Simultaneous Matting and Compositing

Jue Wang
Electrical Engineering
University of Washington
juew@u.washington.edu

Michael F. Cohen
Microsoft Research
Michael.Cohen@microsoft.com

Abstract

Recent work in matting, hole filling, and compositing allows image elements to be mixed in a new composite image. Previous algorithms for matting foreground elements have assumed that the new background for compositing is unknown. We show that, if the new background is known, the matting algorithm has more freedom to create a successful matte by simultaneously optimizing the matting and compositing operations.

We propose a new algorithm, that integrates matting and compositing into a single optimization process. The system is able to compose foreground elements onto a new background more efficiently and with less artifacts compared with previous approaches. In our examples, we show how one can enlarge the foreground while maintaining the wide angle view of the background. We also demonstrate composing a foreground element on top of similar backgrounds to help remove unwanted portions of the background or to re-scale or re-arrange the composite. We compare and contrast our method with a number of previous matting and compositing systems.

1. Introduction

Image matting refers to the problem of estimating an opacity (alpha value) and a foreground color for the foreground element at each pixel in the image. Although the main purpose of matting is to re-compose the foreground onto a new background, previous matting approaches treat matting and compositing as separate tasks by assuming the new background is unknown. We show that, by combining matting and compositing into a single optimization process, the matting algorithm can be more robust and efficient to create a successful composite. We dub this new matting algorithm *compositional matting*.

Besides composing the foreground onto a new background as shown in Figure 5, we also explore the advantages of the proposed algorithm for re-organizing and recomposing elements within a single image. A common photograph we have all seen contains a person standing in front a beautiful outdoor scene. Although the background is nicely framed, the foreground person often looks quite small since the focal length of the camera was set to capture the wide angle scene. This problem becomes more obvious as the image is resized to standard snapshot size or is displayed on a small mobile device, as the foreground character may shrink to the point of being unrecognizable. One could crop the foreground from the original image, however the composite becomes less interesting by losing the background scene.

Based on the compositional matting algorithm we allow the user to set different display ratios for foreground and background objects. As shown in Figure 4, the user roughly indicates the foreground by a few paint strokes, and our system is able to generate a novel version of the photograph where the foreground is enlarged while the background remains as is.

2. Related Work

2.1. Foreground Segmentation

Many systems have been developed to accurately identify the foreground regions with minimal user guidance. Intelligent paint [10] and the object-based image editing system [2] first oversegment the image and then let the user select the regions that form the foreground object. LazySnapping [7] and GrabCut [11] systems provide interactive graph-cut-based segmentation solutions. The GrowCut system [14] employs cellular automation for interactive foreground extraction. In these systems users coarsely indicate foreground and background regions with a few paint strokes of the mouse and the system tries to determine the ideal boundary for segmenting the foreground.

2.2. Foreground Matting

Pixels on the edge of a foreground object usually contain some percentage of the background. These "mixed pixels" can create visual seams when composed onto a new background. Seamlessly composing a foreground object onto

a new background requires estimating an opacity (alpha) value for each pixel as well as the foreground color. Ruzon and Tomasi [12] show how to estimate the alpha matte and foreground color using statistical methods. Chuang et al. [3] extend this approach by employing Bayesian framework for alpha estimation in both images and video. The Poisson matting approach [13] solves Poisson equations for matte estimation, under the assumption that the foreground and background colors are distinct and smooth.

Recently, the problem of foreground segmentation and matting is combined together in the Belief Propagation matting system[15], where an iterative optimization process is employed for estimating foreground matte from sparse user input such as a few scribbles. This quickly becomes a new trend in matting research and some new systems, such as the easy matting [4] and the closed-form matting system[6], have been proposed to achieve more robust and accurate matting results with sparse user inputs.

2.3. Image Compositing

Since our system composes the foreground object in a source image onto a new background image, it belongs to the general framework of "photomontage". Agarwala et al. have proposed an interactive framework [1] for this task by using graph-cut optimization. However, their system only generates hard segmentation of source images thus is not capable of handling partial foreground coverage. The Poisson image editing system [9] uses generic interpolation machinery based on solving Poisson equations for composing a foreground region onto a destination region. It works well when the destination region has a relatively simple gradient field, but is insufficient to handle highly textured regions which are common in many images and in our examples. Recently proposed "Drag-and-Drop Pasting" system[5] improves Poisson image editing by finding optimal boundary conditions, however the system will always change the color of the entire foreground to blend it into the new background, which is often undesirable when the foreground and background have a significant gap in depth.

