Martin Eisemann
eisemann@cg.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Daniel Gohlke
danielgohlke@freenet.de
Computer Graphics Lab, TU Braunschweig

Prof. Dr. Ing. Marcus Magnor
magnor@cg.tu-bs.de
Computer Graphics Lab, TU Braunschweig

Structure-Aware Image Compositing

Technical Report 2010-11-12

November 9, 2010

Computer Graphics Lab, TU Braunschweig

Contents

1 Abstract

The classic task of image compositing is complicated by the fact that the source and target images need to be carefully aligned and adjusted. Otherwise, it is not possible to achieve convincing results. Visual artifacts are caused by image intensity mismatch, image distortion or structure misalignment even if the images have been globally aligned. In this paper we extend classic Poisson blending by a constrained structure deformation and propagation method. This approach can solve the above-mentioned problems and proves useful for a variety of applications, e.g. in de-ghosting of mosaic images, classic image compositing or other applications such as super-resolution from image databases. Our method is based on the following basic steps. First, an optimal partitioning boundary is computed between the input images. Then, features along this boundary are robustly aligned and deformation vectors are computed. Starting at these features, salient edges are traced and aligned, serving as additional constraints for the smooth deformation field, which is propagated robustly and smoothly into the interior of the target image. If very different images are to be stitched, we propose to base the deformation constraints on the curvature of the salient edges for C_1-continuity of the structures between the images. We present results that show the robustness of our method on a number of image stitching and compositing tasks.

2 Introduction

Image compositing, i.e., the arrangement, alignment and adaption of image content from different sources onto one common image plane is one of the fundamental tasks in 2-D computer graphics and image editing. Three fundamental objectives are pursued: The first is the image collage, which serves mainly artistic purposes by arranging different image patches without (or with manual) editing of the content. The second is image stitching, which generates a natural image composite given a set of globally registered images with similar content in the limited overlapping area [Sze06]. The third is seamless image composition of semantically different objects, e.g. swapping faces [BKD$^+$08] or seamless clone brushing [MP08].

Generating a natural transition between one image to another is required to produce a satisfactory result in image stitching. Earlier techniques optimize a blending function to minimize intensity differences in the vicinity of the overlapping areas [BA83, Sze96]. Seamless image compositing is mostly based on exploiting the Poisson equation [PGB03] whose importance grew tremendously in image and video editing during the last decade. Mostly focusing on a smooth color transition there is no guarantee that the image structures will be aligned too. Even small misalignments may produce visual artifacts such as local intensity or structure inconsistency. These misalignments can cause image ghosting or blurring artifacts as one salient edge might fade out as it enters the overlapped area and fades in several pixels apart. Structure deformation is an already established technique in texture synthesis [FH04, WY04], where, in general, only small repetetive structures are to be aligned, and in registration of non-rigid objects, e.g. in medical applications [RJBJ03]. The same technique in general image stitching is less common and still an open problem [JT08], also due to the fact that the human visual system is quite sensitive to sudden deformations of real-world images, such as photographs.

In this paper we address the combined problem of intensity and structure misalignment for real-world footage and general image stitching and propose robust methods to globally eliminate intensity and structure misalignments between the overlapping images which eases the task of seamless image compositing. Our approach is based on, what we would like to call, 1.5-D feature matching and deformation propagation of natural images. First, feature matching along an optimal partition boundary is performed. Second, the salient edges along these features are traced and brought into accordance to form a sparse deformation field which is then propagated throughout the interior of the image. The color propagation is performed in the gradient domain allowing for seamless color transitions between the overlapping images.

The paper is structured as follows. In Sect. 3 we review related work. Sect. 4 presents our technique in detail. Results and comparisons with existing methods are given in Sect. 5. We conclude our paper in Sect. 6.

3 Related Work

Image stitching A very nice survey on image stitching can be found in [Sze06]. Local approaches blend the colors of overlapping images based on precomputed weighting masks, e.g. [Sze96, UES01, EESM10]. The transition is often still visible as it conveys a rather unnatural conversion between the images. The multiresolution spline by Burt and Adelson [BA83] adjusts the transition separately for each band of frequencies, based on a Laplacian pyramid to prevent ghosting artifacts and sudden transitions between the images. Instead of blending between the images it can be more favourable to select only one input image per output pixel in the overlapping area to prevent artifacts such as ghosting. This choice is usually based on an optimal seam between the images. Taking the color difference in the overlapping area into account, an optimal partition based on Graph Cuts [Boy01] or dynamic programming [Bel62] is usually computed. This approach was succesfully applied to texture synthesis [EF01, KSE$^+$03] or semi-automatic image compositing [ADA$^+$04], in the latter the compositing step after the partitioning was done in the gradient domain to remove color inconsistencies. The image stitching algorithm by Levin *et al.* [LZPW03] operates directly in the gradient domain to compute an optimal path based on the gradient strength. Optimal seam methods usually require a good alignment of image features beforehand in order to reveal pleasing results. This, however, cannot be guaranteed in most image stitching applications and misaligned edges or misplaced features are the result.

Han *et al.* [HH10] create a visually smooth mipmap pyramid from already stitched imagery at several scales to hide jarring transitions when zooming into the images. They combine the detail of one image with the local appearance of another and use clipped Laplacian blending to minimize blur for the intermediate levels in the pyramid which need to be created in addition. This way color and structure are conserved, but the method requires images with similar content at different levels of detail in order to produce plausible results.

Poisson Image Editing Poisson image editing was introduced by Pérez *et al.* [PGB03] as a means to produce seamless composites of different images. While the color constraints were taken from the partitioning seam of the target image, the gradient of the source image was used to reconstruct the seamless composite. McCann *et al.* [MP08] proposed an interactive gradient domain painting tool, that could also be used for classic Poisson image editing and gradient domain clone brushing. To prevent some of the unnatural transitions occuring if different backgrounds are stitched together Jia *et al.* [JSTS06] proposed a method to search for the optimal seam around an object which deviates the least from the background. Guo [GS09] extend the classical Poisson Image Editing by allowing the user to draw color constraints on the source image. The color is then integrated into the image only along homogeneous regions and not across strong edges, preventing some of the color bleeding artifacts that might otherwise occur. In [YZC$^+$09] a similar effect is achieved but with a fully automatic method based on a random walk segmentation and pyramid blending. In Sunkavalli *et al.*'s extension [SJMP10] they do not only adjust the color but also contrast, texture, noise and blur. But all of these do not adjust the structure itself and leave this to the user.

