

Natural Image Composition with Inhomogeneous Boundaries

Dong Wang¹, Weijia Jia¹, Guiqing Li², and Yunhui Xiong²

¹ City University of Hong Kong, China

² South China University of Technology, China
donwang@student.cityu.edu.hk,
wei.jia @cityu.edu.hk,
{ligq,yhxiong}@scut.edu.cn

Abstract. Image composition usually floods the composition region of a target image with the same shape as a source image patch. To achieve seamless transition effect, the tone of the boundary in the target image is then transferred to the interior region of the source patch. Traditional approaches usually fail to work for the case that the corresponding boundaries of target and source images don't match well because the tone transformation of all pixels on the boundary are equally propagated to the inner region. This paper presents a new image composition technique based on discrete mean value coordinates(DMVC), which supports the transition of tone transformation of part selected not all pixels on the boundary to the inner region. The approach works as follows. It firstly selects boundary pixels having good matching. The new color of inner pixels is then calculated using DMVC according to those selected pixel pairs from the source and target boundaries. Matting technique is finally introduced to compose the new pixels to the target image. Experiments show that the proposed approach can obtain reasonable results for examples with inconsistent boundaries between source and target images.

Keywords: Image composition, Discrete mean value coordinates, Image editing, Edit propagation.

1 Introduction

Image composition as an interactive image editing operation plays an important role in image processing. Its purpose is to generate a new image by merging the selected image patches from one or more images into one base image naturally and seamlessly. It is widely used in moving making, photo processing, and web design etc. In general, there are two kinds of image composition techniques: image cloning and image blending [1]. For image cloning each pixel in the new composite image is from either the source image or the target image; while image blending deals with a blending operator about alpha between source image pixel and corresponding target pixel. In order to achieve seamless composite effect, we adopt image blending technique by combing alpha matting method.

A trivial image composition is to copy the foreground of the source image patch to some region of the target image without any further processing. It is hard for this method to produce seamless composition results with a unified color tone due to the fact that the target image and source image generally have different hue style. On the other hand, it is not easy for users to exactly extract the foreground of the source image. It is expected that the foreground objects can be quickly specified by sketching a rough region boundary in which not only the foreground objects but also some background pixels are contained. In fact, it is useful to include some background pixels around the objects to be cloned as they can act as the role of a smooth transition from target image to these objects. However, it will exhibit artifacts in the composed image if the corresponding background textures of source and target images are severely different. Moreover, if the selected region is too large, some important content of the target image is possibly deteriorated [2].

As illuminations of the source region and the target image are usually inconsistent, their color tones are generally unmatched and hue style transfer is therefore necessary. Gradient-based methods are a natural choice for conducting such a task [3]. This type of methods generally force the gradient of the composite region accordant with that of the source region while interpolating the boundary pixels from the target image. It finally results in a Laplace equation which can be discredited as a large sparse linear system. As an approximation of harmonic functions, Fleisman et al. employ mean value coordinates (MVC) to interpolate the boundary mismatch between target and source images [4]. This yields a more efficient cloning scheme. The composite boundary plays an important role for both methods. The effectiveness of image composition depends on the selection of boundaries considerably. If the texture and color of the source and target boundaries are consistent, they can achieve satisfactory results. If inconsistent, however, it may bring unrealistic artifacts. This indicates that it is necessary to carefully consider the boundary problem.

A possible way is to automatically find a closed path enclosing the objects to be cloned in the source image and a corresponding region in the target image such that the mismatch is minimized. However, users may usually be willing to specify the boundary of the source objects and the position they should be pasted in the target image. Chen et al. [5] attacked the problem by classifying boundary pixels into consistent and inconsistent sets and employ common and blending boundary conditions, respectively. Ding and Tong [6] further improved the Poisson image composition method by weighting the gradient error which finally yielded a MVC-like scheme. However, both methods do not support controllable boundary matching.

To attack the problem, we put the boundary pairs between source and target regions into four classes: whole-to-whole, part-to-whole, whole-to-part, and part-to-part when propagating the boundary mismatch to the inner source region, where A-to-B indicates that the color tone of A pixels from the source boundary should be transferred to the tone of B pixels from the target boundary. What kind of pairs is chosen depends on the mismatch degree and we provide an

automatic method based on a matching criteria to select a boundary matching style. Of course, it can also be specified by users.

