Vis Comput (2014) 30:1169–1178
DOI 10.1007/s00371-013-0897-y

ORIGINAL ARTICLE

Image copy-and-paste with optimized gradient

Yun Zhang · Jian Ling · Xiaohong Zhang · Hao Xie

Published online: 20 December 2013

Abstract This paper presents a novel approach to image copy-and-paste with optimized gradient. We improve the traditional gradient domain cloning by creating smooth transition from source patches to target images. We first specify a source patch and select its foreground region by strokes. Next, we create a gradient transition map in the cloning area. Finally, we reconstruct the gradient of the source patch according to the gradient transition map, and propose an interpolation based method to efficiently calculate the composition results, which avoids solving a large linear system. Experimental results demonstrate the effectiveness of our method, which can produce more natural and satisfying results.

Keywords Copy-and-paste · Gradient transition map · Smooth transition · Image cloning

1 Introduction

Image copy-and-paste is an important technique that has received much attention in computer graphics. The goal is to generate a new image by selecting a patch from one image and blend it into another image. Recently, many efforts have been devoted to creating composition results with high sense of reality. In general, the method for image composition can be divided into two main categories: alpha blending and gradient domain method. For alpha blending, we first need to carefully select a foreground from one image, and then directly paste it into another image. This method is easy and efficient, which enables many popular editing tools like the quick selection in Photoshop [8]. However, alpha blending requires users to precisely extract foreground objects, which may consume a lot of time and memory. In addition, direct copy-and-paste can not change the color appearance of source patches, so that the illumination and color style might be inconsistent with target images. To solve this problem, the gradient domain method is proposed with two main advantages: (1) users only need to roughly specify the source patch, which avoids the tedious work in precise extraction of the source patch, and makes the object selection easier; (2) the color differences on the cloning boundary can be smoothly diffused into the target image, which ensures the illumination consistency between the source patch and the target image.

The gradient domain method is effective to change the color appearance of source patches. However, it does not perform well when there exits texture inconsistency between source patches and target images. As shown in Fig. 4a, this is an unnatural result by the gradient domain cloning, and the color style of the wood has been adjusted to fit the beach at sunset. However, the texture near the cloning boundary is suddenly changed from the source patch to the target image. (c) is a result by our method, which is more natural by considering the texture transition. The reason for this unnatural result is obvious: the essence of the gradient domain method is to preserve the gradient of source patches, while ensuring that the colors on the cloning boundary are determined by target images. In general, the gradient domain method is more effective for image cloning with similar textures in source patches and target images. Actually, even the source and target textures are similar near the cloning boundary, the gradient domain method may produce unnatural results with-

Y. Zhang (✉) · J. Ling · X. Zhang
Zhejiang University of Media and Communications, Hangzhou 310018, China
e-mail: zhangyun_zju@zju.edu.cn

H. Xie
The State Key Lab of CAD&CG, Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027, China

out dealing with the transition of different textures. To avoid the smudging and discoloration, alpha matte is introduced to eliminate artifacts [11,28]. However, the results might be unnatural without smooth transition from target images to source patches, and it is not easy to get precise alpha mattes.

In this paper, we propose a novel method to improve the traditional gradient domain cloning, which aims to deal with texture inconsistency between source patches and target images. Our main idea is to reconstruct an optimized gradient in the cloning area, which can be applied to creating natural transition from source to target textures. To obtain the optimized gradient, we generate a gradient transition map in the cloning region, which can be calculated by solving a linear system, and requires only a few user interactions to specify the important foreground region and other regions of interest, like the shadow. With the optimized gradient, we propose an interpolation based method to calculate the final cloning result, which avoids solving a large linear system, and enables efficient image cloning.

Contributions of this paper include:

- A novel method is proposed to deal with the texture inconsistency in image cloning.
- An interpolation based method is proposed to efficiently reconstruct the cloning area with optimized gradient, which avoids solving a large linear system.

2 Related work

In image copy-and-paste, there exit two main categories: alpha blending and gradient domain method. Recent works focus on extracting more precise alpha matte, and improving the gradient domain method. Hu et al. [16] presented a survey on internet visual media processing, which provides recent progress on image cloning. Here, we briefly review recent works that closely related to our paper.

Alpha blending

In alpha blending, we need to precisely extract the alpha matte of source patches, which may contain fur, hair and other transparent objects. Most matting techniques first divide a source image into three regions: definite foreground, definite background, uncertain region (the three regions are called '*trimap*'), and focus on solving the alpha values in uncertain regions. Existing matting techniques can be classified into two main categories: *propagation* and *sampling* based methods.

For the *propagation* based method, matting problem can be considered as an interpolation of alpha values from known regions to unknown regions. Levin et al. [20] proposed a closed-form solution, which assumes that foreground and background colors are locally smooth. This method constructs a quadratic energy, and a robust foreground matte can be obtained by solving a sparse linear system. However, the closed-form matting requires constructing a sparse matting Laplacian and solving a large linear system, which may consume a large amount of time and memory. To accelerate the method based on matting Laplacian, He et al. [15] proposed fast matting using large kernel matting Laplacian. They calculated adaptive kernel size by segmenting the trimap based on k-d tree, which largely accelerates the matting calculation.

For the *sampling* based method, the matte of every pixel is estimated by a pair of foreground and background samples selected from source images. Robust matting [26] is the first non-parametric sampling method by matching the color of samples. For an unknown pixel, the foreground and background samples are selected from the neighboring pixels. However, this method is invalid when samples are not in the nearby regions. Gastal et al. [13] proposed Shared Sampling, which expands the search space by emitting rays from different directions of unknown pixels, and then selecting suitable samples which are further shared with the neighboring pixels. However, in their method, the foreground samples are always constrained in the neighborhood of unknown pixels, which may lose suitable samples. To avoid missing suitable samples, He et al. [14] proposed Global Sampling, which first applies the random searching scheme presented in PatchMatch [1] to selecting foreground and background samples globally. However, the performance of this method is degraded when the colors are not definite, e.g. when the unknown pixel is a linear combination of incorrect foreground and background pixels. In our method, precisely extracting the foreground is not necessary, and users only need to roughly specify the foreground region and the cloning boundary.

