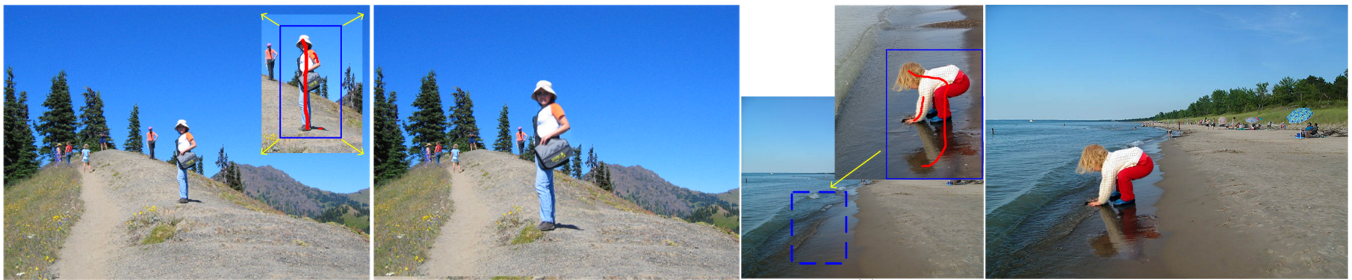
Simultaneous Matting and Compositing

Jue Wang*
University of Washington

Michael F. Cohen†
Microsoft Research



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

Figure 1: A new matting algorithm combines matting and compositing into a single optimization process. Using the algorithm we enlarge the foreground in (a) by 1.8 times and create a novel composition (b), or seamlessly compose the girl in (c) onto a new background image as shown in (d), given only a small amount of user interaction.

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.

This observation was motivated by the problem of recomposing the elements within a single image or within a set of related images. For example, many cameras are by default set for wide angle shots to successfully capture landscapes. Foreground characters often appear quite small relative to the image frame. This issue is exacerbated when one wants to display the image on a small mobile devices.

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 rescale or rearrange the composition. We compare and contrast our method with Bayesian Matting, Iterative Matting, Photomontage, and the Image Retargetting 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 recompose 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 composition. We dub this new matting algorithm *compositional matting*.

Besides composing the foreground onto a new background as shown in Figure 1c and d, we also explore the advantages of the proposed algorithm for reorganizing 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 composition 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 1a and b, 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.

The same method allows composing foreground elements onto new backgrounds. The advantage of compositional matting is most obvious when the new background resembles the old one. When the new background is quite different, the algorithm gracefully degrades to behave like previous methods.

2 Related Work

2.1 Foreground Segmentation

The problem of extracting foreground objects from images has been studied extensively over the last 20 years. The simplest solution is to segment the image by selecting all pixels that match a user-specified image feature such as color. Photoshop's magic wand [INCORP. 2002] takes this approach. It lets users select pixels that match a range of colors. However this approach requires a large amount of user interaction.

Recently, many systems have been developed to accurately identify the foreground regions with minimal user guidance. Intelligent

*e-mail: juew@u.washington.edu
†e-mail:Michael.Cohen@microsoft.com

paint [Reese and Barrett 2002] and the object-based image editing system [Barrett and Cheney 2002] first oversegment the image and then let the user select the regions that form the foreground object. LazySnapping [Li et al. 2004] and GrabCut [Rother et al. 2004] systems provide interactive graph-cut-based segmentation solutions. The GrowCut system [Vezhnevets and Konouchine 2005] 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 image.

Our system employs a similar paint stroke interface for users to roughly specify the foreground object. However, since we directly optimize the final composition instead of extracted foreground, our system does not require accurate foreground boundary identification which can be extremely hard for images with complex foreground and background patterns.

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 [Ruzon and Tomasi 2000] show how to estimate the alpha matte and foreground color using statistical methods. Chuang et al. [Chuang et al. 2001; Chuang et al. 2002] extend this approach by using Bayesian statistics to accurately estimate alpha for partially transparent foreground objects in both images and video. The GrabCut system [Rother et al. 2004] includes a simpler technique called border matting which assumes a strong model for the alpha profile to quickly estimate the alpha matte and produce a smooth matte. This technique has been extended to video objects in the video cutout system [Wang et al. 2005]. In the newly proposed iterative matting system [Wang and Cohen 2005], the problems of foreground segmentation and matting have been combined together by an iterative optimization process. However, it only works for images with distinct foreground and background color distributions and cannot generate good mattes for most examples shown in this paper.

