Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Object-aware Gradient-Domain Image Compositing

Martin Eisemann[1] and Jan Kokemüller[1] and Marcus Magnor[1]

[1]Computer Graphics Lab, TU Braunschweig, Germany

Abstract

We describe an approach to suppress bleeding artifacts without altering the boundary location in gradient-domain compositing*, a technique to create seamless composites. While gradient-domain compositing has become a standard tool for many complex image editing tasks such as seamless cloning, panorama stitching or scene completion, its quality suffers from mismatches in the composited image regions. We propose an approach that is robust to non-optimal region selection by the user without altering his selection which may be neither intended nor possible for certain compositing tasks. In addition, we present an easy-to-use extension to composite interleaving objects. The usability of our approach is demonstrated by several image compositing tasks and comparisons to current state-of-the-art algorithms in gradient-domain image compositing are presented.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.3]: Picture/Image Generation—Computer Graphics [I.3.4]: Paint Systems—Computer Graphics [I.3.6]: Interaction techniques—

1. Introduction

Gradient-domain compositing has become an important technique in computer graphics in fields such as panorama stitching [Aga07, LZPW04, Sze06], seamless image composition [PGB03, ADA*04], video editing and enhancement [BZS*07, GBD*09, BZCC10], scene completion [HE07] and many other applications. It is nowadays "one of the most widely used algorithms in computational photography and video" [Aga07]. The idea is to first delineate the composited regions and then to copy gradients rather than colors from the source images into the composite. In a final step the color composite is reconstructed from the gradients by solving the discretized Poisson equation.

Unfortunately, however, the solution is only approximate as the gradient field may not be integrable and color spilling effects or halos may be the result. Gradient-domain compositing usually works best if the color difference between the source and target image along the boundary is constant or at least smooth. Existing work [LZPW04, JSTS06] therefore aims at optimizing the boundary between regions. This approach is not sufficient in several cases. First, the user may not want the specified region to be altered and second, automatic boundary optimization may not yield acceptable results if the combined regions are largely different.

Another problem encountered in gradient-domain compositing tasks is the composition of overlapping objects. In general the gradient of the source image is pasted on top of the target image without considering the underlying content. Pérez *et al.* [PGB03] proposed some artistic concepts to guide the merging, depending on the desired effect. Pasting the source image *behind* the meaningful content of the target image is not possible. Neither can the user specify which object in the composition should be in front of the other.

This paper describes a simple and efficient approach to minimize bleeding artifacts in gradient-domain image compositing tasks and to enable a multi-layered composition of overlapping or interleaving objects. We therefore propose a two-step algorithm. First, the critical regions, causing the color bleeding artifacts in the source and target images are detected and the Poisson equation is adjusted accordingly to prevent color transitions along these areas. After solving the adjusted Poisson equation, the overlapping areas of the foreground objects in source and target are detected. The ordering is randomly initialized and can then be adjusted by the user according to his needs by a simple point-and-click interface.

2. Gradient-domain Compositing

Since the seminal works of Pérez *et al.* [PGB03] image compositing in the gradient domain has become amazingly pop-

© The Eurographics Association 2011.

DOI: 10.2312/PE/VMV/VMV11/065-071

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

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

Figure 1: We present a seamless image-compositing technique which allows to composite complex interleaving objects without color bleeding artifacts and which is robust to selection inaccuracies. Using our method we are able to seamlessly place objects from the source image even *behind* objects in the target image. From left to right: (a) The source image and target image along with the user defined extraction mask. (b) The result using the method of Tao *et al.* [TJP10]. (c) Our result with the object placed in front. (d) Our result with the object placed in the back of the image.

