Color Adjustment for Seamless Cloning based on Laplacian-Membrane Modulation

Bernardo Henz, Frederico A. Limberger, Manuel M. Oliveira
Instituto de Informática – UFRGS
Porto Alegre, Brazil
{bhenz,falimberger,oliveira}@inf.ufrgs.br

Fig. 1: Compositing elements from a source image into a target one. (Left) Source and target images. (Center) Compositing generated by Poisson cloning [1]. (Right) Compositing created with our method by modulating a Laplacian membrane according to some automatically computed parameter ($w = 0.83$, for this example). Note how our method preserves the original colors while ensuring smooth transitions between the foreground and the background elements.

***Abstract*—Image compositing aims to combine elements from multiple images producing natural-looking results. The Poisson-image-editing framework can generate seamless compositions, but tends to introduce color changes in the inserted objects, which may generate unrealistic and unpleasing results. We present an efficient approach for controlling the amount of color preservation in the inserted objects, while ensuring a seamless transition between the foreground and background elements. Our technique modulates a Laplacian membrane according to some automatically computed parameter, which results in more naturally-looking compositions.**

***Keywords*- Color adjustment, Color preservation, Poisson image editing, Seamless cloning, Image composition**

I. Introduction

Image compositing tries to combine multiple images in a naturally-looking way. Poisson image editing (PIE) [1] can produce seamless compositions and has been adopted by several recent techniques [2]–[5]. However, despite its elegant formulation, PIE may introduce significant color changes in the inserted elements [6], lending to some unpleasing results (Fig. 1 (center)). We present an efficient color-adjustment technique for controlling the amount of color preservation in PIE. Our technique combines Poisson cloning and alpha blending according to some automatically-computed interpolation parameter based on perceptual color differences between the composited objects and the background. Alternatively, the user can manually control the amount of color adjustment. In both cases, it produces naturally-looking results (Fig. 1 (right)).

The flexibility of our technique allows it to benefit from fast solutions for computing Laplacian membranes [7], [8], guaranteeing quick results even when cloning large patches containing over a million pixels.

The **contributions** of this work include:

- An efficient technique to perform seamless image compositing that allows the user to control the degree of color preservation of the inserted patches (Section III);
- An automatic procedure for estimating a parameter value required for preserving the original colors of the composited elements (Section III).

II. Background

Let g be a source image patch to be composited on some target image f^* (Fig. 2). The Poisson image editing framework [1] computes a new image patch f that seamlessly integrates itself into f^* by solving the Poisson equation

$$\Delta f = \nabla \cdot v, \tag{1}$$

with Dirichlet boundary conditions $f|_{\delta\Omega} = f^*$. Here, $\delta\Omega$ is the boundary of the region Ω (the support of both g and f). In general, $v = \nabla g$, enforcing that the gradient of f be as close as possible to the gradient of g. We refer to the process of computing and pasting f into the background image f^* as *Poisson cloning* (Fig. 2).

A. Optimization Methods

The solution of the Poisson equation is obtained by solving an $N \times N$ sparse linear system, where N is the number of pixels in the region Ω. For large values of N, finding the solution of such a system becomes computationally and memory intensive. Trying to address this problem, some techniques had been proposed to efficiently solve large sparse linear systems [9], [10], while others try to reduce the size of the region Ω [11] . While they alleviate the problem, they are still not sufficiently fast for use in interactive applications.

Farbman et al. [7] describe an efficient method to compute a Laplacian membrane $\tilde{f}$ without solving a linear system. This is obtained by approximating the solution of the following Laplacian equation:

$$\Delta \tilde{f} = 0, \ \tilde{f}|_{\delta\Omega} = f^* - g \tag{2}$$

with Dirichlet boundary conditions. f is then computed as

$$f = \tilde{f} + g. \tag{3}$$

Eq. 2 calculates a smooth membrane that interpolates the difference $f^* - g$, computed at $\delta\Omega$, over the entire region Ω (Fig. 3). In a subsequent work, Farbman et al. [8] use a multiscale convolution scheme to approximate the membrane $\tilde{f}$. The main limitation of these methods is that Eqs. 1 and 2 are only equivalent if the vector field v is conservative (*i.e.*, the line integral between any two points in the vector field is independent of the chosen path). Thus, v must be the gradient of some scalar function and, therefore, these methods cannot be used to perform Poisson cloning using mixed gradients [1], for instance. Nevertheless, these optimizations are very handy when one wants to perform seamless cloning of large patches, on several images, or in video sequences.

