High Resolution Image Correspondences for Video Post-Production

Christian Lipski*, Christian Linz*, Thomas Neumann†, Markus Wacker†, Marcus Magnor*

*Computer Graphics Lab
TU Braunschweig
Mühlenpfordtstr. 23, 38106 Braunschweig, GERMANY
email: {lipski,linz,magnor}@cg.cs.tu-bs.de
www: http://www.graphics.tu-bs.de/projects/vvc

†Hochschule für Technik und Wirtschaft Dresden (FH)
Friedrich-List-Platz 1, 01069 Dresden, GERMANY
email: {tneumann,wacker}@informatik.htw-dresden.de
www: http://www.informatik.htw-dresden.de/~wacker/

Abstract

We present an algorithm for estimating dense image correspondences. Our versatile approach lends itself to various tasks typical for video post-processing, including image morphing, optical flow estimation, stereo rectification, disparity/depth reconstruction, and baseline adjustment. We incorporate recent advances in feature matching, energy minimization, stereo vision, and data clustering into our approach. At the core of our correspondence estimation we use Efficient Belief Propagation for energy minimization.

While state-of-the-art algorithms only work on thumbnail-sized images, our novel feature downsampling scheme in combination with a simple, yet efficient data term compression, can cope with high-resolution data. The incorporation of SIFT (Scale-Invariant Feature Transform) features into data term computation further resolves matching ambiguities, making long-range correspondence estimation possible. We detect occluded areas by evaluating the correspondence symmetry, we further apply Geodesic matting to automatically determine plausible values in these regions.

Keywords: Video Post-Production, Optical Flow, Depth Reconstruction, Belief Propagation, Dense Image Correspondences

First presented at the Conference on Visual Media Production 2010, extended and revised for JVRB

1 Introduction

Establishing dense image correspondences between images is still a challenging problem, especially when the input images feature long-range motion and large occluded areas. With the increasing availability of high-resolution content, the requirements for correspondence estimation between images are further increased. High resolution images often exhibit many ambiguous details in places where their low resolution predecessors only show uniformly colored areas, thus the need for smarter and more robust matching techniques arises. Liu et al. [LYT+08] recently proposed a dense matching approach for images possibly showing different scene content. We present an approach for establishing pixel correspondences between two high resolution images and pick up on their idea to incorporate dense SIFT feature descriptors [Low99, Low04],

Figure 1: Belief Propagation Optical Flow. Left to right: two 1920×1080 input images A and B, optical flow from A to B, flow symmetry, rendered in-between image. We specifically tailored our optical flow for image morphing in the presence of large motion and occlusions. We incorporate recent advances in computer vision to produce visually convincing results.

yet we use them for a different purpose. While they identify visually similar regions in low-resolution images, we use them as a descriptor for fine detail in high-resolution images. Our approach provides a versatile tool for various tasks in video post-production. Examples are image morphing, optical flow estimation, stereo rectification, disparity/depth reconstruction, and stereoscopic baseline adjustment.

In order to match fine structural detail in two images, we compute a SIFT descriptor for each pixel in the original high resolution images. To avoid ambiguous descriptors and to speed up computation, we downsample each image by selecting the most representative SIFT descriptor for each $n \times n$ grid cell (typically $n = 4$). An initial lower resolution correspondence map is then computed on the resulting downsampled versions of both images. The 131-dimensional descriptor of each pixel is a combination of the mean RGB color values (3-dimensional) and the representative SIFT descriptor of this cell (128-dimensional). The L^1-norm of this vector describes dissimilarity between two pixels and allows for much clearer distinction between non-corresponding pixels when compared to using just the pixel color as in many previous approaches.

