li^a, Xiaohu Lu^a

^a School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R. China

^b School of Geodesy and Geomatics, Wuhan University, Wuhan, Hubei, P.R. China

†Email: jian.yao@whu.edu.cn Web: <http://cvrs.whu.edu.cn/>

KEY WORDS: Image Completion, Panorama, Graph Cuts, Luminance Compensation, Image Blending, Feathering

ABSTRACT:

In this paper, we propose a unified framework for efficiently completing streetview and indoor 360° panoramas due to the lack of bottom areas caused by the occlusion of the acquisition platform. To greatly reduce the severe distortion at the bottom of the panorama, we first reproject it onto the ground perspective plane containing the whole occluded region to be completed. Then, we formulate the image completion problem in an improved graph cuts optimization framework based on the statistics of similar patches by strengthening the boundary constraints. To further eliminate image luminance differences and color deviations and conceal geometrical parallax among the optimally selected patches for completion, we creatively apply a multi-blend image blending algorithm for perfect image mosaicking from the completed patches and the originally reprojected image. Finally, we back-project the completed and blended ground perspective image into the cylindrical-projection panorama followed by a simple feathering to further reduce artifacts in the panorama. Experimental results on some representative non-panoramic images and streetview and indoor panoramas demonstrate the efficiency and robustness of the proposed method even in some challenging cases.

1. INTRODUCTION

A panorama is an image with a wide angle of view, which has been widely used for street view photos and indoor virtual tour recently. A panorama with a view angle of 360° is common in specific applications. To obtain such 360° high-resolution panoramas, one common way is to capture multiple images synchronously from an integrated platform with multiple cameras whose viewpoints cover the whole 360° except for the ground region occluded by the platform itself and then mosaic these images into a complete 360° panorama. This way has been widely used in industry, for example, Google Street View, and Chinese Baidu and Tencent Street View, by mounting such an integrated platform on a mobile vehicle or a backpack. Another simple and popular way is to capture multiple images at different times with a single camera mounted on a static platform (e.g., a tripod) by rotation and then generate a panorama from these images. This way is widely applied for panorama acquisition in indoor scenes or relatively small spaces. All of these acquisition ways cannot perfectly cover the complete 360° viewpoint due to the ground occlusion caused by the platforms themselves. To quickly obtain a perfect and complete 360° panorama, the image completion and blending techniques can be applied to complete the ground occluded region. In addition, for some privacy protection, we can also apply image completion to conceal the sensitive image regions.

Image completion aims to generate visually plausible completion results for missing or content-concealed image regions. In general, most of image completion algorithms are divided into two categories: diffusion-based and exemplar-based. Earlier works (Ballester et al., 2001; Bertalmio et al., 2000, 2003; Levin et al., 2003; Roth and Black, 2005) typically used the diffusion-based methods, which find the suitable colors in the regions to be completed through solving partial differential equations. These methods usually make use of the consistency of an image and are only suitable for filling narrow or small holes in the image. The ground bottom region to be completed in a 360° panorama is usually

too huge to be effectively completed with these diffusion-based methods.

Another category of image completion methods are exemplar-based. They can obtain satisfactory results even for huge holes in some cases. The basic idea of these methods is to first match the patches in the unknown region with the ones in the known one to find the potential suitable patches and then copy or synthesize the matched ones in the known region under some constraints in color, texture, and structure (Criminisi et al., 2003). These methods are always involved in two issues: (i) how to search for the fittest patches for the image region to be completed; (ii) how to synthesize the patches to keep visual coherence. More specifically, these exemplar-based methods can be further divided into two categories. Some methods (Criminisi et al., 2003; Drori et al., 2003; Efros et al., 1999; Jia and Tang, 2003, 2004) first matched the patches in the unknown region with ones in the known one, and copy those matched patches in the known one to complete the unknown one. The approach proposed by Sun et al. (2005) requires the user's interactions to point out the structure in the missing part to guide the completion process. Another category of exemplar-based methods (He and Sun, 2014) formulate the image completion problem in a graph-based optimization framework. Rather than directly matching patches, they rearranged the patch locations under a Markov random field (MRF) energy optimization framework with some completion constraints. Especially, He and Sun (2014) proposed a graph-cuts-based optimization method based on the statistics of similar patches, which is one representative of the state-of-the-art image completion algorithms. Their proposed method first matched similar patches in the image and found that the statistics of these offsets are sparsely distributed. Referring to dominant offsets as labels to be optimized, He and Sun (2014) defined a MRF energy function, which was optimized via graph cuts (Boykov et al., 2001). The final image completion was implemented with the guidance of the optimized label map. In general, these exemplar-based methods are able to be more suitable for filling huge holes in images, in which the image structural information plays an important role.

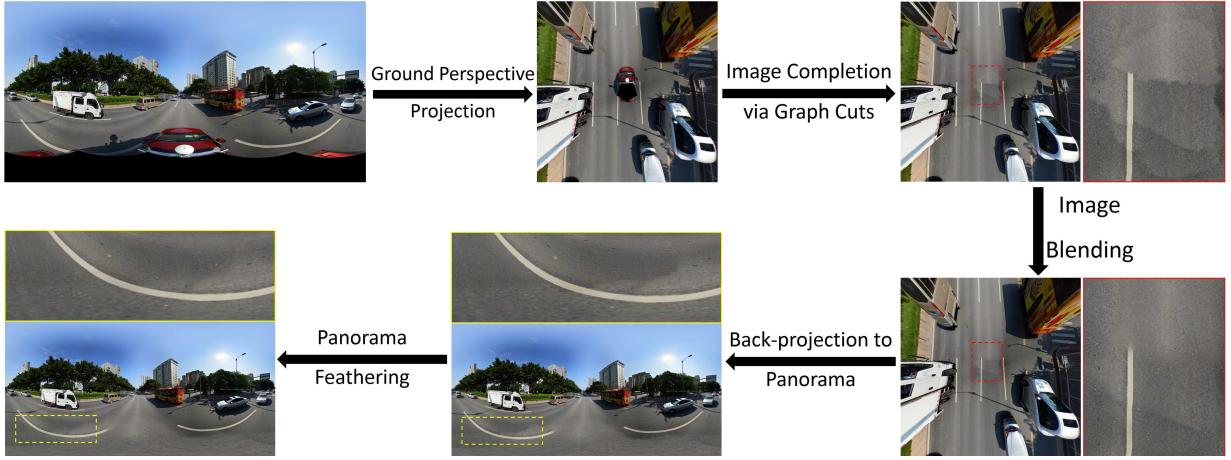


Figure 1: The flowchart of the proposed unified blending framework for panorama completion.

