Drag-and-Drop Pasting

Jiaya Jia[1] Jian Sun[2] Chi-Keung Tang[3] Heung-Yeung Shum[2]
[1]The Chinese University of Hong Kong [2]Microsoft Research Asia
[3]The Hong Kong University of Science and Technology

(a) (b) (c) (d) (e)

Figure 1: Drag-and-Drop Pasting. Given a source image (a), the user draws a boundary that circles the wood log and its shadow in the water, and drags and drops this region of interest onto the target image in (b). The result from Poisson image editing (c) is however not satisfactory. Structures in this target image (e.g., the dark beach) intersect the source region boundary, thus produce unnatural blurring after solving the Poisson equations. Our approach, called *drag-and-drop pasting*, computes an optimized boundary shown in (d), which is then used to generate a seamless image composite (e).

Abstract

In this paper, we present a user-friendly system for seamless image composition, which we call drag-and-drop pasting. We observe that for Poisson image editing [Perez et al. 2003] to work well, the user must carefully draw a boundary on the source image to indicate the region of interest, such that salient structures in source and target images do not conflict with each other along the boundary. To make Poisson image editing more practical and easy to use, we propose a new objective function to compute an optimized boundary condition. A shortest closed-path algorithm is designed to search for the location of the boundary. Moreover, to faithfully preserve the object's fractional boundary, we construct a blended guidance field to incorporate the object's alpha matte. To use our system, the user needs only to simply outline a region of interest in the source image, and then drag and drop it onto the target image. Experimental results demonstrate the effectiveness of our "drag-and-drop pasting" system.

Keywords: Image processing, Poisson image editing, Image compositing

1 Introduction

Image composition is the process of creating a new image by pasting an object or a region from a source image onto a target image. Poisson image editing [Perez et al. 2003] has been proposed recently as an effective approach for seamless image composition. By solving Poisson equations using the user-specified boundary condition, Poisson image editing seamlessly blends the colors from both images without visible discontinuities around the boundary. Poisson image editing not only has an elegant mathematical formulation, but also appears to be easy to use.

The effectiveness of Poisson image editing, however, depends on how carefully the user draws the boundary. As shown in Figure 1, Poisson image editing may not always produce good results. Given source and target images in Figures 1(a) and (b), and the blue boundary casually drawn by the user, Figure 1(c) shows that Poisson image editing may generate unnatural blurring artifacts at places where the boundary intersects with salient structures in the target image (e.g. the dark beach at the top). To obtain seamless composition, we observe that the user needs to take the target image into consideration before drawing the boundary for Poisson image editing.

In this paper, we propose a method to optimize the boundary condition for Poisson image editing. With the optimized boundary shown in Figure 1(d), we again apply Poisson image editing to obtain seamless composition as shown in Figure 1(e). We note that in Figure 1(d) the optimized boundary avoids as much as possible salient image structures on both source and target images.

To optimize the boundary condition, we propose a new objective function which is iteratively minimized using a shortest closed-path algorithm. We search for the optimal boundary in between what the user casually draws (the region of interest) and what the user really cares about (the object of interest). Compared with the original Poisson image editing, our system allows the user to easily drag the region of interest from the source image and to drop it onto the target image, without the need for careful specification of the optimal boundary.

Often, the optimized boundary may intersect with the object of interest. In this case, fine structures of the object may be missing after blending with the target image through Poisson equations. We introduce a blended guidance field to integrate an alpha matte into Poisson equations to faithfully preserve the fractional boundary of the object and produce more natural image composite.

2 Related work

Image matting is a common way to extract an object from a source image which can then be pasted onto a target image naturally using an alpha channel. Most matting methods require a trimap in order to estimate the alpha channel and foreground color. There has been a lot of work on natural image matting using a single image [Berman et al. 2000; Ruzon and Tomasi 2000; Chuang et al. 2001;

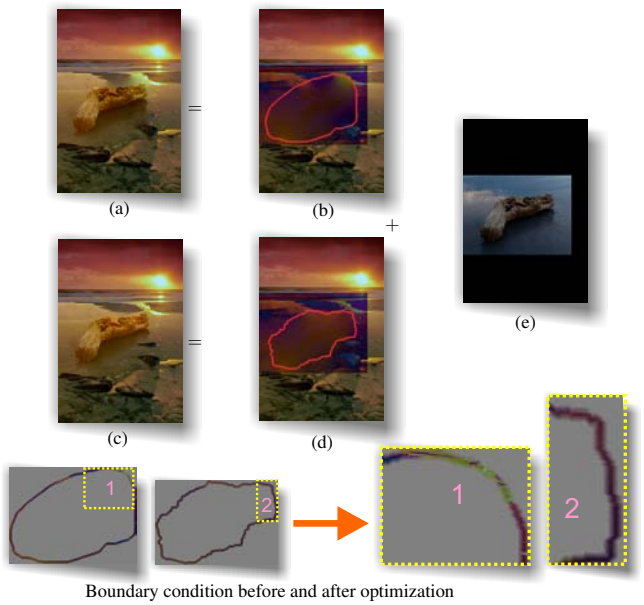


Figure 2: Comparison of the boundary conditions. (a) and (c) are results by solving Poisson equations with different boundary conditions. They are equivalent to adding (b) and (d) to the source image f_s shown in (e) respectively. (b) and (d) are results by solving the corresponding Laplace equations. The red curves are boundaries while the rectangular shaded areas are caused by the difference of the source and result images. The bottom row shows the boundaries in (b) and (d). Note that the color variance along boundary 2 is smaller that along boundary 1.

