Computational Visual Media
DOI 10.1007/s41095-015-0027-z Vol. 1, No. 4, December 2015, 331–341

Research Article

MPB: A modified Poisson blending technique

Mahmoud Afifi[1](✉), **Khaled F. Hussain**[2]

Abstract Image cloning has many useful applications, such as removing unwanted objects, fixing damaged parts of images, and panorama stitching. Instead of using pixel intensities, the gradient domain is used in Poisson image editing; however, it suffers from two main problems: color bleeding and bleeding artifacts. In this paper, a modified Poisson blending (MPB) technique is presented which ensures dependency on the boundary pixels of both target and source images rather than just those of the target. The problem of bleeding artifacts is reduced. This makes the proposed technique suitable for use in video compositing as it avoids the flickering caused by bleeding artifacts. To reduce the problem of color bleeding, we use an additional alpha compositing step. Our experimental results using the proposed technique show that MPB reduces the bleeding problems and generates more natural composited images than other techniques.

Keywords image processing; image inpainting; image blending; image cloning; image enhancement

1 Introduction

Image cloning has many useful applications, such as removing unwanted objects, fixing damaged parts of images, face replacement [1], and panorama stitching. Many techniques have been presented to clone regions of a source image and insert them into a target image. Instead of using pixel intensities, the gradient domain is used in Poisson image editing [2], but this has two main problems: color bleeding and bleeding artifacts [3, 4]. Bleeding problems cause smudges in the generated image (bleeding artifacts) and also lead to the colors of the target image dominating the colors of the source image (color bleeding). This paper presents an image blending technique that generates a natural looking blend between two images. We compare our proposed technique with other image blending techniques, such as Poisson image editing [2], image melding [8], photomontage [6], image harmonization [7], and error-tolerant image compositing [3]. The results show that our modified Poisson blending (MPB) reduces the bleeding problems and generates more natural composited images than other techniques. It can be used for efficient object cloning in video without flickering.

The rest of this paper is organized as follows. In Section 2, the related work is briefly reviewed. A background to gradient domain processing is presented in Section 3. In Section 4, the details of the proposed blending technique are presented. In Section 5, experimental results and comparisons with other techniques are presented. Finally, the paper is concluded in Section 6.

2 Related work

Several image blending techniques have been invented. The Adobe Photoshop healing brush tool allows blending of two images instead of use of straightforward blending using alpha compositing. Other techniques use a multi-scale representation for cloning regions of an image into another one. Interior boundary pixels are interpolated using weighted values by mean value coordinate techniques. Patch-based synthesis is another approach to image cloning based on searching for similar patches in the image

[1] Department of Information Technology, Assiut University, Egypt. E-mail: m.afifi@aun.edu.eg (✉).

[2] Department of Computer Science, Assiut University, Egypt. E-mail: khussain@aun.edu.eg.

Manuscript received: 2015-07-28; accepted: 2015-11-08

to generate a natural composited image. On the other hand, the gradient domain is used in many techniques for cloning the selected pixels of the source image into the target image.

Alpha blending was introduced by Ray and Catmull, who used the alpha channel in image formats to express transparency in images [9]. The alpha channel is used in a direct cloning operation that preserves the original color of the source and the target images. Many approaches have been presented to extract the source object from a given image; both Grabcut for interactive foreground extraction method [10] and the paint selection method in Ref. [11], extract foreground objects using hard cutting mattes. The optimized color sampling method in Ref. [12] generates accurate soft cutting mattes. However, alpha blending gives undesirable results if the source and target images have different illumination.

The healing brush was introduced in Photoshop 7.0 to provide seamless cloning of image regions with different textures and illumination [13]. Georgiev [5] noted that the healing brush tool is based on ensuring continuity of derivatives by use of a fourth-order partial differential equation (PDE). The healing brush tool is the most popular tool used by designers for seamless image cloning.

Multi-scale representation was used by Burt and Adelson [14] to compose multiple images to generate a large mosaic image. The pyramid-based texture analysis method in Ref. [15] matches the textures of images using multi-scale pyramids. The multi-scale image harmonization approach in Ref. [7] uses multi-scale histogram matching to transfer appearance between images with different textures and noise patterns, but it gives unsatisfactory results if the source and target images have no stochastic textures.

Mean value coordinates provide another basis for image cloning, in which the interior boundary pixels are interpolated by weighted values. Coordinates for the instant image cloning method in Ref. [16] achieve good results which preserve the colors of the image from changes at real time in the case of approximate similarity textures of the pixels that close to the boundaries of both target and source images. However, this method has a problem with concave regions. The improved coordinate-based image cloning method in Ref. [18] uses an alternative sampling approach which can handle concave regions. The environment-sensitive cloning in images method in Ref. [17] uses global features of the target image to improve the blending process. Du and Jin [19] combined alpha matting and mean value techniques by using a gradient vector field as a weighted gradient of target and source images. They used mean value coordinates to solve Laplace equations without needing a large linear system.

Patch-based methods have been widely used for image cloning and image inpainting in recent years. Wexler et al. [20] used patch matching for image and video completion. The variational framework for image inpainting in Ref. [21] combines gradient-based techniques with patch-based techniques to provide robust patch matching. The image melding method in Ref. [8] clones two images with completely different structures without losing small details or artifacts in the composited image. However, image melding emphasises quality of results rather than high performance. Wu and Xu [22] used neighborhood search to give accurate patch matching and increased performance with a limited patch search space.

The gradient domain was used instead of pixel intensities in image cloning by Pérez et al. [2], solving Poisson equations with a predefined boundary condition; however, the composited image suffers from color bleeding and bleeding artifact problems. The photomontage method in Ref. [6] and drag-and-drop pasting method in Ref. [23] attempt to find an optimized boundary condition for avoiding boundary artifacts in the composited image. Content-aware copying and pasting [24] preserve the colors of the source image using a combination of the alpha matte and gradient domain approaches. The improved method for color image editing in Ref. [4] uses the gradient norm together with color information in the cloning process to avoid the color bleeding problem. Sketch2Photo [25] presents a hybrid image blending method that optimizes the boundary by specifying texture and color consistency of boundary pixels. Then, hybrid image blending acts on the source and target images on two stages: improved Poisson blending, followed by alpha blending. The intent-aware image cloning method in Ref. [26] extracts interesting objects to clone into the target image instead of working with an entire source patch, to reduce artifacts resulting from texture inconsistencies between the source patch and the target image. The image copy-and-paste with optimized gradient method in Ref. [27] reduces unnatural blending arising in images with different structures using a gradient transition map that requires the user to specify the foreground region of interest in the source image.