Gradient domain method

The core of gradient domain method is to construct a harmonic membrane to diffuse the differences on the cloning boundary into the cloning area, which can make the illumination and color style of the source patch consistent with the target image. Pérez et al. [22] first proposed to apply the gradient domain method to image blending. However, this method requires solving a large sparse linear system, which may consume a huge number of time and memory. In addition, they do not consider the color and texture consistency in the cloning region, which may lead to artifacts in the composition results. Recently, a lot of works focus on improving the performance of gradient domain method. Jia et al. [17] proposed to optimize the cloning boundary, which avoids unnatural results caused by inconsistent textures and colors near the cloning boundary. Lalonde et al. [19] proposed Photo clip art. In their system, a database is set up for the segmented foreground objects, and suitable foregrounds are selected according to the light condition, camera position, et al., which are estimated from the target images. However, it is hard to estimate the scene information from 2D images, like the light condition. Yang et al. [30] proposed a variational model to consider both gradient and color constraints, which allows users to control the artifacts caused by gradient domain fusion. However, their method is computational-intensive, which is far from practical. Chen et al. [4,5] further improved the gradient domain cloning by introducing hybrid boundary. In their method, the cloning region is categorized according to the color and texture consistency, and the final composition results are calculated by solving a Poisson equation which satisfies the hybrid boundary condition. Bai et al. [3] presented a new tool to preserve or freely edit the object appearance by introducing new energy terms, which are integrated into the Poisson image editing framework.

To accelerate the gradient domain cloning, Farbman et al. [12] proposed mean-value coordinates (MVC). Their main contribution is to efficiently construct a harmonic membrane based on MVC, which avoids solving a large linear system. In addition, their method is advantageous in terms of speed, ease of implementation, small memory footage, parallelizability, thus can be used to blend high resolution images. Once proposed, the MVC-based method received much attention. Ding et al. [10] introduced the importance map to preserve the color appearance of foreground area. Du et al. [11] and Xie et al. [28] combined alpha matting and gradient domain cloning to avoid the unwanted artifacts in texture and color diffusion. Zhang et al. [32] considered global and local features of target scenes to enhance the visual effects of gradient domain cloning. Wang et al. [27] proposed harmonic coordinate for image cloning. Compared with MVC [12], their method can better preserve the gradient on the cloning region, thus can produce better interpolation for image cloning. Xue et al. [29] proposed a machine learning and statistics based method to produce cloning results with high sense of reality. However, their method is designed to change the color of 2D images, and more complex operations are needed to enhance the sense of reality, like the light effects, perspective changes. Tong et al. [25] applied the gradient domain method to stereoscopic image cloning, which can efficiently blend 2D images into stereoscopic scenes, and generate visual-pleasing results. Darabi et al. [9] proposed image melding to improve the Poisson blending by PatchMatch [1], which can generate smooth transition of color, texture, structure from one image to another, and produce more natural compositions. Recently, Bai et al. [2] proposed the intent-aware method to deal with the structure conflicts in image cloning. However, their method is only effective when the non-interest regions are similar and different from the region of interest.

3 Algorithm

3.1 Overview

Figure 1 gives an overview of our method, which consists of the following three steps:

1. *Specify foreground region*. In this step, we first roughly specify the source patch by drawing a circle, and then

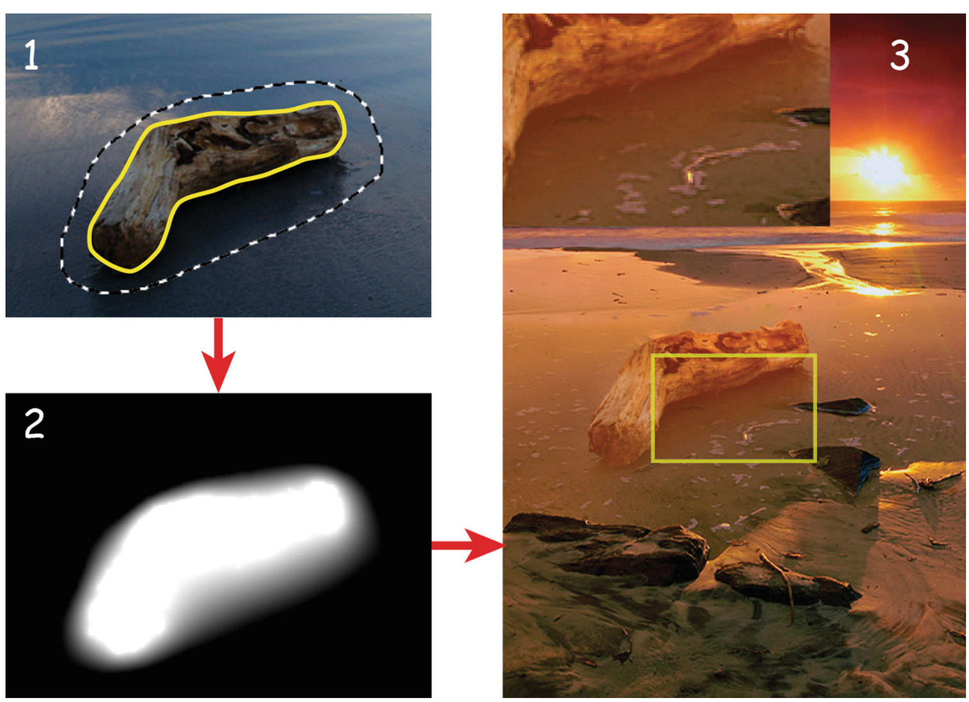


Fig. 1 Algorithm overview. (**1**) specify foreground region; (**2**) generate gradient transition map; (**3**) image cloning with optimized gradient

Fig. 2 Foreground progressive selection. *Left* result of image clustering, and the zoom-in view which shows the links between neighboring clustered regions. *Right* progressive selection result by strokes

progressively select the foreground region by strokes [(see the black–white circle and yellow strokes in Fig. 1(1)].