In contrast, our system seamlessly composes foreground and background images by estimating a matte for the foreground region. The matte is optimized in a sense that it will minimize the visual artifacts on the result image, although it may not be the true matte for the foreground.

2.4. Image Retargeting

Image retargeting means to adapt images for display on devices different than originally intended. Liu and Gleicher have proposed a retargeting algorithm [8] by using nonlinear fisheye-view warping to emphasize parts of an image while shrinking others. Although this method can fit a large image into a small screen, both the foreground and background are very distorted. Instead, we set different display ratios for the foreground and background to achieve the retargeting goal, and only alter a small region around the foreground to eliminate the compositing seams without significant distortion.

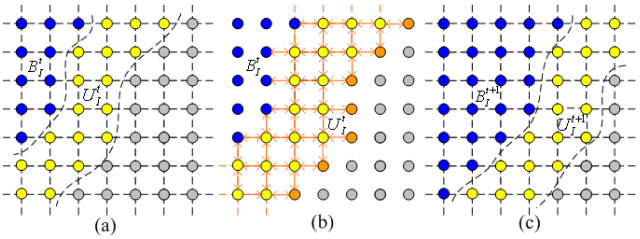

Figure 1. Our algorithm solves the composite in a front-propagation fashion. (a). Dilating background region B_I^t (blue) to create an unknown region U_I^t (yellow). (b). Finding the composite for U_I^t by optimization with boundary conditions. (c). Regions are evolved for the next iteration.

3. The Compositional Matting Algorithm

In the matting equation, the observed image I_z ($z = (x, y)$) is modeled as a convex combination of foreground image $F(z)$ and background image $B(z)$. In this work we assume the new background image B', onto which the extracted foreground will be composed onto, is known. The final composite thus can be calculated as

$$I_z^* = I_z + (1 - \alpha_z)(B'_z - B_z) \quad (1)$$

We can see that the final composite I_z^* is determined by two unknowns: α_z and the original background color B_z. As we will show later, our algorithm starts estimating the matte from the outside of the foreground where $B_z = I_z$, thus mostly the composite will only affected by the single unknown α_z, and the goal of our system is to estimate α_zs to minimize the visual artifacts of I_z^*s.

The most obvious advantage B' provides is that if the new background has very similar regions to the original one, instead of extracting the true foreground matte in these regions which can be erroneous, we can find a good transition between the old background and new background for a good composite. In other words, our matte can be conservative and thus some of the original background is carried into the composed image with the foreground.

3.1. Front Propagation for Matte Estimation

In our system the user first roughly indicates the foreground object on image I by specifying a bounding box R_I around it, and a few paint strokes on it, as shown in Figure 4a. We treat the region outside R_I as the initial background region B_I, the region under paint strokes as the initial foreground region F_I, and all other pixels uncovered as the initial unknown region U_I.

The matte M_I is estimated iteratively. In each iteration B_I is dilated to create a narrow band region U_I^t and update the matte in this region. In early iterations many pixels in U_I^t will have estimated alpha value of 0 since they are very likely to be background pixels. We then expand B_I based on the updated matte and dilate it again to create a new unknown region U_I^{t+1} for the next iteration. The algorithm stops when $U_I^{t+1} \approx U_I^t$. One can imagine that the unknown region is shrinking until becoming stable, as illustrated in Figure 1. The front propagation can stop immediately whenever it finds a good composite, thus can avoid visual artifacts resulting from inaccurate alpha values and foreground colors for real foreground pixels.

3.2. Cost Functions

In this section we describe how we set up the optimization objective in each iteration. A pixel z in the unknown region U_I^t has two possible modes. The first is that the new background color B'_z is very different from the original color I_z, thus an accurate alpha value should be estimated for z in order to fully substitute B_z with B'_z. We call this mode *matting mode*, and design a *matting cost* for it. The second mode, which we call *compositing mode*, is that B'_z is very close to I_z, thus it is a good place for transiting from the original image to the new background image. A *compositing cost* is defined for this case.