ular. Generally, the task at hand is to seamlessly insert a part of a source image I_S into an equally shaped target area Ω in a target image I_T with a given boundary $\partial\Omega$. The basic idea behind Poisson image editing, or more generally gradient-domain reconstruction, is to search for an Image I that best fits the gradient field of ∇I_S with optional color constraints along the boundary $\partial\Omega$ from the target image I_T in a least-squares sense:

$$\underset{I}{\operatorname{argmin}} \iint_{\Omega} ||\nabla I - \nabla I_S||^2 \text{ with } I|_{\partial\Omega} = I_T|_{\partial\Omega} , \qquad (1)$$

where $\nabla . = [\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y}]$ is the gradient operator, $I|_{\partial\Omega}$ depicts the pixels along the boundary $\partial\Omega$. A unique solution to this problem can be found by solving the Poisson equation with Dirichlet boundary conditions:

$$\nabla^2 I = \nabla^2 I_S \text{ with } I|_{\partial\Omega} = I_T|_{\partial\Omega} \qquad (2)$$

While this approach works very well if the color offset along the boundary is constant or very smooth, many authors noted that a manually defined boundary may lead to noticeable reconstruction artifacts. As a result several approaches have been introduced to refine the boundary location [ADA*04, LZPW04, JSTS06]. The inherent problem with these is that it might not be possible to find good positions for the boundary, e.g. if structural misalignments are present. One way to overcome this problem is to smoothly warp the image content in order to make merging successful [JT03, JT08, EGM11]. However, warping is only possible if the task is to stitch similar objects in a common image domain. Also, unrealistic deformations may be the result. And quite often the user may simply not allow to alter the boundary or deform the content. In this case the only option is to alter the integration process. One approach in this direction was just recently proposed by Tao *et al.* [TJP10], which most resembles our own work. In their approach the curl of the boundary is used as a hint on where color bleeding might actually appear in the image and the gradient field is changed accordingly. Additionally, they control the integration residuals such that they are located in textured areas. While the underlying mathematical formulation is very elegant, color bleeding is still present. In contrast to the continuous formulation of Tao *et al.* we propose to use integration barriers which completely block color transitions in critical areas. As we will show in Section 4 this leads to visually superior results in many cases. This was also proposed by Farbman *et al.* [FHL*09] though they required user intervention to mark the regions causing color bleeding artifacts, while our approach can work fully automatically. Lalonde *et al.* [LHE*07] use a presegmentation of the objects before inserting them. In contrast our approach preserves the details of the background in the source image, which gives the user additional freedom for compositing.

However, none of the existing approaches is able to seamlessly insert a new object *behind* a foreground object in the target image. A big drawback of the approach by Tao *et al.* is that the user needs to choose the final position for insertion in the target image before he adjusts the mask. Moving the object afterwards is not possible without tedious adjustment of the mask. An interesting concept in this direction was presented by McCann and Pollard [MP09] called local layering. Instead of providing a single layer per object, the visibility is estimated on a local overlap basis. We will show how such a concept can be integrated into the seamless compositing task to achieve superior image composites which are difficult to achieve with other methods.

3. Our approach

Our approach consists of three main parts. First, we robustly determine foreground objects in both, the source and target image, Section 3.1. Next, we adjust the classic Poisson equation to prevent color bleeding and halos, but assure that the source still seamlessly merges with its subjacent image, Section 3.2. In a last step the user may decide which foreground

© The Eurographics Association 2011.

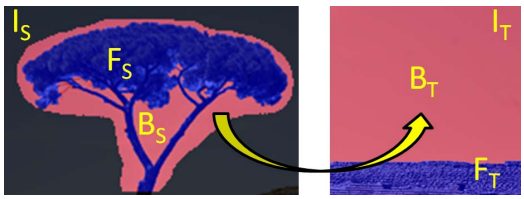

Figure 2: Foreground Estimation: Based on the user defined mask the fore- and background, F_S and B_S, respectively, in the source image I_S are estimated using the GrabCut technique. To robustly initialize the background segmentation in the target image I_T, we transfer the GMM of the source background B_S to the target background B_T.