2. *Generate gradient transition map*. The goal of this step is to calculate a gradient transition map in the source patch. Figure 1(2) shows the gradient transition map, which is used to produce smooth transition of textures from the source patch to the target image.
3. *Image cloning with optimized gradient*. The optimized gradient of the source patch is reconstructed according to the gradient transition map. We propose an interpolation based solver to calculate the composition results, which avoids solving a large linear system. Figure 1(3) is the composition result with high sense of reality.

We first give the notations of our cloning method as follows: Let $S \subset \mathbb{R}^2, T \subset \mathbb{R}^2$ be the domains of source, target images. The intensities of these images are denoted by $g : S \to \mathbb{R}$, $f^* : T \to \mathbb{R}$ respectively. Let $\Omega \subset \mathbb{R}^2$ be the cloning region with the boundary $\partial\Omega_{\text{out}}$, and $\partial\Omega_{\text{obj}}$ be the boundary of the foreground.

3.2 Specifying foreground region

Inspired by the quick selection tool in Photoshop [8], we provide a tool for users to select foreground objects step-by-step, which makes the object extraction intuitive and efficient. Although Photoshop is very popular, it does not perform well when there exist large smooth regions. To reduce the computational cost, we first apply mean-shift [7] to cluster pixels into regions according the similarities in color and space. Then a graph is constructed based on the clustered regions. See Fig. 2, when two regions have a common boundary, an edge is added to link them. The source image is represented by a graph $\mathcal{G} = (\mathcal{P}, \mathcal{N})$, with each clustered region as a node $p \in \mathcal{P}$, and the link between adjacent regions as an edge $\langle p, q \rangle \in \mathcal{N}$. The foreground extraction can be formulated as a binary labeling of Markov Random Fields, and the goal is to specify labels $l_p \in \{0, 1\}$ for every clustered region. The result can be calculated by minimizing the following energy $E(l)$.

$$E(l) = \sum_{p \in \mathcal{P}} D_p(l_p) + \lambda \sum_{\langle p,q \rangle \in \mathcal{N}} V_{p,q}(l_p, l_q) \tag{1}$$

$E(l)$ is a linear combination of data term $D_p(l_p)$ and smooth term $V_{p,q}(l_p, l_q)$, please refer to [21] to get the detailed definition of the two terms. We consider the area of each clustered region as a constraint to enhance the role of larger region in the color model construction of foreground and background, which improves the performance of foreground selection. The energy $E(l)$ can be efficiently minimized by graph cut [18,23]. See Fig. 2, when users draw strokes on the foreground region, the selection is quickly expanded to the neighboring regions.

3.3 Generate the gradient transition map

In order to produce natural transition of different textures in the cloning region, we propose the '*gradient transition map*', which is generated by constructing a harmonic field S_g in the cloning region. As shown in Fig. 3a, $\partial\Omega_{\text{obj}}$ is the boundary of the foreground, and the texture on this boundary belongs to the source image; $\partial\Omega_{\text{out}}$ refers to the cloning boundary, and the texture on this boundary belongs to the target image; for the belt region Ω_B, which refers to the area between $\partial\Omega_{obj}$ and $\partial\Omega_{out}$, the texture is a linear combination of the source patch and the target image. S_g can be calculated by the following Laplacian equation:

$$\triangle S_g = 0 \quad \text{over} \quad \Omega_B, \quad \text{where} \quad S_g|_{\partial\Omega_{\text{obj}}} = 1 \quad \text{and} \quad S_g|_{\partial\Omega_{\text{out}}} = 0 \tag{2}$$

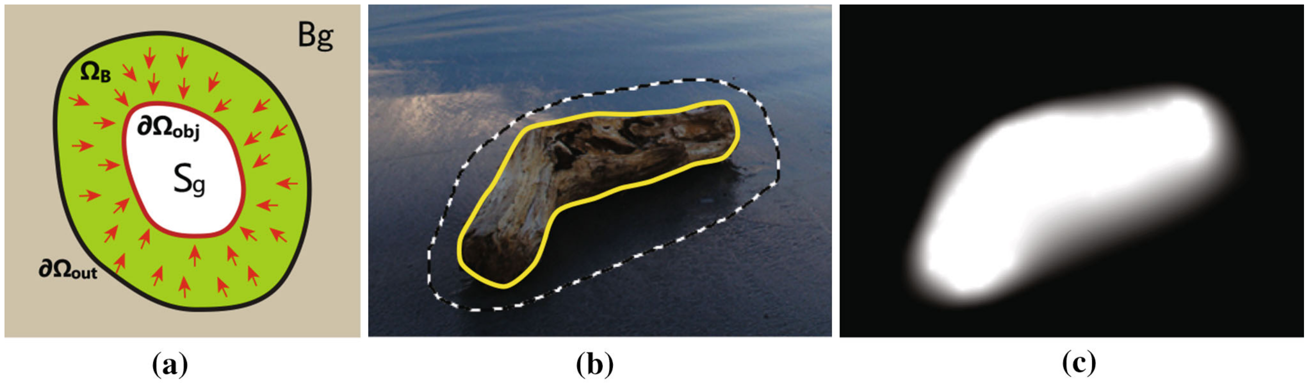


Fig. 3 Construction of gradient transition map. **a** Diagram for the calculation of gradient transition map; **b** the foreground boundary (*yellow*) and the boundary of the source patch (*black–white*); **c** visualization of the gradient transition map