3.2.1 Matting Cost

The matting cost is designed to accurately estimate alpha values. In this case, the new background is simply ignored thus the problem degrades to the traditional matting problem. Inspired by recent optimization approaches for image segmentation and matting [15, 11, 7], our matting cost contains two terms: a *data term* which measures the alpha value based on the pixel's own observations, and a *neighborhood term* which enforces the alpha estimations to be consistent in a local neighborhood area.

Specifically, we sample a group of known background colors from the neighborhood of z, and use them to estimate a Gaussian distribution $G(\overline{B_z}, \Sigma_z^B)$. Since the user marked foreground pixels are usually far away, we use the global sampling method proposed in [15] to gather a group of foreground colors for z, and estimate a Gaussian distribution on them as $G(\overline{F_z}, \Sigma_z^F)$. An estimated alpha value α_z^m is calculated as

$$\alpha_z^m = \frac{d(I_z, \overline{B_z}, \Sigma_z^B)}{d(I_z, \overline{F_z}, \Sigma_z^F) + d(I_z, \overline{B_z}, \Sigma_z^B)} \tag{2}$$

where $d(I, \overline{C}, \Sigma)$ is the Manhattan distance from color I to a Gaussian $G(\overline{C}, \Sigma)$. The data term is then defined to be

$$D_z^{mat} = (\alpha_z - \alpha_z^m)^2 \tag{3}$$

It has been approved in previous approaches that enforcing proper neighborhood constraints can help estimate accurate alpha mattes. The Poisson matting [13] approach enforces the matte gradient to be proportional to the image gradient. The closed-form matting system [6] set the weights between neighboring pixels based on their color differences computed from local color distributions. In this work we choose to use this method for setting the neighborhood term due to its simplicity and efficiency.

Formally, the neighborhood term is defined as

$$J_z^{mat} = \sum_{v \in w_z} (\alpha_v - a_z I_v - b_z)^2 + \epsilon a_z^2 \tag{4}$$

where w_z is a 3×3 window around z, and a_z and b_z are two coefficients which can be eliminated later in the optimization process. ϵ is a regularization coefficient which is set to be 10^{-5} in our system. More details and justifications of this neighborhood term can be found in [6].

The total matting cost is defined by combining the data term and neighborhood term together as

$$C_z^{mat} = w^{mat} \cdot D_z^{mat} + J_z^{mat}. \tag{5}$$

where w^{mat} is a free parameter in the system which balance the data term and the neighborhood term.

3.2.2 Compositing Cost

The compositing cost is defined for image regions where the new background matches well with the original image. Similar to the matting cost, the compositing cost also contains a data term and a neighborhood term. Roughly speaking, the data term determines where to make a transition between the original image and the new background image, and the neighborhood term enforces the transition to be smooth.

Since I_z(which equals to B_z most likely) and B'_z are very close, we can ignore the true alpha of z and make a direct transition from the new background to the foreground image. To do this we examine the color difference between I_z and B'_z, and calculate a transition probability as

$$p_z = exp\left(-\left(B'_z - I_z\right)^T \Sigma_z^{I-1} \left(B'_z - I_z\right)\right), \tag{6}$$

where Σ_z^I is the local color variance computed from a 3×3 window centered at z in the original image.

The data term in the compositing cost is then defined as

$$D_z^{cmp} = (\alpha_z - p_z)^2. \tag{7}$$

If the color difference between B'_z and I_z (normalized by local color variance) is small, the transition probability p_z will be large, which encourages a high alpha value for z to force the front propagation to stop here.

The neighborhood term is defined as

$$J_z^{cmp} = \sum_{v \in w_z} \left[1 - exp\left(-d_t(z,v)\right)\right](\alpha_v - \alpha_z)^2, \quad (8)$$

where $d_t(z,v)$ is the *transition distance* between z and v which is calculated as $[d(I_z, B'_z, \Sigma_z^I) + d(I_v, B'_v, \Sigma_v^I)]/2$. Minimizing J_z^{cmp} will force the transition happens only at places where the transition distance is small, or in other words, only at places where the new background pixels are very close to the original ones. This neighborhood term is similar to the one defined in the photomontage system[1] for seamless compositing, however in the photomontage system the alpha values can only be either 1 or 0.

By combining the data term and the neighborhood term together, the compositing cost is defined as

$$C_z^{cmp} = w^{cmp} \cdot D_z^{cmp} + J_z^{cmp}, \quad (9)$$

where w^{cmp} is another balancing parameter of the system.