Our system employs a similar iterative framework as the iterative matting approach [Wang and Cohen 2005]. However, our compositional matting algorithm has unique optimization goals and additional inputs compared with all previous approaches.

2.3 Image Composition

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 [Agarwala et al. 2004] for this task by using graph-cut optimization. However, their system only calculates hard segmentation of source images thus is not capable of handling partial foreground coverage. The Poisson image editing system [Prez et al. 2003] 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.

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 final composed image, although it may not be the true matte for the foreground.

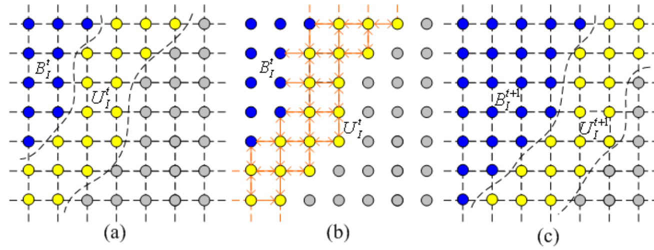


Figure 2: Our algorithm solves the composition in a front-propagation fashion. (a). Dilating background region B_I^t (blue) to create an unknown region U_I^t (yellow). (b). Finding the composition for U_I^t by solving a graph labeling problem using belief propagation. (c). Regions are evolved for the next iteration.

2.4 Image Retargeting

Image retargeting refers to the problem of adapting images for display on devices different than originally intended. Liu and Gleicher recently have proposed a retargeting algorithm [Liu and Gleicher 2005] by using non-linear 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 retargetting goal, and only alter a small region around the foreground to eliminate the compositing seams without significant distortion as shown in Figure 7.

3 The Compositional Matting Algorithm

3.1 Basic Formulations

In the traditional matting problem, the observed image I_z ($z = (x, y)$) is modeled as a linear combination of foreground image $F(z)$ and background image $B(z)$ by an alpha map:

$$I_z = \alpha_z F_z + (1 - \alpha_z) B_z \tag{1}$$

where α_zs can be any value in $[0, 1]$, which we call a *matte*. F_zs and B_zs are foreground and background colors, respectively. Note that α_z, F_z and B_z are all unknown, thus the problem is intrinsically under-constrained.

In this work we assume the new background image I', which the extracted foreground will be composed onto, is known. The final composed image thus can be calculated as

$$I_z^* = \alpha_z F_z + (1 - \alpha_z) I'_z \tag{2}$$

By combining Equation 1 and 2 we have

$$I_z^* = I_z + (1 - \alpha_z)(I'_z - B_z) \tag{3}$$

The above equation demonstrates that by combining matting and compositing together, we can calculate final compositing image by only estimating two unknowns α_z and B_z. This is one of the major advantages of having the new background known a priori.

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

3.2 Front Propagation for Matte Estimation

Inspired by the advantages of combining matting and compositing together, we create a new iterative matting system called *compositional matting*.

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 1a. 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 goal of the algorithm is to estimate a matte M_I for U_I which allows the foreground to be seamlessly composed onto the new background image 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 $U_I^{t+1} = U_I^t$. One can imagine that the unknown region is shrinking until becoming stable, as illustrated in Figure 2.

The advantage of this front propagation algorithm is that in each iteration, the unknown region U_I^t is close to the known background region B_I, thus for a pixel z inside U_t, we can sample background pixels in its local neighborhood area to get an accurate estimation of background color B_z in Equation 3. This allows us to accurately predict the final compositing colors for different possible alpha values. Also, the front propagation can stop immediately whenever it finds a good composition that avoids visual artifacts resulting from inaccurate alpha values and foreground colors for real foreground pixels.