Sun et al. 2004]. Other techniques [Smith and Blinn 1996; McGuire et al. 2005] employ a controlled environment or a special device to reduce the inherent ambiguity of the matting problem, thus produce higher quality results.

In image composition, the multi-resolution spline technique [Burt and Adelson 1983] has long been used to seamlessly blend two different images through interpolations at different levels of Laplacian pyramids. Poisson image editing [Perez et al. 2003], on the other hand, blends two images through Poisson equations with a guidance field and a user-specified boundary. Image stitching in the gradient domain [Levin et al. 2004; Jia and Tang 2005] has also been proposed to reduce the artifacts caused by structure misalignment, and to correct color discrepancy.

Image compositing can also be done by piecing together multiple image patches from a single image or from multiple images. Graph-cut textures [Kwatra et al. 2003], for instance, stitch textures or natural images by finding the best seams using the graph cuts algorithm [Boykov and Jolly 2001]. Interactive digital photomontage [Agarwala et al. 2004] combines different regions of a set of roughly aligned photos with similar content into a single composite. Graph cuts are used to minimize the binary seams between the combined regions, and followed by Poisson image editing to reduce any remaining artifacts. In comparison, our method has the following advantages. The experimental comparisons are shown in section 5.

1. Agarwala et al. [2004] require the user to draw a number of strokes to initialize the graph-cut. For complex examples, the labeling requires a number of strokes be drawn iteratively. One example is the thin parts of the object, the user has to carefully draw lines inside them. No strokes are required for the same examples in our method.

2. Our method optimizes for the new boundary energy based on the minimum-error property in solving the Laplace equations and uses iterative optimization, thus can produce smooth and natural blending results even when the source region and the target images differ largely in color and structure. Moreover, our method can handle fine and transparent structures while solving the Poisson equations.

3 Optimal Boundary

We address the problem of optimizing boundary conditions for Poisson image editing in this section.

3.1 Poisson image editing

To paste a region of interest from the source image f_s to the target image f_t, the following minimization problem [Perez et al. 2003] is solved using the guidance field $v = \nabla f_s$ given the boundary condition defined on the user-drawn region of interest Ω_0:

$$\min_f \int_{p\in\Omega_0} |\nabla f - v|^2 dp \text{ with } f|_{\partial\Omega_0} = f_t|_{\partial\Omega_0}, \quad (1)$$

where f is the resulting image, and $\partial\Omega_0$ is the exterior boundary of Ω_0. We denote $f' = f - f_s$. Since the guidance field $v = \nabla f_s$ is a gradient field, Eqn. 1 can be written as:

$$\min_{f'} \int_{p\in\Omega_0} |\nabla f'|^2 dp \text{ with } f'|_{\partial\Omega_0} = (f_t - f_s)|_{\partial\Omega_0}. \quad (2)$$

The associated Laplace equation is:

$$\Delta f' = 0 \text{ with } f'|_{\partial\Omega_0} = (f_t - f_s)|_{\partial\Omega_0}, \quad (3)$$

where $\Delta = (\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})$ is the Laplacian operator and f' is a membrane interpolation inside Ω_0 for the boundary condition $(f_t - f_s)|_{\partial\Omega_0}$.

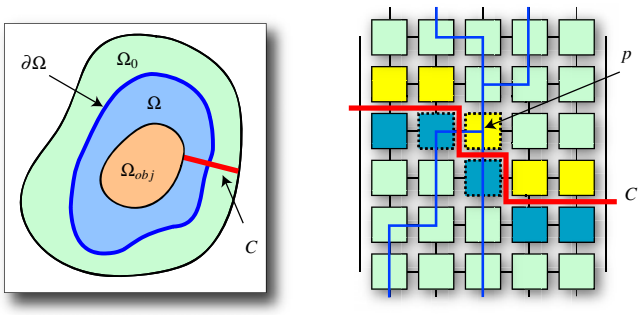
Eqn. 3 is simply a different interpretation of Poisson image editing by solving alternative Laplacian equations instead of Poisson equations. As illustrated in Figure 2, the result from Poisson image editing (a) can be obtained by first solving the corresponding Laplacian equations using the boundary condition $(f_t - f_s)|_{\partial\Omega_0}$ in (b), and then adding back the original source image (e). Similarly, with a different boundary condition, (c) can be obtained from (d) and (e). Both images (a) and (c) are taken from Figure 1.