3 Background

Many methods use the gradient domain instead of pixel intensities in the blending process. Blending pixels in images is achieved by solving the minimization problem below:

$$\min_{f} \iint_{\Omega} \mid \nabla f \mid^{2}, \quad \text{such that } f \mid_{\partial\Omega} = f^{*} \mid_{\partial\Omega} \qquad (1)$$

where f is the unknown scalar function of the generated image, ∇f is the gradient of f, f^{*} is the scalar function of the target image, Ω is the unknown region, and $\partial\Omega$ is the boundary condition. The Poisson image editing technique [2] inserts a vector field v in Eq. (1) to blending pixels in the source image with the corresponding pixels in the target, so the blending of two images requires solution of

$$\min_{f} \iint_{\Omega} \mid \nabla f - v \mid^{2}, \quad \text{such that } f \mid_{\partial\Omega} = f^{*} \mid_{\partial\Omega} \qquad (2)$$

Minimization of Eq. (2) seamlessly clones the source image pixels into the target image using $v = \nabla g$, such that g is a scalar function of the source image. In detail, by minimizing the gradient of the generated image and the gradient of the source image over Ω, taking into account that the pixel intensity over the boundary $\partial\Omega$ of the generated image should equal the boundary pixel intensity of the target image, the specified region in the source image is cloned into the target image with a natural appearance.

Equation (2) can be represented in a discrete form with Dirichlet boundary conditions as follows:

$$\min_{f|_{\Omega}} \sum_{\langle p,q\rangle \cap \Omega \neq 0} (f_p - f_q - v_{pq})^2 \qquad (3)$$

such that $f_p = f_p^{*}$, for all $p \in \partial\Omega$, where f_p is the intensity of the current pixel in the generated image that lies in Ω, f_q is the intensity of the pixel q, one of its four-connected neighbors of the current pixel p, v_{pq} is the first-order derivative of the source image, and f_p^{*} is the intensity of the pixel that lies in the boundary $\partial\Omega$ in the blended region.

To minimize Eq. (3), Pérez et al. [2] made the Laplacian operator $\triangle f$ equals to the Laplacian operator of the source image ∇v, so that

$$\triangle f_p = \nabla v_p, \quad \text{for all } p \in \Omega \qquad (4)$$

Thus, the intensity of each pixel in Ω in the generated image is determined by the following equation:

$$\mid N_p \mid f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^{*} + \sum_{q \in N_p} v_{pq} \qquad (5)$$

where N_p is the set of the four-connected neighbors of the current pixel p. By solving Eq. (5) for each pixel using a well-known iterative solver for linear systems, the intensity values of the unknown pixels in the blended region in the generated image may be found.

The Poisson image editing technique suffers from two main problems: bleeding artifacts and color bleeding [3, 4]. The problem of the bleeding artifacts is the appearance of smudges in the boundary of the blended region in the generated image. The problem of color bleeding is the dominance of target image colors over the source image colors in the generated image: see for example Fig. 1. The reason for bleeding problems is the dependency between the boundary pixel intensities in the target image and the pixel intensities in the unknown region. As shown by Eq. (5), when a pixel p in the unknown region has a neighbor belonging to the boundary pixels in the target image, such as $P_{(1,1)}$ in the unknown region shown in Fig. 2, the only

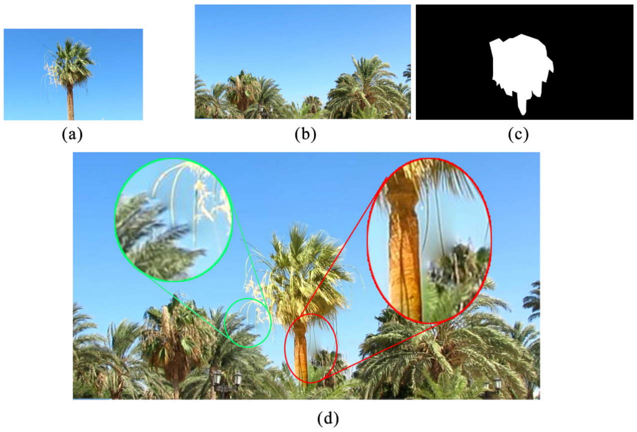

Fig. 1 Bleeding problems with Poisson image editing. (a) Source image. (b) Target image. (c) Binary mask. (d) Result of Poisson image editing. Bleeding artifacts appear in the boundary of the palm tree, and the target image colors overwhelm the source image colors (color bleeding).

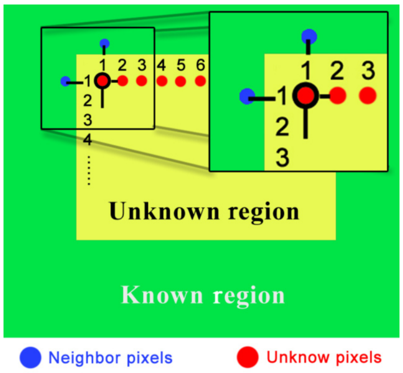

Fig. 2 Bleeding problems in Poisson image editing arise because of the dependency on the boundary pixels in the target image.

known intensity value in the equation is the pixel intensity of the target image f_q^*. This dependency makes the target image colors dominate the colors of the integrated region of the source image. In addition, this dependency leads to the appearance of smudges in the boundary of the blended region in the generated image.

4 Modified Poisson blending

As noted, the bleeding problems occur because of the sole dependency on the intensity of the boundary pixels in the target image. Our *modified Poisson blending* (MPB) technique reduces the problem of the bleeding artifacts by ensuring a more balanced dependency on the boundary pixels in the target and the source images. The problem of color bleeding is reduced by using alpha compositing between two images generated by the MPB process. Firstly, MPB inverts the blending area: it blends pixels in the target image specified by the complement of the binary mask with those pixels in the source image specified by the complement of the mask. The first step creates a composited image I_1 which is used as a source image in the second step of the MPB, which performs Poisson blending as usual, using the uninverted mask, to generate a second image. Finally, MPB composits the two images generated in these two steps: see Fig. 3.