3.3 Solving for the Matte

To seamlessly compose the foreground region I'_F onto the new background I', our a priori expectation for the matte M_F is two-fold. First, we want the foreground object to be as accurate as possible in the final composition, which we call the *semantic constraint*. Secondly, we expect the final composed image to be "natural" and contain no sudden visual discontinuities, which we call the *seam constraint*. In our approach, these two constraints are explicitly treated as optimization objectives. We define a *data energy* for a pixel in U_I^t to respect the semantic constraint, and a *neighborhood energy* for a pixel pair to satisfy the seam constraint.

3.3.1 The Data Energy

We discretize the possible alpha value to 15 levels between 0 and 1, denoted as $\alpha^k, k = 1, ..., 15$. We calculate an energy for each alpha level measuring how well the alpha level fits with the true alpha value at the current pixel.

Similar to previous matting approaches, we first seek for each pixel z in U_I^t a local estimation of the foreground and background color. Given the nature of the front propagation algorithm, we can 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 [Wang and Cohen 2005] 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^s is calculated as

$$\alpha_z^s = \frac{d(I_z, \overline{B_z}, \Sigma_z^B)}{d(I_z, \overline{F_z}, \Sigma_z^F) + d(I_z, \overline{B_z}, \Sigma_z^B)} \tag{4}$$

where $d(I, \overline{C}, \Sigma)$ is the Manhattan distance from color I to a Gaussian $G(\overline{C}, \Sigma)$. Note that although the foreground color estimation may not be very accurate, it is not directly used in our system for computing the final composition.

Despite estimating alpha values from color samples, we also examine the color difference between I'_z and I_z, and calculate a second estimated alpha value as

$$\alpha_z^d = exp\left(-\left(I'_z - I_z\right)^T \Sigma_z^{B-1}\left(I'_z - I_z\right)\right) \tag{5}$$

The idea is that if the color difference between I'_z and I_z is small, this is a good place to transition from I' to I. We then set high alpha values for p_z to force the front propagation to stop here. The final estimated target alpha value is calculated as $\alpha_z^E = max\{\alpha_z^s, \alpha_z^d\}$.

Once the estimated alpha value is determined, the energy for each alpha level α^k is calculated as

$$E_d(\alpha_z^k) = \left|\alpha^k - \alpha_z^E\right|^2 / \sigma_\alpha^2 \tag{6}$$

where σ_α is the covariance which we set to be 0.2 in our system.

3.3.2 The Neighborhood Energy

The neighborhood energy encourages the final compositing image to be locally smooth. For two neighboring pixels z and v, given their alpha levels α_z^k and α_v^j and their sampled background colors, we can estimate their final composed color I_z^* and I_v^* by Equation 3. The neighborhood energy is calculated as

$$E_s(\alpha_z^k, \alpha_v^j) = \left|I_z^* - I_v^*\right|^2 / \sigma_s^2 \tag{7}$$

where σ_s^2 is calculated as $(tr(\Sigma_z^B) + tr(\Sigma_v^B))/6$ ($tr()$ is the trace operation) to respect the local image characteristics. If the local region where z and v lies is highly textured, then σ_s^2 will be large thus loosening the requirement for the color matching between $I^*(z)$ and $I^*(v)$ since the transition made in textured regions will be less noticeable.

Some previous image matting and compositing approaches also have defined neighborhood energies. The iterative matting approach [Wang and Cohen 2005] defines them based on the difference of alpha levels to encourage alpha values vary smoothly, but there is no guarantee that the final composition will not contain artifacts. In contrast, our neighborhood energy directly minimizes the discontinuities in the final composition. The photomontage approach [Agarwala et al. 2004] also defines a neighborhood term based on color differences between pixels of the composed image, however it only allows the alpha values to be 0 or 1 thus is not capable of handling partial foreground coverage.

3.3.3 Energy Minimization

The total energy is defined as a combination of the data and neighborhood energy:

$$E = \sum_z E_d(\alpha_z^k) + w \cdot \sum_{z,v} E_s(\alpha_z^k, \alpha_v^j) \tag{8}$$

where w is the weight balancing the data and neighborhood energy. We set w to be lower when the foreground and background colors are similar where the data cost is less reliable, and set it to be higher when the foreground and background have distinct color distributions. By defining the total energy we transfer the matting problem into a graph labeling problem, and the total energy is minimized by choosing the proper alpha level α_z^k for pixel z.