Structure deformation Structure deformation for image alignment has been heavily researched in medical image registration, which commonly deals with non-rigid registration errors by first roughly aligning the images (if this is not inherently done), matching prominent features and smoothly interpolating these sparse correspondences to compute a global deformation field. This approach of constraining the deformation and enforcing interior smoothness was first proposed by Bajcsy and Kovacic [BK89]. An overview of the large amount of existing literature for medical image registrations is given in [MV98].

Matching features and distorting the image accordingly has already been used in a variety of applications as texture synthesis, medical image registration or especially image morphing. Wu and Yu's [WY04] texture synthesis algorithm deforms texture patches by matching sparse features and interpolating the deformation vectors based on thin-plate splines [Mei79] or Shepard's method [HL93]. But as the algorithm is designed for example-based texture synthesis no intensity correction is applied or necessary in their case.

The work with the closest resemblance to ours is the approach by Jia *et al.* [JT05] and their follow-up work in [JT08]. They aim at correcting the mismatch along the partitioning border by feature matching. The computed deformation along the seam is then smoothened out into the interior of the source, without taking any further structural information into account. Our approach aims at overcoming these limitations, by not only matching features along the seam, but also by tracing salient edges into the source and target image, which are used as additional deformation constraints to create more plausible transitions.

4 Our method

Consider the basic task of stitching together two images I_S and I_T which have already been roughly aligned and overlap in an area called Ω, Fig. 1. The partitioning seam is called $\partial\Omega$ with $\partial\Omega_S$ and $\partial\Omega_T$ depicting the pixels along the border in I_S and I_T respectively. Our algorithm proceeds in six steps:

1. An optimal partitioning is computed between the roughly aligned images I_S and I_T.

2. Features along the partitioning seam $\partial\Omega$ are matched and brought into alignment.

3. The outgoing edges along these features are traced and brought into alignment. Optionally, if no edges can be traced (as image information might be missing or is erroneous), a spline is fit to the edges available in both source and target image, and these are brought into accordance to fulfill approximate C_1-continuity.

4. The sparse deformation field derived from the matching is propagated throughout the area of the source image I_S which is warped accordingly.

5. To take the warping into account, we compute a new optimal partitioning.

6. The color values of I_S are adjusted subject to a constraint Poisson equation.

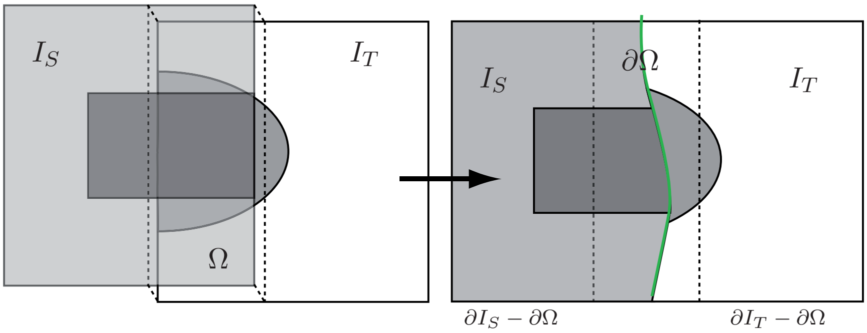

I_S
I_T
Ω
I_S
$\partial\Omega$
I_T
$\partial I_S - \partial\Omega$
$\partial I_T - \partial\Omega$

Figure 1: Optimal partitioning for two overlapping images: (left) The images I_S and I_T have been globally aligned and overlap in the area Ω. (right) The partitioning divides Ω into two parts. Based on the found seam $\partial\Omega$ the images are combined into a common image space. The border around each of the image excluding the seam $\partial\Omega$ is denoted $\partial I_S - \partial\Omega$ and $\partial I_T - \partial\Omega$ respectively.

4.1 Optimal partition

As Poisson blending works best if the color discrepancy along the seam is relatively constant we employ the *Drag-and-Drop Pasting* method [JSTS06] to find an optimal partitioning seam in Ω. Starting with an arbitrary path this iterative technique can find the seam optimizing the following energy function:

$$E(\partial\Omega, k) = \sum_{p\in\partial\Omega} ((I_T(p) - I_S(p)) - k)^2, \text{s.t. } p \in \Omega \cup (\partial I_S - \partial\Omega), \quad (1)$$

where k is the average color difference on the boundary seam $\partial\Omega$ computed as the L_2-norm on the rgb-triplets.

For the case where I_S is fully surrounded by I_T, which is usually the case for object insertion tasks, we define a foreground object Ω_{Obj} in I_S by applying the GrabCut technique of Rother *et al.* [RKB04].This defines a foreground area in I_S which may not be crossed by the optimal seam, Fig. 2. If I_S only partially overlaps I_T we can force the seam to cross the boundary pixels of $\partial I_S - \partial\Omega$ by enforcing $I_S - \Omega$ to belong to the foreground object except for the border pixels $\partial I_S - \partial\Omega$ and Ω.