In Eq. 2, we set $S_g = 1$ in the definite foreground area, $S_g = 0$ in the definite background area, and $S_g \in [0, 1]$ in the belt region Ω_B. In Fig. 3b shows the $\partial\Omega_{\text{obj}}$ (yellow boundary) and $\partial\Omega_{\text{out}}$ (black–white boundary), (c) is the visualization of the gradient transition map S_g. To reduce the computational complexity, we only need to calculate S_g in the belt region Ω_B. We utilize Green's functions [24] to efficiently solve this Laplacian equation.

3.4 Image cloning with optimized gradient

With the gradient transition map S_g, the optimized gradient of the source patch can be reconstructed by a linear combination of the foreground and background gradient, and the final cloning result f can be obtained by minimizing the following energy function, see Eq. 3.

$$\arg\min_f \iint_\Omega \|\nabla f - \mathbf{v}\|^2 \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$
$$\text{and} \quad \mathbf{v} = S_g \nabla g + (1 - S_g)\nabla f^* \tag{3}$$

Equation 3 can be minimized by solving the following Poisson equation with Dirichlet boundary conditions.

$$\triangle f = \text{div} \;\; \mathbf{v} \quad \text{with } f|_{\partial\Omega_{\text{out}}} = f^*|_{\partial\Omega_{\text{out}}} \tag{4}$$

In general, solving a Poisson equation involves solving a large linear system whose size is proportional to the number of pixels in the input data, which may consume a large number of memory and time, and makes it difficult to blend large scale images. We hope to utilize an efficient method like the MVC [12] interpolation to avoid the expensive computation.

We modify Eq. 3 by introducing the gradient transition map S_g to control the process of image blending, see Eq. 5. For the foreground area whose appearance should be affected by the target image, a larger factor ($S_g = 1$) is used. For the area in the belt region whose appearance should smoothly transit from the target to the foreground, a smaller factor ($S_g \in [0, 1]$) is used.

$$\arg\min_f \iint_\Omega \left\| \nabla \left(\frac{f - gS_g - f^*(1 - S_g)}{S_g} \right) \right\|^2$$
$$\text{with } f|_{\partial\Omega_{\text{out}}} = f^*|_{\partial\Omega_{\text{out}}} \tag{5}$$

In Eq. 5, after defining $\tilde{f} = (f - gS_g - f^*(1 - S_g))/S_g$, we can get a Laplacian equation as follows:

$$\triangle \tilde{f} = 0 \quad \text{with} \quad \tilde{f}|_{\partial\Omega_{\text{out}}} = f^*|_{\partial\Omega_{\text{out}}} \tag{6}$$

$\tilde{f}$ can be efficiently calculated by the MVC [12] interpolation, which calculates a membrane to smoothly diffuse the differences on the cloning boundary to the cloning region.

$$\tilde{f}(\mathbf{x}) = \sum_{i=0}^{m-1} \lambda_i(\mathbf{x})(f^* - g)(\mathbf{p}_i) \tag{7}$$

where λ_i is the mean value coordinates (MVC) for every point $\mathbf{x}$ in the cloning region, please refer to [12] for detailed definition. The final result f can be obtained by

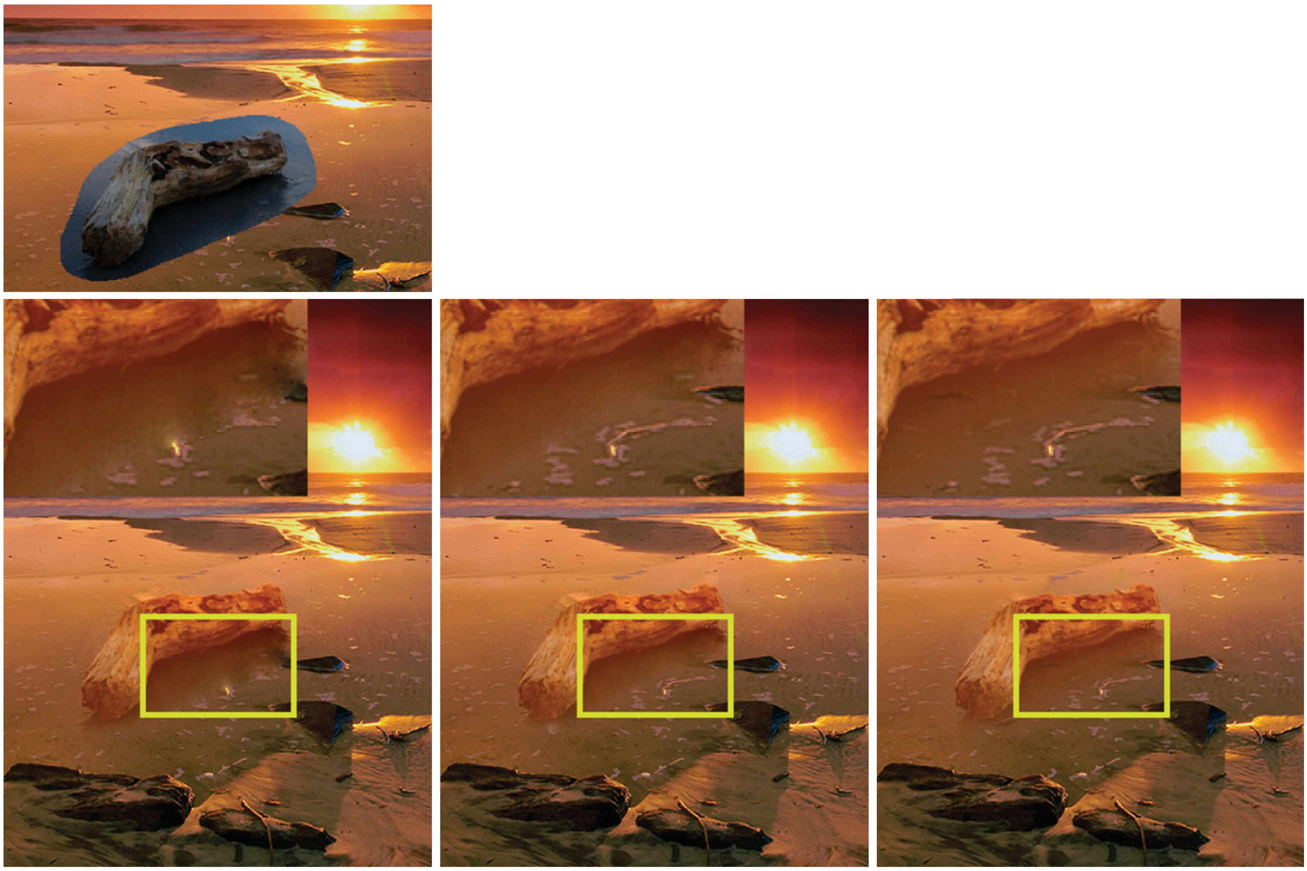
$$f = S_g(\tilde{f} + g) + (1 - S_g)f^* \tag{8}$$