Similar energy minimization formulations have been proposed in previous segmentation, matting and composition approaches, and optimization algorithms such as min-cut [Boykov et al. 2001] and loopy belief propagation [Weiss and Freeman 2001] have been extensively used to find approximate solutions to them. In our system

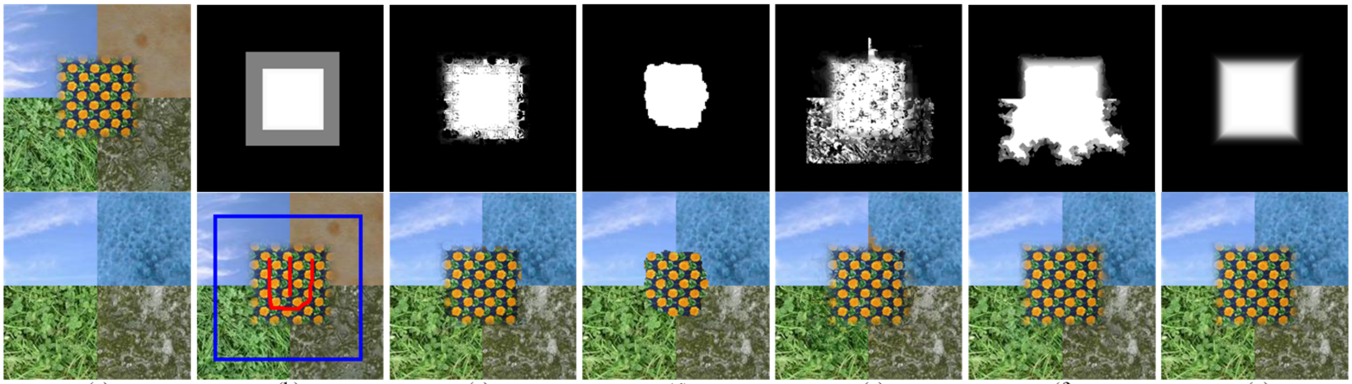


Figure 3: (a). The original image I(top) and the new background I'(bottom). (b). User input: a trimap for Bayesian matting (top) and paint strokes for photomontage, iterative matting and our system (bottom). (c). Matte and composition created by Bayesian matting. (d). Matte and composition created by photomontage. (e). Matte and composition created by iterative matting. (f). Matte and composition created by our system. (g). The ground truth matte and composition.

since the energy defined in Equation 7 does not necessarily satisfy the energy constraints for min-cut algorithm [Kolmogorov and Zabih 2004]. We thus chose to use loopy belief propagation to approximately minimize the energy.

Specifically, as shown in Figure 2, pixels in U_I^t (shown in yellow) and boundary pixels in U_B^t are included in the graph. By using belief propagation, each pixel in the graph sends messages to its neighbors, and finally at each pixel the alpha decision is made based on the property of the current pixel and all the messages it receives from it neighbors. A good practical introduction to loopy belief propagation can be found in [Sun et al. 2005].

3.4 Automatic Refinement

Up to this point, composing the foreground onto a new background has relied on the user to set the exact size and position of the foreground object on the new background. The composition result can be further improved by allowing the system slightly alter the scale and position of the foreground to find a better composition based on the user's rough specification.

We find a better size and position by first creating an approximate foreground segmentation. We then vary the size and position examining the difference between a small shell around the foreground and the new background. We choose a size and position that minimizes this difference, and then run the compositional matting.

This process is illustrated in Figure 10. In Figure 10b, after the foreground region is specified by the user, we first apply a graph-cut based binary segmentation [Li et al. 2004] in the foreground bounding box. A dilation of the foreground segment creates a boundary region, as shown as the yellow region in Figure 10b. The average pixel difference between the foreground image and the new background in the boundary region is used as the indicator for the fit of the current size and location. We sample the scale and translation in a small range around the user specified values (the purple rectangle in Figure 10b), and record the scale and position that minimizes the pixel difference (shown as the red rectangle in Figure 10b). The expectation is that when the pixel difference is small in the boundary region, the old background and new background will result in a good match, thus our algorithm better take advantage of the new background.