It should be pointed out that the last three classes result in open boundaries. It is difficult to directly apply gradient-based/MVC-based methods to these cases as they require a closed boundary. This motivates us to adopt a DMVC interpolation to propagate the mismatch. Our contributions are as follows: (1) A new composition algorithm based on DMVC is devised, which can deal with the propagation of open boundaries; (2) It can generate a reasonable composition even when the boundaries of target and source regions do not match well while it is difficult to do this for previous methods.

2 Related Work

There are two basic image composition approaches: alpha blending and gradient-based method [6]. Alpha blending approaches are mainly focused on alpha calculation. In [7], the matte is directly reconstructed from matte gradient field by solving Poisson equation with boundary information from a user-supplied trimap. [8] proposed a closed-form matting approach by explicitly deriving a cost function from local smoothness assumptions on the foreground and background colors, where each foreground and background is a linear mixture of two colors over a small neighborhood around each pixel. [9] introduced a graph cut technique to develop a method of interactive foreground extraction and alpha-matting of color images. The algorithm builds a Gaussian mixture model of color image based on a given trimap, and iteratively alternates between graph cuts and the updated model parameters until it satisfies a specified convergence criterion. If we divide the composition process into two parts: cut and paste, alpha blending mainly concerns about the cut, how to get the expected region.

Gradient-based methods pay more attention to the pasting effect. [3] gave a Poisson image editing method where the gradient field inside the cloned region is taken from the source image and the Dirichlet boundary conditions for the equation is defined by the boundary of the cloned region where the pixel color is from the corresponding target image. For gradient domain techniques need to solve a large sparse linear system, a number of works proposed fast Poisson solvers for various scenarios [10,11] and for solving the Poisson equation on the GPU [12]. [4] gave a different fast algorithm by introducing mean value coordinates(MVC) image cloning. The new color of cloned region is the sum of the base color from the source image patch and the offset. The offset along the boundary of the cloned region is the difference between the corresponding pixel of the source image and the target image, while the offset of each interior pixel is given by a weighted combination of values along the boundary.

To preserve the color fidelity of the foreground objects of source images, [1,6] gave their different solutions. Based on the Poisson image editing framework, [1] proposed a variational model that considers both the gradient constraint and the color fidelity. They did a soft segmentation to the source image to obtain the foreground and the new gradient of cloned region is the weight sum of the

corresponding gradient of the source image and the target image with the soft segmentation as weight. [6] discussed a content-aware image blending technique. Based on MVC image cloning, the offset is controlled by a coefficient related to the alpha, where they can preserve the color of foreground region unchanged.

When the corresponding boundary of source image and target image do not match, it may generate blending artifacts in the composite image by gradient-based methods, as discussed in [2] and [13]. In [2] they proposed a new objective function to compute an optimized boundary condition. A shortest closed-path algorithm is designed to search for the location of the boundary. But their method may produce local optimization. In [13] they improved standard gradient-domain compositing technique. They redefine the boundary gradients to ensure the produced gradient field is nearly integrable and spread the residuals to those regions where they are less conspicuous. They do not overcome but disperse the bleeding artifacts.

We cast image composition as a discrete MVC interpolation problem which diffuses the difference between pixels from source and target boundaries. The method does not pose any special requirements on initial conditions. The approach can produce more natural composition results than gradient-based and MVC-based composition techniques for the case with complicated boundary color tones.

3 Discrete MVC Image Composition

In MVC-based image cloning [4], a new pixel in the composite region is obtained by adding an offset to the corresponding pixel of the source region. The offset on boundary pixels is defined as the difference of the corresponding boundary pixels. The offset inside composite region is calculated by an MVC-based interpolatory function interpolating the boundary offsets. In our discrete MVC image composition, we allow offsets of a portion of boundary pixels to be diffused while other boundary pixels are treated using blending technique. In this section, we firstly describe our problem formulation, then derive the discrete MVC offset propagation, and finally outline the composition algorithm.