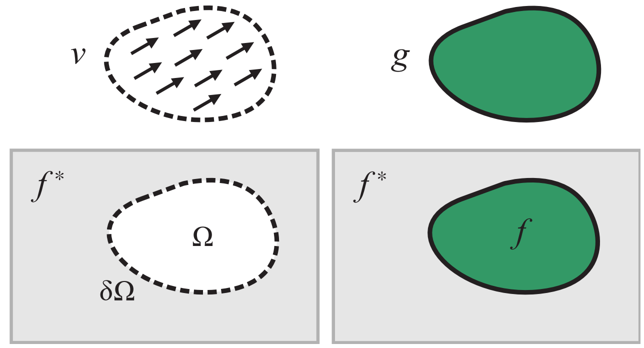

Fig. 2: Poisson cloning [1] places a source patch g, with an associated vector field v and support Ω, over a source image f^*. To produce a seamless result, f^* is composited with f (a modified version of g computed by the algorithm).

B. Color-Correction Techniques

Since Poisson image editing works in the gradient domain, it has little control over changes in color space. Thus, if the tones of the inserted patch and the background image differ significantly, the colors of the inserted elements may be severely affected. Many methods have been developed specifically to control this color problem. Dizdaroglu and Ikibas use information from each color channel to built a *mixed guided vector field* that tries to preserve the foreground color [12]. Li et al. [13] also build a new guided vector field as a weighted average between the gradient of the source image and a matting composition. Yang et al. propose a multi-resolution framework that uses a variational model in a minimization equation to generate more natural and realistic compositions [14]. Their method also modifies the guidance vector field. Guo and Sim modify the Poisson equation by adding different weights for groups of image pixels [15]. The user specifies regions that should be preserved (using markups), and the technique adjusts their weights accordingly. Wu and Xu use geodesic-distance maps for computing a color-belief measure for each pixel in the source image [16]. This measure is used as a weight in a closed-form energy function that is minimized by solving a linear system.

While all these methods achieve color preservation by solving new and modified minimization equations, or by building new guidance vector fields, the resulting systems cannot be written as Laplace equations because there is no assurance that the new vector fields will be conservative. Therefore, they cannot take advantage of fast methods that approximate the Laplace membrane [7], [8]. Our technique performs color preservation while taking advantage of these fast methods.

Some color-transfer techniques focus on matching the mean and variance values of color histograms of the source and target images [17], [18]. Lalonde and Efros proposed a method to recolor cloned image regions [19]. Recently, Xue et al. performed statistical and visual perception experiments to determine factors that influence realism on image composites, proposing an automatic method to try to improve them [20].

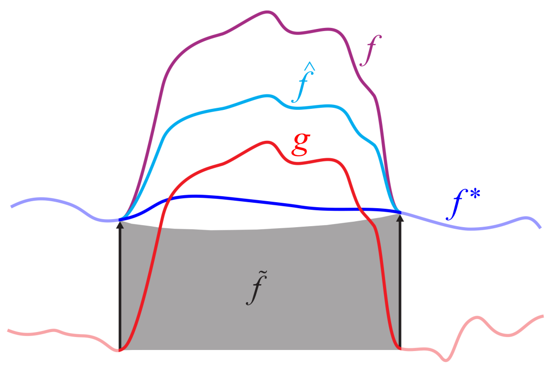

Fig. 3: Poisson cloning computes $f = \tilde{f} + g$. Thus g (red curve) is shifted by $\tilde{f}$ to f (purple curve). Our technique modulates the contribution of $\tilde{f}$, allowing the user to obtain results that range from alpha blending (red curve) to Poisson cloning (purple curve). The cyan curve shows a typical intermediate result produced by our technique. Note how its borders nicely blend with the target image f^*.

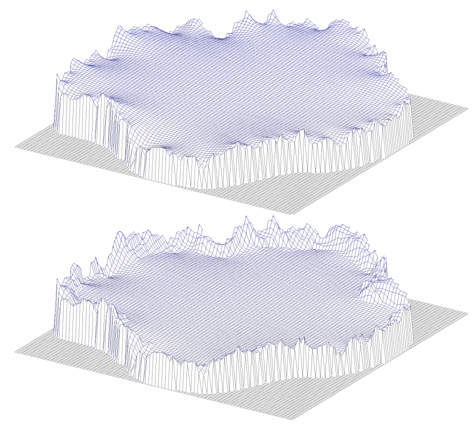

Fig. 4: The Laplacian membranes. (top) $\tilde{f}$ membrane computed using [8]. (bottom) $\tilde{f}_m$ membrane generated by our color-adjustment approach. The lower values of the membrane $\tilde{f}_m$ reduce the difference between $\hat{f}$ and g.