4.2 1-D feature detection and matching

Optimal partitioning may find a very subtle seam to switch between the input images, but it is powerless against structural discontinuities at the

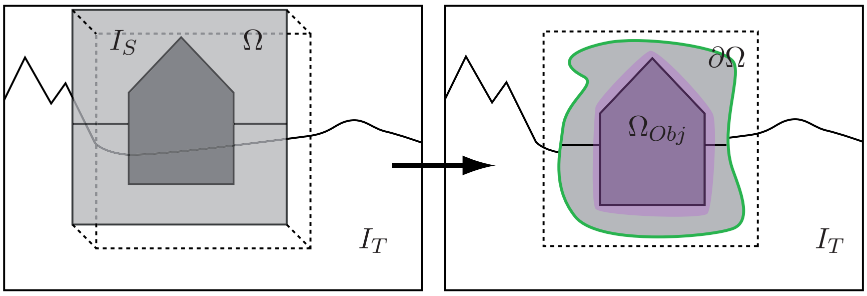


Figure 2: Optimal partitioning for two images, one enclosing the other: (left) The source image I_S is completely surrounded by its target I_T. (right) A foreground region Ω_{Obj} (lilac) is defined through which the seam may not pass, and the optimal seam is computed around it (green).

seam. To deal with these one needs to align the features of both images. To preserve the general applicability of our method, we do not assume that I_S and I_T actually have any common features, unlike in most image stitching applications. Therefore direct feature matching, e.g., using SIFT [Low04] is not possible. An example for such a case is given in Fig. 8, where the larger brush is attached to a smaller brush.

In our observation the most prominent artifacts are produced by mismatching salient edges in both images. Therefore, the first step is to detect these edges. We start by removing noise in the image by applying a bilateral filter [TM98]. Using the Canny edge detector [Can86] we find all important edges in the images and thin them out, to assure there are at most two neighboring pixels for each edge pixel, which is beneficial for the latter edge matching. Assuming, without loss of generality, that there are n edges found along $\partial\Omega_S$, m edges along $\partial\Omega_T$ and $n \geq m$, an optimal edge matching can be found by dynamic programming:

$$E' = min \sum_{0 \leq i < m} (p_T(i) - p_S(k_i))^2 \ , \qquad (2)$$
$$\text{s.t. } 0 \leq k_0 < k_1 < \ldots < k_{m-1} < n \ ,$$

where $p_S(.)$ and $p_T(.)$ are the pixel positions of the salient edges along $\partial\Omega_S$ and $\partial\Omega_T$, respectively. For each of the matched edges e_i, a deformation vector $\mathbf{d}$ pointing from its pixel position along $\partial\Omega_T$ towards the position of its match along $\partial\Omega_S$ is defined as a constraint for a deformation field $\mathbf{D}$ for I_S, Fig. 3. If the automatic alignment fails, e.g. if semantically very different images are to be stitched, we allow for manual correction of the correspondences.

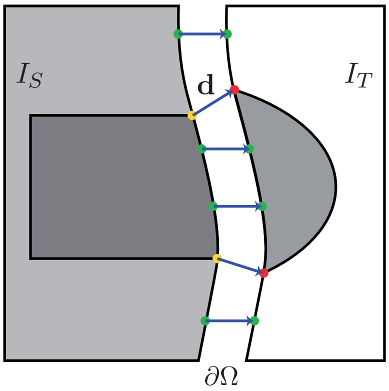


Figure 3: Structure misalignments might still persist along the border of an optimal partitioning. To remove these misalignments we compute the most salient edge pixels (red and yellow) along $\partial\Omega$ and compute a deformation vector for each of these. Additional zero vectors (green) are added to prevent excessive deformations.

4.3 1.5-D feature matching

Once a matching of the salient edge pixels along $\partial\Omega$ has been computed, we need to propagate these deformations in a meaningful manner to the rest of the pixels. To restrict the deformation of I_S, we set additional zero deformation vectors $\mathbf{O} = (0, 0)$ for those pixels along $\partial\Omega$ that are 10% of the seams overall length away from any previously computed deformation vector $\mathbf{d}$. The 10% are chosen empirically but give good results in our test cases. For the rest of the unassigned pixels along $\partial\Omega$ we linearily interpolate the values of the two neighbouring deformation vectors to the left and to the right.

Matching of the salient edges avoids structural mismatches along the seam, one can think of this as C_0-continuity, but the edge direction can still change rather abruptly, so there is no real C_1-continuity along the edges. We will therefore trace the edges further into I_S and I_T and match these as well.

To trace an edge starting at the edge's pixel position p, we create an edge path $\mathbf{P}$ of a user-specified length l. Due to the preprocessing we can usually simply walk along the edges already found in Sect. 4.2 by the Canny edge detector [Can86]. In case of ambiguities we follow the strongest gradient strength. If a manual correction was set at a pixel which is no valid edge pixel of the preprocess, we use the following algorithm. We describe it exemplarily for I_S.

1. Set $\mathbf{P} = \{p\}$.

2. Initialize the set of potential edge candidates to $\mathbf{C} = \emptyset$.

3. Set the edge direction to be orthogonal to $\partial\Omega$ and pointing inwards I_S.

4. Add all neighbouring pixels of the active pixel p to $\mathbf{C}$, which are in positive edge direction.

5. If $|\mathbf{C}|$ is 0 return $\mathbf{P}$.

6. Compute the gradient strength for all candidates in $\mathbf{C}$.

7. If the strongest gradient strength is beneath a threshold $\tau = 50$, return $\mathbf{P}$.

8. Add the candidate with the strongest gradient to $\mathbf{P}$ and remove all others from $\mathbf{C}$.

9. Set the edge direction to be $p_{n-1} - p_n$, where p_{n-1} and p_n are the next to last and last edge pixel added to $\mathbf{P}$ respectively.

10. If $|\mathbf{P}| < l$, goto step 4.

11. Return $\mathbf{P}$.

τ is chosen as proposed in Ref. [JT08]. Edge tracing for I_T is done accordingly, but the edge is additionally traced into the direction of I_S in step 3. As a convention we will use $\mathbf{P}_{I_A \to I_B}$ to denote the set of edge pixels in I_A starting at p and going in the direction of I_B, Fig. 4.