3.1 Nomenclature

Suppose a source image $\iota_s : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ and a target image $\iota_t : \mathbb{R}^2 \rightarrow \mathbb{R}^d$ are given, where d is the color feature dimension. A source image region $\Omega_s \subset \mathbb{R}^2$ to be cloned onto $\Omega_t \subset \mathbb{R}^2$ is selected by interactively specifying its closed boundary $\partial\Omega_s$. The cloned region $\Omega_t \subset \mathbb{R}^2$ in the target image ι_t is uniquely determined by a translation $o(o_x, o_y) : \Omega_t = \{q' | q' = q + o, q \in \Omega_s\}$. Let $P_s \subset \partial\Omega_s$ be a set of pixels on the boundary of the source region, and $P_t \subset \partial\Omega_t$ the corresponding pixel set on the boundary of the target image. Furthermore, denote the composition image by $\iota_c : \mathbb{R}^2 \rightarrow \mathbb{R}^d$. Finally, $\alpha : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a mask such that

$$\alpha(p) \begin{cases} = 0, p \in \Omega_b, \\ = 1, p \in \Omega_f, \\ \in (0, 1), \textit{otherwise.} \end{cases}$$

where $\Omega_b(\subset \Omega_s)$ is the background and $\Omega_f(\subset \Omega_s)$ is the foreground, which will be used as blending weight. To achieve good composition quality, α should be the sampling of a smooth function for which α maps calculated using image matting methods [7,8,9] are good candidates.

3.2 Discrete MVC Interpolatory Function

For given planar polygon P and function f defined on the boundary of the polygon, the MVC interpolation function on the plane is defined as [14,15,16]

$$f(p) = \frac{\int_{q \in \partial P} \omega(p, q) f(q) dq}{\int_{q \in \partial P} \omega(p, q) dq},$$

where $\omega(p, q) = \frac{1}{\|p-q\|}$ which we replace with $\omega(p, q) = \frac{1}{\|p-q\|^2}$ for efficiency consideration.

In our setting, only a set of discrete pixels are known for interpolation, therefore we introduce a so-called discrete MVC interpolatory function

$$f_P(p) = \frac{\sum_{q \in \partial P} \omega(p, q) f_P(q)}{\sum_{q \in \partial P} \omega(p, q)},$$

where P is a discrete point (pixel) set.

3.3 Image Composition

Now we can produce an intermediate image ι_i by diffusing the color offsets between P_s and P_t in order to transfer the color tone of Ω_t to ι_s using the discrete MVC interpolatory function

$$\iota_i(p) = \iota_s(p) + f_P(p), \forall p \in \mathbb{R}^2$$

with $f_P(q) = \iota_t(q + o) - \iota_s(q)$ for $q \in P_s$ which is the color difference between target and source images.

After this processing, ι_i will have a similar appearance to that determined by pixels in P_t . We then embed the corresponding region in ι_i to Ω_t of the target image smoothly with alpha blending:

$$\iota_c(q) = \alpha(q - o)\iota_i(q - o) + (1 - \alpha(q - o))\iota_t(q), \forall q \in \mathbb{R}^2$$

4 Boundary Matching Analysis

In image composition, the boundary, as initial conditions, its color in the source image patch is replaced by that of the corresponding target region. And the color inside the composition region should be renewed following the change on boundaries. But in some cases, the source boundary and the target boundary may be greatly different. Just like the case in Fig. As an example, let us see 1(a) in which the source boundary consists of two parts of pixels: one is a leaf part and the other is a sky part. If we only select leaf pixels of the source region to compute edit propagation, the flower should follow a transformation between leaves instead of sky to leaves, and therefore receive little change. But if the sky part of pixels is chosen, the flower should greatly be changed following the difference between between sky pixels and the leaf pixels. So in order to make the composite image more natural and realistic, we need to select the proper initial conditions from the boundary by analyzing the source image and the target image (see 1(c) for example). We categorize the matching types of the source and target boundaries into four classes: (1) whole-to-whole, (2) part-to-whole, (3) whole-to-part, and (4) part-to-part, where the former refers to the boundary status of the source region while the latter stands for the boundary status of the target boundary. For the first class composition that the two boundaries completely match, previous methods can produce ideal results. All other three cases, which belong to partially matching, are be discussed in the following subsections.