Their method uses machine learning for choosing zones in the histograms of the source and target images for matching luminance, CCT (Correlated Color Temperature), and saturation over the images. While these methods focus on achieving realistic composites, there is no assurance of color preservation, i.e., object colors may significantly change. In addition, they do not offer control over the adjustments.

III. Our Color-Adjustment Method

According to Eq. 3, the Laplace membrane $\tilde{f}$ is the difference between the computed patch f and the original one g. Therefore, the bigger the values over the membrane, the bigger the difference between f and g. Thus, the **central idea** of our method is to *control this difference by adjusting the membrane accordingly*. Figs. 3 and 4 illustrate the concept.

Similarly to other approaches for color preservation in seamless cloning [6], [13], our method uses two masks: a pasting mask M_p (same as the mask used for Poisson image editing, and used for computing the membrane $\tilde{f}$ in Eq. (2)), and an alpha mask M_α for the object(s) we would like to preserve colors. M_p includes the object plus some transition zone, and can be defined manually, using some interactive tools [21], [22], or obtained as a dilated version of M_α.

Our method modulates the Laplacian membrane using

$$\tilde{f}_m = (1 - w^* M_\alpha)\tilde{f}, \tag{4}$$

where $\tilde{f}$ is the Laplace membrane computed using [8], $\tilde{f}_m$ is the modulated membrane, and $w \in [0, 1]$ controls the modulation. Fig. 4 shows the membrane before and after the modulation (Eq. 4). The compositing is obtained as

$$\hat{f} = \tilde{f}_m + g. \tag{5}$$

Note the influence of the parameter w in Eq. 4. When $w = 0$, the equation reduces to Poisson image editing, with no color adjustment (i.e., $\tilde{f}_m = \tilde{f}$). When $w = 1$, Eqs. 4 and 5 produce an alpha blending of the patch g and the Poisson result $\tilde{f}$. $w \in [0, 1]$ controls how much of the color will be changed. Bigger values will make the colors on the inserted object(s) to take values similar to the original ones; lower values will produce results closer to Poisson image editing [1].

We now describe a simple way to automatically compute w. Note that, at the transition zones, if the colors in the target image differ too much from the colors in the source image, Poisson image editing will diffuse such difference into f. Our idea is to use a measure of this difference to compute a suitable w value. As compositing results are strictly related to human perception, we use Pele and Werman's perceptual color-difference measure [23] for this purpose.

Consider a transition zone Γ (Fig. 5 bottom left) defined as $\Gamma = M_p - M'_\alpha$, where M'_α is a binary mask obtained as

$$M'_\alpha(x, y) = \begin{cases} 1, & \text{if } M_\alpha(x, y) > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{6}$$

We evaluate the perceptual difference between the target and source images. Let c_T be the average color over Γ in the target image; likewise, let c_S be the average color over Γ in the source image. We compute $\text{diff}_{pw} = |c_T - c_S|_{pw}$, where $|..|_{pw}$ is the Pele and Werman's measure. As diff_{pw} falls in the $[0, 1]$ interval, with lower values indicating similar colors, we make $w = \text{diff}_{pw}$. We have applied this strategy to a variety of examples and found the results satisfactory.

IV. Results and Discussion

Our color-adjustment solution can be easily combined with the techniques proposed by Farbman et al. [7], [8] to compute an approximation to $\tilde{f}$. For all results shown in the paper, our method used Convolution Pyramids [8] for acceleration. The computation cost for obtaining the new membrane $\tilde{f}_m$ is linear in the number of pixels. Thus, the total cost of the compositing process is bounded by the computation of $\tilde{f}$. We now present several results obtained with our technique using different values for w. In these examples, the alpha masks