The optical flow between the two images is computed via Efficient Belief Propagation. While the original Belief Propagation implementation by Felzenszwalb et al. [FP06] might not retain crisp borders due to the grid-based message passing scheme, we employ a non-grid-like regularization technique as proposed by Smith et al. [SZJ09] in the context of stereo matching. As memory consumption of Belief Propagation on this scale is still too high for long-range correspondence estimation, we introduce a simple minima-preserving data term compression. For each row and column as well as each pixel of a downsampled version of the data term window, we store the minimal matching cost. Decompression is done by obtaining the maxima of the respective minima of a pixel. During Belief Propagation, a symmetry term ensures consistent results. Occluded regions are identified and inpainted, i.e., plausible correspondence values are estimated where no valid symmetric correspondences can be found. Assuming that each occluded area is surrounded by two independently moving regions, we use Geodesic Matting [BS09] to propagate correspondence information. The resulting image correspondence map is upsampled to its original size and refined locally.

The paper is structured as follows: After a brief survey of related work in Sect. 2, we present our correspondence estimation algorithm in Sect. 3. Various possible applications are presented in Sect. 4 and results are presented in Sect. 5. In our accompanying video, we further give a qualitative demonstration of our rendered results. Finally, we conclude in Sect. 6.

2 Related Work

Belief Propagation has been introduced to the computer vision community by Felzenszwalb and Huttenlocher [FP06]. Since then, it has received considerable attention: Several extensions have been proposed to speed up the calculation, e.g., [LTWS11, GGC11]. It was used by Liu et al. [LYT+08] in combination with SIFT features for dense correspondence estimation between similar images. However, these correspondences were only established for thumbnail-sized images and did not suffice for the tasks our approach can cope with (i.e., occlusion handling, symmetric correspondences, high resolution data).

Optical flow algorithms are closely linked to our approach. A recent survey of state-of-the-art algorithms has been conducted by Baker et al. [BSL+07]. The key difference in concept is that optical flow algorithms typically derive continuous flow vectors instead of discrete pixel correspondences. Additionally, optical flow computation typically assumes

color/brightness constancy between images. We qualitatively compare our method with a state-of-the-art optical flow algorithms [WPB10, BM11, CP11] and show an additional extensive comparison with [SPC09] in our accompanying video. Recently, perception-based image interpolation algorithms have been presented [SLAM08, SLW+11] that concentrate on matching visible edges in image pairs. However, these algorithms neglect fine texture details available in high-resolution images.

Recently, commercial tools for stereo footage post-production have reached a mature stage of development (e.g. Ocula [Hal08]). Our image correspondence algorithm could be easily integrated into any stereoscopic post-production pipeline, since it works in an unsupervised fashion, only requires image pairs as input and can be applied to various post-production tasks, including image rectification and disparity estimation.

3 Belief Propagation for image correspondences

Belief Propagation estimates discrete labels for every vertex in a given graph, i.e., for every pixel in a given image. Although we do not achieve sub-pixel accuracy with Belief Propagation, its robustness makes it an appealing option for discrete energy minimization problems. In a nutshell, establishing pixel correspondences between two images with Belief Propagation works as follows: Matching costs for every possible pixel match in a given search window are computed for each pixel. Typically, the L^1 norm of the difference in pixel color serves as a basic example for this matching cost. Neighboring pixels iteratively exchange their (normalized) matching costs for potential correspondences. This message passing process regularizes the image correspondences and finally converges to a point where consensus about final pixel correspondences is reached. A combination of the individual matching costs and the exchanged messages determines the final pixel correspondences. As a result, a discrete correspondence vector $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ is assigned to every pixel location $\mathbf{p} = (x, y)$ that encodes the correspondence to pixel location $\mathbf{p}' = (x + u(\mathbf{p}), y + v(\mathbf{p}))$ in the second image. For a thorough introduction we would like to refer to [FP06].