4.1 Mathematical model

To interpolate pixels over the boundary in the composited image, the following function minimization is performed:

$$\min_{f|_{\Omega}} \sum_{\langle p,q\rangle \cap \Omega \neq 0} (f_p - f_q - w_{pq})^2 \tag{6}$$

such that $f_p = f_{t_p}$ for all $p \in \partial\Omega$, where f_t is a scalar function of the target image H and w is the divergence of the image I_1, given by the following minimization problem:

$$\min_{f_2|_{\Omega'}} \sum_{\langle p,q\rangle \cap \Omega' \neq 0} (f_{2p} - f_{2q} - t_{pq})^2 \tag{7}$$

such that $f_{2p} = f_{s_p}$ for all $p \in \partial\Omega'$, where f_s is a scalar function of image I_1, t is the divergence of the target image H, intensities of pixels in the source image O are represented by the scalar function f_s, and Ω' is the inverse of the unknown region specified by the given binary mask.

According to Pérez et al. [2], if the Laplacian of the generated image over the unknown region is set equal to the Laplacian of the integrated region, taking into account that the intensity of each boundary pixel in the generated image over the unknown region is equal to the intensity of the corresponding boundary pixel in the original image, the integrated region is thereby blended in a natural way with the original image. Thus, to minimize Eq. (6), the following equation is used:

$$\triangle f_p = \nabla w_p, \text{ for all } p \in \Omega \tag{8}$$

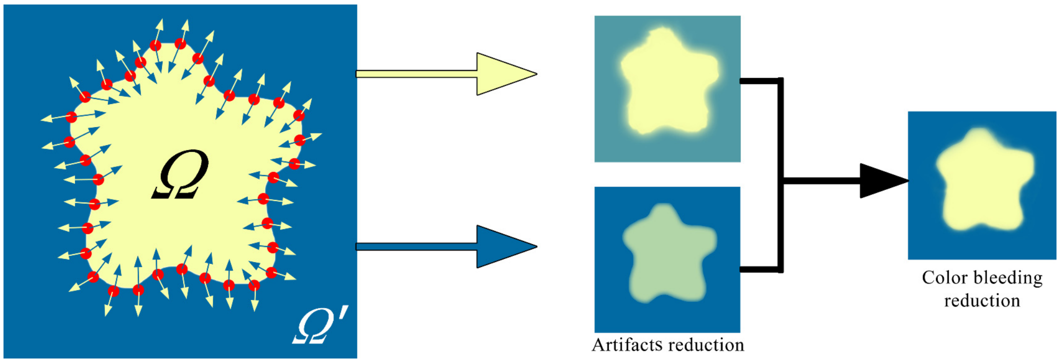

Fig. 3 MPB pipleine, reducing artifacts and color bleeding in the composited image.

while to minimize Eq. (7), the following equation is used:

$$\triangle\, f_{2p} = \nabla t_p, \text{ for all } p \in \Omega^{'} \tag{9}$$

Thus, the intensity of each pixel in the composited image is determined by the following equations:

$$\mid N_p \mid f_p - \sum_{q\in N_p\cap\Omega} f_q = \sum_{q\in N_p\cap\partial\Omega} f_{\mathrm{t}_q} + \sum_{q\in N_p} w_{pq} \tag{10}$$

$$\mid N_p \mid f_{2p} - \sum_{q\in N_p\cap\Omega'} f_{2q} = \sum_{q\in N_p\cap\partial\Omega^{'}} f_{\mathrm{s}_q} + \sum_{q\in N_p} t_{pq} \tag{11}$$

where N is the current pixel's set of four-connected neighbors. Solving these equations using Gaussian elimination gives the intensities of the pixels in the unknown region $\Omega^{'}$ of image I_1. The Laplacian of the scalar function of the image I_1 is used in Eq. (10) to find the intensities of the composited image I_2 within the unknown region Ω.

As shown in Eqs. (10) and (11), we use a balanced dependency on f_{s} and f_{t} that represents the intensities of boundary pixels in the source and target images respectively. The usage of boundary pixels of both the target and source images reduces the artifacts that appear in the composited image I_2.

To address the problem of color bleeding, we use the generated images I_1 and I_2 for prevent the source image colors from being dominated by the target image colors in the composited image I_3. We use an additional alpha blending step given by the following equation:

$$I_{3(x,y)} = (1 - M_{A(x,y)})I_{2(x,y)} + M_{A(x,y)}I_{1(x,y)} \tag{12}$$

where M_A is a binary mask generated from the target image H and the composited image I_2 by the following equation:

$$S_{(x,y)} = |I_{2(x,y)} - H_{(x,y)}| \tag{13}$$

where $H_{(x,y)}$ is the intensity of the target image H, $I_{2(x,y)}$ is the intensity of the composited image I_2, and $S_{(x,y)}$ is the intensity of the difference image S. We determine the binary mask by a filtering process given by the following equation:

$$M_{(x,y)} = \begin{cases} 0, & \text{if } S'_{(x,y)} \leqslant T \\ 1, & \text{if } S'_{(x,y)} > T \end{cases} \tag{14}$$

where S' is generated by quantizing colors in the difference image S by use of the median cut algorithm [28], and T is a predefined threshold determined experimentally. Undesired holes in M are filled by using morphological operations:

$$M = (M \oplus L) \ominus L \tag{15}$$

where L is a symmetric structuring element, $\oplus$ is the dilation operation, and $\ominus$ is the erosion operation. An averaging kernel is then used to create a new mask M':

$$M' = M * K \tag{16}$$

where K is an $h_A \times h_A$ arithmetic mean kernel used in a convolution process. The final alpha mask M_A is created by applying a symmetric Gaussian low-pass filter G of size $n_1 \times n_2$ with standard deviation σ as follows:

$$G_{(n_1,n_2)} = \frac{h_g(n_1,n_2)}{\sum_{n_1}\sum_{n_2} h_g} \tag{17}$$

where

$$h_g(n_1,n_2) = \exp\left(\frac{-(n_1^2+n_2^2)}{2\sigma^2}\right) \tag{18}$$

4.2 Implementation

We have implemented MPB in Matlab; Algorithm 1 gives details. There are three steps. In the first step, the source image is used as a target image in the original Poisson blending operation and the target image is used as a source image. The original binary mask is inverted, so the blended region is the inverse of the specified area in the original binary mask. The first step creates a composited image I_1 which is used as a source image in the second step of the MPB. We use the original target image as a target image in the Poisson blending operation and we use the original binary mask, to get the second composited image I_2. In this step, we reduce the problem of artifacts by ensuring dependency on both source and target images. In the third step, the generated images from the previous steps are blended using an alpha blending operation. We use the original target image and the generated image from the second step to determine the binary mask. The third step generates image I_3 avoiding the problem of color bleeding: see Fig. 4.

```
Algorithm 1:   MPB
Input:  Target image $H$, source image $O$, binary mask $B$,
        Boolean $bleeding$, float $h$, $n_1$, $n_2$, $\sigma$.
Output:  The composited image $F$.
$I_1 \leftarrow$ Poisson($H$, $O$, $B$);
$I_2 \leftarrow$ PoissonGray($I_1, H, B^{'}$);
if $bleeding$ then
    $S = (H - I_2) + (I_2 - H)$;
    $M_A = \text{filter}(S, h, n_1, n_2, \sigma)$;
    $I_3 = M_A \times I_1 + M_A^{'} \times I_2$;
    $F = I_3$;
else
    $F = I_2$.
```