4.1 Part-to-Whole Matching Composition

Let's observe Fig. 1 again. We want to replace the red flower in the target image (1 (b)) with the yellow rose in the source image. To avoid the inconsistent matching between whole-to-whole boundaries, we only select leaf pixels from the source image which can match the boundary pixels of the target image very well and generate the composition image by propagating the difference between the selected source pixels and the corresponding target image pixels to the inner pixels of the source patch. To extract those pixels quickly, some reference pixels are provided interactively. We sketch the contour from the leaves and choose a portion of pixels (one percent of the contour pixels for example) as reference pixels. The remaining pixels of the contour are classified by a matching method. The matching idea is that similar color appearance belongs to the same class, and the matching function is defined as

$$s_i = \max_j (\exp(-\|f_i - f_j\|^2) / \delta_f), \quad (1)$$

where i and j are both boundary pixels in the source image. i is the i th pixel whose matching value s_i will be calculated and j is the reference pixel. The meaning of the function is that pixel i matches with all reference pixels. If the maximum value $s_i > \varepsilon$ ($\varepsilon = 0.9$ in our experiments), the corresponding pixel will be selected as the initial condition to calculate the new composite image. The selected pixels are shown in Fig. 1(c) for the example in Fig. 1.

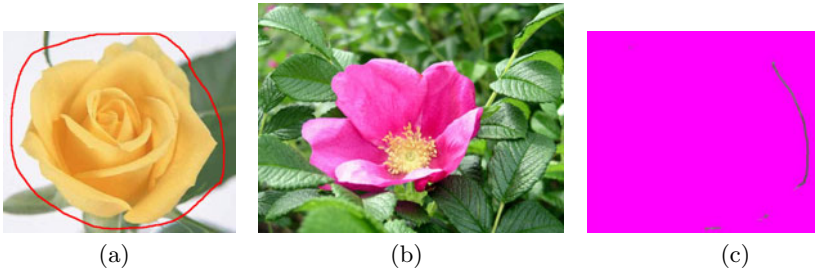


Fig. 1. A part-to-whole example: (a) original image, and selected region, (b) target image, (c) selected boundary pixels as initial conditions for image composition

4.2 Whole-to-Part Matching Composition

In Fig. 2, we hope to move the sand pyramid in the source image into the right top corner of the target image, as shown in Fig. 2(a). The sand pyramid is only surrounded by a kind of pixels, sand pixels, in the source image patch, but there are two kinds of pixels, sand pixels and sky pixels in the target boundary. Noticing that sand pixels and sky pixels are very different in color, the sky pixels will play a negative role in composition if we use the whole boundary pixels to calculate the result as shown in Fig. 6 (d). In this case, we need to extract the pixels from the target boundary which corresponds to sand. Automatic discrimination is difficult too. We again address the problem by manually selecting some reference pixels, (see the red curve in Fig. 2(b) as an example). The difference is that the reference pixels should be selected from the target boundary but not the source boundary. Figure 2(c) illustrates the matching result.

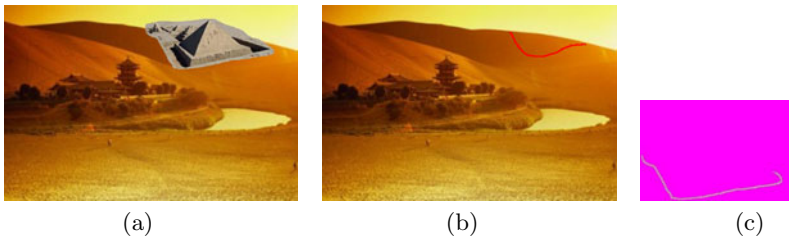


Fig. 2. A whole-to-part example: (a) original image patch and its composition position, (b) reference pixels for boundary pixel selection (marked with red), (c) selected boundary pixels

4.3 Part-to-Part Matching Composition

Figure 3 shows a part-to-part boundary matching example. We want to put two people in on the beach under the coco of the target image. It is a sunning day and there is shadow. The shadow map is created (at the right top of Fig. 3(a))