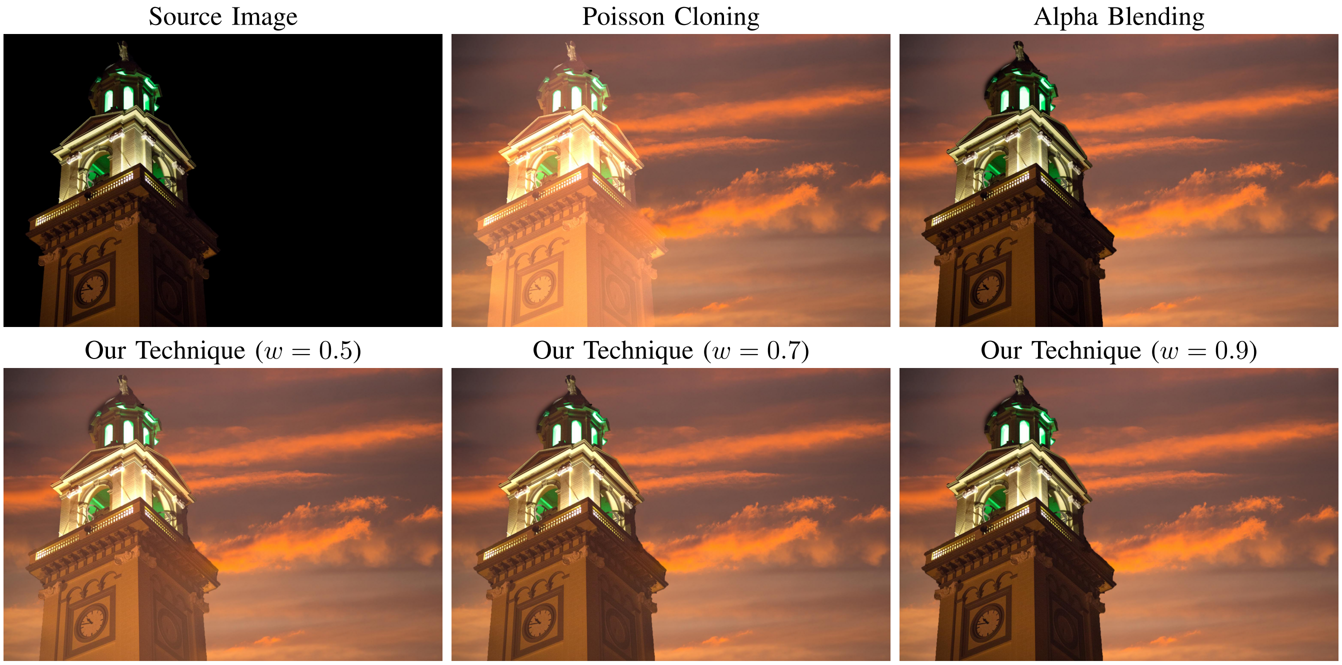

Fig. 6: Compositing a bell tower from a photograph taken at night onto a sunset background. Comparison of the results produced by conventional Poisson cloning, alpha blending, and our technique using different values for the parameter w. Poisson cloning introduces significant color shift in the bell tower, producing some unnatural result. Although alpha blend preserves the original colors of the tower, the result also does not look natural, as there is no interaction of the ambient light with the tower. By changing the value of the parameter w, our technique allows the user to choose among different results.

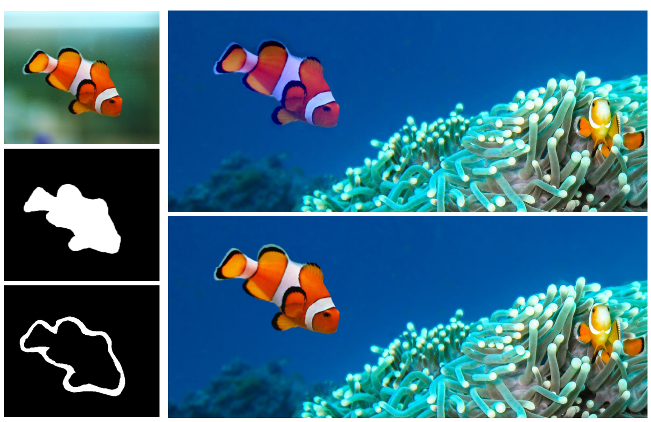

Fig. 5: Seamless cloning of a clownfish. (left) From top to bottom: source image, alpha mask, and transition zone $\Gamma = M_p - M'_\alpha$, respectively. (right) Results produced by Poisson image editing (top) and by our technique using automatically computed $w = 0.97$ (bottom).

were computed by an available MATLAB implementation of Shahrian et al.'s technique [24]. The corresponding pasting masks M_p were obtained from the alpha masks M_α, after applying binarization and dilation filters.

Fig. 5 shows the result of cloning a clownfish using Poisson image editing (top) and our method with an automatically computed $w = 0.97$ (bottom). Our result properly preserves the original fish colors.

The example in Fig. 6 shows the compositing of a bell tower (top left) from a picture taken at night, into a sunset background. Note how Poisson cloning yields a significant color shift to the tower (top center). The use of alpha blending (top right) preserves its original colors, but produces some unnatural compositing. Our results (bottom row) range between these two extremes, achieving good color-preservation and smooth transitions. For these images, the values of w were manually chosen to exhibit distinct results.

Fig. 7 shows the compositing of a portrait into a background image with unusual ambient illumination. Fully preserving the original skin tones (top right) makes the person to stand out from the rest of the image, as if the subject were not originally part of the scene. Strongly changing the skin color (bottom left) tends to produce unpleasant results. Our method allows for smooth diffusion of the green color as ambient light over the subject. Fig. 8 provides another example comparing our technique to Poisson cloning and to alpha blending. It shows two planes inserted on a sunset sky. By diffusing sunset's color over the steam while controlling the colors of the planes, our technique produces more natural-looking results.