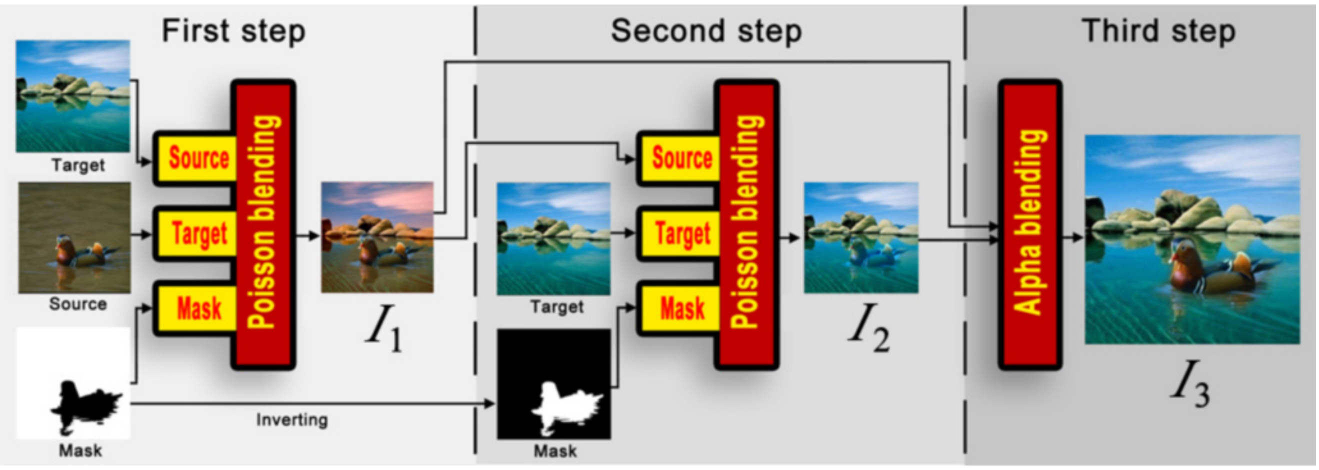

Fig. 4 MPB details.

More equitable dependency on the boundary pixels of source and target images leads to reduced artifacts: compare our results to those from Poisson image editing in Fig. 5. The problem of color bleeding caused by using the intensities of pixels of the source image are shown in Fig. 5(d) and of pixels of the target image in Fig. 5(e). By using the additional alpha compositing step, we reduce the problem of color bleeding. The alpha mask used preserves the original colors of the source image well after the compositing process. Figure 6 shows the internal steps used to create the final alpha mask M_A. By varying T, h_A, n_1, n_2, and σ, we get different versions of the composited image as shown in Fig. 7. The value of T is directly proportional to the size of mask; higher values of T generate larger masks. Higher values of h_A, n_1, n_2, and σ make the mask proportionately more blured.

5 Results and discussion

We have tested MPB on a variety of source and target images, using an Intel core$^{\mathrm{TM}}$2 quad CPU Q9550 @ 2.83 GHz machine. The test images have 1920 × 1080 pixels. For comparisons with other cloning techniques, we used images presented by those techniques.

Our MPB technique achieves good results in image cloning in comparison with other methods as shown in Fig. 8. In MPB, preserving original colors of the source image is achieved by combining alpha compositing with gradient domain techniques: see Fig. 9. Figure 10 compares MPB with the method

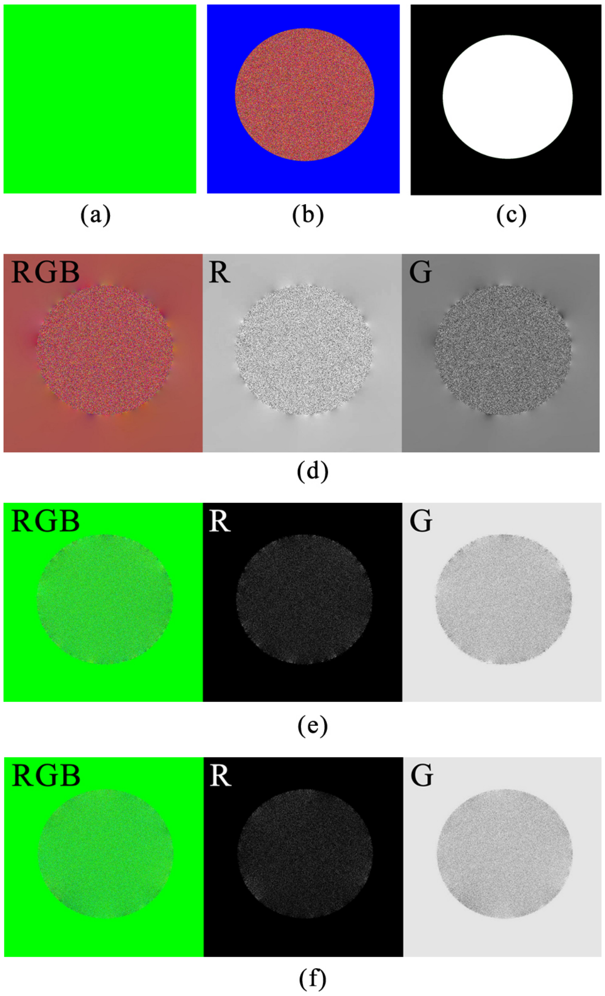

Fig. 5 MPB implementation. (a) Target image. (b) Source image. (c) Binary mask. (d) Poisson image editing result based on intensities of the source image. (e) Poisson image editing result based on intensities of the target image. (f) MPB result which depends fairly on pixel intensities in both source and target images.