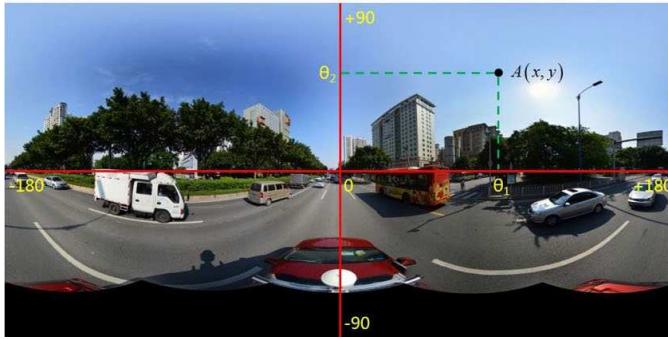


Figure 2: The perspective projection transformation from a cylindrical-projection panorama to a virtual sphere.

Although the state-of-the-art image completion algorithms can generate satisfactory completion results in most cases, the ground bottom area in a 360° panorama to be completed is usually large with severe distortion, which makes existing image completion algorithms very challenging or even impossible for completing it. In this paper, we propose a novel unified blending framework for image completion, especially completing panoramas with a view angle of 360°. The whole framework is comprised of five main steps as shown in Figure 1. Firstly, we reproject the cylindrical-projection 360° panorama onto the ground perspective plane containing the whole occluded region to be completed, which greatly reduces the distortion and makes the exemplar-based image completion methods feasible. Secondly, inspired by the method proposed by He and Sun (2014), we formulate the image completion problem in an improved graph cuts optimization framework based on the statistics of similar patches by strengthening the boundary constraints in the smooth and data energy terms. Thirdly, to further eliminate image luminance differences and color deviations and conceal geometrical parallax among the optimally selected patches for completion, we propose to first apply the global luminance compensation followed by applying a multi-blend image blending algorithm for perfect image mosaicking from the completed patches and the originally reprojected image. Fourthly, we back-project the completed and blended ground bottom image into the cylindrical-projection panorama. Finally, to further reduce artifacts and keep the good resolution in the panorama, we propose to apply an image feathering on the original panorama and the back-projected one.

The remainder of this paper is organized as follows. The whole unified panorama completion framework will be detailed introduced in Section 2. Experimental results on some representative non-panoramic images and streetview and indoor panoramas are

presented in Section 3 followed by the conclusions drawn in Section 4.

2. OUR APPROACH

Our proposed unified blending framework for panorama completion will be introduced in the following five subsections. In Section 2.1, we first introduce a theoretical foundation that we can reproject a 360° panorama onto the ground perspective plane containing the whole occluded region. Inspired by the method proposed by He and Sun (2014), the proposed image completion method under an improved graph cuts optimization framework will be presented in Section 2.2. To generate perfect completion results, we first perform a global luminance compensation operation on all the optimally selected patches for completing and the original background image and then apply a multi-blend image blending on all luminance corrected patches and the originally reprojected image, which can greatly eliminate image luminance differences and color deviations, and conceal geometrical parallax among the optimally selected patches. These two operations are introduced in Section 2.3. The back-projection is the reverse process of panorama perspective projection and they share the same theoretical foundation. The completed and blended image is back-projected to the cylindrical-projection panorama, which is introduced in Section 2.4. Finally, to further reduce artifacts and keep the good resolution in the panorama, we propose to apply an image feathering on the original panorama and the back-projected one, which is introduced in Section 2.5.

2.1 Panorama Perspective Projection

To reproject a 360° panorama onto the ground perspective plane, the main work is to find a transformation from a pixel on the

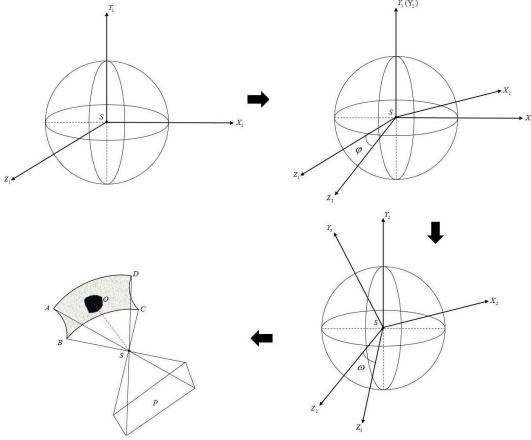


Figure 3: An illustration of two rotation processes to get a perspective projected image from a cylindrical-projection panorama. Along the arrow: original sphere coordinate system $S - X_1 Y_1 Z_1$, rotating φ about the Y_1 axis to get $S - X_2 Y_2 Z_2$, rotating ω about the X_2 axis to get $S - X_3 Y_3 Z_3$, and final projection process.