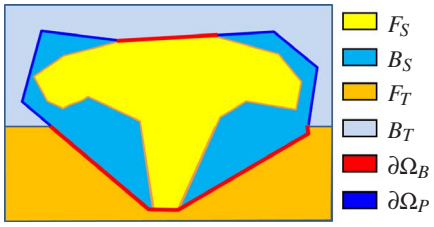

Figure 3: Seam classification: We detect those pixels $\partial\Omega_B$ which are potential candidates for color bleeding artifacts along the seam based on the constellation of fore- and background in source and target. Only the regions where B_S is connected to B_T are safe areas for the integration.

objects should be in front and which behind, based on a local layering concept. Section 3.3.

3.1. Foreground Estimation

The user starts by selecting a region from the source image I_S which he wants to seamlessly paste into the target image. To segment the source image patch into meaningful regions, we apply the GrabCut technique by Rother *et al.* [RKB04] giving us an estimate of the fore- and background regions in the selected patch. The GrabCut automatically models fore- and background as an optimized set of Gaussian Mixture Models (GMM) using iterative Graph Cuts [BVZ01]. Assuming that source and target have similar colored backgrounds, which is not an uncommon assumption, think of a blue sky as background, we extract the GMM from the source's background and apply it to the target's background estimate. For each pixel in the target image we compute its probability to belong to the extracted GMM. If it is higher than 0.5 the pixel is considered as belonging to the background, otherwise as belonging to the foreground. Evaluating the GrabCut again on the target image with this initialization leads to a robust segmentation in most cases. Obviously, such an initialization cannot always be free of errors. Therefore, as it was proposed by Rother *et al.*, we allow for manual correction of the foreground estimation by letting the user draw simple strokes on the source or target image pixels, constraining them either to be definitely foreground or definitely background. This is only necessary if the previously described automatic GrabCut estimation fails and generally takes no more than a few seconds per image. This procedure is also visualized in Figure 2 If not stated otherwise, all results in this paper have been produced without manual refinement.

3.2. Non-bleeding Compositing

Once the segmentation of I_S and I_T is completed the user chooses an appropriate position to insert the source patch I_S into the target image I_T. This is no final decision. Our system allows to change the position again later in the procedure which is beneficial for the artist as he can easily adjust the image to his needs later on.

Color bleeding and halo effects in Poisson image editing occur, whenever the gradient field becomes non-integrable [TJP10]. Therefore the task is to find an estimate of these regions along the user-specified boundary $\partial\Omega$ and in a second step adjust the Poisson equation, Equation (2), to prevent these artifacts.

We divide the boundary pixels $\partial\Omega$ into two disjoint sets: $\partial\Omega_P$ for which the classic Poisson equation should reveal sufficient results and $\partial\Omega_B$ for those pixels which are potential candidates for artifacts. Due to the segmentation we can distinguish between four constellations along the boundary $\partial\Omega$ and how they possibly influence the solution of the Poisson equation. An illustration is also given in Figure 3.

1. B_S-B_T: If the source background touches the target background along the boundary no change is necessary, as our assumption is that the gradient field is integrable in these parts and we add these pixels to $\partial\Omega_P$
2. B_S-F_T: The color of the foreground of the target image potentially varies strongly compared to the background of the source image, therefore the Poisson equation needs to be adjusted to prevent color bleeding. These pixels are added to $\partial\Omega_B$.
3. F_S-B_T: Equivalent to 2.
4. F_S-F_T: As the foreground in the source and target image do not necessarily belong to the same object, we need to adjust the Poisson equation here as well and add these pixels to $\partial\Omega_B$

As from time to time the GrabCut technique is slightly imprecise, we also add those pixels to $\partial\Omega_B$ where F_S or F_T are closer than m pixels to the boundary $\partial\Omega$. In all our experiments we set $m = 3$ pixels.