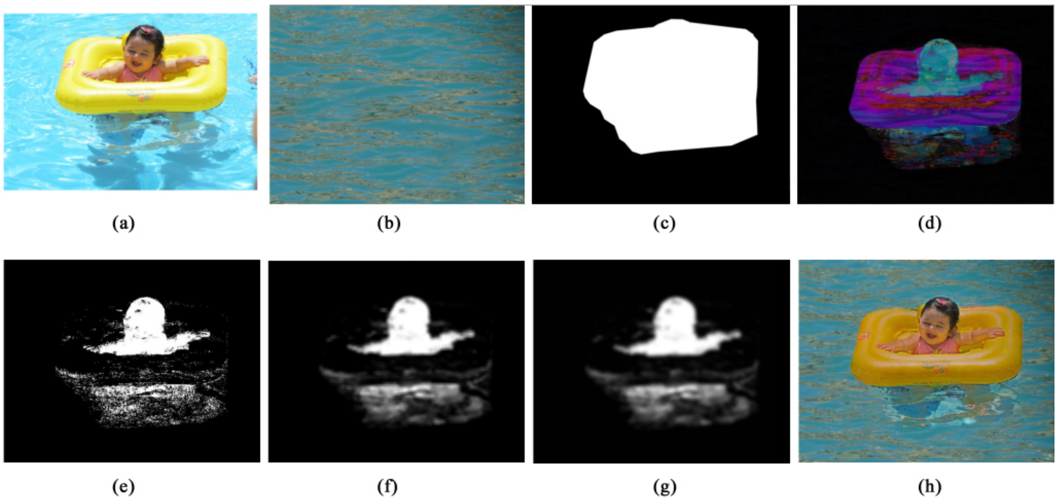

Fig. 6 The third step in the MPB technique. (a) Source image. (b) Target image. (c) Predefined mask. (d) Difference image S. (e) Binary mask M. (f) Updated binary mask M'. (g) Final mask M_A. (h) Compositing result.

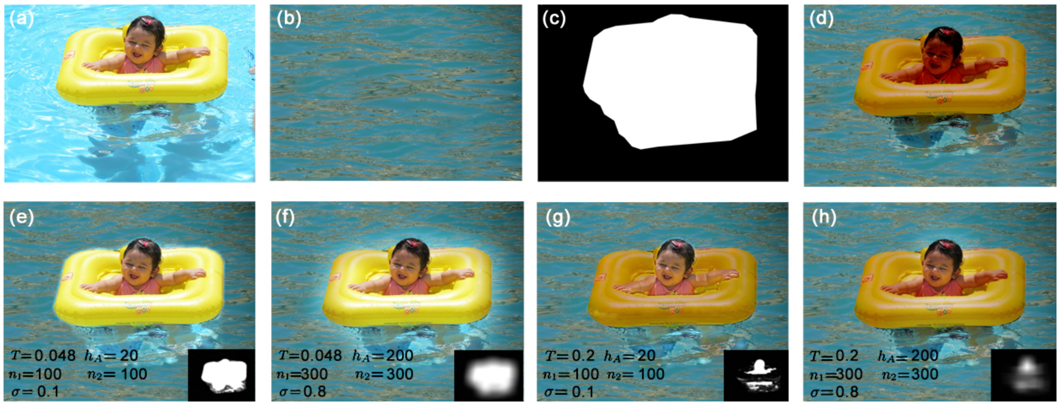

Fig. 7 MPB controls the level of preservation of the source color from color bleeding problems by using an alpha matte. (a) Source image. (b) Target image. (c) Predefined mask. (d) Result of Poisson image editing. (e)–(h) Results of our method with different values of T, h_A, n_1, n_2, and σ.

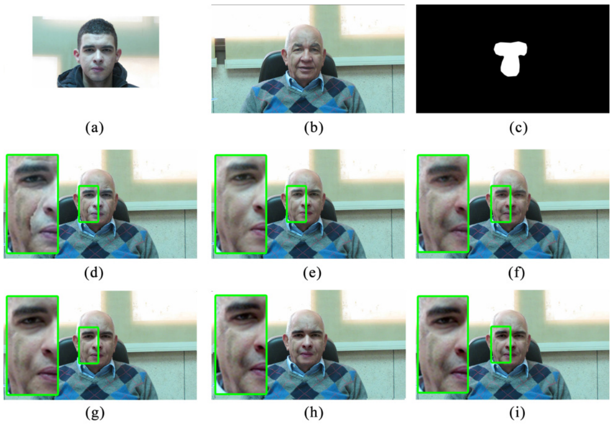

Fig. 8 Comparison between MPB and other methods. (a) Source image. (b) Target image. (c) Binary mask. (d) Result of photomontage method [6]. (e) Result of Adobe Photoshop healing brush tool [5]. (f) Result of multi-scale image harmonization approach [7]. (g) Result of image melding method [8]. (h) Result of object cloning using constrained mean value interpolation [19]. (i) Result of our MPB technique.

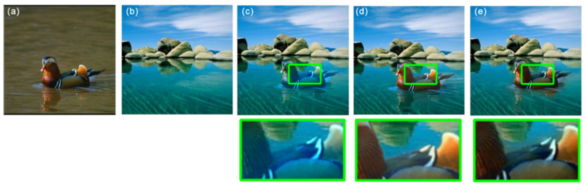

Fig. 9 Comparison between our MPB approach and image compositing using dominant patch transformations [22]. (a) Source image. (b) Target image. (c) Poisson image editing result. (d) Image compositing using dominant patch transformations result. (e) MPB result.

used in Sketch2Photo [25] using $T = 0.02$, $h_A = 5$, $n_1 = 10$, $n_2 = 10$, and $\sigma = 0.03$. It is clear that MPB reduces color bleeding and bleeding artifacts arising in Poisson blending [2]. However, in the Sketch2Photo method the colors of the blended region are merged with the new background, which does not happen in MPB as it focuses on preserving

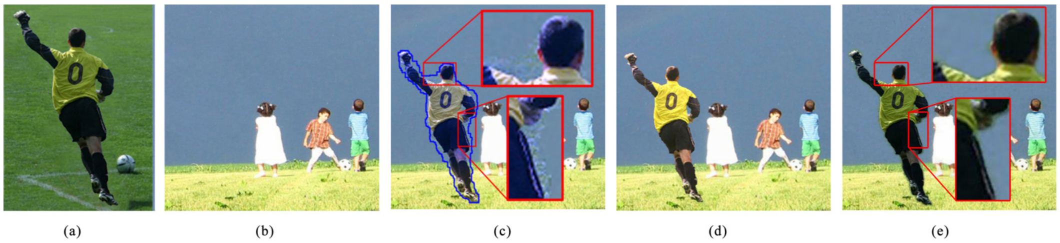

Fig. 10 Comparison between the MPB technique, Poisson image editing, and the Sketch2Photo method [25]. (a) Source image. (b) Target image. (c) Poisson image editing result. (d) Sketch2Photo result. (e) MPB result.

the source image colors without change. Doing so is useful in many cases, such as the example shown in Fig. 11, where MPB prevents the color of the eyes from changing, using $T = 0.2$, $h_A = 20$, $n_1 = 100$, $n_2 = 100$, and $\sigma = 0.1$. Compared to the content-aware copying and pasting method in Ref. [24], the third step of MPB preserves the original colors of the source image more effectively: see Fig. 12.