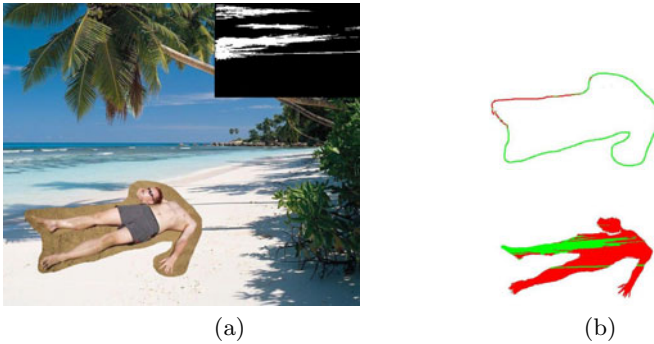


Fig. 3. A part-to-part example:(a) the shadow, original image patch and its composition position, (b) up: the divided boundary, shadow boundary red and the remaining green; bottom: the divided foreground, shadow region red and the remaining green

using Photoshop, which is a binary image. The shadow divides the people and the boundary in two cases, as Fig. 3(b). One is in shadow (marked with red) and the other is under the sun (marked with green). From Fig. 3(b), we can notice each case is made up of several parts, which presents complex boundary. The two cases will be dealt with separately as two composition regions.

5 Experiments and Discussion

In this section, we show our composite results by the proposed approach with a wide variety of source and target images and synchronously compare it with direct pasting, Poisson image editing and error-tolerant image compositing approaches. In our experiments, besides the initial conditions discussed in Section 4, we also need an alpha map for the source image with copied object as foreground in addition to the source and target images. With all these as input, the composition can be computed automatically and quickly.

Figure 4 depicts an example of part-to-whole type corresponding to images in Fig. 1. An alpha map is generated by Levins method [8] for the source image. The illumination of the target image is stronger than that of the source image. Fig. 4(c) presents the result by directly pasting the rose of the source image onto a region of the target image. It is obvious that color tone of the rose cannot match that of the background well as its illumination is weak. Figure 4(d) gives the result by Poisson editing. The color tone of the rose changes unevenly in this case and result exhibits obvious inharmony between the left top corner and the right bottom corner of the composite region. This is caused by unequal color difference between the source image and the target image boundaries. This phenomenon is improved in Figure 4(e) by error-tolerant image compositing method, which disperses the difference into the rose. However, the color appearance of the main object (rose) is almost completely different from its natural color in both results. Figure 4(f) illustrates our result in which only illumination is changed and the

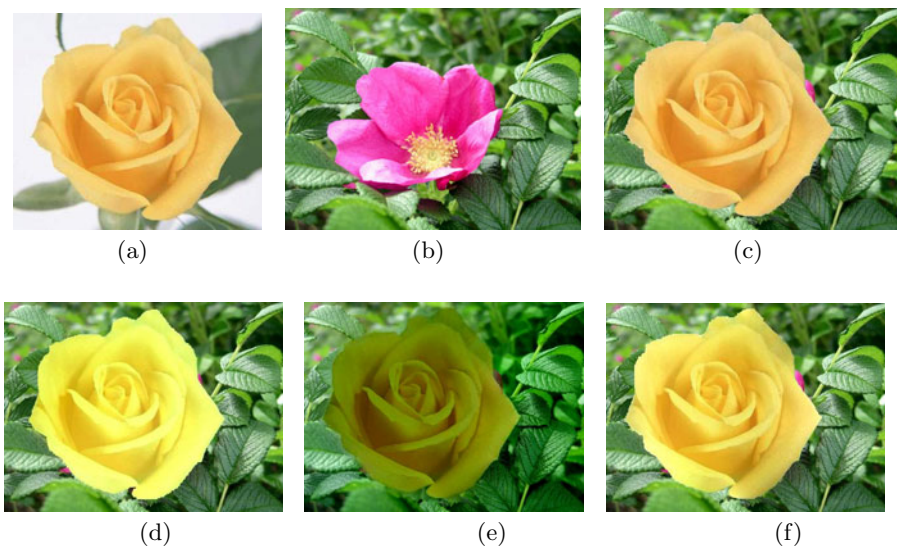


Fig. 4. A part-to-whole image composition example:(a) original image, (b) target image, (c) direct cut-and-paste,(d) Poisson method, (e)Tao's method [13], (f)our method

natural color of the rose is preserved very well. Figure 5 presents another part-to-whole example.