Our technique requires the specification of an alpha mask. This, however, is not a big constraint, as there are many high-quality alpha-matting techniques available [24], [25], including one that works in real time [26]. Although refined alpha masks are preferred, our technique still achieves good results when

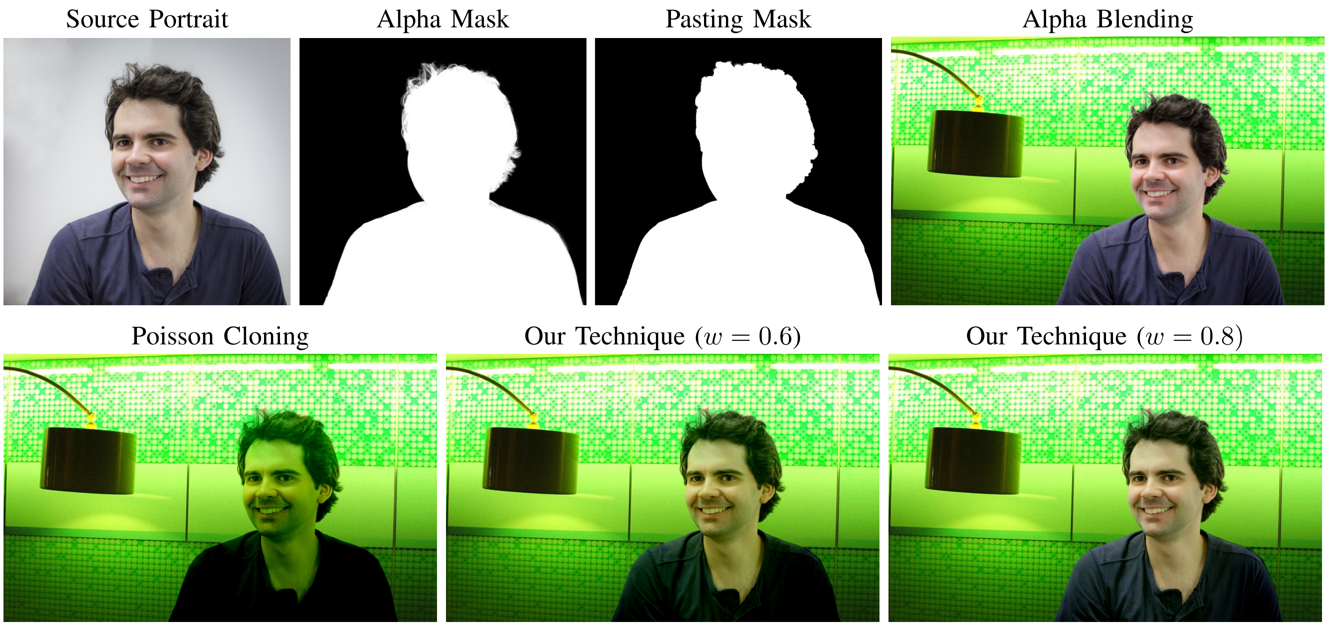

Fig. 7: Compositing a portrait on a background with unusual ambient illumination. Pure alpha blending or Poisson cloning produces unrealistic results. Our technique allows for smooth diffusion of the green color as ambient light over the subject.