The MPB technique reduces the problems of Poisson image editing by using a more equitable dependency on target and source boundary pixels for seamless cloning: see Fig. 13. Compared to other methods, MPB reduces artifacts as shown in Figs. 14–16.

Table 1 compares the time required to blend two images using our MPB technique and other techniques, for different sizes of target and source images. Figure 17 shows the relation between the number of blended pixels of the source image and the time taken for Poisson image editing [2] and our MPB technique. MPB takes approximately the same time, whether the source image is large or small, and is similar to that taken by Poisson image editing for large source images. This is because of the small size of the unknown region in the second step of MPB. The time complexity of computing the color image gradient norm in Poisson image editing [2] is $O(nL)$, where n is the number of color channels in the image, and L is the number of pixels in the unknown region. In the MPB, the time complexity to compute the color image gradient norm is $O(nN)$, where N is the number of pixels in the image.

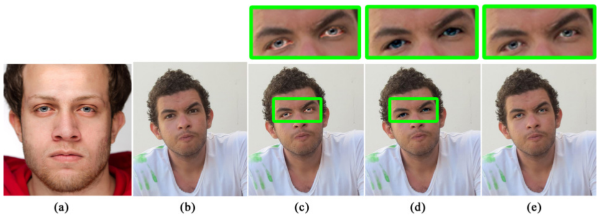

Fig. 11 Preserving color using MPB. (a) Source image. (b) Target image. (c) Eye replacement using alpha blending. (d) Poisson image editing result. (e) MPB result.

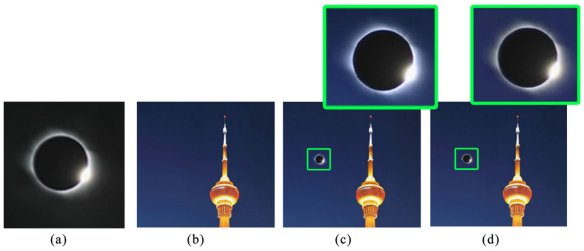

Fig. 12 Comparison between the MPB technique and the content-aware copying and pasting method in Ref. [24]. (a) Source image. (b) Target image. (c) Content-aware copying and pasting result. (d) MPB result.

By reducing the bleeding artifacts in the composited image, flickering in video is minimized. Figure 18 compares the results from MPB with those from other techniques. MPB creates a much less noticable blending in videos compared to other techniques. Inpainting is an expensive process for video because of the large number of frames. We crop the area of interest in both source and target images, and apply the blending process only to the cropped area. The performance of MPB compared to other blending techniques is shown in Fig. 19, which gives the time needed for 200 frames of the video shown in Fig. 18 (24 fps, 1920 × 1080 pixels). In this video, we track the face in the target video to align the source face with it.

6 Conclusions

A new gradient domain technique for image blending has been presented, called modified Poisson blending (MPB). It uses boundary pixels of both the source and the target images in the blending process to reduce the bleeding artifacts that occur using traditional gradient domain techniques. The MPB

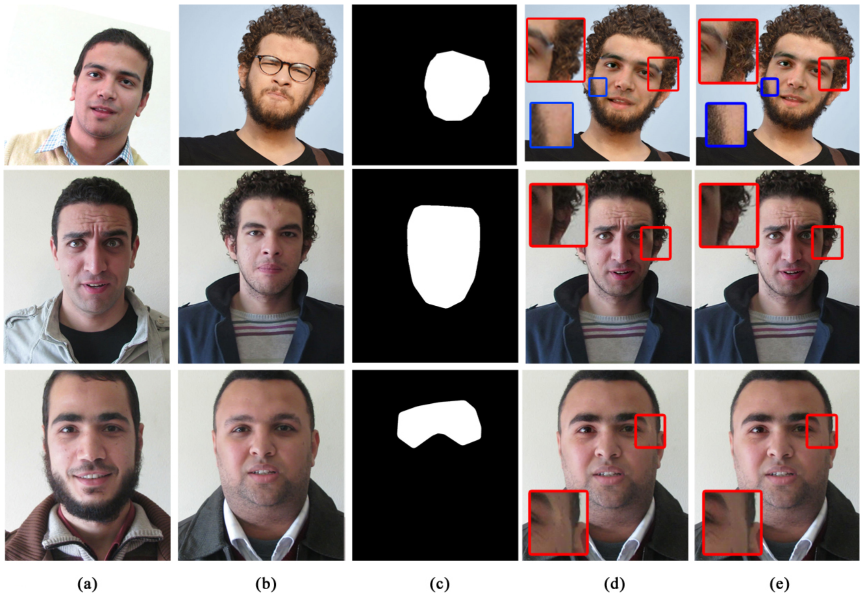

Fig. 13 Comparison between the MPB technique and Poisson image editing for face replacement. The source face (a) is replaced with the target face (b) using the binary mask (c). (d) Poisson image editing results suffer from artifacts at boundaries. (e) MPB results are better.

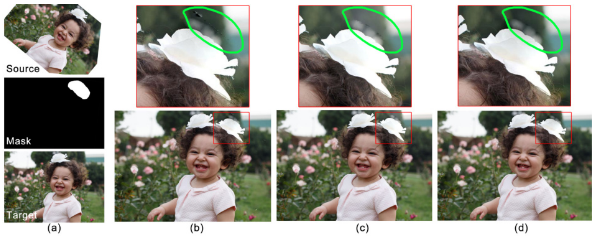

Fig. 14 Comparison between MPB and error-tolerant image compositing [3]. (a) Source, mask, and target images. (b) Poisson image editing result. (c) Error-tolerant image compositing result. (d) MPB result.