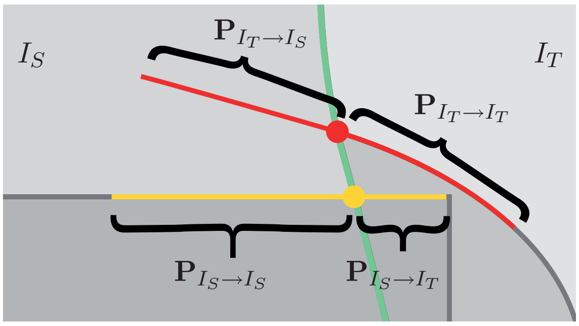

I_S
$\mathbf{P}_{I_T\to I_S}$
I_T
$\mathbf{P}_{I_T\to I_T}$
$\mathbf{P}_{I_S\to I_S}$
$\mathbf{P}_{I_S\to I_T}$

Figure 4: Naming convention of the traced edges. The in- and outgoing edges in I_T are marked in red, the respective edges in I_S are marked in yellow.

We propose three different methods according to three different edge configurations to match the pixels along the traced edges, described in the following.

Static correspondence Let p be an edge pixel on the seam $\partial\Omega_T$ with the assigned deformation vector $\mathbf{d}_p$. For each edge pixel position in $\mathbf{P}_{I_T \rightarrow I_S}$ we add an additional deformation vector equal to $\mathbf{d}_p$. The intention behind this procedure is to later warp the content of I_S according to the edge in I_T. This procedure is especially useful if the corresponding edge in I_S is hard to trace, corrupted or differs largely from the target's edge. An example is given in Fig. 5.

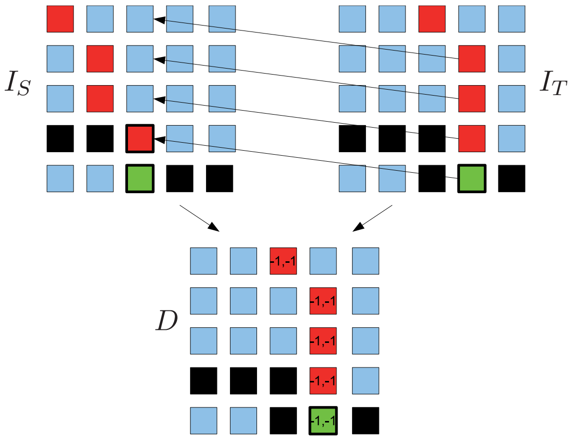


Figure 5: Static correspondence: The deformation of edge e (green) on the seam $\partial\Omega$ (black) is propagated along $\mathbf{P}_{I_T \rightarrow I_S}$ (red pixels on the right). The same deformation is assigned to all edge pixels along $\mathbf{P}_{I_T \rightarrow I_S}$.

Variable correspondence Our second approach takes both edge paths $\mathbf{P}_{I_S \rightarrow I_S}$ and $\mathbf{P}_{I_T \rightarrow I_S}$ into account. Without loss of generality we assume that $|\mathbf{P}_{I_S \rightarrow I_S}| \geq |\mathbf{P}_{I_T \rightarrow I_S}|$. For each pixel position in $\mathbf{P}_{I_T \rightarrow I_S}$ we set $\mathbf{d}_{p_n} = \mathbf{P}_{I_S \rightarrow I_S}(n) - \mathbf{P}_{I_T \rightarrow I_S}(n)$, where $\mathbf{d}_{p_n}$ is the deformation vector at pixel position $\mathbf{P}_{I_T \rightarrow I_S}(n)$. The n-th pixel position in $\mathbf{P}_{I_T \rightarrow I_S}(n)$ is matched with the n-th pixel position in $\mathbf{P}_{I_S \rightarrow I_S}(n)$. This approach is beneficial whenever the edges in the source and target image are comparable, and direct matching improves the *semantical* transition. This is depicted in Fig. 6.

Approximated correspondence If the images I_S and I_T have already been cropped beforehand, there is no way to use the previous methods, as the edges in I_T are missing in the area of I_S. To handle this case, we fit a quadratic B-spline to the edges in $\mathbf{P}_{I_T \rightarrow I_T}$ and extrapolate in the direction of I_S to estimate a new set of edge pixels $\mathbf{P}_{I_T \rightarrow I_S}$ which can then be matched to $\mathbf{P}_{I_S \rightarrow I_S}$, similar to the variable correspondence method, Fig. 7. To fit the quadratic B-spline to the edge pixels in $\mathbf{P}_{I_T \rightarrow I_T}$ we use a least squares method [PL07] which takes as input all pixel positions in $\mathbf{P}_{I_T \rightarrow I_T}$.

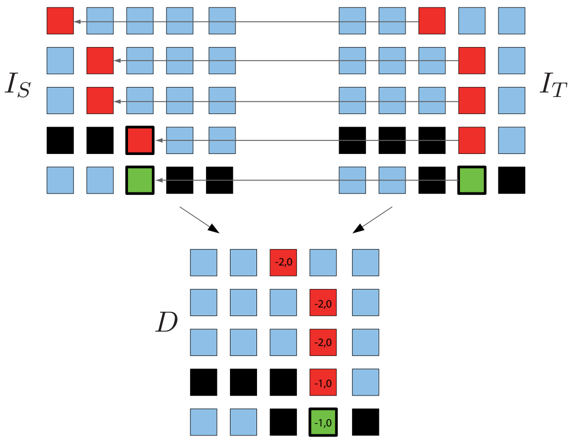


Figure 6: Variable correspondence: (top) The deformation of edge e (green) on the seam $\partial\Omega$ (black) is propagated along $\mathbf{P}_{I_T \rightarrow I_S}$ (red pixels on the right). To take the variable edge directions in both images into account, the n-th edge pixel in $\mathbf{P}_{I_T \rightarrow I_S}$ is matched with the n-th edge pixel in $\mathbf{P}_{I_S \rightarrow I_S}$ (red pixels on the left). (bottom) The appropriate deformation vector is written to the corresponding position in the deformation field D.