See Fig. 4c, the result by our method is more satisfying than other gradient domain methods.

4 Results and performance

4.1 Results

We now show several cloning results to demonstrate the effectiveness of our method. Figure 4 shows the cloning result based on the optimized gradient, and comparisons with other methods. The first row shows the source patch and the target image, and the second row shows the results by different methods: (a) is the result by MVC [12] cloning, which is unnatural due to the texture smudging. (b) is the result by



(a) **(b)** **(c)**

Fig. 4 Results and comparisons. **a** *First row* is the source patch and the target image, *second row* is the result by MVC [12]; **b** result by Drag-and-drop [17]; **c** result by our method. The zoom-in view shows the comparison of the three methods

Drag-and-drop [17]. Although the cloning boundary is optimized to avoid the artifacts caused by inconsistent textures, there still exists unnatural transition of different textures. (c) is the result by our method, which is more natural than (a) and (b) after considering the smooth transition from the target image to the source patch. The zoom-in view shows the advantages of our method, and we find that the smooth transition from the beach at sunset to the shadow of wood greatly enhances the sense of reality.

Compared with MVC [12], which is sensitive to different textures, our method can produce natural transition of different textures, and enhance the sense of reality. Figure 5 shows another example and comparison with the gradient domain method. (a) shows the source patches and the target images; (b) gives results by the MVC [12] cloning. The zoom-in view shows the artifacts caused by inconsistent color and texture, like the region near the wood and water. (c) gives results by our method, which remove the artifacts by inconsistent textures, as shown in the zoom-in view. Figure 6 gives more results. The first column shows results by MVC [12] cloning (up-left corner gives the source patches); the second column shows results by content-aware copying and pasting [10]; the last column shows results by our method, which are more satisfying.

Figure 7 shows an example which contains shadows in the source patch. We hope that the source patch can blend into the target image seamlessly, and the shadow of the source patch can naturally cast on the target image which contains the texture of desert. (a) shows the source patch and the target image; (b) shows the result by photo-clip-art [19], which is obviously unnatural; (c) is the result by MVC [12] cloning, and the result is not visual-pleasing because of the inconsistent textures. To generate more satisfying result, we need to produce more effective gradient transition map, which not only produces natural transition of different textures, but pays attention to the shadow region. As shown in the up-left corner of (a), the black–white boundary is the cloning boundary, and the yellow boundary roughly specifies the shadows. In the shadow region, the gradient of the source patch is set to be that of the target image. (d) is the result by our method, which naturally blends the shadow into the target image, and generates a more satisfying result. The zoom-in view shows the advantages of our method.