Eqn. 2 leads to the following important property. The variational energy $\int_{\Omega_0} |\nabla f'|^2$ will approach zero if and only if all boundary pixels satisfy $(f_t - f_s)|_{\partial\Omega_0} = k$, where k is a constant value [Zwillinger 1997]. In other words, the membrane interpolation is constant if and only if the boundary condition is constant.

As illustrated in Figure 2, the boundary conditions $(f_t - f_s)|_{\partial\Omega_0}$ determine the final results. At the bottom left of Figure 2, the color difference between the source and target images, $(f_t - f_s)$ along the user-drawn boundary $\partial\Omega_0$ (on the left) and another new boundary $\partial\Omega$ (on the right) are shown. From the zoomed-in views at the bottom right, we can observe that pixel colors along the boundary $\partial\Omega$ have much less variation than those along $\partial\Omega_0$. A smoother boundary condition produces smaller variational energy in solving the Laplacian equations, thus improves the quality of the resulting composite.

Where is the optimal boundary? Obviously it should be inside the region of interest Ω_0 that the user has drawn. It should also be outside of the object of interest Ω_{obj} which is what the user really wants to paste onto the target image. However, an ordinary user would prefer not to carefully trace Ω_{obj} but only to casually specify Ω_0 instead. Fortunately, recent interactive segmentation techniques such as GrabCut [Rother et al. 2004] and Lazy Snapping [Li et al. 2004] can be used to produce Ω_{obj} once Ω_0 is given. For most results in this paper, Ω_{obj} is automatically computed by GrabCut using Ω_0 as initialization.

The question now is how to construct a color-smooth boundary condition $(f_t - f_s)|_{\partial\Omega}$ in the region $\Omega_0 \backslash \Omega_{obj}$, where $\partial\Omega$ is a closed boundary to be estimated, as shown in Figure 3 (a).



(a) (b)

Figure 3: Boundary optimization. (a) The region of interest Ω_0 is pasted onto the target image, which completely encloses the object of interest Ω_{obj}. The optimized boundary $\partial\Omega$ lies inside $\Omega_0 \backslash \Omega_{obj}$. The cut C is shown in red. (b) Zoom-in view of the cut C. The yellow and blue pixels are on different sides of C. The dashed yellow pixel p is adjacent to two blue pixels. Two shortest paths, shown as blue lines, are simultaneously computed.

3.2 Boundary energy minimization

The optimized boundary $\partial\Omega$ should lie in between Ω_0 and Ω_{obj}. To reduce the color variance along the boundary, the following objective function or boundary energy is minimized:

$$E(\partial\Omega, k) = \sum_{p \in \partial\Omega} ((f_t(p) - f_s(p)) - k)^2, \text{ s.t. } \Omega_{obj} \subset \Omega \subset \Omega_0, \quad (4)$$

where k is a constant value to be determined. Note that in all our experiments, we take the value of each color pixel $f(p)$ as a ternary set in $\{r, g, b\}$ color space. $(f(p) - f(q))$ is computed as the L2-norm in color spaces. $(f_t(p) - f_s(p)) - k$ represents the color deviation of the boundary pixels with respect to k.

Iterative optimization Since the optimal boundary may pass through all pixels in $\Omega_0 \backslash \Omega_{obj}$, simultaneously estimating the optimal k and the boundary $\partial\Omega$ is intractable. In the following, we propose an iterative optimization algorithm to optimize them.

1. Initialize Ω as Ω_0.
2. Given the current boundary $\partial\Omega$, the optimal k is computed by taking the derivative of Eqn. (4) and setting it to zero:

$$\frac{\partial E(\partial\Omega, k)}{\partial k} = 0$$
$$\Leftrightarrow \quad k = \frac{1}{|\partial\Omega|} \sum_{p \in \partial\Omega} (f_t(p) - f_s(p)), \quad (5)$$

   where $|\partial\Omega|$ is the length of the boundary $\partial\Omega$. So k is the average color difference on the boundary.
3. Given the current k, we optimize the boundary $\partial\Omega$.
4. Repeat steps 2 and 3 until the energy of Eqn. 4 does not decrease in two successive iterations.

The convergence of the above algorithm is guaranteed in step 4 by constraining the energy defined in Eqn. 4 to be monotonically decreasing.

In step 3, computing an optimal boundary is equivalent to finding a shortest path in a graph $\mathcal{G}$ defined in $\Omega_0 \backslash \Omega_{obj}$ (in short, the band).

3.3 A shortest closed-path algorithm

The nodes in $\mathcal{G}$ are pixels within the band while the edges represent 4-connectivity relationships between neighboring pixels. The cost $((f_t(p) - f_s(p)) - k)^2$ is defined on each node as the color difference with respect to k. The accumulated cost of a path sums up the costs of all nodes on the path. For a single object, the estimated Ω_{obj} in our method can be regarded as genus-0 region and $\Omega_0 \backslash \Omega_{obj}$ is of genus-1 type, as shown in Figure 3 (a).