To prevent color bleeding along $\partial\Omega_B$ we introduce integration barriers into the classic Poisson equation. These barriers are similar to the diffusion blockers in [BEDT10] and the image border handling in [PGB03]. I.e., we remove from our system of linear equations any part that relates two pixels

© The Eurographics Association 2011.

across $\partial\Omega_B$. For a better understanding Figure 4 illustrates such a situation. For each pixel p let N_p be the set of its 4-connected neighbors and let (p,q) denote a pixel pair such that $q \in N_p$. The finite difference discretization of Equation (1) combined with our special border handling yields:

$$\min_{I|_\Omega} \sum_{(p,q)\cap\Omega\neq\emptyset} ((I_p - I_q) - (I_{S,p} - I_{S,q}))^2, \text{ with } I|_{\partial\Omega_P} = I_T|_{\partial\Omega_P} \tag{3}$$

Its solution satisfies the following linear equations:

$$\forall p \in \Omega: \quad |N_p| I_p - \sum_{q\in N_p\setminus\partial\Omega_B} I_q = |N_p| I_{S,p} - \sum_{q\in N_p\setminus\partial\Omega_B} I_{S,q} \tag{4}$$

where $|N_p|$ is the number of pixels in the 4-connected neighborhood of pixel p that are not equal to any pixel in $\partial\Omega_B$. By solving the adjusted Poisson equation the color of the source patch I_S is adjusted so that the color transition is smooth across areas where the backgrounds touch each other, i.e., along $\partial\Omega_P$, while we preserve a crisp transition along $\partial\Omega_B$, as desired. Please note that the colors along $\partial\Omega_B$ are still adjusted, but the influencing pixels are different.

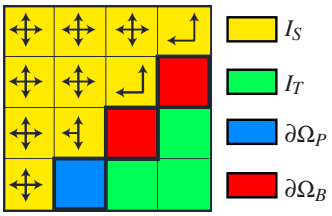

Figure 4: Integration barrier: We adjust the classic Poisson equation so that pixels across the barrier $\partial\Omega_B$ are unrelated. Dirichlet boundary conditions for pixels on $\partial\Omega_P$ are handled as usual.

Ill-conditioned or non-unique solutions are possible in the rare case that $\partial\Omega_B$ is equal to the user-specified mask, i.e., only a foreground object without background is copied or the background part is completely enclosed by the foreground object. To handle such a case we initialize our solution with the colors from the source image. This way, the method would boil down to a simple copy and paste without color adjustment still revealing plausible results. Previous methods, on the other hand, would try to adjust the foreground object which is usually undesired. As soon as the interacting boundary is at least one pixel in size, a unique seamless solution is found.

3.3. Composition of Interleaving Objects

To allow for interleaving objects we assign an ordering to the different classified regions on a local basis. We divide the image into overlapping regions and track the ordering of layers in each region separately. This is done by first creating a binary mask, setting each pixel to white where an overlap of the foregrounds is detected. Each other pixel is set to zero. We then search for connected components using a standard connected component analysis [CSRL01]. We define the foreground objects to be visually more important than the background objects and therefore we initially place F_S in front of F_T and F_T in front of B_S (B_T is never visible in the composited regions). We mark the regions where F_S and F_T overlap and for each distinct overlapping region the user can choose which one should be in front and which one in the back by a simple point-and-click interface. For complex interweavings, such as the tree branches in Figure 7, we also provide a randomized ordering for each overlap region. For each region a random number is computed. If it is smaller than 0.5 we assign F_S as being in front, otherwise F_B. This way we provide also the possibility to place objects *behind* others. No recomputation of the image colors is necessary if the ordering is changed, making the approach fast to work with. An example is given in Figure 5.

(a) (b) (c)

Figure 5: Interleaving objects: (a) Our system marks overlapping regions of foreground objects in red. (b) and (c) The user may then choose which part should be in front by simply clicking on the appropriate region.

4. Experimental Results

We implemented an interactive framework to test and compare our approach to other techniques. The only computationally expensive procedure in our approach are the GrabCut which can take a few seconds to compute in our current implementation and to solve the Poisson equation for which any solver of the appropriate sparse linear system can be used. In our implementation we use the UMFPACK solver [Dav09] which takes two or three seconds to compute the results for a 0.5 megapixel image, making our approach reasonably fast to work with. In all comparisons we used the same user drawn mask to extract I_S and placed it at the same position. Specifically we compare our approach to the following methods:

Copy & Paste: I_S is simply drawn over I_T.