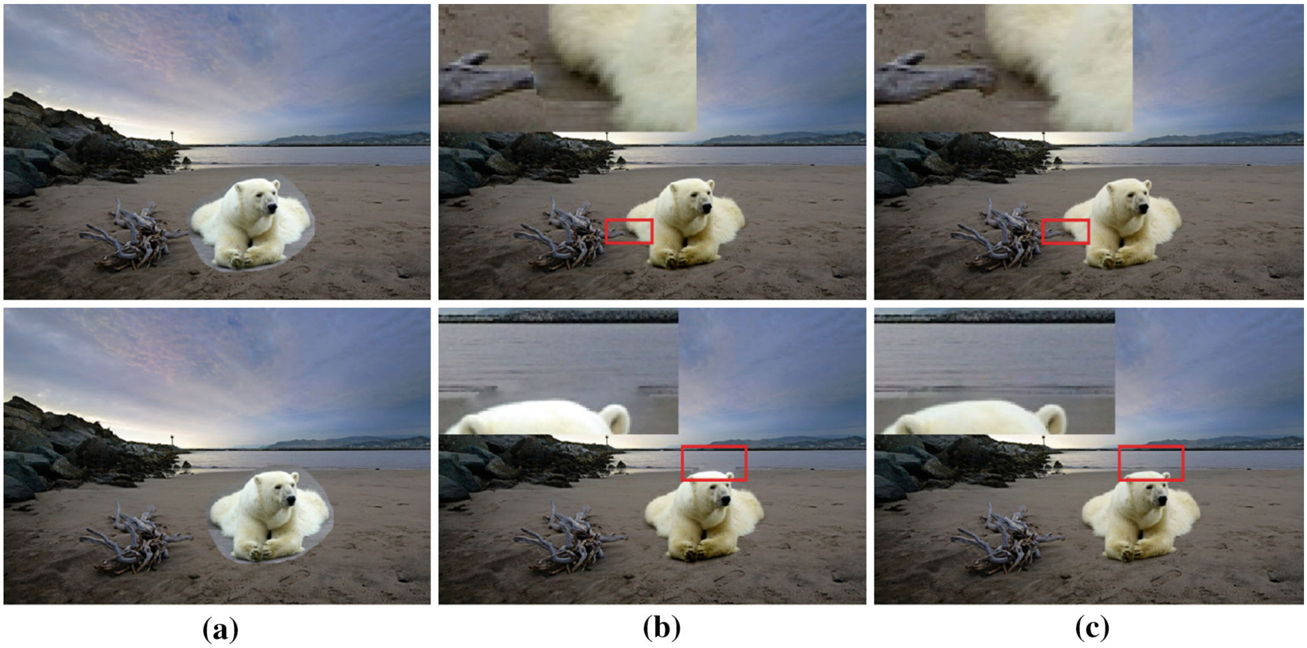


Fig. 5 Another example and comparisons. **a** Source patches and target images; **b** results by MVC [12] cloning; **c** results by our method. The zoom-in view shows the comparison of the two methods

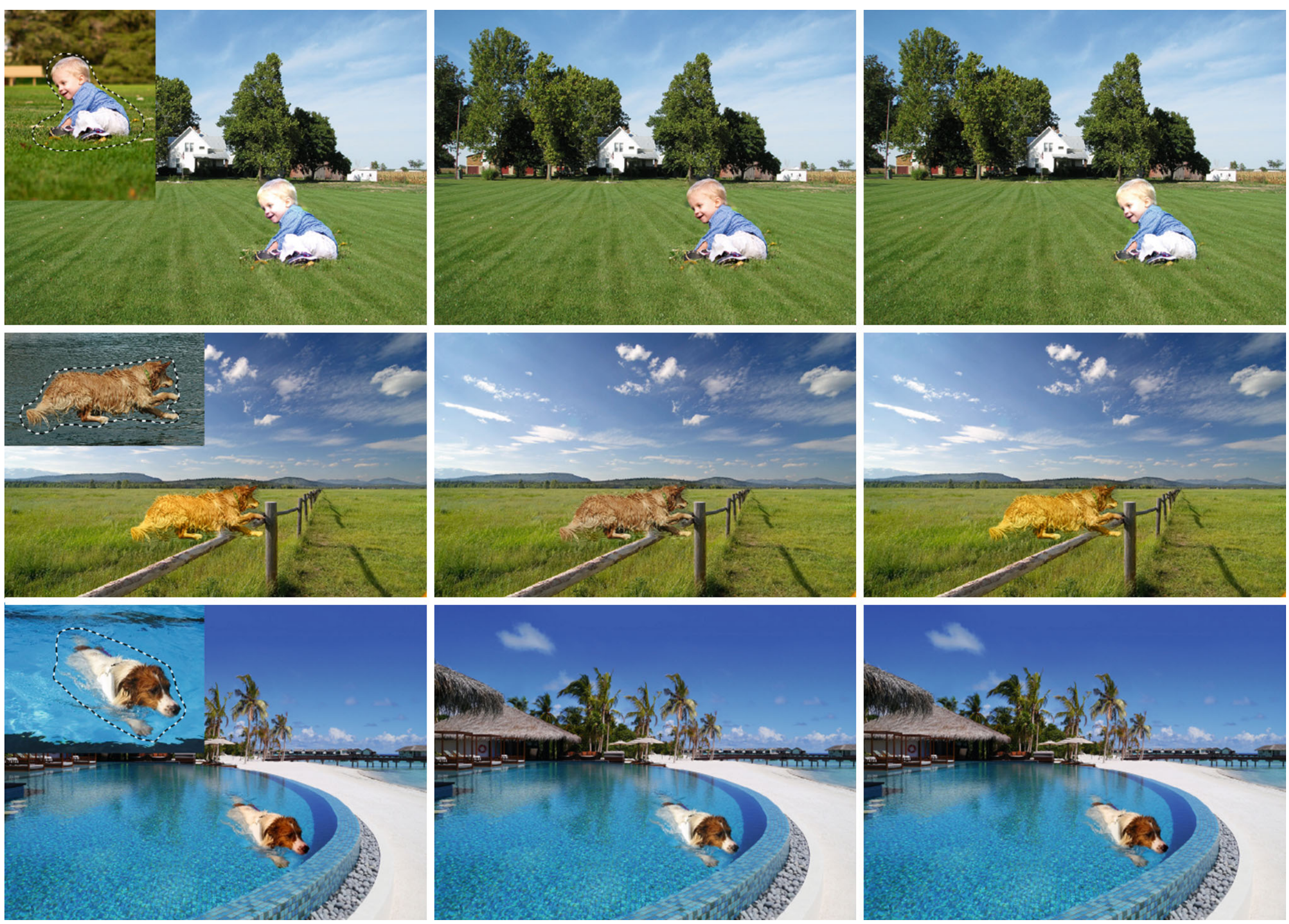


Fig. 6 More results and comparisons. *Left to right* results by MVC cloning [12], content-aware copying and pasting [10], and our method respectively