Unlike a standard shortest path problem, $\partial\Omega$ is a closed curve enclosing Ω_{obj}, which complicates the optimization. To make it tractable, we first change the type of region $\Omega_0 \backslash \Omega_{obj}$ from genus-1 to genus-0. This can be done by breaking the band connectivity using a cut C, as shown in red in Figure 3 (a). In the corresponding representation in graph $\mathcal{G}$, we remove all edges crossing the cut. In the following, we show how to compute a closed shortest-path using 2D dynamic programming.

A shortest closed-path algorithm

- As illustrated in Figure 3(b), for each pixel p on one side of the cut C (shown in yellow), we compute the shortest paths to all adjacent pixels on the other side of the cut (shown in blue). Since graph $\mathcal{G}$ is a 2D grid, computing the shortest path from any node to all others in the band can be achieved by 2D dynamic programming [Dijkstra 1959; Mortensen and Barrett 1995] with a complexity $O(N)$, where N is the number of pixels in the band. Among all the shortest paths starting from pixel p and ending at the neighboring pixels on the other side of the cut, $Path(p)$ is the one with minimum cost. In Figure 3(b), for two dashed blue pixels that are neighbors to p in the image plane, their corresponding shortest paths are computed and shown as blue lines.
- We repeat the previous computation for all pixels on the yellow side of the cut C, and get a set of $Path$. The optimized boundary $\partial\Omega$ is assigned as one that gives the globally minimum cost. Suppose that there are M pixels on the yellow side of the cut in graph $\mathcal{G}$, the overall computational complexity is $O(MN)$.

If the optimal boundary passes the cut C only once, the above algorithm will reach the global minimum of the energy defined in Eqn. 4. Indeed, the path with the minimum accumulated cost seldom twists in our experiments.

The cut C intersects the band at two pixels on $\partial\Omega_0$ and $\partial\Omega_{obj}$ respectively. There are many possibilities how it is initialized. In our method, to achieve better performance, we compute a shortest straight line segment among all pixel pairs connecting $\partial\Omega_{obj}$ and $\partial\Omega_0$ by computing the Euclidian distance. This line segment is then rasterized into a pixel list in a 2D image plane. The cut C is drawn adjacent to the pixel list on any side. There are two benefits in computing the shortest cut C. First, the short length reduces the probability that the optimal boundary passes the cut more than once. Second, with fewer pixels adjacent to the cut C, the value of M will be small, which speeds up the computation.

4 Fractional boundary

An optimized boundary condition reduces the variational energy in solving the Laplacian equations, and avoids unnatural blurring in the composite. However, the optimized boundary may intersect with the object with a fractional boundary and break up subtle and fine details. Figure 4 depicts a scenario where $\partial\Omega$ is close to $\partial\Omega_{obj}$ and breaks the hairy structures (shown in (c) and (d)). To faithfully preserve the object's transparent boundary, we propose to incorporate an alpha matte in a blended guidance field for Poisson equations, by detecting the regions where alpha blending should be applied along the boundary.

4.1 A blended guidance field

Our goal of introducing transparency is to preserve precise fractional boundary of the object of interest from the source image when it is composited onto the target image while being capable of blending the color of the object seamlessly with the target image. Conventional alpha blending techniques cannot modify the color of the source object. To combine alpha blending with Poisson image editing, we define a binary coverage mask M to indicate where alpha blending should be applied ($M(p) = 1$) and vice versa, which

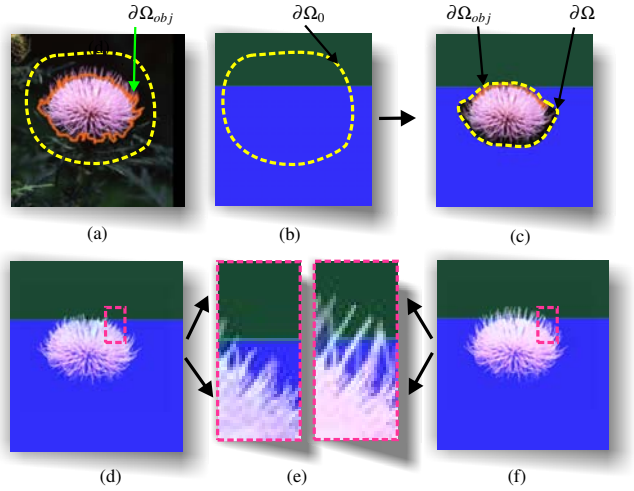


Figure 4: Preserving fractional object boundary. Given the input source image of a flower in (a) and target image in (b), the optimized boundary $\partial\Omega$ snaps closely to the top of the flower's silhouette $\partial\Omega_{obj}$, thus intersects the fine details as shown in (c). Using the optimized boundary to solve the Poisson equation, as shown in (d), the fractional boundary cannot be well preserved as depicted in the zoom-in view (e). Our approach integrates the object's alpha matte in the Poisson equation, and faithfully preserves the fine details and produces an improved composite of the flower shown in (f). The corresponding zoom-in view is shown in (e).

partitions the image into regions. However, when directly applying the blending techniques in separate regions, the pixels in adjacent region boundaries may have color discontinuities since they are processed by two methods respectively without an appropriate spatial transition. To eliminate the artifact caused by the color discontinuity, we integrate the alpha matte in the blended guidance field in the Poisson equation.