3.5 Comparisons

We first compare the proposed algorithm with previous matting and compositing approaches on a synthetic data set shown in Figure 3. More comparisons on real images will be shown later. Figure 3a shows the original image I where the foreground textured is composed onto a background texture using a pre-defined matte. Below this is the new background image I' where the bottom half is similar but the top has changed.

Figure 3b-e shows that previous approaches have difficulties dealing with this data set. Bayesian matting [Chuang et al. 2001] cannot generate a good matte for the bottom half of the foreground since the foreground and background patterns are complex, thus in the final composition the foreground edges are destroyed. The photomontage system [Agarwala et al. 2004] is not able to deal with partial coverage thus the composition contains visual discontinuities. The iterative matting approach [Wang and Cohen 2005] also cannot generate a good matte for this complex image thus the final composition are erroneous.

Figure 3f shows the composition generated by our system, which has significantly higher visual quality than compositions created by other approaches. Our algorithm achieves this 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 composition, 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 previous matting algorithms to extract a good matte for the foreground. Figure 3g shows the ground truth which is quite similar to our result.

4 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 I' are differently scaled versions of the original photograph. This result is similar to virtually pulling the foreground towards the camera.

Using previous methods, one could achieve this by first extracting a high-quality matte for the foreground object. Then hole filling methods would be needed to repair the background. One could

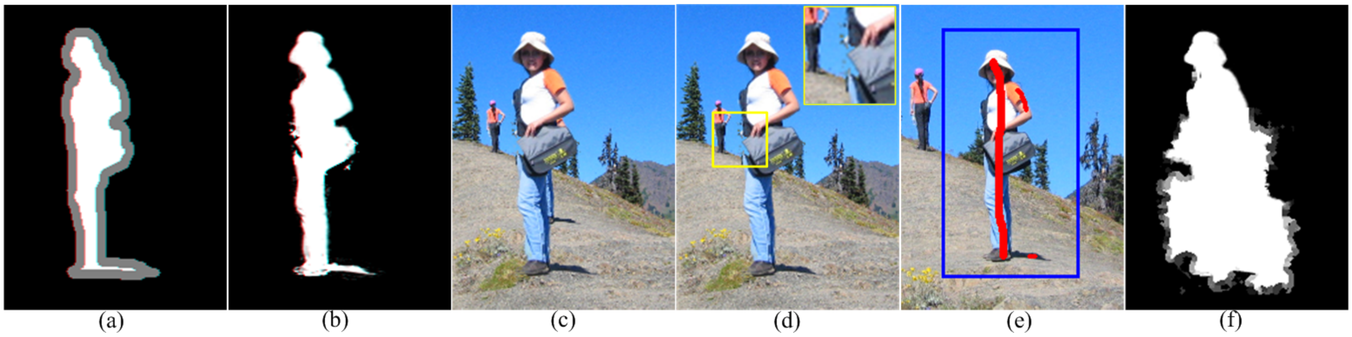


Figure 4: (a). The user specified trimap for example in Figure 1a. (b). Extracted matte by Bayesian matting. (c). Composition with matte b. (d). Composition with matte b after hole filling. (e). The input to our system. (f). The matte extracted by our system to create the composition in Figure 1b.

Figure 5: (a). Scaled original image. (b). Composition created by Bayesian matting with provided trimap and extracted matte. (c). Composition created by the photomontage system with input strokes and extracted matte. (d). Composition created by our system with input strokes and extracted matte. Yellow arrows highlight artifacts.

then compose a scaled up version of the foreground matte onto the background. However, extracting a perfect matte for the foreground object is difficult for general images, as is hole filling, and the composed image may contain visual artifacts.

For example, we attempt to use Bayesian matting to enlarge the foreground shown in Figure 1a, the user needs to specify a good trimap as shown in Figure 4a. Based on this trimap Bayesian matting approach generates the matte in Figure 4b, resulting in a composition in Figure 4c. 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 [Yamauchi et al. 2003] to fill the holes on the background, resulting in a better composition in Figure 4d. However, as shown in the highlighted region, the errors in the matte estimation still cause noticeable visual artifacts.

In constrast, our compositional matting algorithm utilizes the advantages the new background provides to generate a good composition in one pass, as shown in Figure 1e. The user input and the extracted matte are shown in Figure 4e and f, respectively. Note that our system requires much less user input than Bayesian matting to generate a better composition. More results and comparisons are demonstrated in Section 5.