3.2.3 Combined Cost

For an unknown pixel z, we classify it to be in either matting mode or compositing mode based on its local image characteristics. The classification is achieved by defining a function δ_z as

$$\delta_z = \begin{cases} 1 & : \quad \alpha_z^m > \alpha_z^c \\ 0 & : \quad \alpha_z^m < \alpha_z^c \end{cases} \quad (10)$$

The total cost for tth iteration is defined as as

$$\begin{aligned} C^t &= \sum_{z \in U_I^t} \left\{\delta_z C_z^{mat} + (1-\delta_z) C_z^{cmp}\right\} \\ &= \sum_{z \in U_I^t} \left\{\delta_z [w_z^{mat} \cdot D_z^{mat} + J_z^{mat}]\right. \\ &\quad \left. + (1-\delta_z)[w_z^{cmp} \cdot D_z^{cmp} + J_z^{cmp}]\right\} \end{aligned} \quad (11)$$

For better justification of the total cost we defined, we can examine some extreme situations. If the new background B' is totally different from the original image I, then δ_zs will be all ones since α_z^cs defined in equation 6 will be very small. Then only the matting cost will be minimized and the compositional matting degrades to a general matting algorithm, which can generate competing (and often times better) mattes compared with previous matting approaches. Similarly, if B' is very close to I' over the whole image, thus only the compositing cost will be minimized and our system works in a similar way as the photomontage system[1], however since our system allows smooth alpha transition, it can generate more smooth composites than the photomontage system. For most of the cases, our system performs mixed matting and compositing.

3.3. Optimization

As we can see that each term in equation 11 is defined as a quadratic function over alpha values, thus minimizing the total cost could be achieved by solving a sparse linear system.

Specifically, the matte is calculated as

$$\alpha = argmin\ \bar{\alpha}^T L \bar{\alpha}, \ \ s.t.\ \alpha_{bi} = s_{bi} \quad (12)$$

where $\bar{\alpha} = [1, 0, \alpha_{z1}, ..., \alpha_{zn}, \alpha_{b1}, ..., \alpha_{bm}]^T$, and $z1, ..., zn$ are all pixels in U_I^t, and $b1, ..., bm$ are boundary pixels. s_{bi} represents the boundary condition. In our system if the boundary pixel bi is on the outer boundary of the unknown region then $s_{bi} = 0$, otherwise it is set to be $\max(\alpha_{bi}^m, \alpha_{bi}^s)$.

L is a sparse, symmetric, positive definite Laplacian matrix with dimensions $(2+n+m) \times (2+n+m)$, given by

$$L(1, \alpha_z) = -\delta_z w_z^{mat} \alpha_z^m - (1-\delta_z) w_z^{cmp} \alpha_z^c \quad (13)$$

$$L(0, \alpha_z) = -1 - L(1, \alpha_z) \quad (14)$$

$$\begin{aligned} L(\alpha_z, \alpha_v) = -\Bigg\{ &(1 - \delta_z \delta_v)\left[1 - exp(-d_t(z,v))\right] + \\ &\sum_k^{(z,v) \in w_k} \left[\frac{\delta_z \delta_v}{9}(1 + (I_z - u_k)^T (\Sigma_k + \frac{\epsilon \cdot I_3}{9})^{-1}(I_v - u_k)\right] \Bigg\} \end{aligned} \quad (15)$$

$$L(\alpha_z, \alpha_z) = -\left[L(1, \alpha_z) + L(0, \alpha_z) + \sum_{v \neq z} L(\alpha_z, \alpha_v)\right] \quad (16)$$

Specifically, $L(1, \alpha_z)$ and $L(0, \alpha_z)$ are derived from the data terms in Equation 11, and $L(\alpha_z, \alpha_v)$ is derived from the neighborhood terms in Equation 11. Note that the first term in $L(\alpha_z, \alpha_v)$ is derived from J_z^{cmp}, and the second term is derived from J_z^{mat}, which is provided in [6]. In this term $u_k(3 \times 1)$ and $\Sigma_k(3 \times 3)$ are color mean and variance in the local window, and I_3 is the 3×3 identity matrix. Note that the parameters a and b in Equation 4 have been eliminated in the final optimization, as demonstrated in [6].

The definition of $L(\alpha_z, \alpha_z)$ ensures that every row of L sums to zero. In contrast to the "matting Laplacian" defined in [6], we call the matrix L defined here as *compositional matting Laplacian*.