We denote $\Phi = \{p|0 < \alpha(p) < 1\}$ as the fractional object boundary, where α is computed automatically within a few pixels surrounding Ω_{obj} by coherence matting [Shum et al. 2004]. Comparing to Bayesian matting, coherence matting cooperates a prior of the alpha value in its formulation. In our method, we model it as a Gaussian distribution with respect to the median axis of Φ. Taking Figure 5(b) as an illustration, Φ is of the shape of a narrow blue belt. The blended guidance field is $v' = (v'_x, v'_y)$. For each pixel $p = (x,y)$, $v'_x(x,y)$ is defined as:

$$v'_x(x,y) = \begin{cases} \nabla_x f_s(x,y) & M(x,y) = M(x+1,y) = 0 \\ \nabla_x(\alpha f_s + (1-\alpha) f_t) & M(x,y) = M(x+1,y) = 1 \\ 0 & M(x,y) \neq M(x+1,y) \end{cases} \quad (6)$$

$v'_y(x,y)$ is defined in a similar way. It shows that, depending on whether the alpha matte is applied, $v'_x(x,y)$ is defined as the alpha blended gradient or source image gradient in regions $M = 1$ and $M = 0$ respectively. However, in between these two regions, the gradient has no exact definition in image space. So we assign value zero to these pixel gradients to smoothly fuse the two regions and eliminate color discontinuity.

Given the new blended guidance field, we minimize the following variational energy:

$$\arg\min_f \int_{p \in \Omega \cup \Phi} ||\nabla f - v'||^2 dp \text{ with } f|_{\partial(\Omega\cup\Phi)} = f_t|_{\partial(\Omega\cup\Phi)}, \quad (7)$$

where $\Omega \cup \Phi$ includes pixels either within the optimized boundary or inside the fractional object boundary, and $\partial(\Omega \cup \Phi)$ is $\Omega \cup \Phi$'s exterior boundary.

Based on the above analysis, to solve the boundary problem, we construct the binary coverage mask M within $\Omega \cup \Phi$ before solving Eqn. 7. Consider the pixels inside the object where $\alpha(p) = 1$, *the guidance field v' will always be ∇f_s regardless of the values of* $M(p)$. Therefore, we do not need to compute M in these pixels.

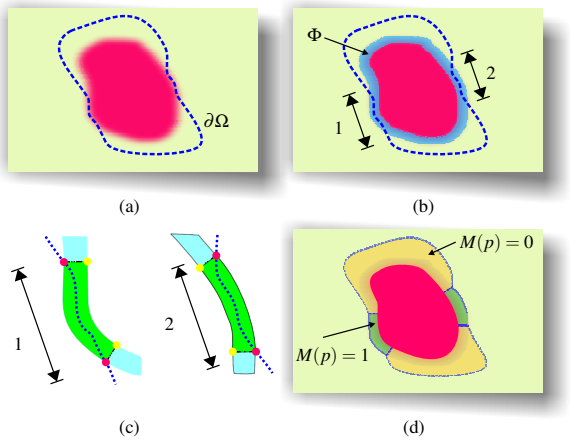


Figure 5: Construction of the binary coverage mask M. (a) A source object with fractional boundary is pasted onto a target image. The dashed curve is the optimized boundary $\partial\Omega$. (b) The blue belt shows the region $\Phi = \{p|0 < \alpha(p) < 1\}$. As indicated by arrows 1 and 2, $\partial\Omega$ penetrates into the belt where matte compositing should be applied. (c) Zoom-in views of segments 1 and 2, around which the region $\{p|M(p) = 1\}$ is computed. (d) The resulting binary coverage mask where $\{p|M(p) = 0\}$ and $\{p|M(p) = 1\}$ are shown in yellow and green.

Figure 5 illustrates how we estimate M. In Figures 5(a) and 5(b), $\partial\Omega$ penetrates into the fractional object boundary in two segments 1 and 2, where some pixels with fractional alpha value are left outside Ω. This breaks the structure of the object boundary. Around segments 1 and 2, matte compositing is applied in the blended guidance field and M is set to 1. In the following, we list the main steps to construct the mask M:

- We first compute the head and tail intersections between $\partial\Omega$ and the belt Φ in each segment, indicated as red dots in (c). Two segments that contain the intersections are indicated by arrows 1 and 2 in Figures 5(b) and 5(c).
- To get the region where alpha blending should be applied, we compute the nearest points on the other side of the belt Φ, as shown by the yellow points, and connect the corresponding red and yellow points by straight line segments. Thus, the belt Φ is partitioned into blue and green in Figure 5(c).
- We set the green region in Figure 5(d) as $\{p|M(p) = 1\}$, and set $M(p) = 0$ for the remaining pixels p in $\Omega \cup \Phi$.