4.4 Deformation propagation

As the human visual system is susceptible to sudden changes in a warped image, we fill out the rest of the deformation field D as smooth as possible. We solve the following diffusion equation with Dirichlet boundary conditions:

$$\begin{aligned} D^*(x,y) &= D(x,y) \text{ , if } D(x,y) \neq \emptyset \\ \nabla^2 D^*(x,y) &= 0 \text{ , otherwise} \end{aligned} \qquad (3)$$

where ∇^2 is the Laplace operator. This is equivalent to

$$D^* = \underset{D}{\operatorname{argmin}} \int\int_{p \in I_S} ||\nabla D||^2 dp \qquad (4)$$

This can be efficiently solved with multigrid or conjugate gradient solvers [PTVF07].

4.5 Warping and color adjustment

After minimization, each pixel in our image domain is associated with a deformation vector. Performing an inverse mapping with bilinear interpolation on I_S, we obtain the warped image I_S^*. As the image was deformed during this process, we compute a new optimal partition to refine the seam as described in Sect. 4.1. Finally, we adjust the colors of I_S^* by solving the following Poisson equation to compute our final result I_R:

$$\begin{aligned} I_R(x,y) &= I_T(x,y) \ , \text{ if } (x,y) \in \partial\Omega \\ \nabla^2 I_R(x,y) &= \nabla^2 I_S^*(x,y) \ , \text{ otherwise} \end{aligned} \tag{5}$$

5 Results

In this Section we show some results achieved with our technique and compare it to other state-of-the-art methods. The computation time in our non-optimized C/C++ implementation for a standard one megapixel image is around a few seconds on a Intel Core i7 950, 3,06GHz. Most time, around 90%, is spent on the seam finding and solving the different equation systems. As both is easily parallelizable, we expect interactive behaviour if ported to the GPU. The manual interaction time, if needed is usually just about a few seconds to mark the correct matchings or delete distracting edges.

Image compositing In Fig. 8 we compare our method to a variety of other techniques using a test scene from Ref. [JT08]. Here only a single input image, Fig. 8(a), was used, and the lower brush was copied to approximately align with the upper brush. The user-drawn mask is shown in (b) (yellow region). Feathering (c), computing an optimal seam (d), the GIST operator by Levin *et al.* [LZPW03] (e) and direct Poisson blending [PGB03] (f) all result in visible transitions in color and structure. Aligning the brush using textural structure deformation as proposed in Ref. [WY04] (g) is difficult as this example has diverse features at various and multiple scales that confuses the warping process. The result by Jia *et al.* [JT08] (h) nicely matches the structure and color but fails to plausibly propagate the structural deformation, though the transition is smooth, but still quite noticeable. Our results in (i) show a consistent propagation of the top edge, but unfortunately could not match the lower edges due to the feature complexity of the brush. In (j) we allowed the user to erase disturbing edges, which enforces a stronger congruence of the brushes and therefore provides a more natural transition.

In our second example, Fig. 9, we tested our algorithm by replacing the blossom of a flower with another blossom and use our method to adjust the stem in order to create a plausible transition between the two images. In a first step the images are roughly aligned by hand. The yellow pixels in the left images show the traced edges, the seam is shown in magenta in the top left image and is used for partitioning in this experiment. We compare our method to two other established techniques for seamless image stitching, namely *Poisson Blending* by Pérez *et al.* [PGB03] (middle left) and *Image Stitching Using Structure Deformation* by Jia *et al.* [JT08]. In the bottom row, we show close-ups of the transition area along the stem. The images in (b) show the result using only the Poisson Blending technique of Pérez *et al.*Although the color discrepancy is alleviated, the transition between the two stems is clearly visible due to their differing width. Strong color bleeding artifacts are the result. For the images in the middle right we used the method of Jia *et al.*which nicely adjusts the stem along the seam, but the transition area is still annoyingly visible due to the fact that the sparse deformation constraints are only computed along the seam. Using the same deformation vectors along the seam, but our variable correspondence method to match the interior edges, we can create a much more natural looking transition without noticeably deforming the blossom. Note that this effect would be quite difficult to achieve even by manual adjustment.

In the third test scene, Fig. 10, we want to composite a statue of a woman on top of a bronze statue by Nina Akamu. The images were cropped beforehand and roughly aligned manually with very little overlap. Misalignments are still visible, even after optimizing the seam by the Drag & Drop method [JSTS06], Fig. 10 (a) and (b). Using our approximated correspondence method, (c) and (d), to extrapolate splines along the salient edges of the target (horse) into the source region (woman), we can align these edges and create a more natural transition.

Database super-resolution In this example we want to increase the perceived resolution of certain parts of an image, Fig.11 (b) and (c), by replacing it with downloaded image footage from an Internet database. We downloaded the first 35 images from Flickr using the phrase *statue of liberty*. To find corresponding images and globally arranging them we used the technique described in Eisemann *et al.* [EESM10] and chose the matching image shown in Fig. 11 (a). Poisson blending [PGB03] can adjust the color of the source image, but structures like the clouds or the basement cannot be handled properly, Fig. 11 (d). Using *Drag & Drop Pasting* [JSTS06] an optimal seam is created so that the transition in the clouds is less visible, Fig. 11 (e). The structural misalignments, however, are still not handled well. Our approach handles both color and structural discrepancies sufficiently well: visible seams are removed while the applied deformation is hardly noticeable.

Limitations Due to the complexity of many natural images, robust automatic feature detection along the seam is still an open problem, as too many fine scale structures in the images can lead to erroneous matchings. Inherently related to this problem is our edge tracing. In Fig. 8, satisfactory results could only be obtained after erasing some of the edges produced by the water droplets. In addition, the transition of the shadow in the same image is still noticeable in all of the tested approaches. This is an inherent drawback of the Poisson Blending technique which works best if the color difference along the seam varies smoothly, which is not the case in this example, as the shadow of the larger brush is darker.

6 Conclusion and Discussion

We presented a novel method to deal with intensity inconsistencies and structural misalignment for general image compositing and stitching tasks. A global matching of salient features in a 2-D image plane might be impossible due to different semantic content. We thus propose to first match features only along an optimal partitioning of the images. We then propagate them along the salient edges and match these to construct a set of sparse deformation vectors. This reduces the misalignment problem from 2-D to two 1-D problems. From these sparse features we smoothly propagate the deformation into the area of the source image. In addition to subsequently applied Poisson blending, we correct both color and structural misalignment, alleviating the user from carefully aligning the source and target image along an optimal seam. Our method can serve as an efficient way to address the general problem of natural image stitching. In this paper we only discussed the problem of aligning two images. Extending this approach to multiple images, video or even stereo / multiview image editing is a prolific area for future research.