One thing that should be mentioned is that 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 the estimated alpha value in Equation 4 for pixel z, we find its corresponding location z' on the new background I'. The correspondence is naturally built in the image lattice since both I and I' are differently scaled versions of the same photograph. If $\alpha^s_{z'}$ is smaller than α^s_z, we then let $\alpha^s_{z'}$ equals to α^s_z. In other words, we assign high initial alpha values to pixels insides the holes to encourage them to be occluded in the final composition. In this way our system achieves hole filling, foreground matting and compositing in a single optimization procedure.

5 Results

Figure 5 shows another example of enlarging the foreground. The compositions generated by Bayesian matting and photomontage both contain significant visual artifacts on the foreground object since accurately segmenting the foreground is difficult for this image. Our system cannot extract an accurate matte either, however it still finds a good matte to create a successful composition. Note how our system capture the soft shadow on the ground to make the composition more realistic.

Figure 6 shows another example where we want to create a more impressive waterfall from the original one. To do this 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

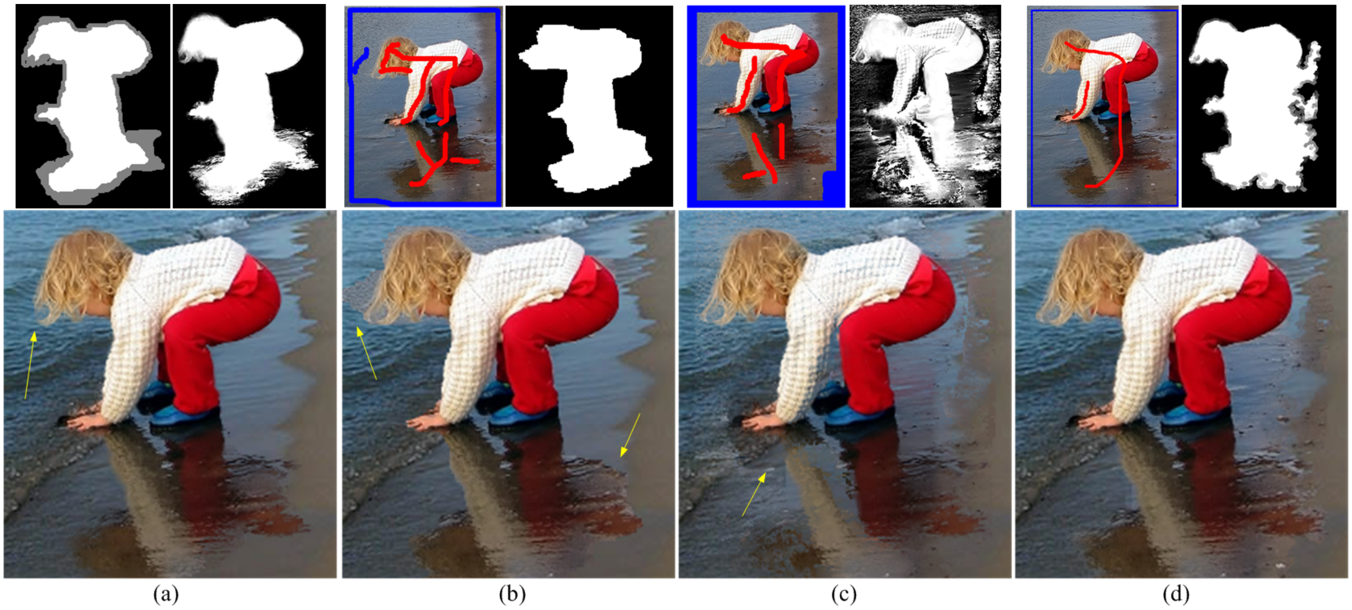


Figure 8: Comparisons on the example shown in Figure 1c. (a). Input trimap, extracted matte and composition of Bayesian matting. (b). Input strokes, extracted matte and composition of photomontage. (c). Input strokes, extracted matte and composition of iterative matting. (d). Input strokes, extracted matte and composition of our system. Yellow arrows highlight artifacts.