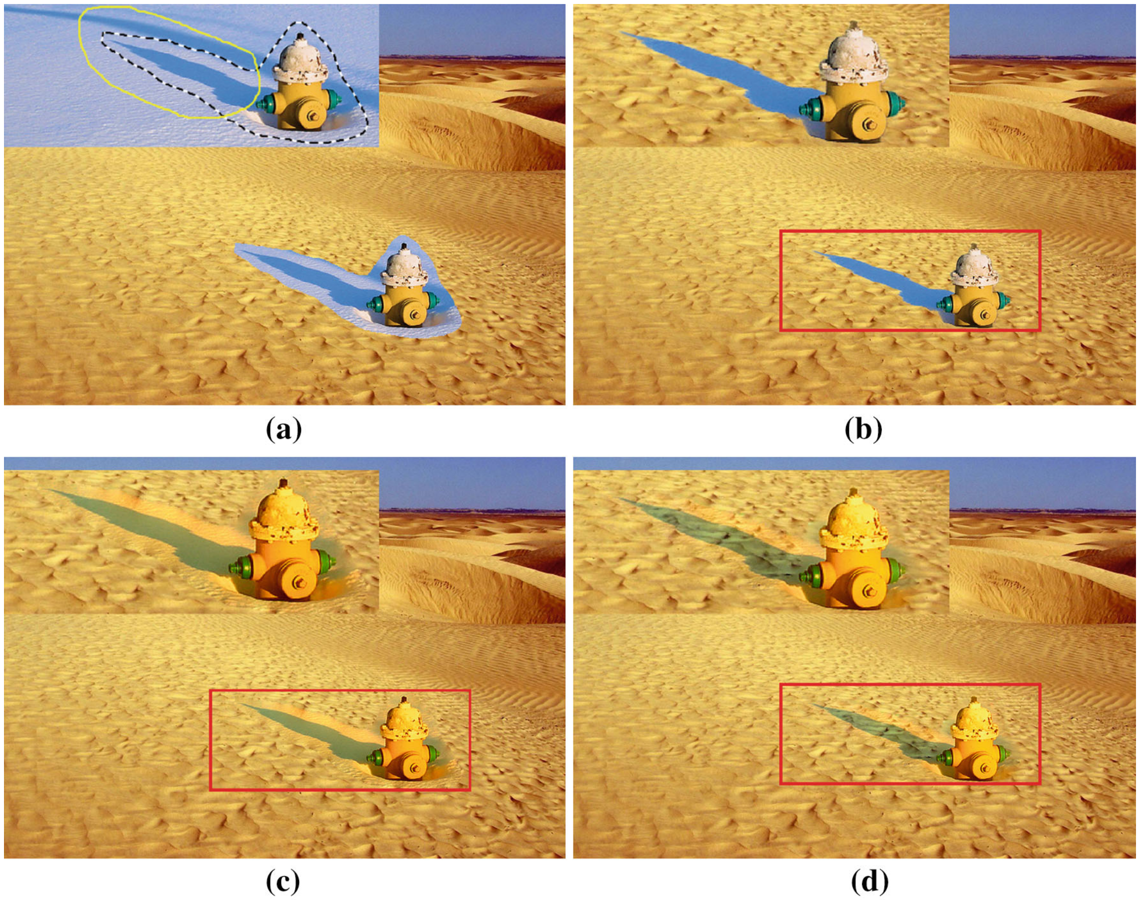


Fig. 7 An example with shadow in the source patch. **a** the source patch and the target image; **b–d** results by photo-clip-art [19], MVC cloning [12], and our method respectively. The zoom-in view shows the comparison of the three methods

4.2 Performance

Our experiments are performed on a PC with Intel Duo CPU E8400 3GHz, 4G RAM, NVIDIA GeForce 9600GT. Take Fig. 4 for example, the size of the source image is 400×450. As we paint one stroke on the foreground, the segmentation result is calculated within 20∼30 ms. The time to generate the gradient transition map is 32 ms, and the time to calculate the cloning result by solving a Poisson equation is 640 ms. In our implementation, the time cost for the MVC [12] interpolation based cloning is only 18 ms (not include the calculation of MVC), and the memory cost is much less, which makes it possible to blend large scale images. Since the MVC [12] and the gradient transition map are fixed once the source patch and its foreground are specified, the calculation of them are not needed when we move the source patch in the target image, which makes our image cloning more efficient and practical. We developed a software system for image cloning, and asked 3∼5 college students with no experiences in image processing to use our system. User study shows that even non-professional users can easily select the foreground region, and produce visual-pleasing compositions.

5 Conclusions

This paper has presented a novel method for image cloning, which improves previous gradient domain methods by constructing an optimized gradient in the cloning area. We first roughly specify the source patch by drawing a circle, and select the foreground region progressively by strokes. Then, we generate a gradient transition map in the source patch. Finally, we reconstruct the optimized gradient of the source patch according to the gradient transition map, and calculate the cloning result by the MVC [12] based interpolation, which avoids solving a large linear system. Experimental results show the advantages of our method over previous gradient domain methods.

In the future, we will explore more effective method to produce gradient transition map for more satisfying results. In addition, we will further solve the problem of texture inconsistency in image cloning, and plan to combine the texture mixture [31] and image saliency [6] for better results.

Acknowledgments We thank all anonymous reviewers for their valuable comments. This work was supported by National Basic Research Program of China (No. 2011CB302205), National High-Tech Research and Development Program of China (No. 2013AA013903), Zhejiang Provincial Natural Science Foundation of China (No. LY13F020036) and (No. LY12F02032).