Once the matte converges, we use locally sampled background colors to estimate a background color B_z for a pixel z whose alpha value is between 0 and 1, and then use Equation 1 to generate the final composite.

4. Comparisons

We first compare the proposed algorithm with previous matting and compositing approaches on a synthetic example shown in Figure 2. More comparisons on real images

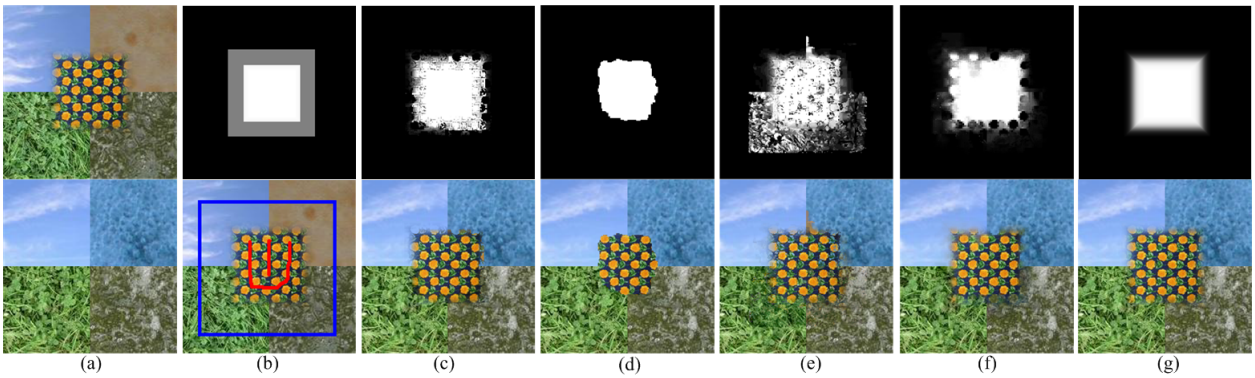

Figure 2. (a). The original image I(top) and the new background I'(bottom). (b). User input: a trimap for Bayesian matting (top) and paint strokes for scribble-based systems. (c)-(f): Matte and composite created by Bayesian matting(c), photomontage(d), iterative BP matting(e) and closed-form matting(f). (g). The ground truth matte and composite.

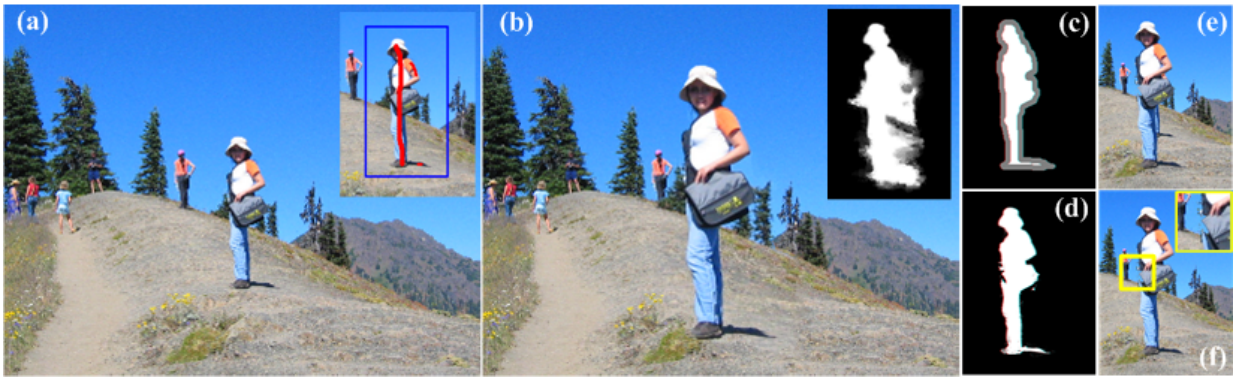

Figure 4. (a). Zoomed original image and user inputs. (b). Novel composite created by our system. (c). Trimap for Bayesian matting. (d). Matte computed by Bayesian matting. (e). Composite with matte in d. (f). Composite with matte in d after hole filling still contains visual artifacts.