cylindrical-projection panorama to the corresponding pixel on the perspective projection image. We introduce a virtual sphere coordinate system as a transition of this problem. As shown in Figure 2, a pixel A on a 360° panorama has a simple correspondence with the pixel A' . These two corresponding points A and A' are linked by two longitude and latitude angles (θ_1, θ_2) . The point A' also has three-dimensional coordinates (X_S, Y_S, Z_S) in this virtual sphere coordinate system. From the virtual sphere to the re-projected perspective image, it needs getting through two rotation processes with two rotation angles (φ, ω) and one perspective transformation as shown in Figure 3. The specific rotation angles (φ, ω) denote the sight direction looking at the occluded bottom region of a panorama. If we denote the three-dimensional perspective coordinates (X_P, Y_P, Z_P) , according to the principles of rotation transformation, the perspective transformation can be formulated as:

$$\begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} = \mathbf{R}_\varphi \mathbf{R}_\omega \begin{pmatrix} X_S \\ Y_S \\ Z_S \end{pmatrix}, \quad (1)$$

where (X_P, Y_P, Z_P) denotes the perspective projected point from the point (X_S, Y_S, Z_S) in the panorama, and \mathbf{R}_φ and \mathbf{R}_ω stand for two rotation matrices with the rotation angles φ and ω , respectively. When reprojecting the panorama onto the ground perspective plane, apart from the sight direction, the field of view and the focal length should also be specified properly to obtain a projected image in a suitable size to contain the whole occluded bottom region.

2.2 Image Completion via Graph Cuts

Recently, He and Sun (2014) proposed a graph-cuts-based image completion algorithm, which is one representative of the state-of-the-art image completion algorithms and achieves great performances in filling huge holes in images. Similarly, we formulate the image completion problem in such a graph cuts optimization framework based on the statistics of similar patches but with two improvements strengthening the boundary constraints.

He and Sun (2014) first matched patches in the known region of the image using a Propagation-Assisted KD-Trees (He and Sun, 2012). Then, they calculated dominant offsets based on these matched patches. These dominant offsets present the self-similarities of the image, which are referred as a set of labels to

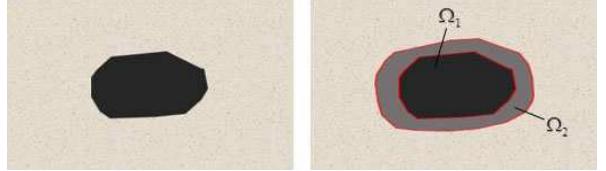


Figure 4: An illustration of an unknown region to be completed with a dilation operation: (Left) the original missing part; (Right) the dilated result.

be optimized in graph cuts. After that, they defined a Markov random field (MRF) energy function as follows:

$$E(L) = \sum_{\mathbf{x} \in \Omega} E_d(L(\mathbf{x})) + \sum_{(\mathbf{x}, \mathbf{x}') | \mathbf{x} \in \Omega, \mathbf{x}' \in \Omega} E_s(L(\mathbf{x}), L(\mathbf{x}')), \quad (2)$$

where Ω stands for the unknown region to be completed, \mathbf{x} denotes a point in Ω , $(\mathbf{x}, \mathbf{x}')$ denotes a pair of four-connected neighbouring points, and L refers to the label map, which assigns each unknown pixel $\mathbf{x} \in \Omega$ a label $L(\mathbf{x})$. Each label corresponds to one of the pre-selected dominant offsets. If a label i is assigned to the pixel \mathbf{x} , i.e., $L(\mathbf{x}) = i$, it means that the color of \mathbf{x} will be completed by that of the pixel at $\mathbf{x} + \mathbf{o}_i$ where \mathbf{o}_i denotes the offset coordinates of the label i .

In (He and Sun, 2014), the data energy term E_d in Eq. (2) is defined as:

$$E_d(L(\mathbf{x})) = \begin{cases} 0, & \text{if the label } L(\mathbf{x}) \text{ is valid for } \mathbf{x}, \\ +\infty, & \text{if the label } L(\mathbf{x}) \text{ is invalid for } \mathbf{x}. \end{cases} \quad (3)$$

And the smooth energy term E_s is defined as:

$$E_s(L(\mathbf{x}), L(\mathbf{x}')) = \|\mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}))) - \mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}')))\|^2 + \|\mathbf{I}(\mathbf{x}' + \mathbf{o}(L(\mathbf{x}))) - \mathbf{I}(\mathbf{x}' + \mathbf{o}(L(\mathbf{x}')))\|^2, \quad (4)$$

where $\mathbf{I}(\cdot)$ denotes the color values of a pixel in the image \mathbf{I} , and $\mathbf{o}(\cdot)$ denotes the offset coordinates of some label.

For the data energy term, we think its penalty is so weak to constrain label optimization. Although the unknown region has been expanded by one pixel to include boundary conditions, the imposed boundary constraints still fail to get satisfactory results in some cases. Hence, we propose to further strengthen the constraint of the data energy. As shown in Figure 4, Ω_1 is the missing region to be completed and we expand it to Ω_2 with a simple dilation operation. Let Ω_3 be the extra region caused by dilation, i.e., $\Omega_3 = \Omega_2 - \Omega_1$. Different from setting the labels of pixels in Ω_1 , all pixels in Ω_3 are assigned with a new label at an offset $(0, 0)$. In this way, we further optimize the whole expanded region Ω_2 . We impose boundary constraints by considering the differences between the optimized output and the original data in Ω_3 . The data energy term is modified as follow:

$$E_d(L(\mathbf{x})) = \begin{cases} D_S, & \text{if } \mathbf{x} \in \Omega_3, \\ 0, & \text{if } \mathbf{x} \in \Omega_1 \text{ and } L(\mathbf{x}) \text{ is valid for } \mathbf{x}, \\ +\infty, & \text{if } \mathbf{x} \in \Omega_1 \text{ and } L(\mathbf{x}) \text{ is invalid for } \mathbf{x}, \end{cases} \quad (5)$$

where D_S is used to measure the difference of Ω_3 before and after optimization and imposes important boundary constraints. Here, we consider both color intensities and gradient magnitudes in D_S , which is defined as:

$$D_S = \|\mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}))) - \mathbf{I}(\mathbf{x})\|^2 + \|\nabla \mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}))) - \nabla \mathbf{I}(\mathbf{x})\|^2$$