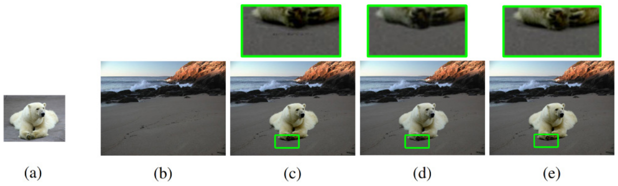

Fig. 15 Comparison between MPB and object cloning using constrained mean value interpolation [19]. (a) Source image. (b) Target image. (c) Poisson image editing result. (d) Object cloning using constrained mean value interpolation result. (e) MPB result.

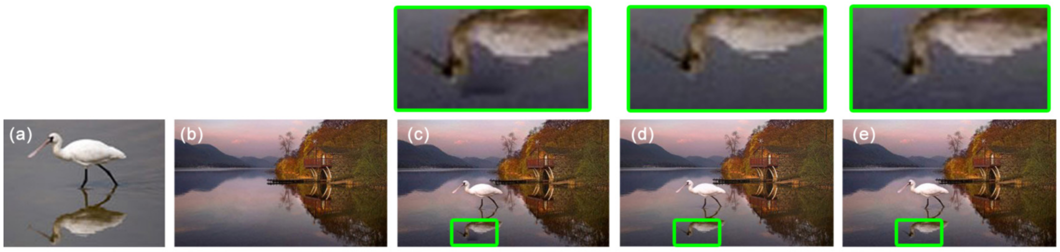

Fig. 16 Comparison between MPB and drag-and-drop pasting [23]. (a) Source image. (b) Target image. (c) Poisson image editing result. (d) Drag-and-drop pasting result. (e) MPB result.

creates two versions of blended images which are used to reduce the problem of color bleeding that occurs because of the dependency on the boundary pixels of the target image only in traditional methods. Our results show that MPB reduces bleeding problems that lead to undesirable results in many cases. It generates much better blending results in video inpainting in comparison to other image blending techniques.

Acknowledgements

The authors thank Prof. Nasser Afify, Amr Hassan, Abd El-rahman Kamel, Mohammed Bakly, Abd El-rahman Hosny, and Malika M. Afifi for their cooperation for allowing use of their footage in the experimental results.

Table 1 Performance comparison for our MPB technique, Poisson image editing [2], image melding [8], and an error-tolerant method [3]. The size of the source image refers to the number of pixels in the source image blended with corresponding pixels in the target image. Small source images (S) are 20% of the size of the target images, while medium source images (M) and large source images (L) are 40% and 80% resepctively. Time is given in second for each technique

Target image	1280 × 720 pixels			640 × 360 pixels			320 × 180 pixels		
Source image	S	M	L	S	M	L	S	M	L
Poisson [2]	5.11	13.08	40.02	1.1	1.89	5.94	0.31	0.56	1.79
Image melding [8]	3338.0	3313.0	3139.6	1144.2	879.8	809.7	519.3	479.9	383.8
Error-tolerant [3]	16.29	13.14	13.04	2.82	3.17	4.63	0.878	1.09	1.66
MPB	47.12	47.18	47.26	9.58	9.56	9.68	2.71	2.72	2.78

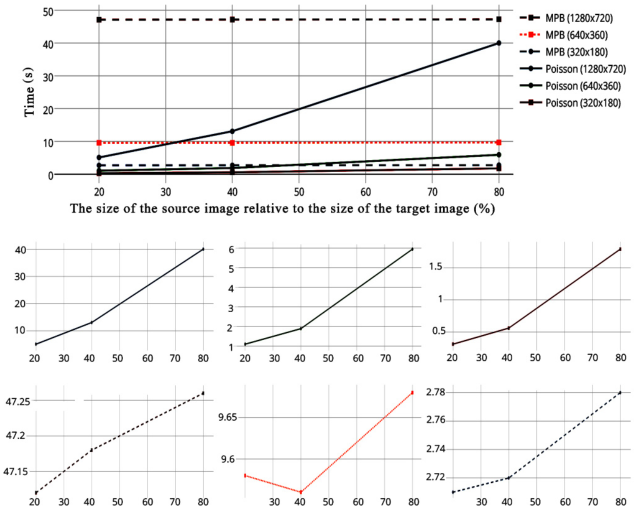

Fig. 17 Time taken by Poisson image editing and the MPB for varying numbers of blended pixels from the source image.

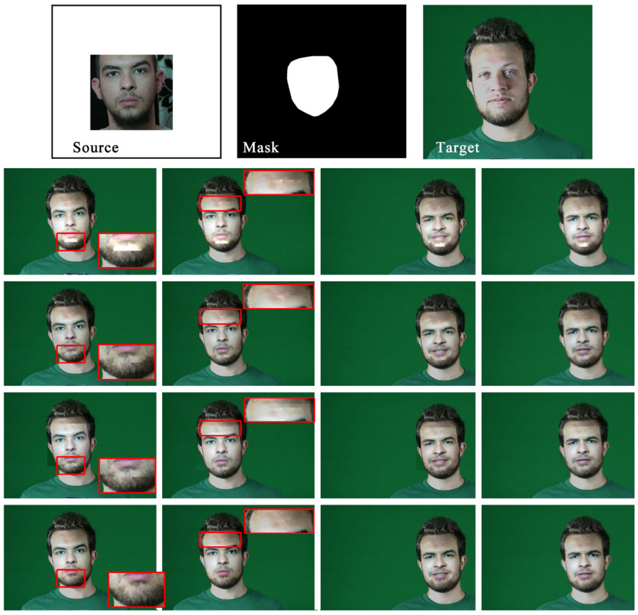

Fig. 18 Top: source, mask, and target. Second row: four frames of Poisson image editing results. Third row: image melding results. Fourth row: error-tolerant image compositing results. Bottom row: MPB results. See the accompanying video in the Electronic Supplementary Material (ESM).

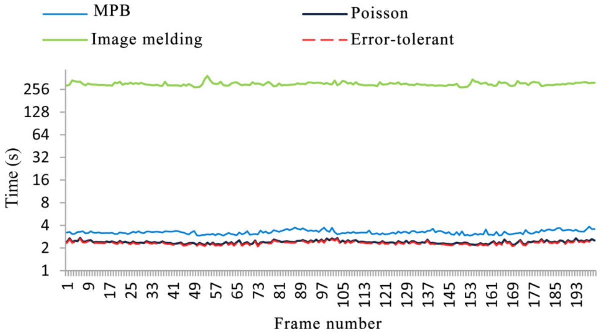

Fig. 19 Time analysis for 200 frames of a video.

medium, provided the original author(s) and the source are credited.