Poisson: Here I_S is adjusted according to the classic Poisson image editing by Pérez *et al.* [PGB03].

Drag & Drop: Before solving the Poisson equation we optimize the boundary $\partial\Omega$ according to the energy functional proposed by Jia *et al.* [JSTS06] to find the *optimal*

© The Eurographics Association 2011.

path around the foreground object in I_S. The necessary foreground mask is the same as in our approach.

Error-tolerant: The gradient field is adjusted according to Tao *et al.* [TJP10] creating low curl boundaries and concentrating integration residuals in textured areas, where they are less conspicuous.

In Figure 6 several image compositing tasks are shown and compared. None of the masks were manually corrected in these examples. Simple Cut & Paste is obviously not able to create visually pleasing results due to the color differences, Figure 6a. As expected Poisson image editing [PGB03] cannot cope with erroneous masks and colour bleeding and halos are the result, Figure 6b. Especially visible at the boundary between the pyramids and palms in the third row of Figure 6b. The same scene also poses a big problem to boundary refinement methods [JSTS06] as an alteration may result in semantically incorrect composites, like the flying palms in Figure 6c. The method of Tao *et al.* [TJP10] proves to work sufficiently well for most examples but still reveals some color bleeding in regions where the curl is close to zero but the color differs between I_S and I_T, as can be seen in the first row of Figure 6d. On the other hand Tao *et al.*perform better in the pyramid example, where a slight color bleeding is visible on the left side of the palm. But this could be easily corrected in our approach with a single stroke in the user-correction step of the foreground mask.

Our method is the only technique that allows to place objects *behind* other objects in the target image, Figure 1. Note that none of the previously mentioned methods is able to reproduce this effect. While a similar effect would be possible with matting techniques [CCSS01, GCL*06, GO10], matting still adds an additional manual color correction step, in order for the object to nicely fit in the image, which is automatically achieved by Poisson image editing techniques. As the segmentation in our approach is independent of the placing, the artist is free to move the dove around and the algorithm will correctly adjust the colors and overlapping areas. Only a few strokes of the user were necessary to adjust the fore- and background mask of the target image. Even very complex compositions of trees are possible, Figure 7. Note the interleaving pattern of the branches created by our randomization procedure. More examples can be found in the additional material.

5. Discussion & Conclusion

In this paper we have presented an image-compositing algorithm which advances classic Poisson image editing in several directions. First, we do not directly modify the user selected boundary, which is important for various image-compositing tasks. Second, our method is robust to selection inaccuracies and reduces color bleeding artifacts and halos around the boundary. And third, it is capable of sophisticated image compositing tasks with interleaving objects, which would be very tedious by manual intervention and even impossible with previous seamless cloning techniques. The complexity is very similar to the standard Poisson equation as our integration barriers can be directly incorporated into the sparse linear system.

While it might seem less sophisticated to completely block the diffusion process at specific locations instead of using a non-linear diffusion [PM90, TJP10], we chose to do so, as a reweighting will still introduce some color bleeding into the result and we found that robust and high quality results are easier obtainable with our approach. In addition no tuning of coefficients is necessary, making our technique easy to work with. For future work we plan to integrate border matting into our formulation to deal with the problem of matting in the presence of blur and mixed pixels along smooth object boundaries, which could otherwise produce small artifacts along the object.

Acknowledgments

The authors gratefully acknowledge funding by the German Science Foundation from project DFG MA2555/4-2.