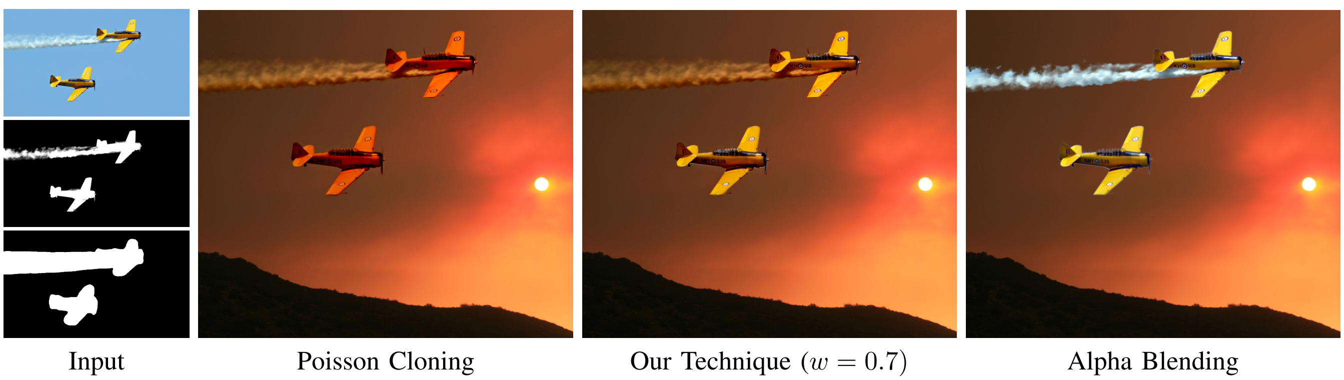

Fig. 8: Image composition of two planes into a sunset background. (first column) Input: source image, alpha mask, and pasting mask, respectively. The remaining columns show compositing results obtained using Poisson cloning, our method ($w = 0.7$), and alpha blending. Our method diffuses the background color over the inserted patches, simulating the effect of ambient illumination. Note the colors of the planes and of the steam.

using rough M_α masks. For instance, the masks in Fig. 6 were produced manually using a quick-selection tool. In fact, we show that even when using very precise alpha masks, our technique tends to produce more naturally-looking results than pure alpha blending. This is illustrated in Figs.7 and 8, where the shading of the objects inserted using pure alpha blending is not "influenced" by the lighting of the target environment.

Our color-adjustment method works over a Laplacian membrane and can handle both conservative and non-conservative vector fields. In the first case, the membrane is computed using fast seamless-cloning approaches [7], [8]. If the vector field is non-conservative, we compute the membrane as $\tilde{f} = f - g$, where f is obtained solving a linear system in the conventional way. Fig. 9 shows our color-preservation method applied with a variation of the mixed-gradient approach [1]. In this example, we have used mixed gradients only in the transition zone (region in the pasting mask not covered by the alpha mask). This solution avoids blurring the transition zone, while also preserving the gradient of the bear fur. Such a result cannot be obtained using [7], [8].

The user should be careful to not introduce textures discontinues at the boundaries of Ω. Although mixed gradients can be used for treating this problem in some cases, there is no assurance of obtaining good results in general. An alternative for these cases is to use our method with techniques that treat texture discontinuities [3], [16], [27], [28].

We tried to compare the quality of our results with other color-adjustment techniques. Unfortunately, we could not find any available implementation for these techniques. The only method we have found figures with sufficient resolution for

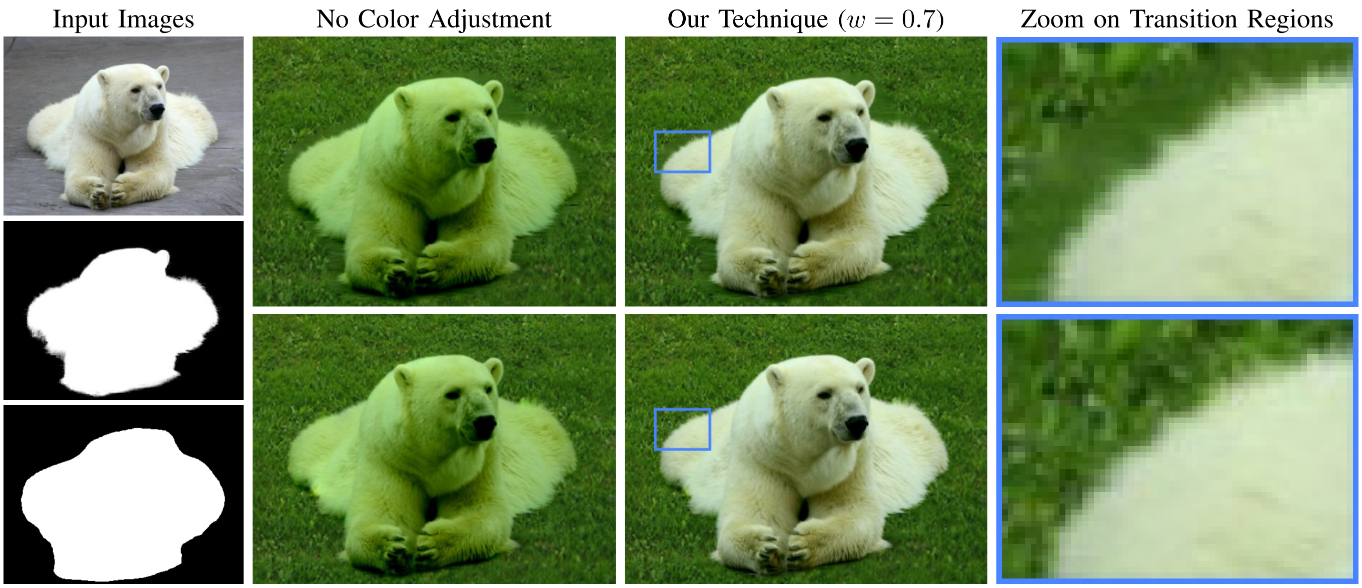

Fig. 9: Color adjustments with the use of different gradient strategies at the transition zone, obtained with our technique. Single gradient (top row) and mixed gradients (bottom row). Input images are shown on the left: source image, alpha mask, and pasting mask, respectively. Note how the mixed gradients are able to match the background texture in the transition regions.