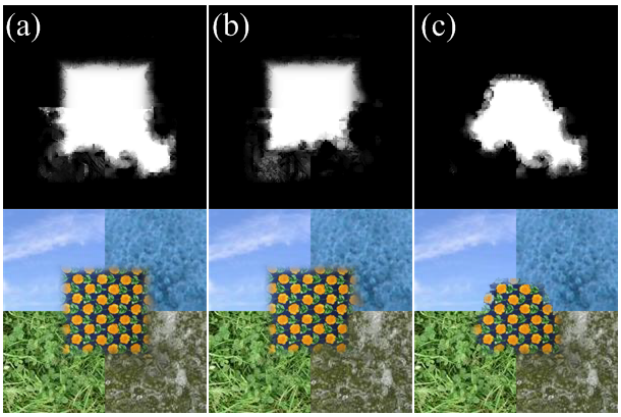

Figure 3. (a). Matte and composite created by our system. (b). The results when the compositing cost is disabled. (c). The results when the matting cost is disabled.

will be shown later. Figure 2a shows the original image I where the foreground texture is composed onto a background texture using a pre-defined matte. Below this is the new background image B' where the bottom half is similar but the top has changed.

Figure 2b-e shows that previous approaches such as Bayesian matting [3], photomontage [1], iterative BP matting [15] and closed-form matting [6] all have difficulties dealing with this data set. Given the complex foreground and background patterns, these algorithms fail to extract an accurate matte thus the final composites are erroneous.

Figure 3a shows the composite generated by our system, which has higher visual quality than composites created by others, and is quite similar to the ground truth shown in Figure 2g. This is achieved by implicitly treating different regions in different ways. For the bottom half of the image where old and new backgrounds are similar, the front propagation stops earlier when it finds a good transition, thus the hard problem of finding an accurate matte in this region is avoided. For the upper half of the image where the old and new backgrounds are different, our algorithm works in a similar fashion as traditional matting algorithms to try to extract a good matte for the foreground.

Figure 3b shows the results if we disable the compositing cost in Equation 11 by letting all δ_zs to be 1. And Figure

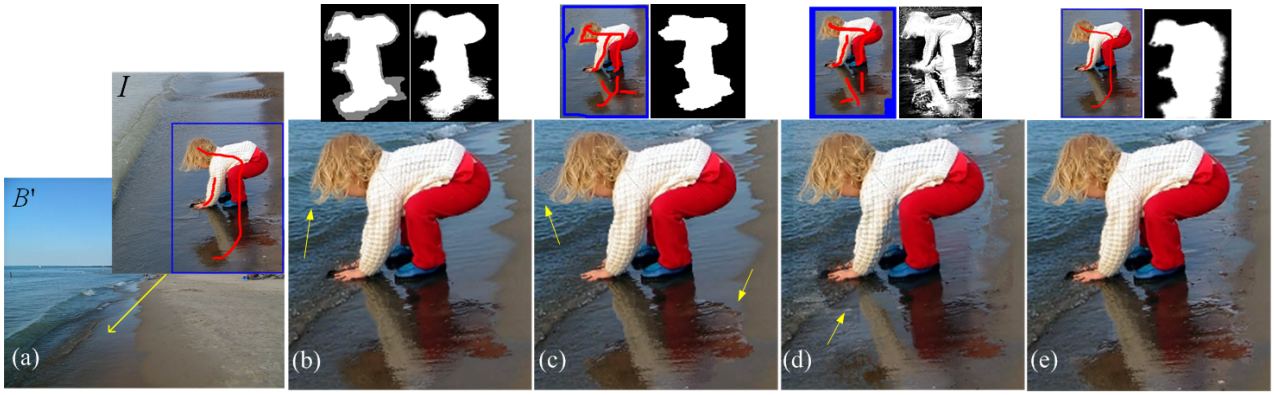

Figure 5. (a). I, B' and user input. (b)-(e). Input strokes, extracted matte and composite of Bayesian matting, photomontage, iterative matting and our system, respectively. Yellow arrows highlight artifacts.

3c shows the results of disabling the matting cost. These results demonstrate that minimizing a single cost is not sufficient for creating a successful composite.

5. Foreground Zooming

We apply the compositional matting algorithm to the task of recomposing a single image by varying the size ratio between the foreground and background within a single image. In general, we set a higher display ratio for the foreground relative to the background to emphasize the foreground. In this case, both the image I and the new background B' are differently scaled versions of the original photograph. This result is similar to virtually pulling the foreground towards the camera, as shown in Figure 4.