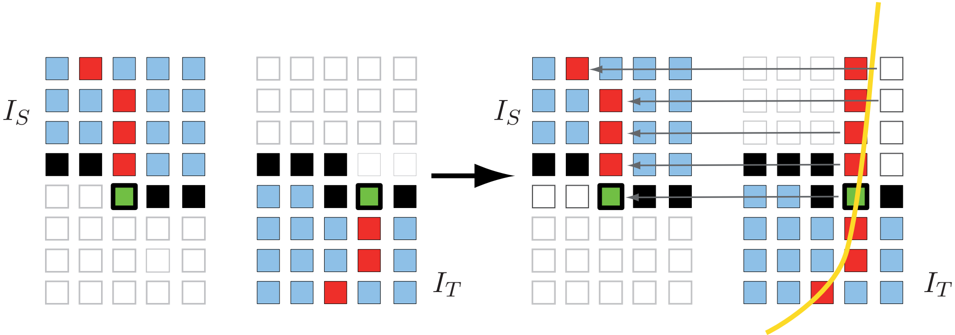
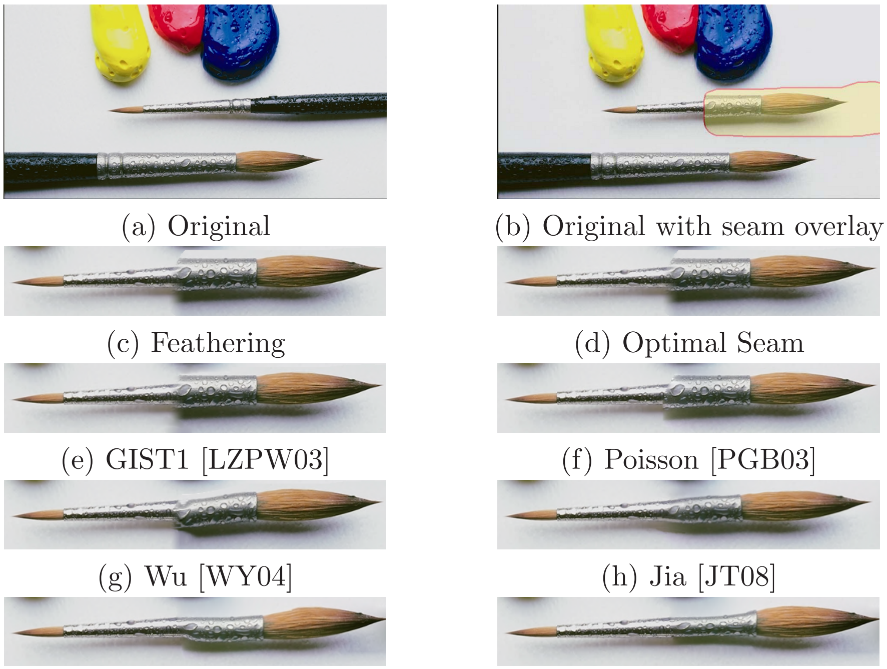


Figure 7: Approximated correspondence: (left) The approximated correspondence method does not assume the existence of $\mathbf{P}_{I_S \to I_T}$ or $\mathbf{P}_{I_T \to I_S}$. This case can happen if the images to composite have already been cropped. (right) $\mathbf{P}_{I_T \to I_S}$ is estimated by fitting a quadratic B-spline (orange) onto $\mathbf{P}_{I_T \to I_T}$ and extrapolating the spline into the area of I_S.



(a) Original

(b) Original with seam overlay

(c) Feathering

(d) Optimal Seam

(e) GIST1 [LZPW03]

(f) Poisson [PGB03]

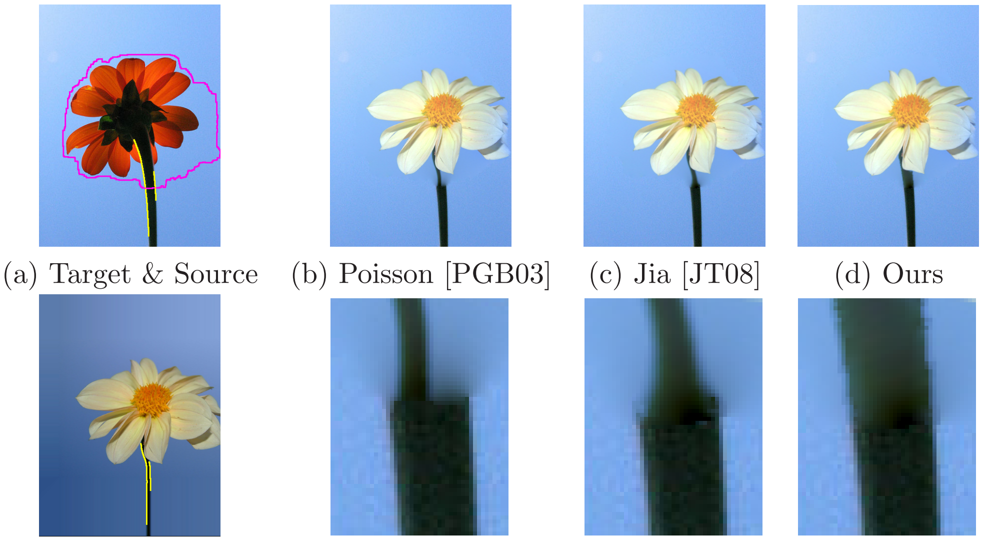
(g) Wu [WY04]

(h) Jia [JT08]