Assuming that the search space is the whole image, computational complexity and memory usage are as high as $O(L^4)$, where L is the image width. By decoupling u and v (the horizontal and vertical component of the correspondence vector), the complexity for message passing can be reduced to $O(L^3)$, as proposed by Liu et al. [LYT+08]. Still, the evaluation of the matching costs runs in $O(L^4)$. We formulate the correspondence estimation as an energy minimization problem, our energy functional is based on the one proposed by Liu et al. [LYT+08]

$$\begin{aligned} E(\mathbf{x}) &= \sum_{\mathbf{p}} \|d_1(\mathbf{p}) - d_2(\mathbf{p} + \mathbf{w}(\mathbf{p}))\|_1 \\ &+ \sum_{(\mathbf{p},\mathbf{q})\in\epsilon} \min(\alpha|u(\mathbf{p}) - u(\mathbf{q})|, d) \\ &+ \sum_{(\mathbf{p},\mathbf{q})\in\epsilon} \min(\alpha|v(\mathbf{p}) - v(\mathbf{q})|, d) \quad (1) \end{aligned}$$

Where $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ is the correspondence vector at pixel location $\mathbf{p} = (x, y)$. In contrast to the original SIFT flow implementation, $d_i(\mathbf{p}) = [c_i(\mathbf{p}), s_i(\mathbf{p})]$ is a 131-dimensional descriptor vector, containing both color information $c_i(\mathbf{p}) \in \mathbb{R}^3$ and the SIFT descriptor $s_i(\mathbf{p}) \in \mathbb{R}^{128}$. Each descriptor entry has a value between 0 and 255. We set $\alpha = 160$ and $d = imagewidth \times 5$. In addition, the pixel neighborhood ϵ is not simply defined by the image lattice, as we will show in Sect. 3.3. In contrast to SIFT flow, we do not penalize large motion vectors, since we explicitly try to reconstruct such scenarios.

As Liu et al. [LYT+08] demonstrated, this energy functional can be minimized with Efficient Belief Propagation. Before we start with a detailed description of our pipeline, we would like to present our extension of the energy functional to symmetric correspondence maps.

3.1 A Symmetric Extension

Since we want to enforce symmetry between bidirectional correspondence maps, we introduce a symmetry term similar to the one proposed by Alvarez et al. [ADPS07].

To our energy functional we add a symmetry term:

$$\begin{aligned} E(\mathbf{x}) &= \sum_{\mathbf{p}} \|d_1(\mathbf{p}) - d_2(\mathbf{p} + \mathbf{w_{12}})\|_1 \\ &+ \sum_{(\mathbf{p},\mathbf{q})\in\epsilon} \min(\alpha|u(\mathbf{p}) - u(\mathbf{q})|, d) \end{aligned}$$

$$+ \sum_{(\mathbf{p},\mathbf{q})\in\epsilon} \min(\alpha|v(\mathbf{p}) - v(\mathbf{q})|, d)$$
$$+ \sum_{\mathbf{p}} \min(\alpha\|\mathbf{w_{12}} + \mathbf{w_{21}}(\mathbf{p} + \mathbf{w_{12}})\|_2, d)$$

Please note that now two correspondence maps coexist: $\mathbf{w_{12}}$ and $\mathbf{w_{21}}$. They are jointly estimated and evaluated after each Belief Propagation iteration. It proved to be sensible to assign the same weighting and truncation values α and d to the symmetry term that are also used for message propagation. When comparing SIFT descriptors, the L^1 distance is used as proposed by Lowe [Low04]. Due to decoupling of vertical and horizontal flow introduced by Liu et al. [LYT+08], the smoothness term also uses the L^1 norm. The construction of the symmetric term does not impose any constraints on the distance function, we empirically determined that the L^2 norm works well when evaluating symmetry.

Our correspondence estimation consists of six consecutive steps. The first three steps (Sect. 3.2, 3.3 and 3.4) are preprocessing steps for the actual Belief Propagation optimization. After an initial low resolution solution has been computed, possibly occluded parts are inpainted, as described in Sect. 3.5. As a last step, the correspondence map is upsampled and locally refined as described in Sect. 3.6.