References

[ADA*04] AGARWALA A., DONTCHEVA M., AGRAWALA M., DRUCKER S., COLBURN A., CURLESS B., SALESIN D., COHEN M.: Interactive Digital Photomontage. *ACM Transactions on Graphics 23*, 3 (2004), 294–302. 1, 2

[Aga07] AGARWALA A.: Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics 26*, 3 (2007), 1–5. 1

[BEDT10] BEZERRA H., EISEMANN E., DECARLO D., THOLLOT J.: Diffusion constraints for vector graphics. In *International Symposium on Non-Photorealistic Animation and Rendering* (2010), NPAR '10, ACM, pp. 35–42. 3

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239. 3

[BZCC10] BHAT P., ZITNICK L., COHEN M., CURLESS B.: GradientShop: A Gradient-Domain Optimization Framework for Image and Video Filtering. *ACM Transactions on Graphics 27*, 3 (2010), 1–14. 1

[BZS*07] BHAT P., ZITNICK C. L., SNAVELY N., AGARWALA A., AGRAWALA M., CURLESS B., COHEN M., KANG S. B.: Using photographs to enhance videos of a static scene. In *Eurographics Symposium on Rendering* (2007), pp. 327–338. 1

[CCSS01] CHUANG Y.-Y., CURLESS B., SALESIN D. H., SZELISKI R.: A bayesian approach to digital matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2001), vol. 2, pp. 264–271. 5

[CSRL01] CORMEN T. H., STEIN C., RIVEST R. L., LEISERSON C. E.: *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001. 4

[Dav09] DAVIS T.: Umfpack, 2009. http://www.cise.ufl.edu/research/sparse/umfpack/. 4

© The Eurographics Association 2011.

(a) Copy & Paste (b) Poisson (c) Drag & Drop (d) Error-tolerant (e) Ours

Figure 6: Comparison of different image compositing techniques. (a) Simple Copy & Paste; (b) Seamless cloning by Pérez *et al.* [PGB03]; (c) Drag & Drop Pasting by Jia *et al.* [JSTS06]; (d) Error-tolerant image compositing by Tao *et al.* [TJP10]; (e) Our approach.

[EGM11] EISEMANN M., GOHLKE D., MAGNOR M.: Edge-constrained image compositing. In *GI '11: Proceedings of Graphics Interface 2011* (2011), Canadian Information Processing Society. to appear. 2

[FHL*09] FARBMAN Z., HOFFER G., LIPMAN Y., COHEN-OR D., LISCHINSKI D.: Coordinates for Instant Image Cloning. *ACM Transactions on Graphics 28*, 3 (2009), 1–9. 2

[GBD*09] GUPTA A., BHAT P., DONTCHEVA M., CURLESS B., DEUSSEN O., COHEN M.: Enhancing and experiencing space-time resolution with videos and stills. In *International Conference on Computational Photography* (2009), pp. 1–9. 1

[GCL*06] GUAN Y., CHEN W., LIANG X., DING Z., PENG Q.: Easy matting - a stroke based approach for continuous image matting. *Computer Graphics Forum 25*, 3 (2006), 567–576. 5

[GO10] GASTAL E. S. L., OLIVEIRA M. M.: Shared sampling for real-time alpha matting. *Computer Graphics Forum 29*, 2 (2010), 575–584. 5

[HE07] HAYS J., EFROS A. A.: Scene completion using millions of photographs. In *ACM Transaction on Graphics* (2007), pp. 1–8. 1

[JSTS06] JIA J., SUN J., TANG C., SHUM H.: Drag-and-Drop Pasting. *ACM Transactions on Graphics 25*, 5 (2006), 631–637. 1, 2, 4, 5, 6

[JT03] JIA J., TANG C.-K.: Image Registration with Global and Local Luminance Alignment. In *ICCV '03 (Proceedings of the Ninth IEEE International Conference on Computer Vision)* (2003), pp. 156–163. 2

[JT08] JIA J., TANG C.-K.: Image stitching using structure deformation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*, 4 (2008), 617–631. 2

[LHE*07] LALONDE J.-F., HOIEM D., EFROS A. A., ROTHER C., WINN J., CRIMINISI A.: Photo clip art. *ACM Transactions on Graphics 26*, 3 (2007), 1–10. 2

[LZPW04] LEVIN A., ZOMET A., PELEG S., WEISS Y.: Seamless image stitching in the gradient domain. In *European Conference on Computer Vision* (2004), pp. 377–389. 1, 2