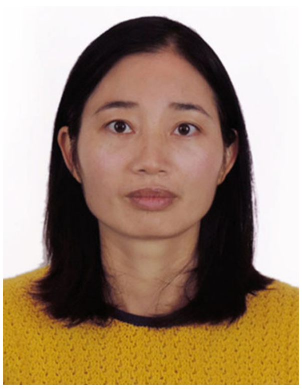
References

1. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24:1–24:11 (2009)
2. Bie, X., Wang, W., Sun, H., Huang, H., Zhang, M.: Intent-aware image cloning. Vis. Comput. **29**(6–8), 599–608 (2013)
3. Bie, X.H., Huang, H.D., Wang, W.C.: Free appearance-editing with improved poisson image cloning. J. Comput. Sci. Technol. **26**(6), 1011–1016 (2011)
4. Chen, T., Cheng, M.M., Tan, P., Shamir, A., Hu, S.M.: Sketch2photo: internet image montage. ACM Trans. Graph. **2**8(5), 124:1–124:10 (2009)
5. Chen, T., Tan, P., Ma, L.Q., Cheng, M.M., Shamir, A., Hu, S.M.: Poseshop: human image database construction and personalized content synthesis. IEEE Trans. Vis. Comput. Graph. **19**(5), 824–837 (2013)
6. Cheng, M.M., Zhang, G.X., Mitra, N.J., Huang, X., Hu, S.M.: Global contrast based salient region detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 409–416 (2011)
7. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
8. Corporation, A.: Photoshop. http://helpx.adobe.com/photoshop.html (2012)
9. Darabi, S., Shechtman, E., Barnes, C., Goldman, D.B., Sen, P.: Image melding: combining inconsistent images using patch-based synthesis. ACM Trans. Graph. **31**(4), 82:1–82:10 (2012)
10. Ding, M., Tong, R.F.: Content-aware copying and pasting in images. Vis. Comput. **26**(6–8), 721–729 (2010)
11. Du, H., Jin, X.: Object cloning using constrained mean value interpolation. Vis. Comput. **29**(3), 217–229 (2013)
12. Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., Lischinski, D.: Coordinates for instant image cloning. ACM Trans. Graph. **2**8(3), 67:1–67:9 (2009)
13. Gastal, E.S.L., Oliveira, M.M.: Shared sampling for real-time alpha matting. Comput. Graphics Forum. **29**(2), 575–584 (2010)
14. He, K., Rhemann, C., Rother, C., Tang, X., Sun, J.: A global sampling method for alpha matting. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2049–2056 (2011)
15. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2165–2172 (2010)
16. Hu, S.M., Chen, T., Xu, K., Cheng, M.M., Martin, R.R.: Internet visual media processing: a survey with graphics and vision applications. The Vis. Comput. **29**(5), 393–405 (2013)
17. Jia, J., Sun, J., Tang, C.K., Shum, H.Y.: Drag-and-drop pasting. ACM Trans. Graph. **25**(3), 631–637 (2006)
18. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 147–159 (2004)
19. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J.M., Criminisi, A.: Photo clip art. ACM Trans. Graph. **2**6(3), 3:1–3:10 (2007)
20. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 228–242 (2008)
21. Liu, J., Sun, J., Shum, H.Y.: Paint selection. ACM Trans. Graph. **28**(3), 69:1–7 (2009)
22. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans. Graph. **22**(3), 313–318 (2003)
23. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23**(3), 309–314 (2004)
24. Sun, X., Xie, G., Dong, Y., Lin, S., Xu, W., Wang, W., Tong, X., Guo, B.: Diffusion curve textures for resolution independent texture mapping. ACM Trans. Graph. **31**(4), 74 (2012)
25. Tong, R., Zhang, Y., Cheng, K.L.: Stereopasting: interactive composition in stereoscopic images. IEEE Trans. Vis. Comput. Graph. **19**(8), 1375–1385 (2013)
26. Wang, J., Cohen, M.F.: Optimized color sampling for robust matting. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
27. Wang, R., Chen, W., Pan, M., Bao, H.: Harmonic coordinates for real-time image cloning. J. Zhejiang Univ. Sci. C. **11**(9), 690–698 (2010)
28. Xie, Z., Shen, Y., Ma, L., Chen, Z.: Seamless video composition using optimized mean-value cloning. Vis. Comput. **26**(6–8), 1123–1134 (2010)
29. Xue, S., Agarwala, A., Dorsey, J., Rushmeier, H.E.: Understanding and improving the realism of image composites. ACM Trans. Graph. **31**(4), 84:1–84:10 (2012)
30. Yang, W., Zheng, J., Cai, J., Rahardja, S., Chen, C.W.: Natural and seamless image composition with color control. IEEE Tran. Image Process. **18**(11), 2584–2592 (2009)
31. Zhang, F.L., Cheng, M.M., Jia, J., Hu, S.M.: Imageadmixture: putting together dissimilar objects from groups. IEEE Trans. Vis. Comput. Graph. **18**(11), 1849–1857 (2012)
32. Zhang, Y., Tong, R.: Environment-sensitive cloning in images. Vis. Comput. **27**(6–8), 739–748 (2011)

Yun Zhang is an assistant professor of Zhejiang University of Media and Communications. He received his B.S. and M.S. from Hangzhou Dianzi University, China in 2006 and 2009, and a Ph.D. from Zhejiang University, China in 2013. His research interests include image/video editing and computer vision.

Jian Ling is an associate professor of Zhejiang University of Media and Communications. He received his B.S. and Ph.D. from Zhejiang University, China in 1990 and 2007. His research interests include image processing and computer vision.

Hao Xie is a Ph.D candidate of Zhejiang University. He received his B.S. from Xi'an Jiaotong University, China in 2005. His research interests include image/video editing and computer vision.



Xiaohong Zhang is an assistant professor of Zhejiang University of Media and Communications, China. She received her B.S. and Ph.D. in Department of Computer Science and Engineering, Zhejiang University in 2004 and 2012. Her research interests include image processing, and computer graphics, etc.