comparison was Guo and Sim's [15]. Fig. 10 compares one of their results with ours. Note how the colors in the transition zone (window internal wall) in their result look desaturated.

Table I compares the execution times (in seconds) for different techniques: Convolution Pyramids [8] (*Conv Pyram*) using an implementation provided by the authors; our automatic computation of w and $\tilde{f}_m$ (*Color Adjust*); and Poisson image editing (*PIE*) using an implementation of the bi-conjugate gradient method. All these implementations are written in MATLAB. The total cost of our color-preservation seamless-cloning technique is given by *Conv Pyram* + *Color Adjust*. The majority of the measured time for color adjustment (over 70%) is used for automatically estimating the appropriate value for w, while the rest is used for membrane modulation. These measurements were performed on an i7 3.5GHz CPU with 16 GB of RAM. As the size of the cloning area increases, the use of PIE becomes impractical.

Our project website [29] provides a MATLAB implementation of our method, as well as images used in the paper. We encourage the readers to try their own compositions, confirming the effectiveness of our technique.

V. Conclusion

We presented an efficient technique to perform seamless image compositing that allows users to control the degree of color preservation of the inserted patches. Our technique combines Poisson cloning and alpha blending according to some automatically-computed interpolation parameter, which is obtained based on perceptual color differences between the composited objects and the background. By changing the value of the interpolation parameter, the user can interactively adjust the degree of color preservation.

TABLE I: Performance Comparison. Cloning area (in pixels) and time (in seconds) for different techniques: Convolution Pyramids [8] (*Conv Pyram*), our automatic computation of w and $\tilde{f}_m$ (*Color Adjust*), and Poisson image editing (*PIE*) using a MATLAB implementation of the bi-conjugate gradient method. The total time of our color-adjustment technique is given by *Conv Pyram* + *Color Adjust*.

Image source		**Times (in seconds)**		
Example	**Clone area**	**Conv Pyram**	**Color Adjust**	**PIE**
Bear	60 K	0.56	0.28	19.19
Fish	93 K	1.03	0.38	43.32
Plane	162 K	0.98	0.43	122.35
Portrait	359 K	1.01	0.40	552.39
Church	765 K	1.57	0.566	2,473.40
Dolls	1,983 K	3.87	1.104	17,946.55

Our technique takes advantage of fast solutions for estimating Laplacian membranes, allowing it to provide instant feedback on the compositing results. It can also be used with non-conservative vector fields, by solving the Poisson equation using a linear system in the conventional way.

We have demonstrated the effectiveness of our technique by comparing it with traditional Poisson cloning [1] and alpha blending. Through these comparisons, we have shown our method's clear benefits. Compared to Poisson cloning, it avoids the excessive diffusion of background colors into the inserted elements. With respect to alpha blending, our technique lends to more natural compositions, which approximate the influence of ambient illumination on the shading of the composited objects (Figs. 7 and 8). In addition, our technique is the first to support color adjustment with fast seamless cloning approaches [7], [8], a requirement when working with high-resolution images (Table I).

Fig. 10: Comparison between Guo and Sim's method [15] and ours. (left) Source and target images, (center) their composition, (right) our result using automatically computed $w = 0.98$.

As future work, it would be desirable to automatically handle texture discontinuities over the transition zones.

Acknowledgements

This work was sponsored by CAPES and CNPq-Brazil (grants 306196/2014-0, 134134/2013-3, and 482271/2012-4). The source image of Fig. 1 is from *http://www.alphamatting.com/*. Figs 10 (left) and (center) are reproduced from [15] (Springer License 3640240696190). The remaining images are from Flickr users bernardohenz, Don McCullough, Bud, slworking, salemstudioz, simens, wonderlane, and Wonderlane, and are under Creative-Commons license.