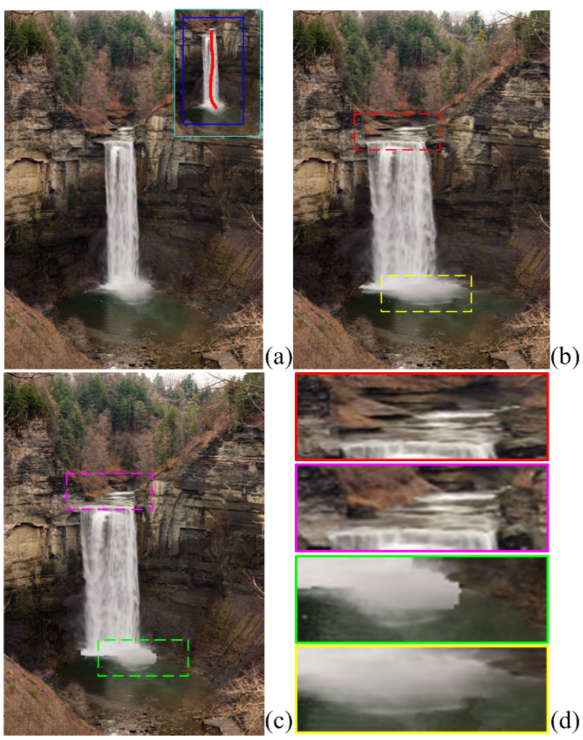


Figure 6: (a). Scaled original image with the user's input to expand the waterfall. (b). Composition generated by our system. (c). Composition generated by photomontage system. (d). Details of compositions.

creating a trimap for matting is erroneous. Instead, we compare our system with the photomontage system. Since photomontage cannot deal with partial coverage, the composition generated from it con-

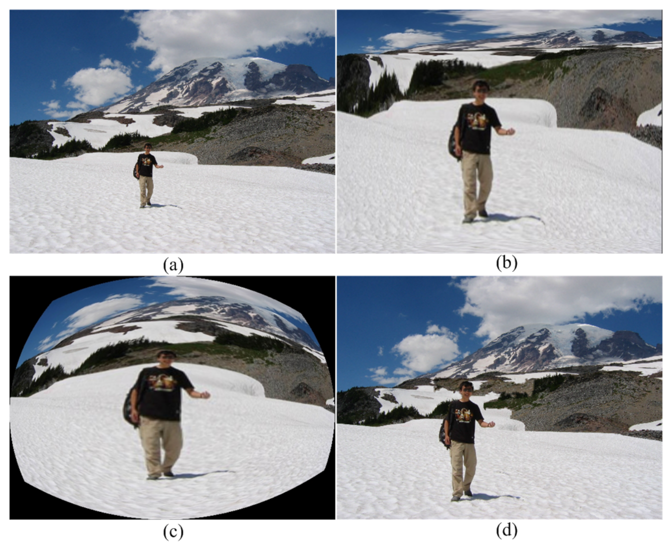


Figure 7: (a). Scaled original Image. (b). Simulated fish-eye image used by the image retargeting system. (c). A normal fish-eye image. (d). Enlarging foreground by 1.9 times using our system.

tains more visual artifacts than the one generated from our system.

Figure 7 compares our system with the image retargeting system [Liu and Gleicher 2005]. 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 7c, 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.

Figure 8 compares different approaches on extracting the foreground shown in Figure 1c and composing it onto the new back-

(a) (b) (c)

Figure 10: (a). Top: Original image. Bottom: new background. (b). Top: user input and graph-cut segmentation result. The yellow region Is used to deteremine the best position and scale for the foreground. Bottom: user specified initial composition position (purple rectangle) and automatically refined position (red rectangle). (c). Final composition. The purple patch is from the composition without local stitch, and the red patch is from the one with local stitch. Artifacts are highlighted by yellow arrows.

Figure 9: The matte and composition generated by our system when the new background is provided as a solid blue.

ground. It clearly demonstrates that our system is able to create a more satisfying composition than previous approaches. Additionally, Figure 9 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 10a shows a photograph of a girl standing before a beautiful landscape however the white fence is quite distracting. Using our system the user can take another picture without the fence and easily compose the foreground to the new background. We also demonstrate the local foreground stitching algorithm as a pre-processing step to improve the final composition.