3.2 SIFT Descriptor downsampling

Liu et al. [LYT+08] designed their SIFT flow with the goal in mind to match images that may only be remotely similar, which comes close to the original intention to find only a few dominant features [Low04]. Our goal, on the other side, is to match very similar images. We are faced with the challenge to discard possible matching ambiguities that occur when only color information is used as a dissimilarity measure between pixels. We use SIFT features to capture detail information about the scene, hence we generate one feature for every pixel of the full resolution images and search at the bottom layer of the SIFT scale-space pyramid. In order to only capture the most prominent details, a single representative feature $s_i(\mathbf{p_g})$ is kept for every $n \times n$ grid $\mathbf{g}$ of pixel locations.

The search for a representative descriptor is inspired by the work of Frey et al. [FD07], who use their Affinity Propagation technique to search for clusters in data and simultaneously identify representatives for these clusters. Since we have a pre-defined arrangement of clusters, i.e. we want to roughly preserve the $n \times n$ pixel block structure, we used our fixed clusters and only adopt their suggestion that a cluster's representative should be the one most similar to all other members of the cluster.

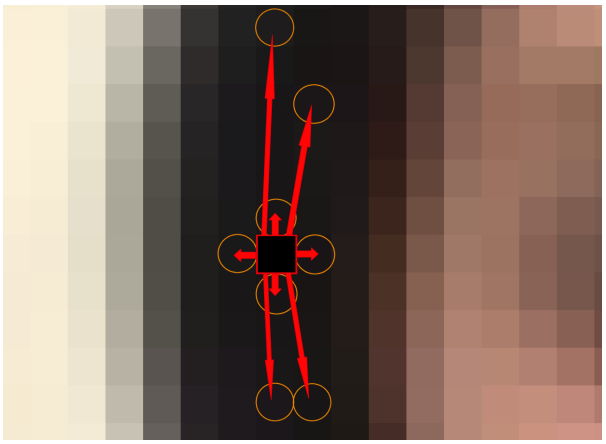


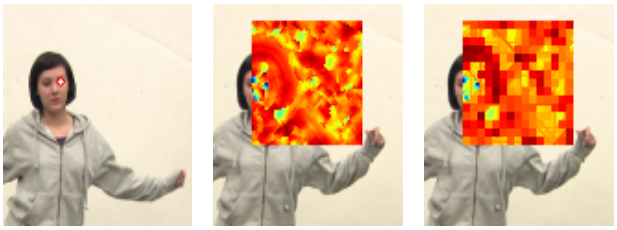
Figure 2: In our Belief Propagation scheme a single pixel (red square) exchanges messages with its spatial neighbors as well as pixels of similar color (orange circles). The underlying graph structure is obtained by computing minimal spanning trees.

Hence, the representative descriptor for each pixel block is the one in the $n \times n$ pixel cell that has the lowest cumulative L^1 distance to all other descriptors.

A downsampled representation of the image is then computed, where every grid cell in the downsampled image is represented by a single descriptor. This descriptor consists of the mean color value of the cell and the representative SIFT descriptor.

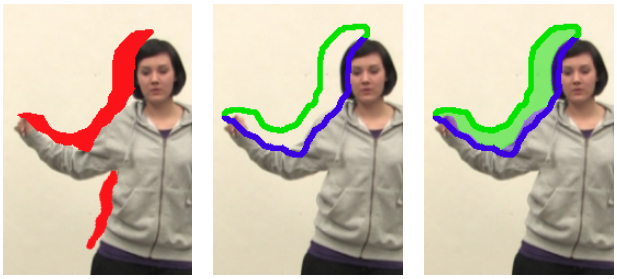
3.3 Construction of Message Passing Graph

The fact that image regions of similar color often share common properties, e.g. similar motion, is often exploited in regularization techniques. Typically, this is achieved by applying an anisotropic regularization, i.e., neighboring pixels with different colors exert less influence on each other than pixels with a similar color. This technique has two drawbacks: First, regularization is decelerated. Second, the grid-aligned regularization still manifests in jaggy borders around correspondence discontinuity edges. Recently, Smith et al. [SZJ09] proposed the construction of a non-grid-like regularization scheme. While they applied this technique to stereo matching with a variational approach, we adapt their idea to our Belief Propagation approach, see Fig. 2. We build an initial graph where each vertex represents a pixel location of the



(a) (b) (c)

Figure 3: Data term compression. For each pixel in a source image (a), matching costs have to be evaluated for Belief Propagation. One common approach is to precompute matching costs in a predefined window (b). However, this leads to very high memory load. Our approach uses a simple minima-preserving compression of these matching cost windows (c). The minima of each pixel block, each column, row and the diagonal lines of the search window are stored. During decompression, the maximum of these values determines the matching cost for a given location. While regions with high matching costs are not recovered in detail, local minima are preserved with high accuracy.