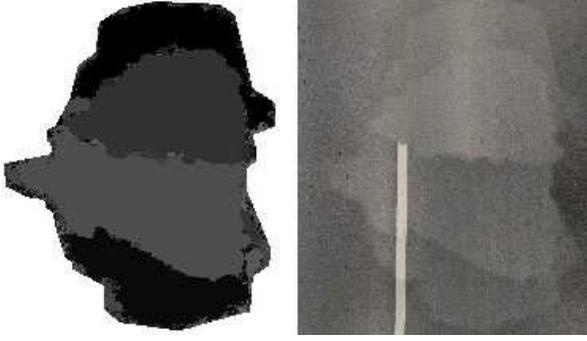


Figure 5: An illustration of the optimal label map (Left) and its correspondingly completed image region at the bottom of a streetview panorama.

where $\nabla \mathbf{I}$ stands for the gradient map of the image \mathbf{I} .

For the smooth energy term, only color intensities are considered in (He and Sun, 2014). When the image of the scene is complex enough, it may find a pixel that has a similar color but is not suitable for the missing pixel anymore. As we know, gradient contains more information about image structure. So, we combine it with color intensities to construct our smooth energy to reduce the artifact, which is defined as:

$$E_S(L(\mathbf{x}), L(\mathbf{x}')) = E_S^{color} + \alpha \times E_S^{gradient}, \quad (6)$$

where α is a weight parameter to balance color intensity and gradient terms ($\alpha = 2$ was used in this paper),

$$\begin{aligned} E_S^{color} &= \| \mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}))) - \mathbf{I}(\mathbf{x} + \mathbf{o}(L(\mathbf{x}'))) \|^2 + \\ &\quad \| \mathbf{I}(\mathbf{x}' + \mathbf{o}(L(\mathbf{x}))) - \mathbf{I}(\mathbf{x}' + \mathbf{o}(L(\mathbf{x}'))) \|^2, \end{aligned} \quad (7)$$

and

$$\begin{aligned} E_S^{gradient} &= \| \nabla \mathbf{I}(\mathbf{x} + L(\mathbf{x})) - \nabla \mathbf{I}(\mathbf{x} + L(\mathbf{x}')) \|^2 + \\ &\quad \| \nabla \mathbf{I}(\mathbf{x}' + L(\mathbf{x})) - \nabla \mathbf{I}(\mathbf{x}' + L(\mathbf{x}')) \|^2. \end{aligned} \quad (8)$$

With both improved data and smooth energy terms, the total energy function can be optimized in a graph cuts energy optimization framework with a publicly available multi-label graph-cuts library¹. The optimized output is a label map L , which assigns each unknown pixel with an offset label.

2.3 Image Blending

Different labels guide the pixels in the unknown region to be completed to find the optimally corresponding one at different parts of an image. Different patches generated by the optimal label map maybe result in luminance differences, color deviations, and even geometrical parallax among the finally completed image region, as shown in Figure 5. To greatly alleviate these issues, we choose to apply a multi-bland image blending algorithm (Burt and Adelson, 1983) for perfect image mosaicking from the completed patches and the originally reprojected image.

To make use of the multi-bland image blending algorithm, we first generate the label mask map for each single label. Then, we expand these label mask maps with a dilation operation to make them having overlaps, which is the basic requirement of using this blending algorithm. For each expanded label mask map, we individually generate its corresponding image according to the patch offset the label refers to. Figure 6 presents three selected label

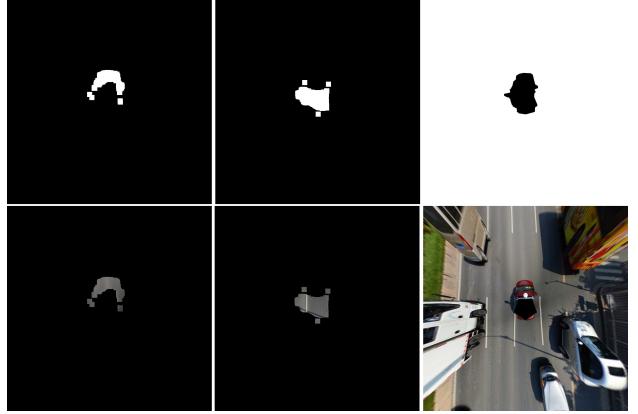


Figure 6: The mask maps corresponding to three labels in the top row and the corresponding sub-images in the bottom row.

mask maps with their corresponding sub-images. Especially, we refer to the whole background mask, which is not required to be completed, as another label and the background of the uncompleted projected image as a sub-image, as shown in the last column in Figure 6. In this way, we not only blend sub-images generated by different labels and but also blend them with the background to suppress the seams between completed patches and the original background image.

However, while the luminance differences between neighboring patches or one between completed patches and the original background image are very apparent, only applying a multi-bland image blending is not enough to eliminate the artifacts. In these extreme cases, we choose a simple gain model as the luminance refinement model to correct those patches used for completion before performing the multi-bland image blending. As for each pair of adjacent sub-images \mathbb{P}_i and \mathbb{P}_j , as illustrated in the bottom row in Figure 6, we transfer them to the $l\alpha\beta$ color space, in which the l channel denotes the main luminance component. The difference between \mathbb{P}_i and \mathbb{P}_j after the adjustment can be written as:

$$e_{ij} = (a_i \mu(\mathbb{P}_i) - a_j \mu(\mathbb{P}_j))^2 + \alpha((a_i - 1)^2 + (a_j - 1)^2), \quad (9)$$

where $\mu(\cdot)$ denotes the mean luminance of valid pixels in some sub-image, α is a fixed weight for preserving the average variance of the images after adjustment. and a_i and a_j stand for the gain coefficients of \mathbb{P}_i and \mathbb{P}_j , respectively. Considering the joint adjustment for all the pairs of adjacent sub-images in all N labels, the total difference can be written as:

$$E = \sum_{\mathbb{P}_i \text{ and } \mathbb{P}_j \text{ are adjacent}}_{i=1 \dots N, j=1 \dots N} e_{ij}. \quad (10)$$

We aim to find such a set of coefficients $\mathcal{A} = \{a_i\}_{i=1}^N$, which minimizes the total difference between each pair of images in overlaps. It can be achieved by solving Eq. (10) through the linear least square algorithm.