5 Experimental Results

We apply drag-and-drop pasting on a wide variety of source and target images, and compare it with Poisson image editing, matte compositing, and digital photomontage. The results are automatically computed by applying boundary optimization and the blended guidance field consecutively.

Sand pyramid. If the source and target images vary significantly in color and structure, as shown in Figures 6(a) and (b), Poisson image editing method cannot produce satisfactory results (Figure 6(c)). When applying "digital photomontage" [Agarwala et al. 2004] using the user-drawn strokes in Figure 6(d), the result still does not faithfully preserve the object structures (Figure 6(e)). Our approach computes an optimized boundary around the sand pyramid (Figure 6(f)). No blurring, color mismatches or broken structures are produced (Figure 6(g)), even though the top of the pyramid intersects with the river in the target image.

Motorcycle. In Figure 7, the pasted motorcycle touches the cars and the Toyota sign in the target image. Figure 7(d) shows the result using Poisson image editing with a user-drawn boundary. It is not surprising that the result looks unnatural, because the pasted region does not match well with the target image. Figure 7(e) is the matte compositing result. Note that the alpha matte between the two

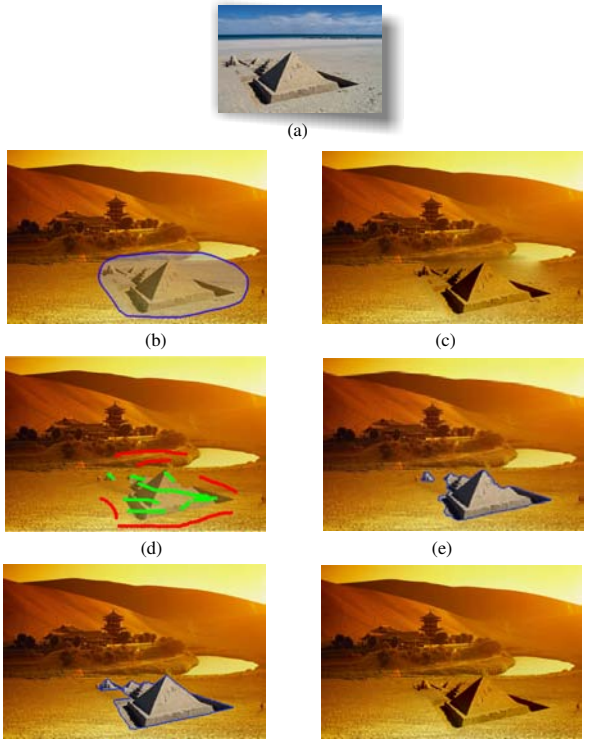


Figure 6: Sand pyramid. (a) Source image. (b) User drags a region of interest to include the pyramids, and drops it onto the target image. (c) Poisson image editing result using the user-drawn boundary. (d) The user-drawn strokes used to initialize segmentation in [Agarwala et al. 2004]. (e) The refined boundary using the method in [Agarwala et al. 2004]. (f) The optimized boundary from our method. (g) Our blending result. No unnatural occlusion or broken structures are observed.

wheels in the source image is difficult to be extracted automatically. In addition, the color of the source object does not match well with the target scene. Our automatically optimized boundary is shown in Figure 7(f), in which the alpha blending is appropriately applied to faithfully preserve the fractional boundary of the source object. Note that the optimized boundary preserves the inherent structure of the target image as well, as shown in Figure 7(g).

Chimney. Figure 8(a) and (b) show a chimney pasted into a sea line image where the alpha matte is needed at the intersection pixels. Figure 8(e) is the alpha matte automatically computed. Note that the matte of the smoke is not accurate. Both Poisson image editing using the mixing gradients and matte compositing cannot produce good results. Our method does not rely on the entire matte of the object, but only part of it where the optimized boundary snaps close to the silhouette of the object. Figure 8(g) shows the satisfactory result produced using our method.

Surfing. We show in Figure 9 a difficult surfing example. Poisson image editing using the user-drawn region cannot avoid boundary misalignment (Figure 9(b)). "Digital photomontage" requires the user to draw a number of strokes to initialize the graph-cut. Even with the carefully marked strokes (Figure 9(c)), their method is susceptible to labeling errors due to the thin arms of the surfer. Our method using the optimized boundary and the blended guidance field works well in this difficult example. The optimized boundary passes through a highly textured area while not breaking any salient structure in the target image, e.g., the body of the lower surfer.

6 Conclusion and Discussion

In this paper, we have proposed a user-friendly approach to achieve seamless image composition without requiring careful initialization by the user. Compared with the original Poisson image editing, our system is more practical and easy to use. Moreover, our approach can also preserve fractional object boundary by introducing a blended guidance field that incorporates the object's matte.

Our method uses GrabCut and image matting methods to automatically compute Ω_{obj} and α respectively. The alpha matte is only used in the region where $M = 1$ as described in section 4. Therefore, we do not require precise alpha value for all pixels. Only if the automatically computed Ω_{obj} and α contain large error, the user interactions are needed.