(a) (b) (c)

Figure 4: Occlusion removal. Regions with asymmetric correspondences (a), red, are processed in a two-step algorithm. First, a k-means clustering ($k = 2$) reveals the two predominant offset directions (b), blue and green. These two sets of images are used as input for Geodesic Matting (c). Depending on which label is assigned to an occluded pixel, the local median foreground or background motion is assigned.

image. Edges connect pixels that have a certain maximal distance. Typically, we set this maximal distance to 20 pixels. Each edge is assigned a weight that corresponds to the L^1 norm of the color and position of the connected pixels. As in [SZJ09], a minimum spanning tree is calculated using Kruskal's algorithm [Kru56]. The edges of the spanning tree are added as neighbors of the pixel and removed from the initial graph. Repeating this procedure on the initial graph once more leads to an average number of 4 neighbors per pixel. We further add the 4 direct image grid neighbors from the image grid to the neighborhood ϵ of a pixel location.

3.4 Data Term Compression

One bottleneck in Belief Propagation with SIFT features is the computation of matching costs $\|s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w})\|_1$ between pixel locations $s_1(\mathbf{p})$ and $s_2(\mathbf{p}+\mathbf{w})$. Liu et al. [LYT+08] precompute the matching costs before message passing. The alternative is to re-evaluate matching costs on demand, which happens at least once during each iteration when the data term is evaluated. This results in $262 (= 131 \times 2)$ memory lookups per pixel comparison. Since storing data terms is not an option with our high resolution data and on-the-fly evaluation leads to run-times of several days, we design a simple data term compression technique. We precompute all possible matching costs for a single pixel $s_1(\mathbf{p})$ and its potential matching candidates, see Fig. 3. Since it is quite likely that a pixel will be finally matched to a candidate with a low dissimilarity (i.e., low matching cost), we employ a minima-preserving compression technique that loses detail in areas where high matching scores prevail. For each $m \times m$ grid cell of the original data term, the minimum is stored. In addition, for each row and column of the matching window, the respective minimum is stored. The same applies to the minima along the two diagonal directions.

During decompression, the maximum of the these minima is evaluated, resulting in 5 memory lookups (the minimal grid cell value, the minimal row and column values and the minimal values of the two diagonals). When setting $m = 4$, at a data term window size of typically 160×160 pixels, the memory usage per term is reduced from $160 \times 160 = 25600$ float values to $40 \times 40 + 4 \times 160 = 2240$ float values.

3.5 Occlusion Removal

It can be observed that the introduction of a symmetry term leads to quasi-symmetric warps in non-occluded areas. Hence, we use the symmetry $\|\mathbf{w_{12}} + \mathbf{w_{21}}(\mathbf{p} + \mathbf{w_{12}})\|_2$ of two opposing correspondence maps $\mathbf{w_{12}}$ and $\mathbf{w_{21}}$ as a measure of occlusion.

For each of these two simultaneously estimated maps, asymmetric correspondence regions are identi-

fied and treated independently, see Fig. 4. First, all occluded regions are filled with correspondence information values using diffusion. All pixels which would lie outside the actual image boundaries according to their correspondence vector are discarded as boundary occlusions and their diffused values are kept.