Combining the luminance adjustment under a gain model and a multi-bland image blending algorithm can improve the luminance consistency of the completed projected image, even when the luminance difference is very large at the beginning, and generate a perfectly completed image as seamless as possible.

2.4 Back-Projection to Panorama

After mosaicking the completed image region, we back-project it onto the original panorama model to fill the occluded region at the bottom of the panorama. In Section 2.1, we have described

¹ Available at <http://www.csd.uwo.ca/faculty/olga/>



Figure 7: An illustration of obvious artifacts in a completed panorama by just back-projecting the completed image region in the ground perspective plane.

the transformation from the cylindrical-projection panorama to the perspective projected image. Back-projection is its reverse process. By inverting \mathbf{R}_φ and \mathbf{R}_ω , the transformation of back-projection can be formulated as:

$$\begin{pmatrix} X_S \\ Y_S \\ Z_S \end{pmatrix} = \mathbf{R}_\omega^{-1} \mathbf{R}_\varphi^{-1} \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix}. \quad (11)$$

2.5 Panorama Feathering

When we back-project the whole mosaicked image region onto the panorama, the resolution has declined in the area that would be covered when the whole projected image is back-projected onto the panorama. Thus, we would just back-project the completed image region corresponding to the missing one rather than back-projecting the whole mosaicked image. However, just back-projecting the completed image region would cause an obvious artifacts in the finally completed panorama due to different resolutions between the back-projected image region and the original panorama. In addition, it also introduces another issue. Although the blending result on the projected image is perfect, it changes the luminance and color of the image regions close to the completed region but not inside it due to that the multi-band blending adjusts the whole image in the whole. As shown in Figure 7, if we only back-project the completed image region to the panorama, there exist obvious artifacts in the finally completed panorama.

The above mentioned two issues can be efficiently solved by applying a simple feathering on the original panorama and the completely back-projected one. As shown in Figure 8, we do feathering by adjusting the color intensities between lines A and B where the line A denotes the boundary of the occluded region to be completed and the line B is far away from A with a fixed range. Given a pixel \mathbf{x} between the lines A and B , the color intensities after feathering are calculated as follows:

$$\mathbf{I}_p(\mathbf{x}) = \alpha \times \mathbf{I}_p^o(\mathbf{x}) + (1 - \alpha) \times \mathbf{I}_p^b(\mathbf{x}), \quad (12)$$

where \mathbf{I}_p , \mathbf{I}_p^o , and \mathbf{I}_p^b stand for the feathered panorama, the original one, and the back-projected one completely from the mosaicked image in the ground perspective plane, respectively, and the feathering coefficient α is calculated as:

$$\alpha = d(\mathbf{x}, A) / (d(\mathbf{x}, A) + d(\mathbf{x}, B)),$$

where $d(\mathbf{x}, A)$ and $d(\mathbf{x}, B)$ represent the vertical distances from the pixel \mathbf{x} to the lines A and B , respectively.

3. EXPERIMENTAL RESULTS

To sufficiently evaluate our proposed unified blending framework for panorama completion, we tested our algorithm on a set of



Figure 12: Comparative results of performing image blending without luminance compensation in the top-right corner and combining two methods in the bottom-right corner

representative streetview and indoor 360° panoramas, provided by Baidu and Tencent.

Recently, He and Sun (2014) proposed a graph-cuts-based image completion algorithm which is considered as one state-of-the-art method. However, their proposed method only considers color intensities in the definition of MRF energy function and its constraints upon boundary areas are too weak to obtain some satisfactory results in some cases. To prove that our improvements do work well even in challenging cases, we tested it on some representative non-panoramic images. Figure 9 shows the completion results on two images with obvious self-similarities but with slight deviations to some extent. From Figure 9, we observed that the completion results generated via our improved graph cuts is better than the original method proposed by He and Sun (2014). The misplacement between the whole completed image region and the boundary one of the original background image was obviously reduced as shown in the first row in Figure 9. Also, the misplacement between different patches inside the completed image region was alleviated to some extent as shown in the second row in Figure 9. This improvement benefits from strengthening the boundary and gradient constraints in our improved graph cuts optimization framework.

Figure 10 shows the comparative results with a multi-band blending operation and without it applied in the ground perspective images of the bottom regions of three streetview panoramas, from which we observed that the proposed blending strategy greatly eliminates the image luminance and color deviations, and also conceals small misplacements. The used luminance gain model calculated the gain coefficients through the linear least square algorithm and adjusted the luminance of all the patches (i.e., sub-images) according to their own gain coefficients. Figure 11 shows comparative results, from which we observed that our used luminance compensation strategy is able to correct all the completed patches and the original background image into a more consistent luminance. In some extreme cases with severe luminance differences, directly applying the multi-band image blending would not work very well. In this condition, we first performed the luminance compensation and then applied the multi-band image blending, whose advantage is illustrated in Figure 12. From Figure 12, we observed that the combined strategy slightly improves the blending result than just applying the multi-band blending.

Panorama feathering is a post-processing work to suppress the artifacts on the completed panoramas. Figure 13 shows the comparative results on three streetview panoramas before and after feathering. From Figure 13, we observed that our proposed panorama



Figure 8: An illustration of the feathering area in a panorama.