(i) Our result with static correspondences

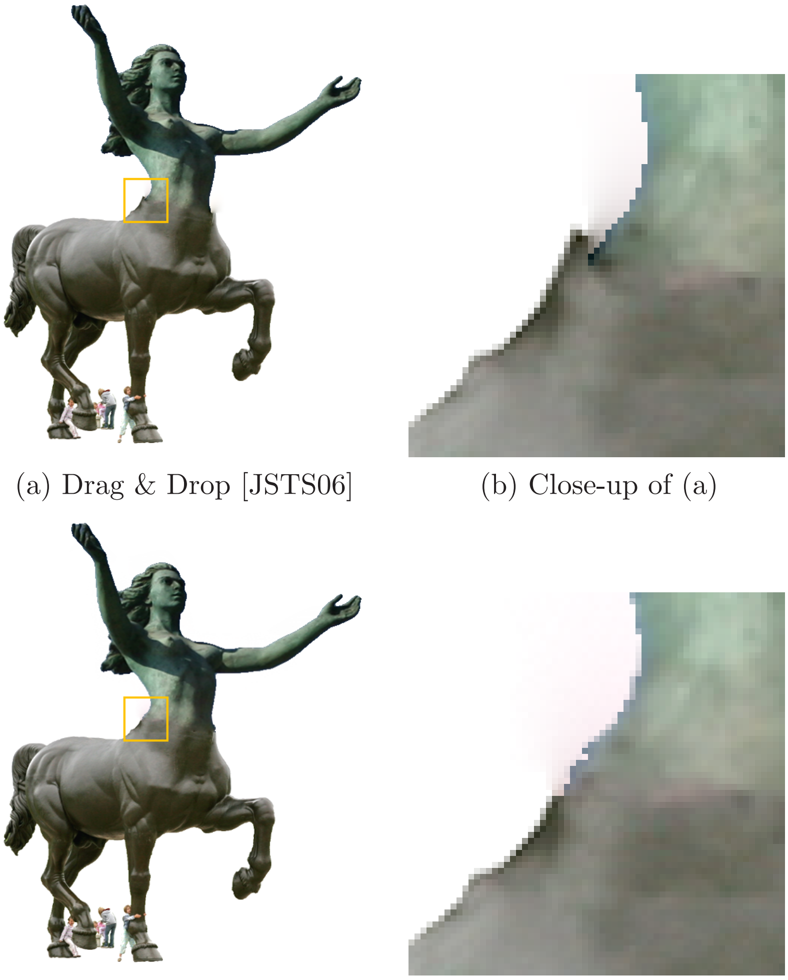
(j) Our refined result

Figure 8: In this example the lower brush of the input image (a) is copied and pasted onto the upper one, shown in the transparent region (b). (c) Feathering result. (d) Optimal seam result (computed with the technique presented in [JT05]). (e) GIST1 result [LZPW03]. (f) Poisson blending [PGB03]. (g) Structure deformation result [WY04]. (h) Image stitching and structure deformation [JT08]. The images are nicely stitched, but the brush still shows unnatural-looking shape deformations. (i) Our result with automatic edge detection and the static correspondence method. The structural transition is more smoothly at the top of the brush by enforcing the direction of the brush's edge in the source area to match the target. The bottom edge could not be matched properly due to the complexity of the edge features. (j) Our result after interactively removing distracting edges.



(a) Target & Source (b) Poisson [PGB03] (c) Jia [JT08] (d) Ours

Figure 9: In this example, the blossom of the white flower replaces the red one. (a) The traced edges of the stem are marked in yellow, the optimized seam in magenta. (b) After roughly aligning the images, Poisson blending [PGB03] reveals a strong structural misalignment at the flower's stem. (c) The technique of Jia *et al.* [JT08] is able to match the corresponding edges, but results in a rather disturbing structural transition at the seam. (d) Our variable correspondence method automatically propagates the necessary deformation of the stem more faithfully into the rest of the image. The according deformation of the blossom is unnoticable to the human observer.



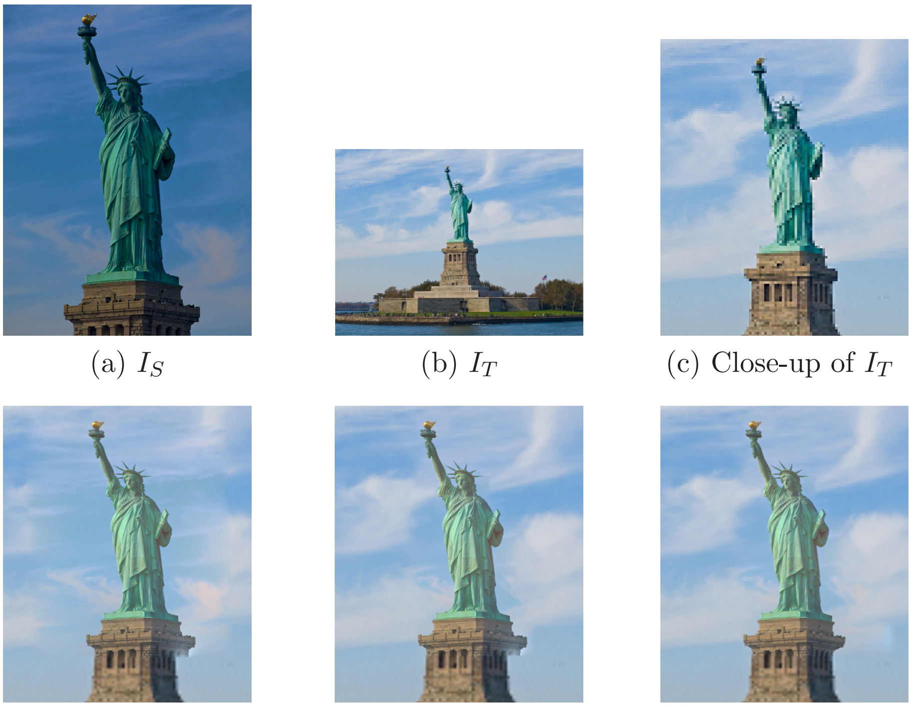
(a) Drag & Drop [JSTS06]

(b) Close-up of (a)

(c) Ours

(d) Close-up of (c)

Figure 10: Compositing of two cropped images (not shown here). Using our approximated correspondence method we can extrapolate splines along the salient edges in the target image into the source image and align these with the source. (a) Even sophisticated methods like the Drag & Drop Pasting method [JSTS06] fail at finding a good seam, due to the misalignment and the small overlap area. (b) Close-up view of the marked transition area in (a) . (c) Our result. By warping the source image slightly the salient edges could be matched and the transition shows less artifacts. (d) Close-up of (c).