Figure 6 deals with the whole-to-part case. The color appearance of the sand of the target image is obviously different from that of the source image region whose boundary is homogeneous. Fig. 6(c) presents the result by the direct pasting technique. Figure 6(d) gives the result synthesized by the Poisson image editing technique. Since a closed boundary is demanded, the color tone of the top of sand pyramid is closer to the sky color tone and far away from the sand color tone. It is unrealistic. To alleviate the artifact, the boundary gradients are adjusted by error-tolerant image compositing method. The result shown in Fig. 6(e) is created by this improved method. It disperses the color difference



Fig. 5. Another part-to-whole image composition example:(a) original image, (b) direct cut-and-paste,(c) Poisson method, (d) our method

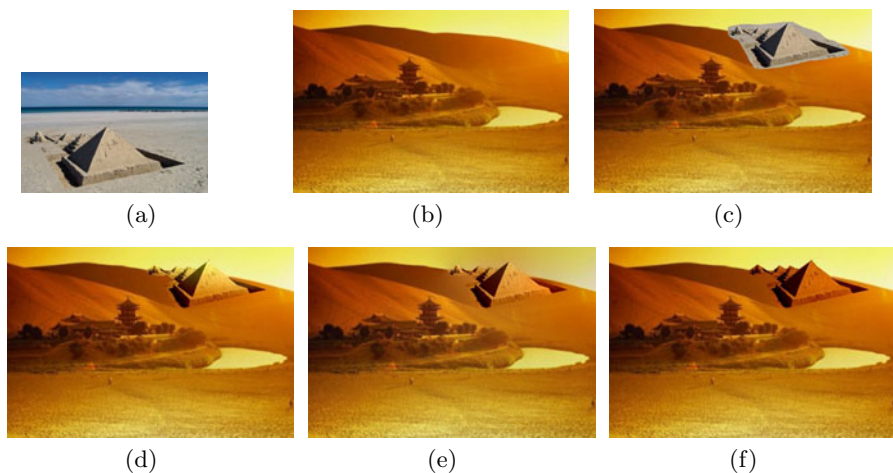


Fig. 6. A whole-to-part image composition example:(a) original image, (b) target image, (c) direct cut-and-paste,(d) Poisson method, (e)Tao's method [13], (f)our method

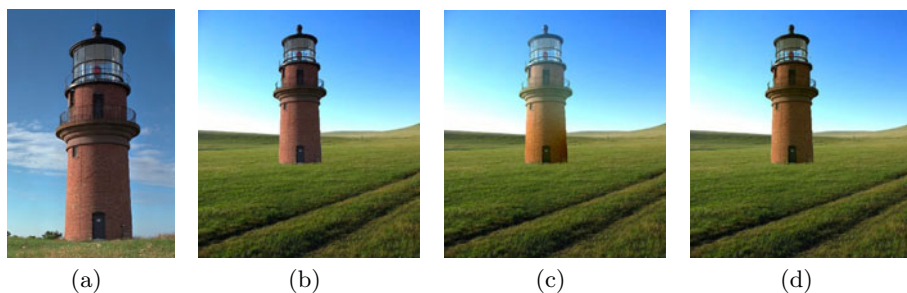


Fig. 7. Another whole-to-part image composition example:(a) original image, (b) direct cut-and-paste,(c) Poisson method, (d) our method

between the sky and the sand into the whole sand pyramid and therefore changes the sky color tone. Our approach clearly outperforms both methods as shown in Fig. 6(f). The color tone of the sand pyramid is well consistent with that of the target region with color tone of the left area deeper than that of the right area. Furthermore, its color tone has nothing with that of the sky color. Figure 7 present another example of this case.

Finally, Figure 8 demonstrate is a part-to-part example corresponding to images in Fig. 3. The color tone between the source image and the target image is very different for the latter includes a set of shadow pixels while the former hasn't. Again, the Poisson image editing composition cannot clearly reproduce the shadow on the man body as shown in Fig. 8 (b) due to the whole boundary color difference propagation while our result can nicely reproduce the sunshine

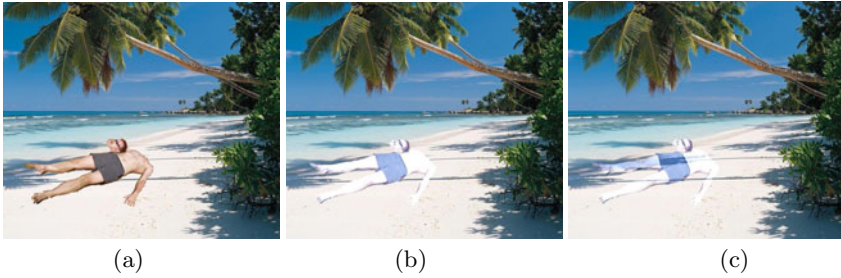


Fig. 8. Image composition with shadow:(a) direct cut-and-paste,(b) Poisson method, (c) our method

and shadow as shown in Fig. 8(d) attributing to intelligently selecting boundary pixels for propagation.