We propose future research directions are as follows: 1) Investigating the degree of diffusion or color change controlled within the Poisson image editing framework. When the source and target images differ significantly in color, solving the Poisson equations changes the source object's color in the composite. This may not be always desirable. An example is to paste a dog from a white beach onto green grass. The dog's color in the composite will have a green shade after solving the Poisson equations. 2) If the target image has complex structures, no matter how we refine the pasted region boundary, the structure of the source region and target scene cannot be precisely aligned. One possible solution is the modification of the boundary conditions.

Acknowledgements. The authors would like to thank Michael S. Brown for providing his voiceover in the video and Ruonan Pu for her help in preparing the paper. Jiaya Jia's research was funded by the grant from Microsoft Research Asia and Research Grant Council of Hong Kong special Administration Region, China.

References

AGARWALA, A., DONTCHEVA, M., AGRAWALA, M., DRUCKER, S., COLBURN, A., CURLESS, B., SALESIN, D., AND COHEN, M. 2004. Interactive digital photomontage. *Proceedings of ACM SIGGRAPH 23*, 3, 294–302.

BERMAN, A., VLAHOS, P., AND DADOURIAN, A. 2000. Comprehensive method for removing from an image the background surrounding a selected object. *U.S. Patent 6,134,345*.

BOYKOV, Y., AND JOLLY, M. P. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Proceedings of ICCV*.

BURT, P. J., AND ADELSON, E. H. 1983. A multiresolution spline with application to image mosaics. In *ACM Transactions on Graphics*, vol. 2, 217–236.

CHUANG, Y., CURLESS, B., SALESIN, D., AND SZELISKI, R. 2001. A bayesian approach to digital matting. In *Proceedings of CVPR01*, vol. 2, 264–271.

DIJKSTRA, E. W. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 269–270.

JIA, J., AND TANG, C.-K. 2005. Eliminating structure and intensity misalignment in image stitching. In *Proceedings of ICCV*.

KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: image and video synthesis using graph cuts. *Proceedings of ACM SIGGRAPH 22*, 3, 277–286.

LEVIN, A., ZOMET, A., PELEG, S., AND WEISS, Y. 2004. Seamless image stitching in the gradient domain. In *Proceedings of ECCV*, Vol IV: 377–389.

LI, Y., SUN, J., TANG, C., AND SHUM, H. 2004. Lazy snapping. *Proceedings of ACM SIGGRAPH*, 303–308.

MCGUIRE, M., MATUSIK, W., PFISTER, H., HUGHES, J. F., AND DURAND, F. 2005. Defocus video matting. In *Proceedings of ACM SIGGRAPH*, vol. 24, 567–576.

MORTENSEN, E. N., AND BARRETT, W. A. 1995. Intelligent scissors for image composition. *Proceedings of ACM SIGGRAPH*, 191–198.

PEREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *Proceedings of ACM SIGGRAPH*, 313–318.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "grabcut" - interactive foreground extraction using iterated graph cuts. *Proceedings of ACM SIGGRAPH*, 309–314.

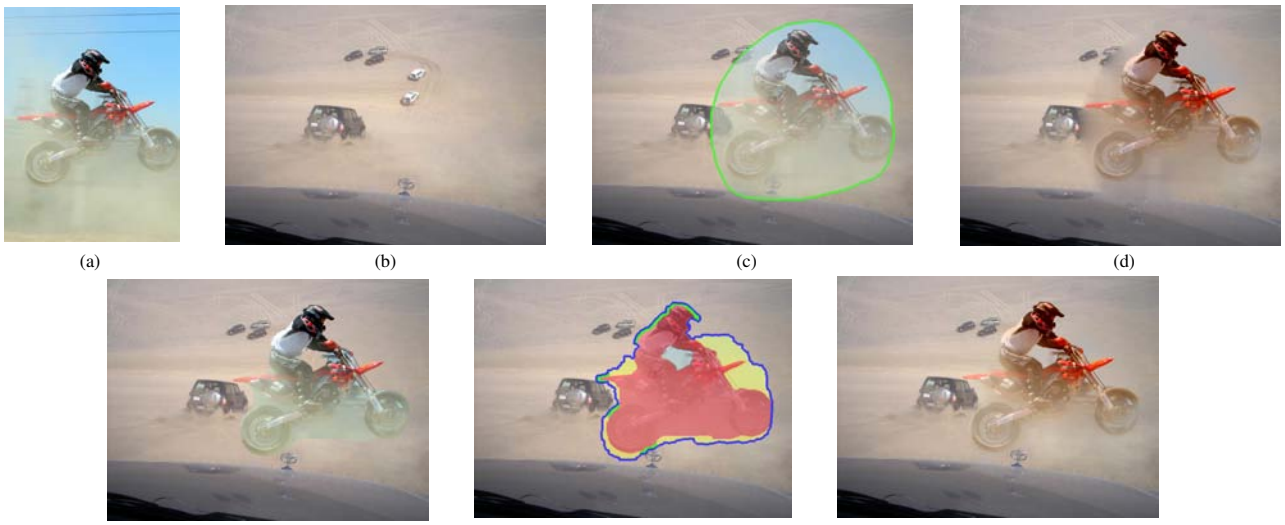
RUZON, M., AND TOMASI, C. 2000. alpha estimation in natural images. In *Proceedings of CVPR00*, 18–25.