References

[1] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM TOG*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[2] S. Jeschke, D. Cline, and P. Wonka, "A gpu laplacian solver for diffusion curves and poisson image editing," *ACM TOG*, vol. 28, no. 5, pp. 116:1–116:8, Dec. 2009.
[3] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister, "Multi-scale image harmonization," *ACM TOG*, vol. 29, pp. 125:1–125:10, July 2010.
[4] M. W. Tao, M. K. Johnson, and S. Paris, "Error-tolerant image compositing," in *Proc. ECCV*, 2010, pp. 31–44.
[5] M. Farid and A. Mahmood, "An image composition algorithm for handling global visual effects," *Multimedia Tools and Applications*, vol. 71, no. 3, pp. 1699–1716, 2014.
[6] X. Li and D. Kim, "A color adjustment method for automatic seamless image blending," in *Proc. of HCI*, 2007, pp. 82–91.
[7] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski, "Coordinates for instant image cloning," *ACM TOG*, vol. 28, no. 3, pp. 67:1–67:9, Jul. 2009.
[8] Z. Farbman, R. Fattal, and D. Lischinski, "Convolution pyramids," *ACM TOG*, vol. 30, no. 6, pp. 175:1–175:8, Dec. 2011.
[9] J. Bolz, I. Farmer, E. Grinspun, and P. Schröder, "Sparse matrix solvers on the gpu: conjugate gradients and multigrid," in *ACM SIGGRAPH 2005 Courses*, 2005.
[10] F. Wei and H. Yang, "Directed transmission method, a fully asynchronous approach to solve sparse linear systems in parallel," in *Proc. SPAA*, 2008, pp. 365–365.
[11] R. Szeliski, M. Uyttendaele, and D. Steedly, "Fast poisson blending using multi-splines," in *Proc IEEE ICCP 11*, 2011.
[12] B. Dizdaroglu and C. Ikibas, "A seamless image editing technique using color information," in *IPCV'09*, 2009, pp. 216–221.
[13] Y. Li, Z. Xie, B. Sheng, Y. Gui, and L. Ma, "Seamless image composition with coherent tone adjustment," in *Proc. VRCAI*, 2011, pp. 149–154.
[14] W. Yang, J. Zheng, J. Cai, S. Rahardja, and C. W. Chen, "Natural and seamless image composition with color control," *Trans. Img. Proc.*, vol. 18, no. 11, pp. 2584–2592, Nov. 2009.
[15] D. Guo and T. Sim, "Color me right - Seamless image compositing," in *Proc. CAIP*, 2009.
[16] H. Wu and D. Xu, "Image compositing using dominant patch transformations," *Computers and Graphics*, vol. 38, no. 0, pp. 277 – 285, 2014.
[17] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE CGA*, vol. 21, no. 5, pp. 34–41, Sep 2001.
[18] T. Pouli and E. Reinhard, "Progressive histogram reshaping for creative color transfer and tone reproduction," in *Proc of NPAR*, 2010, pp. 81–90.
[19] J.-F. Lalonde and A. A. Efros, "Using color compatibility for assessing image realism," in *IEEE ICCV*, 2007.
[20] S. Xue, A. Agarwala, J. Dorsey, and H. Rushmeier, "Understanding and improving the realism of image composites," *ACM TOG*, vol. 31, no. 4, pp. 84:1–84:10, Jul. 2012.
[21] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *SIGGRAPH '95*, 1995, pp. 191–198.
[22] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum, "Drag-and-drop pasting," *ACM TOG*, vol. 25, no. 3, pp. 631–637, Jul. 2006.
[23] O. Pele and M. Werman, "Improving perceptual color difference using basic color terms," *arXiv*, vol. abs/1211.5556, 2012.
[24] E. Shahrian, D. Rajan, B. Price, and S. Cohen, "Improving image matting using comprehensive sampling sets," in *Proc. IEEE CVPR*, 2013, pp. 636–643.
[25] E. Varnousfaderani and D. Rajan, "Weighted color and texture sample selection for image matting," *IEEE TIP*, vol. 22, no. 11, pp. 4260–4270, Nov 2013.
[26] E. S. L. Gastal and M. M. Oliveira, "Shared Sampling for Real-Time Alpha Matting," *CGF*, vol. 29, no. 2, pp. 575–584, 2010.
[27] J. Sun, H. Zhu, Z. Xu, and C. Han, "Poisson image fusion based on markov random field fusion model," *Inf. Fusion*, vol. 14, no. 3, pp. 241–254, Jul. 2013.
[28] Y. Zhang, J. Ling, X. Zhang, and H. Xie, "Image copy-and-paste with optimized gradient," *The Visual Computer*, vol. 30, no. 10, pp. 1169–1178, 2014.
[29] B. Henz, F. A. Limberger, and M. M. Oliveira, "Color adjustment for seamless cloning based on laplacian-membrane modulation website," http://www.inf.ufrgs.br/~bhenz/projects/color_adjustment/, 2015.