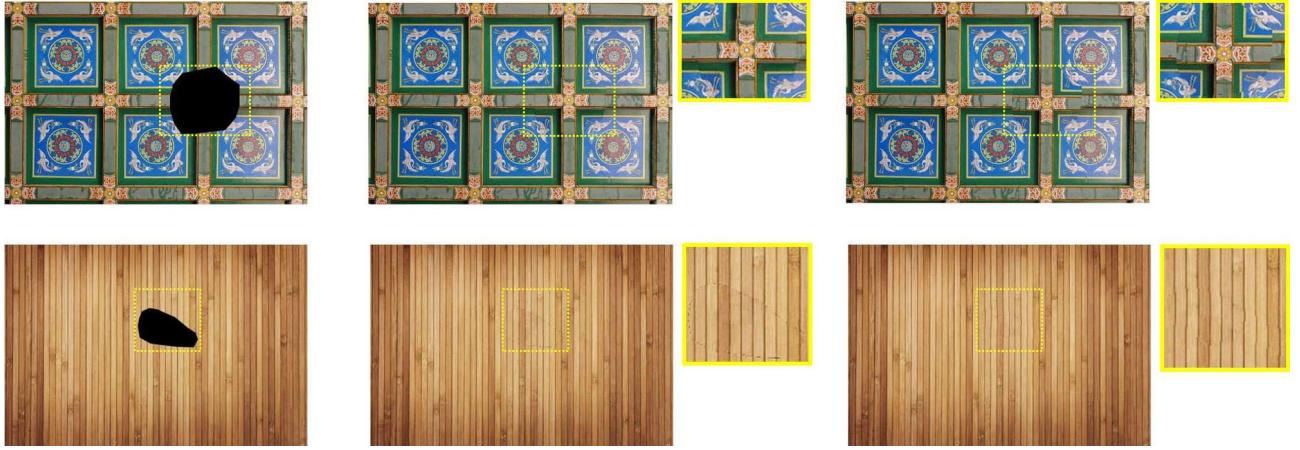


Figure 9: Comparisons between our improved graph cuts optimization and the original method proposed by He and Sun (2014): input images with black mask regions to be completed, the completion results by our method and the original one, from left to right.

feathering strategy can obviously eliminate the artifacts in the finally completed panoramas.

Figure 14 shows two finally completed panoramas by our proposed unified framework, from which we observed that the completed results are very satisfactory as a whole not only for indoor panoramas but also for outdoor ones.

4. CONCLUSION

In this paper, we propose a unified framework for efficiently completing streetview and indoor 360° panoramas due to the lack of bottom areas caused by the occlusion of the acquisition platform. The whole unified framework is comprised of five main stages. Firstly we reprojected the panorama onto the ground perspective plane containing the whole occluded region to be completed. Secondly, the image completion problem was solved by formulating it in an improved graph cuts optimization framework. Thirdly, we proposed to apply the luminance compensation and a multi-blend blending operations to greatly eliminate the luminance differences of different patches used for completion, which has great effect in completing streetview and indoor 360° panoramas with complex lighting conditions, and generate a perfectly mosaicked image as seamless as possible. Fourthly, we back-projected the completed and mosaicked ground perspective image into the cylindrical-projection panorama. Finally, we did feathering on the completed panorama to further reduce artifacts in the panorama. This proposed unified framework is helpful to quickly complete a perfect and completed 360° panorama. Experimental results on a large amount of representative non-panscopic images and panoramas demonstrate the efficiency and

robustness of our proposed panorama completion algorithm. Although the satisfactory results can be achieved in most cases, the current framework is not very feasible for the ground projection images with severe perspective distortion or with very cluttered backgrounds, which will be further studied in the future. In addition, we implemented the current completion optimization strategy with a single CPU, which can be further improved with multiple CPU cores and even multiple GPU ones.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Project No. 41571436), the Jiangsu Province Science and Technology Support Program, China (Project No. BE2014866), the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027), and the South Wisdom Valley Innovative Research Team Program.

References

- Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G. and Verdera, J., 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing* 10(8), pp. 1200–1211.
- Bertalmio, M., Sapiro, G., Caselles, V. and Ballester, C., 2000. Image inpainting. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, p. 417–424.