6 Conclusion

We present a discrete mean value coordinates based image composition technique which can generate natural composition images even if the boundaries of the source and target images cannot match well. Compared with previous methods, our method wisely selects an open boundary as the initial condition, and therefore can provide a good solution to the composition of some complex scenes with inhomogeneous boundary color tones.

In our current implementation, our method needs interaction in selection of boundary pixels. An automatic technique is expected in the future work by analyzing boundaries of the given source image patch and the corresponding target region, for example. In fact, there exist many special complex cases to be handled in image composition, such as shadow treatment, texture structure preserving or transferring, and so on. Finally, it is also interesting to investigate methods evaluating the pros and cons of existing composition techniques.

Acknowledgement. The work is partially supported by NSFC (60973084), NSF of Guangdong (915106410-1000106), and Fundamental Research Funds for the Central Universities (2009zz0016).

References

1. Yang, W., Zheng, J., Cai, J., Rahardja, S., Chen, C.: Natural and seamless image composition with color control. *IEEE Transactions on Image Processing* 18(11), 2584–2592 (2009)
2. Jia, J., Sun, J., Tang, C.-K., Shum, H.-Y.: Drag-and-drop pasting. *ACM Siggraph* 25(3), 631–636 (2006)
3. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. *ACM Siggraph* 22(3), 313–318 (2003)

4. Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., Fattal, R., Lischinski, D.: Coordinates for instant image cloning. *ACM Trans. on Graphics* 28(3), 67:1–67:10 (2009)
5. Chen, T., Cheng, M.-M., Tan, P., Shamir, A., Hu, S.-M.: Sketch2photo:Internet image montage. *ACM Trans. Graph.* 28(5) (2009)
6. Ding, M., Tong, R.: Content-aware copying and pasting in images. *The Visual Computer* 26, 721–729 (2010)
7. Sun, J., Jia, J., Tang, C.-K., Shum, H.-Y.: Poisson matting. *Proceedings of ACM SIGGRAPH* 23(3), 315–321 (2004)
8. Levin, A., Lischinski, D., Weiss, Y.: A closed form solution to natural image matting. In: *Proceedings of IEEE CVPR* (2006)
9. Rother, C., Kolmogorov, V., Blake, A.: Grabcut - interactive foreground extraction using iterated graph cut. *Proceedings of ACM SIGGRAPH* 23(3), 309–314 (2004)
10. Agarwala, A.: Efficient gradient-domain compositing using quadrees. *ACM Trans. Graph.* 26(3), 94:1–94:6 (2007)
11. Kazhdan, M., Hoppe, H.: Streaming multigrid for gradient-domain operations on large images. *ACM Trans. Graph.* 27(3), 21:1–21:10 (2008)
12. Bolz, J., Farmer, I., Grinspun, E., Schroder, P.: Sparse matrix solvers on the GPU: conjugate gradients and multigrid. *ACM Trans. Graph.* 22(3), 917–924 (2003)
13. Tao, M.W., Johnson, M.K., Paris, S.: Error-Tolerant Image Compositing. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part I*. LNCS, vol. 6311, pp. 31–44. Springer, Heidelberg (2010)
14. Floater, M.S.: Mean value coordinates. *CAGD* 20(1), 19–27 (2003)
15. Hormann, K., Floater, M.S.: Mean value coordinates for arbitrary planar polygons. *ACM Transactions on Graphics* 25(4), 1424–1441 (2006)
16. Lipman, Y., Kopf, J., Cohen-Or, D., Levin, D.: GPU-assisted positive mean value coordinates for mesh deformations. In: *SGP 2007*, pp. 117–123 (2007)