Electronic Supplementary Material Supplementary material is available in the online version of this article at http://dx.doi.org/10.1007/s41095-015-0027-z.

References

[1] Afifi, M.; Hussain, K. F.; Ibrahim, H. M.; Omar, N. M. Video face replacement system using a modified Poisson blending technique. In: Proceedings of 2014 International Symposium on Intelligent Signal Processing and Communication Systems, 205–210, 2014.

[2] Pérez, P.; Gangnet, P. M.; Blake, A. Poisson image editing. *ACM Transactions on Graphics* Vol. 22, No. 3, 313–318, 2003.

[3] Tao, M. W.; Johnson, M. K.; Paris, S. Error-tolerant image compositing. *International Journal of Computer Vision* Vol. 103, No. 2, 178–189, 2013.

[4] Dizdaroğlu, B.; İkibaş, C. An improved method for color image editing. *EURASIP Journal on Advances in Signal Processing* Vol. 2011, 98, 2011.

[5] Georgiev, T. Photoshop healing brush: A tool for seamless cloning. In: Proceedings of Workshop on Applications of Computer Vision, 2004. Available at http://www.tgeorgiev.net/Photoshop_Healing.pdf.

[6] Agarwala, A.; Dontcheva, M.; Agrawala, M.; Drucker, S.; Colburn, A.; Curless, B.; Salesin, D.; Cohen, M. Interactive digital photomontage. *ACM Transactions on Graphics* Vol. 23, No. 3, 294–302, 2004.

[7] Sunkavalli, K.; Johnson, M. K.; Matusik, W.; Pfister, H. Multiscale image harmonization. *ACM Transactions on Graphics* Vol. 29, No. 4, Article No. 125, 2010.

[8] Darabi, S.; Shechtman, E.; Barnes, C.; Goldman, D. B.; Sen, P. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics* Vol. 31, No. 4, Article No. 82, 2012.

[9] Smith, A. R. Alpha and the history of digital compositing. Microsoft Technical Memo 7, 1995. Available at https://www.cs.princeton.edu/courses/archive/fall00/cs426/papers/smith95c.pdf.

[10] Rother, C.; Kolmogorov, V.; Blake, A. "Grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics* Vol. 23, No. 3, 309–314, 2004.

[11] Liu, J.; Sun, J.; Shum, H.-Y. Paint selection. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 69, 2009.

[12] Wang, J.; Cohen, M. F. Optimized color sampling for robust matting. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2007.

[13] Adobe INCORP. Adobe photoshop user guide. 2002. Available at http://www.adobe.com/aboutadobe/pressroom/pressmaterials/pdfs/photoshop/ps7_rev_guide.pdf.

[14] Burt, P. J.; Adelson, E. H. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* Vol. 2, No. 4, 217–236, 1983.

[15] Heeger, D. J.; Bergen, J. R. Pyramid-based texture analysis/synthesis. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, 229–238, 1995.

[16] Farbman, Z.; Hoffer, G.; Lipman, Y.; Cohen-Or, D.; Lischinski, D. Coordinates for instant image cloning. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 67, 2009.

[17] Zhang, Y.; Tong, R. Environment-sensitive cloning in images. *The Visual Computer* Vol. 27, No. 6, 739–748, 2011.

[18] Lee, S.-Y.; Lee, I.-K. Improved coordinate-based image and video cloning algorithm. In: Proceedings of ACM SIGGRAPH ASIA 2009 Posters, Article No. 48, 2009.

[19] Du, H.; Jin, X. Object cloning using constrained mean value interpolation. *The Visual Computer*, Vol. 29, No. 3, 217–229, 2013.

[20] Wexler, Y.; Shechtman, E.; Irani, M. Space–time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 3, 463–476, 2007.

[21] Arias, P.; Facciolo, G.; Caselles, V.; Sapiro, G. A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision*, Vol. 93, No. 3, 319–347, 2011.

[22] Wu, H.; Xu, D. Image compositing using dominant patch transformations. *Computers & Graphics* Vol. 38, 277–285, 2014.

[23] Jia, J.; Sun, C.; Tang, C.-K.; Shum, H.-Y. Drag-and-drop pasting. *ACM Transactions on Graphics* Vol. 25, No. 3, 631–637, 2006.

[24] Ding, M.; Tong, R.-F. Content-aware copying and pasting in images. *The Visual Computer* Vol. 26, No. 6, 721–729, 2010.

[25] Chen, T.; Cheng, M.-M.; Tan, P.; Shamir, A.; Hu, S.-M. Sketch2Photo: Internet image montage. *ACM Transactions on Graphics* Vol. 28, No. 5, Article No. 124, 2009.

[26] Bie, X.; Wang, W.; Sun, H.; Huang, H.; Zhang, M. Intent-aware image cloning. *The Visual Computer* Vol. 29, No. 6, 599–608, 2013.

[27] Zhang, Y.; Ling, J.; Zhang, X.; Xie, H. Image copy-and-paste with optimized gradient. *The Visual Computer* Vol. 30, No. 10, 1169–1178, 2014.

[28] Heckbert, P. Color image quantization for frame buffer display. In: Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques, 297–307, 1982.

Mahmoud Afifi received his B.S. degree in information technology in 2009, and M.S. degree in information technology in 2015, both from the Faculty of Computers and Information, Assiut University, Egypt. He is currently a teaching assistant in the Department of Information Technology, Assiut University, Egypt. Since June 2015, he has been the vice chair of the ACM student chapter, Assiut. In addition, he is a CG artist. His research interests include multimedia systems, image processing, and computer vision.

Khaled F. Hussain gained his B.S. and M.S. degrees in electrical engineering both from Assiut University, Assiut, Egypt, in 1994 and 1996, respectively, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, USA, in 2001. From 2002 to 2006, he worked as a visiting assistant professor at the University of Central Florida. From 2006 to 2007, he worked as an assistant professor in the Department of Electrical Engineering, Assiut University. Since 2007, he has been with the Department of Computer Science, Assiut University, and is currently an associate professor, executive director of the Multimedia Laboratory, and head of the Department of Multimedia, Assiut University. His major research interests include computer vision, computer graphics, augmented reality, and computer animation.

Other papers from this open access journal are available free of charge from http://www.springer.com/journal/41095. To submit a manuscript, please go to https://www.editorialmanager.com/cvmj.