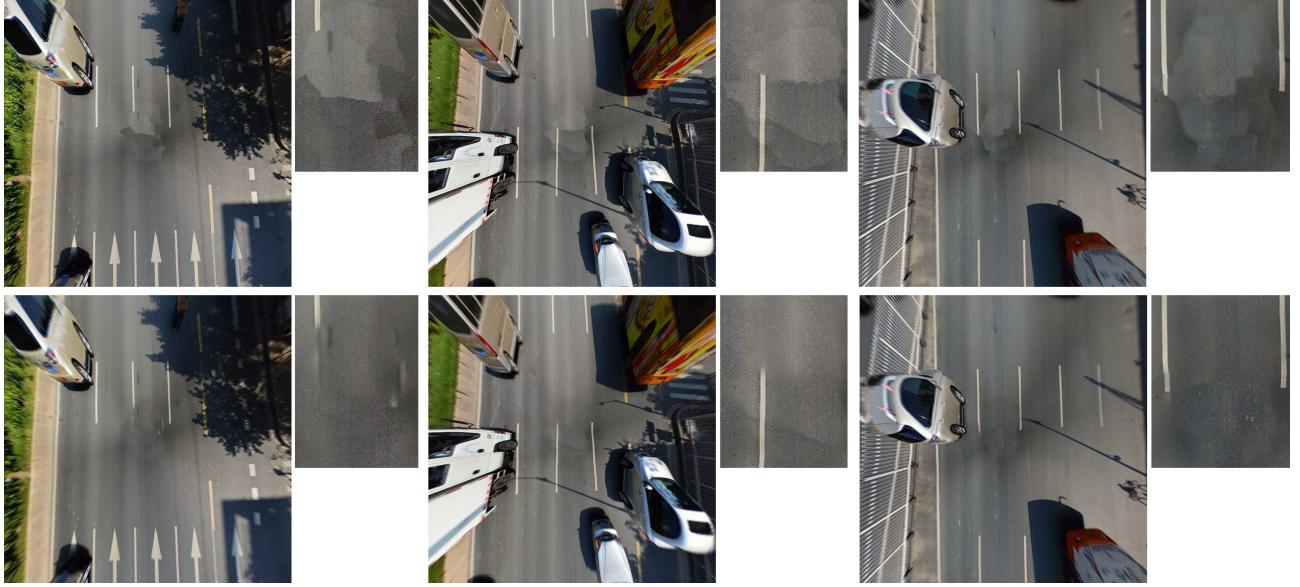


Figure 10: Comparative image completion results without applying a multi-band blending operation in the top row and with applying it in the bottom row.

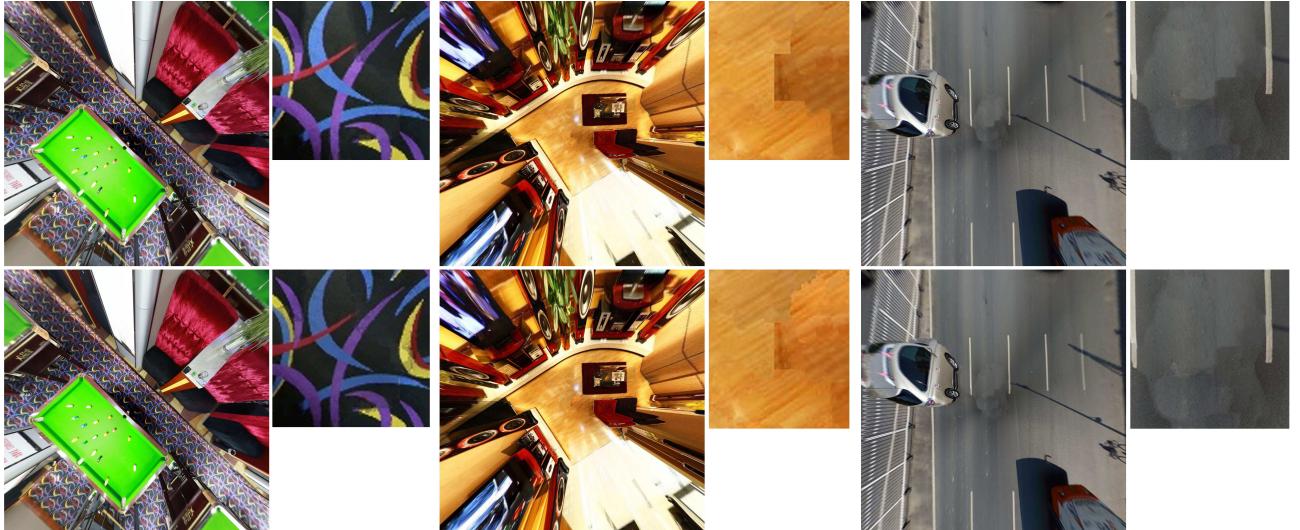


Figure 11: Comparative results without the luminance compensation in the top row and with it in the bottom row.

- Bertalmio, M., Vese, L., Sapiro, G. and Osher, S., 2003. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing* 12(8), pp. 882–889.
- Boykov, Y., Veksler, O. and R.Zabih, 2001. Efficient approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(12), pp. 1222–1239.
- Burt, P. and Adelson, E. H., 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* 2(4), pp. 217–236.
- Criminisi, A., Perez, P. and Toyama, K., 2003. Object removal by exemplar-based inpainting. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
- Drori, I., Cohen-Or, D. and Yeshurun, H., 2003. Fragment-based image completion. In: ACM Transactions on Graphics (TOG), Vol. 22number 3, pp. 303–312.
- Efros, A., Leung, T. K. et al., 1999. Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision (ICCV), Vol. 2, pp. 1033–1038.
- He, K. and Sun, J., 2012. Computing nearest-neighbor fields via propagation-assisted kd-trees. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 111–118.
- He, K. and Sun, J., 2014. Image completion approaches using the statistics of similar patches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(12), pp. 2423–2435.
- Jia, J. and Tang, C.-K., 2003. Image repairing: Robust image synthesis by adaptive nd tensor voting. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. I–643.
- Jia, J. and Tang, C.-K., 2004. Inference of segmented color and texture description by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(6), pp. 771–786.
- Levin, A., Zomet, A. and Weiss, Y., 2003. Learning how to inpaint from global image statistics. In: IEEE International Conference on Computer Vision (ICCV), pp. 305–312.
- Roth, S. and Black, M. J., 2005. Fields of experts: A framework for learning image priors. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, pp. 860–867.