5.1 Limitations

Although our system works well on most of the examples we have tested, it does not always give satisfying compositions. When the new background differs significantly from the original background, the compositional matting has few advantages over older methods. Difficult and successful examples are shown in Figures 12 and 11. In the original image, Figure 11a, 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 composition as shown in 11d. If unfortunately, the new background is substantially different from the original one, our system along with previous approaches all fail to give good compositions, as shown in Figure 12.

6 Timings

The compositional matting algorithm we proposed runs in roughly 20 seconds for a foreground object of around 200 by 300 pixels in size. The processing time will increase linearly as the foreground increases in size. To place the method in context, it is slower than Bayesian matting and Photomontage, and is faster than the iterative matting approach. The local foreground stitching can improve the composition results, but itself is computational expensive and will add additional processing time to our system if turned on.

However, in our tests we found that our system is quite efficient in terms of total time compared with other approaches. As shown in our examples our system takes the minimum user input to generate a satisfying result, thus our system has advantages in terms of user time to other matting approaches which require an accurate trimap to generate a good result. Also, when recomposing an enlarged foreground onto the original background, our system combines hole filling, foreground matting and composition into a single process. Other techniques require additional hole filling operations, which substantially slow the overall process.

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

Figure 11: (a). Original image. Extracting a good matte for the foreground is very hard in this case. (b). Matte extracted by Beyesian matting. (c). Composition created by Bayesian matting. (d). Composition created by our algorithm.

Figure 12: A failure example. From left to right: composing the foreground in Figure 11a onto a substantially different background using our system; matte extracted by our system; composition created by Bayesian matting.

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

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

BARRETT, W. A., AND CHENEY, A. S. 2002. Object-based image editing. In *Proceedings of ACM SIGGRAPH*, 777–784.

BOYKOV, Y., VEKSLER, O., AND ZABIH, R. 2001. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Analysis and Machine Intelligence 23*, 11, 1222–1239.

CHUANG, Y.-Y., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2001. A bayesian approach to digital matting. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition*, 264–271.

CHUANG, Y.-Y., AGARWALA, A., CURLESS, B., SALESIN, D., AND SZELISKI, R. 2002. Video matting. In *Proceedings of ACM SIGGRAPH*, 243–248.

INCORP., A. S. 2002. Adobe photoshop user guide.

KOLMOGOROV, V., AND ZABIH, R. 2004. What energy functions can be minimizedvia graph cuts? *IEEE Trans. on Pattern Analysis and Machine Intelligence 26*, 2, 147–159.

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

LIU, F., AND GLEICHER, M. 2005. Automatic image retargeting with fisheye-view warping. *Proceedings of ACM UIST*, 153–162.

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

REESE, L. J., AND BARRETT, W. A. 2002. Image editing with intelligent paint. In *Proceedings of Eurographics*, 714–724.

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

RUZON, M., AND TOMASI, C. 2000. Alpha estimation in natural images. In *Proceedings of IEEE CVPR*, 18–25.

SUN, J., YUAN, L., JIA, J., AND SHUM, H.-Y. 2005. Image completion with structure propagation. In *Proceedings of ACM SIGGRAPH*, 861–868.

VEZHNEVETS, V., AND KONOUCHINE, V. 2005. Grow-cut - interactive multi-label n-d image segmentation. In *Proceedings of GraphiCon 2005*.

WANG, J., AND COHEN, M. 2005. An iterative optimization approach for unified image segmentation and matting. In *Proceedings of ICCV 2005*.

WANG, J., BHAT, P., COLBURN, A. R., AGRAWALA, M., AND COHEN, M. F. 2005. Interactive video cutout. In *Proceedings of ACM SIGGRAPH*, 585–594.

WEISS, Y., AND FREEMAN, W. 2001. On the optimality of solutions of the max-product belief propagation algorithm in arebitrary graphs. *IEEE Trans. on Information Theory 47*, 2, 303–308.

YAMAUCHI, H., HABER, J., AND SEIDEL, H.-P. 2003. Image restoration using multiresolution texture synthesis and image inpainting. *Proceedings of Computer Graphics International*, 120–125.