Using previous methods, one could achieve this by first extracting a high-quality matte for the foreground. Then hole filling methods would be needed to repair the background. One could then compose a scaled up version of the foreground matte onto the background. However, extracting a perfect matte for the foreground is difficult for general images, as is hole filling, and the composed image may contain visual artifacts.

For example, to attempt to use Bayesian matting to enlarge the foreground in Figure 4a, the user needs to specify a good trimap as shown in Figure 4c, which generates the matte in Figure 4d and results in a composite in Figure 4e. We can see "ghost" artifacts since the enlarged foreground does not fully cover the original foreground. We then use the image inpainting technique proposed in [16] to fill the holes on the background, resulting in a better composite in Figure 4f. However the errors in the matte estimation still cause noticeable visual artifacts.

In our system instead of using image inpainting techniques to fill-in the holes, we simply modify one step of our algorithm to avoid introducing holes as shown in Figure 4c. Once we calculate α_z^m in Equation 2 for pixel z, we find its corresponding location z' on the new background B'. If $\alpha_{z'}^m$ is smaller than α_z^m, we then let $\alpha_{z'}^m$ equals to α_z^m. In other words, we set high alpha values to pixels insides holes to encourage them to be occluded in the final composite. In this way our system achieves hole filling, foreground matting and compositing in a single optimization procedure.

Figure 6. The matte and composite generated by our system when the new background is provided as a solid blue.

6. More Results

Figure 5 compares different approaches on extracting the foreground from I and composing it onto B'. It shows that our system is able to create a more satisfying composite than previous approaches. Additionally, Figure 6 shows that if we use a totally different new background such as a solid blue, our system will try to extract an accurate matte since no useful new background information can be used in this case. This demonstrates that our system will work just as a normal matting algorithm when the new background is different from the original one instead of generating unpredictable results.

Figure 7 shows another example where we want to create a more impressive waterfall from the original one. We stretch the waterfall in the horizontal direction and recompose it onto the original image. Using previous matting approaches to achieve this is particularly hard since the foreground object is semi-transparent thus creating a trimap for

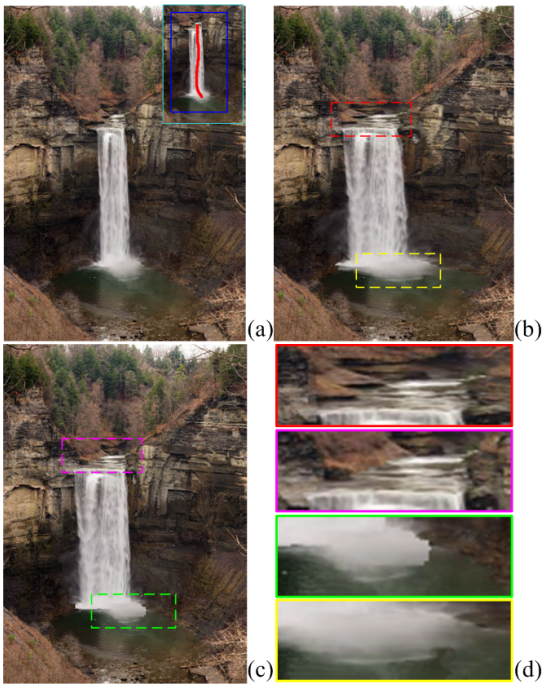

Figure 7. (a). Scaled original image with user inputs to expand the waterfall. (b). Composite generated by our system. (c). Composite generated by the photomontage system. (d). Details of composites.

matting is erroneous. Instead, we compare our system with the photomontage system. Since photomontage cannot deal with partial coverage, the composition generated from it contains more visual artifacts than the one generated from our system.

Figure 8 compares our system with the image retargeting system [8]. In the zoomed out fish-eye image created by the image retargeting system the mountain behind the person is unacceptably distorted. As shown in Figure 8c, a true fish-eye image is even worse since the foreground character is also unacceptably distorted. In contrast, our system is able to enlarge the foreground while keeping both the foreground and the background in as original a state as possible.