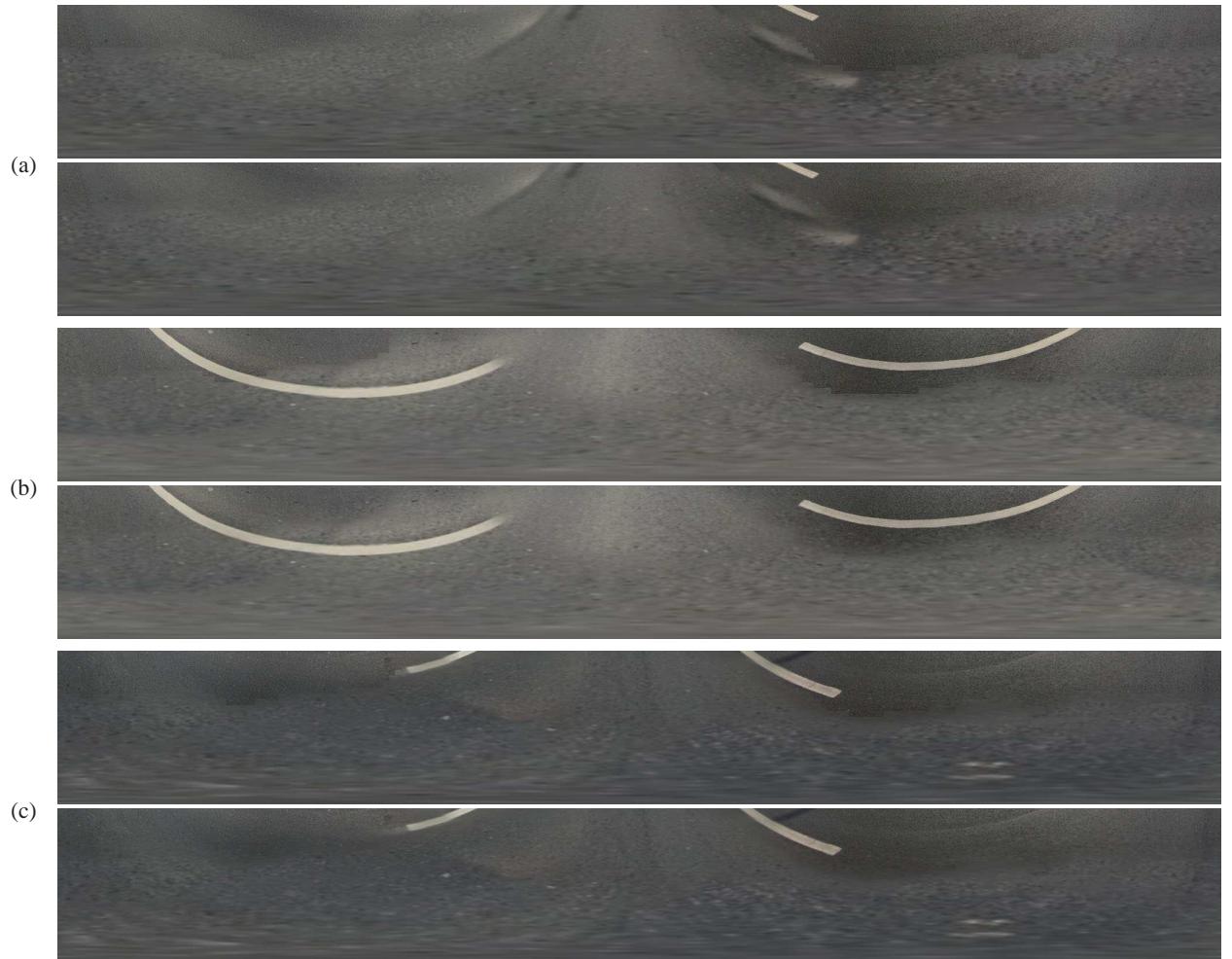


Figure 13: Comparisons of three completed streetview panoramas before feathering (top images) and after feathering (bottom images).

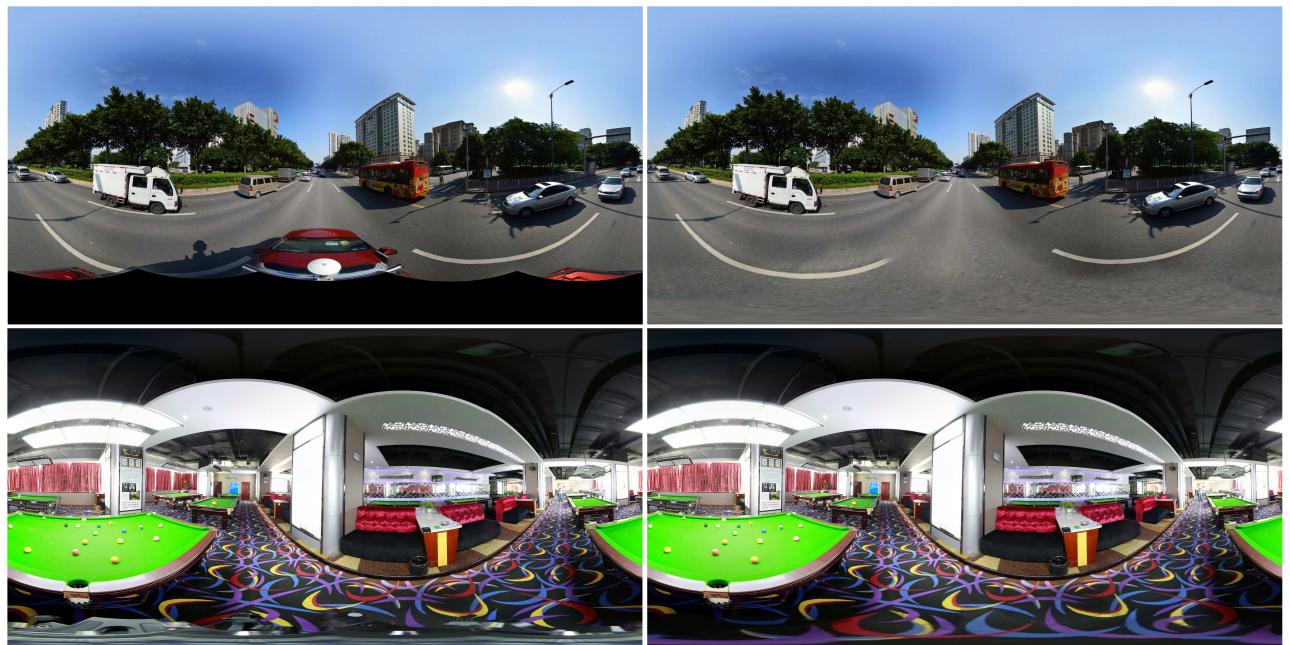


Figure 14: Two finally completed panoramas by our proposed unified framework.