SHUM, H.-Y., SUN, J., YAMAZAKI, S., LI, Y., AND TANG, C.-K. 2004. Pop-up light field: An interactive image-based modeling and rendering system. *ACM Trans. Graph. 23*, 2, 143–162.

SMITH, A., AND BLINN, J. 1996. Blue screen matting. *Proceedings of ACM SIGGRAPH*, 259–268.

SUN, J., JIA, J., TANG, C., AND SHUM, H. 2004. Poisson matting. *Proceedings of ACM SIGGRAPH*, 315–321.

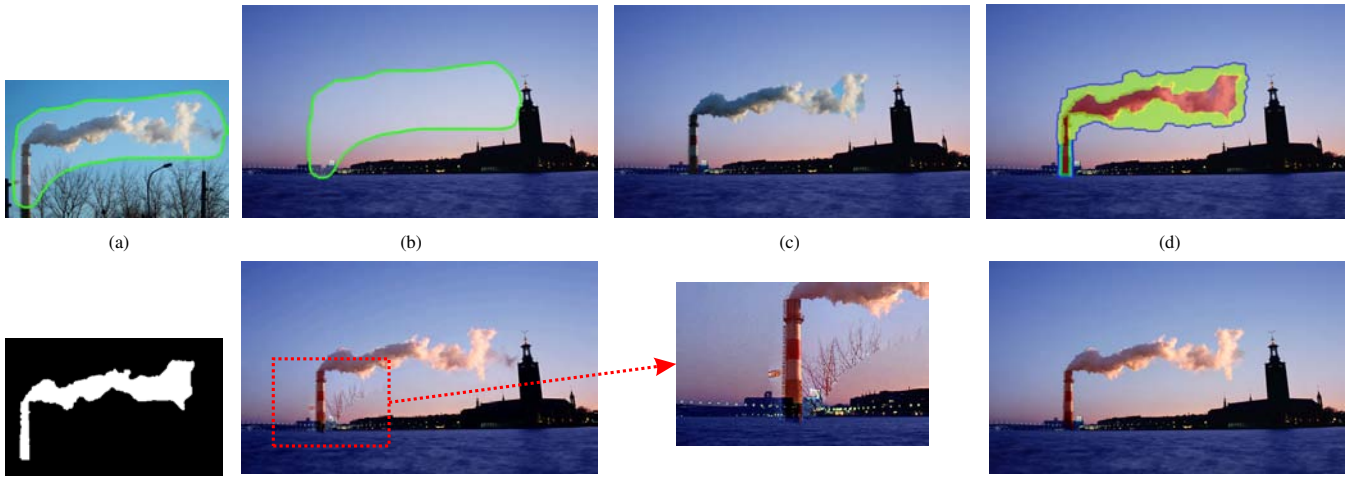
ZWILLINGER, D. 1997. *Handbook of Differential Equations, 3rd ed.* Boston, MA: Academic Press.



(a) (b) (c) (d)

(e) (f) (g)

Figure 7: Motorcycle. (a) Source image. (b) Target image. (c) The user-drawn region on the source image is overlaid on the target image. (d) Poisson image editing result with the user-drawn region. The boundary intersects with salient structures in the target image. (e) Matte compositing result. The color of the motorcyclist does not match well with the target scene. (f) The optimized boundary and the binary coverage mask to preserve fractional boundary. The color code is the same as Figure 5 (d). (g) Our result. The object is naturally blended into the target image without unnatural blurring.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8: Chimney. (a) Source image and the user-specified region of interest. (b) The user-specified region on the source image is overlaid onto the target image. (c) Matte compositing result. Note the color of the chimney does not match well with the target scene. (d) Optimized boundary and binary coverage mask. The color code is the same as Figure 5. (e) The alpha matte computed by the matting method. (f) Poisson image editing result using mixing gradients in seamless cloning [Perez et al. 2003]. (g) Zoom-in view of the mixing gradients result. It is noted that the unwanted tree branches and the occluded sea-line can be seen because of their strong gradients. (h) Our result. The chimney is seamlessly blended with the target image without structure and color mismatch.

(a) (b) (c) (d) (e)

Figure 9: Surfing. (a) Source and target images of surfers. (b) Poisson image editing result using the user-drawn boundary, showing unnatural occlusion and other obvious artifacts. (c) The user-drawn strokes used to initialize graph-cut in [Agarwala et al. 2004], and the refined boundary from graph-cut. The thin parts of the object, e.g., the arms of the surfer, are missing. (d) Our optimized boundary where the object structures are faithfully preserved. (e) Our blending result. The two surfers look natural in the final composite. The matte is applied around the surfboard to faithfully preserve the fractional boundary.