Although our system works well on most of the examples we have tested, it does not always give satisfying composites. When the new background differs significantly from the original one, the compositional matting faces the same difficulties as traditional matting algorithms do. Difficult and successful examples are shown in Figures 10 and 9. In the original image, Figure 9a, the foreground is so similar to the background that extracting an accurate matte is almost impossible. However, if the new background is similar to the original one, our system is able to create a good composite as shown in 9d. If unfortunately, the new background is substantially different from the original one, our system along with previous approaches all fail to give

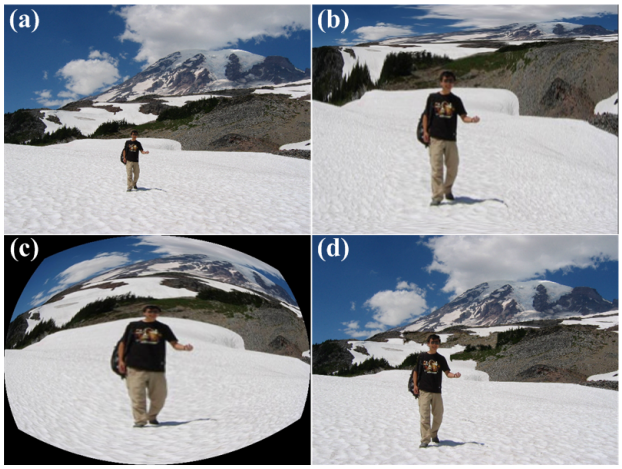

Figure 8. (a). Scaled original Image. (b). Simulated fish-eye image used by the image retargeting system. (c). A normal fish-eye image. (d). Enlarging foreground by 1.9 times using our system.

good composites, as shown in Figure 10.

7. Conclusion

A key lesson to take from our work is that such image processing methods should take advantage of all information known in a real application. Matting in the absence of the knowledge of the new background may not describe the full task.

In this paper we have demonstrated a compositional matting algorithm by taking the advantage of knowing the new background image which the foreground is to be composed on to. Experimental results show that our algorithm outperforms previous proposed matting and composition algorithms when the new background has similar regions with the old one. Based on the new matting algorithm we show how to recompose images by displaying foreground and background with different scales.

In the future we hope to consider how one might take temporal coherence into account for recomposing video objects. One could create a similar unified optimization framework, but computational considerations would certainly need to be addressed.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *Proceedings of ACM SIGGRAPH*, pages 294–302, 2004.

[2] W. A. Barrett and A. S. Cheney. Object-based image editing. In *Proceedings of ACM SIGGRAPH*, pages 777–784, 2002.

Figure 9. (a). Original image and matte extracted by Beyesian matting. (b). Composition created by Bayesian matting. (c). Composition created by our algorithm.

[3] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *Proceedings of IEEE CVPR*, pages 264–271, 2001.

[4] Y. Guan, W. Chen, X. Liang, Z. Ding, and Q. Peng. Easy matting. In *Proceedings of Eurographics*, 2006.

[5] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum. Drag-and-drop pasting. In *Proceedings of ACM SIGGRAPH*, 2006.

[6] A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *Proceedings of IEEE CVPR*, 2006.

[7] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *Proceedings of ACM SIGGRAPH*, pages 303–308, 2004.

[8] F. Liu and M. Gleicher. Automatic image retargeting with fisheye-view warping. *Proceedings of ACM UIST*, pages 153–162, 2005.

[9] P. Prez, M. Gangnet, and A. Blake. Poisson image editing. In *Proceedings of ACM SIGGRAPH*, pages 313–318, 2003.

[10] L. J. Reese and W. A. Barrett. Image editing with intelligent paint. In *Proceedings of Eurographics*, pages 714–724, 2002.

[11] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cut. In *Proceedings of ACM SIGGRAPH*, pages 309–314, 2004.

[12] M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *Proceedings of IEEE CVPR*, pages 18–25, 2000.

[13] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. In *Proceedings of ACM SIGGRAPH*, pages 315–321, 2004.

Figure 10. A failure example. From left to right: composing the foreground in Figure 9a onto a substantially different background using our system; matte extracted by our system; composition created by Bayesian matting.

[14] V. Vezhnevets and V. Konouchine. Grow-cut - interactive multi-label n-d image segmentation. In *Proceedings of GraphiCon 2005*, 2005.

[15] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of ICCV 2005*, pages 936–943, 2005.

[16] H. Yamauchi, J. Haber, and H.-P. Seidel. Image restoration using multiresolution texture synthesis and image inpainting. *Proceedings of Computer Graphics International*, pages 120–125, 2003.