(a) I_S

(b) I_T

(c) Close-up of I_T

(d) Poisson [PGB03]

(e) Drag & Drop [JSTS06]

(f) Ours

Figure 11: Internet super-resolution: The source image I_S (a) out of a collection of 35 images downloaded from the image database Flickr is used to upscale a different image I_T (b) and to provide a higher level of detail for the statue. (c) Close-up of the target image I_T. (d) Close-up of the result using Poisson blending by Pérez *et al.* [PGB03]. Note the mismatches at the pedestal. (e) Close-up of the result with an optimized seam using Drag & Drop Pasting [JSTS06]. The clouds look better, but the mismatch at the pedestal is still present. (f) Close-up of our result using variable correspondences which can also handle the structural inconsistencies (f).

References

[ADA+04] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive Digital Photomontage. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):294–302, 2004.

[BA83] P. J. Burt and E. H. Adelson. A Multiresolution Spline With Application to Image Mosaics. *Computer Graphics (Proceedings of SIGGRAPH 1983)*, 17(3):217–236, 1983.

[Bel62] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9(1):61–63, 1962.

[BK89] Ruzena Bajcsy and Stane Kovačič. Multiresolution elastic matching. *Computer Vision, Graphics and Image Processing*, 46(1):1–21, 1989.

[BKD+08] D. Bitouk, N. Kumar, S. Dhillon, P. N. Belhumeur, and S. K. Nayar. Face Swapping: Automatically Replacing Faces in Photographs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 27(3):1–8, 2008.

[Boy01] Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[EESM10] M. Eisemann, E. Eisemann, H. Seidel, and M. Magnor. Photo Zoom: High Resolution from Unordered Image Collections. In *GI '10 (Proceedings of Graphics Interface 2010)*, pages 71–78, 2010.

[EF01] A. A. Efros and W. T. Freeman. Image Quilting for Texture Synthesis and Transfer. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2001)*, 20(3):341–346, 2001.

[FH04] H. Fang and J. C. Hart. Textureshop: Texture Synthesis as a Photograph Editing Tool. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):354–359, 2004.

[GS09] Dong Guo and Terence Sim. Color me right – seamless image compositing. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 444–451. Springer-Verlag, 2009.

[HH10] Charles Han and Hugues Hoppe. Optimizing continuity in multiscale imagery. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2010)*, 29(5):to appear, 2010.

[HL93] Josef Hoschek and Dieter Lasser. *Fundamentals of computer aided geometric design.* A. K. Peters, Ltd., Natick, MA, USA, 1993. Translator-Schumaker, Larry L.

[JSTS06] J. Jia, J. Sun, C. Tang, and H. Shum. Drag-and-Drop Pasting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, 25(5):631–637, 2006.

[JT05] J. Jia and C. Tang. Eliminating Structure and Intensity Misalignment in Image Stitching. In *ICCV '05 (Proceedings of the Tenth IEEE International Conference on Computer Vision)*, pages 1651–1658, 2005.

[JT08] Jiaya Jia and Chi-Keung Tang. Image stitching using structure deformation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):617–631, 2008.

[KSE+03] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):277–286, 2003.

[Low04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[LZPW03] Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss. Seamless Image Stitching in the Gradient Domain. In *ECCV '04 (Proceedings of the Eighth European Conference on Computer Vision)*, volume 4, pages 377–389, 2003.

[Mei79] J. Meinguet. Multivariate interpolation at arbitrary points made simple. *Journal of Applied Mathematics and Physics*, 5:439–468, 1979.

[MP08] J. McCann and N. S. Pollard. Real-time gradient-domain painting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008)*, 27(3):1–7, 2008.

[MV98] J. Maintz and M. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1):1–36, 1998.

[PGB03] P. Pérez, M. Gangnet, and A. Blake. Poisson Image Editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003)*, 22(3):313–318, 2003.

[PL07] F. Pighin and J. P. Lewis. Practical least-squares for computer graphics. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, pages 1–57, New York, NY, USA, 2007. ACM.

[PTVF07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, September 2007.

[RJBJ03] Torsten Rohlfing, Calvin R. Maurer Jr., David A. Bluemke, and Michael A. Jacobs. Volume-preserving nonrigid registration of mr breast images using free-form deformation with an incompressibility constraint. *IEEE Transactions on Medical Imaging*, 22:730–741, 2003.

[RKB04] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut" - Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):309–314, 2004.

[SJMP10] K. Sunkavalli, M. K. Johnson, W. Matusik, and H. Pfister. Multiscale Image Harmonization. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010)*, 29(4):1–10, 2010.

[Sze96] R. Szeliski. Video Mosaics for Virtual Environments. *Computer Graphics and Applications, IEEE*, 16(2):22 –30, 1996.

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

[TM98] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *ICCV '98 (Proceedings of the Sixth International Conference on Computer Vision)*, pages 839–646, 1998.

[UES01] M. Uyttendaele, A. Eden, and R. Szeliski. Eliminating Ghosting and Exposure Artifacts in Image Mosaics. In *CVPR '01 (Proceedings of IEEE Conference on Computer Vision and Pattern Recognition 2001)*, pages 509–516, 2001.

[WY04] Q. Wu and Y. Yu. Feature Matching and Deformation for Texture Synthesis. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004)*, 23(3):364–367, 2004.

[YZC$^+$09] Wenxian Yang, Jianmin Zheng, Jianfei Cai, Susanto Rahardja, and Chang Wen Chen. Natural and seamless image composition with color control. *Transactions on Image Processing*, 18(11):2584–2592, 2009.