[MP09] MCCANN J., POLLARD N.: Local layering. *ACM Transactions on Graphics 28* (2009), 84:1–84:7. 2

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson Image Editing. *ACM Transactions on Graphics 22*, 3 (2003), 313–318. 1, 3, 4, 5, 6

[PM90] PERONA P., MALIK J.: Scale-Space and Edge Detection using Anisotropic Diffusion. *Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (1990), 629–639. 5

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "GrabCut" - Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics 23*, 3 (2004), 309–314. 3

[Sze06] SZELISKI R.: Image alignment and stitching: a tutorial. *Found. Trends. Comput. Graph. Vis. 2*, 1 (2006), 1–104. 1

[TJP10] TAO M. W., JOHNSON M. K., PARIS S.: Error-tolerant image compositing. In *European Conference on Computer Vision* (2010), pp. 1–14. 2, 3, 4, 5, 6

© The Eurographics Association 2011.

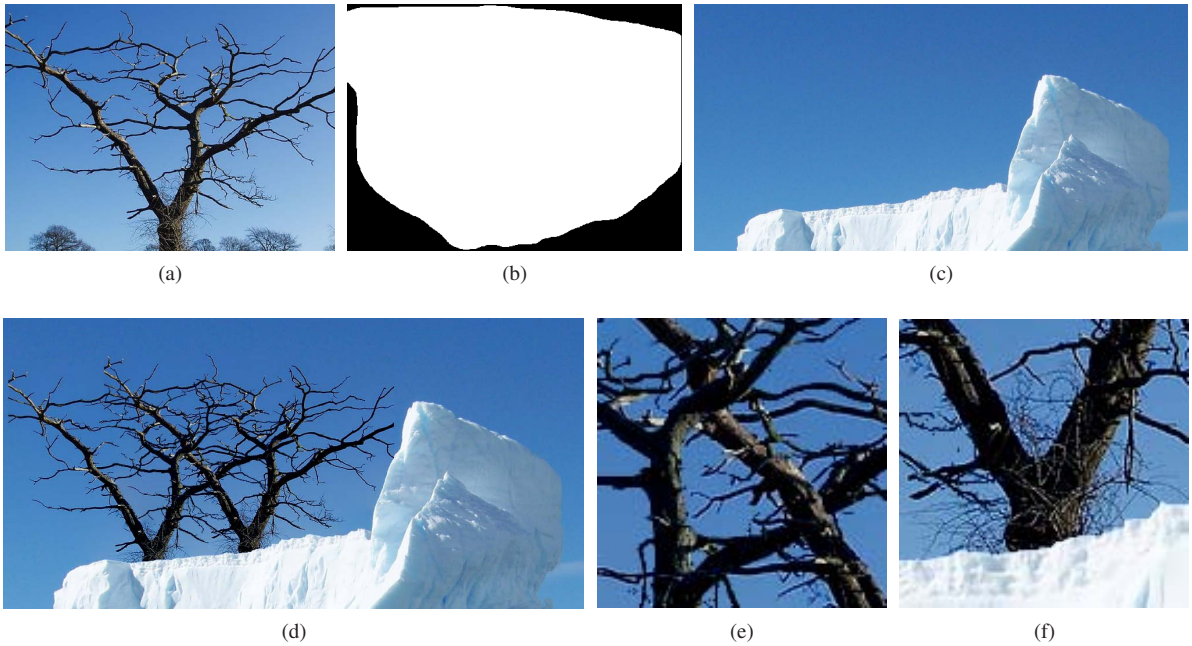

Figure 7: Interleaving composites: Our approach allows to create very complex, interleaving composites with almost no user interaction at all. (a) The source image. (b) The user defined mask. (c) The target image. (d) The source image was composited twice into the target image using our algorithm. (e) Close-up view of the branches. Note the interleaving pattern. (f) Close-up view of the stem. Note that any color bleeding from the ground is suppressed. The user is free to change the ordering of the branches at any time.

© The Eurographics Association 2011.