Assuming that each of the remaining occlusion regions is confined by a foreground and a background region that move incoherently, we perform a k-means clustering ($k = 2$) of the border region outside each occluded area, all pixels in these border regions are clustered according to their two correspondence vector components $u(\mathbf{p})$ and $v(\mathbf{p})$. The resulting pixel sets serve as input data for a binary Geodesic Matting [BS09] that assigns each pixel in the occluded area a foreground or background label. Depending on the label, each pixel is either assigned the foreground or background flow. After the labeling is computed, the median value of the $\mathbf{n}$ nearest neighbors of the foreground or backround region is assigned. Typically, we set $\mathbf{n} = 20$ to get smooth results.

3.6 Upsampling and Refinement

The low resolution correspondence is upsampled as follows. On the high-resolution map, the pixel that was chosen as the representative SIFT descriptor is assigned the values that results from the low-resolution Belief Propagation (scaled by factor n). For all remaining pixels, the value of the nearest representative pixel in gradient space is assigned. These assigned correspondence values serve as a prior for a local refinement. Like on the low resolution level, symmetric Belief Propagation is used to obtain the final per-pixel correspondence. The crucial difference is that the search window is set to a very small size (typically $(n * 2 + 1) \times (n * 2 + 1)$ pixel) and that it is located around a correspondence prior $\mathbf{p}+\mathbf{w_{12}}$ and not around the pixel location $\mathbf{p}$ itself.

4 Applications

We identified two main showcase applications for our algorithm. First, we use our dense correspondences for image morphing. While traditional approaches usually employ a user-assisted workflow [BN92], we strive to compute motion vectors between images automatically. We use a simple forward warping scheme to seamlessly render intermediate views between two frames.

Figure 5: Quasi-Euclidean stereo image rectification [FI08] using a randomly sampled subset of correspondences. For visualization purposes, only 300 out of the $10,000$ sampled correspondences are rendered on top of the rectified image.

A GPU rendering approach, inspired by Stich et al. [SLAM08], is used. We create a dense vertex mesh for each input image and forward warp it according to the high-resolution flow field. We discard fragments whose local divergence in the correspondence map exceed a threshold of 4 pixels. If ambiguities arise (i.e. two fragments of a mesh overlap), the fragment with lower symmetry is discarded. The two forward-warped meshes are alpha-blended, pixel fragments with very low symmetry are again discarded in the presence of pixels with symmetric correspondence. Fig. 6 shows some image warping results.

Our second application is post-processing of stereo footage. Although disparity computation is a 1-D search problem in theory, actual stereoscopic footage often violates the epipolar constraint. This is always true for a convergent camera rig and may also happen when a parallel setup of cameras is slightly misaligned. Temporal misalignment of stereo recordings will also violate the epipolar constraint. Even if the two video streams are synchronized up to a single frame [MSMP08], the subframe offset between cameras remains. In addition, temporal misalignment will violate the epipolar constraint in the presence of moving objects. Dense pixel correspondences for each stereo pair are computed using our method. Since we expect that the epipolar constraint is violated, we allow for small displacements along the y-axis. For an ad hoc rectification of an image pair, we use the quasi-euclidean method proposed by Fusiello et al. [FI08]. Typically, this algorithm expects a set of sparse correspondences as its input. Robust rectification is possi-

ble using only a subset of our symmetric correspondences. We use a set of $10,000$ randomly sampled symmetric correspondences for rectification. Now, per-pixel disparities can be deduced from the correspondence maps and the rectification homographies. Disparity-based effects such as artificial depth of field, fog, baseline modification, and layer segmentation can be applied to the footage (see Fig. 8) and combined with traditional flow-based effects, such as slow motion rendering. We present a selection of results in our accompanying video.

Numerous other application scenarios for our image correspondence algorithm exist. One of the most interesting is the integration into free-viewpoint-video systems that may depend on disparity [ZKU+04] or optical flow estimation [CW93, LLB+10, MLSM10].

5 Results and Discussion

For a qualitative assessment of our proposed approach, we would like to refer to the accompanying video which can be found on the project website.[1] We estimated correspondences for the Middlebury optical flow data set and rendered image morphs of the Backyard, Dumptruck, and Evergreen sequence. Among these, the Backyard scene is the most challenging one, since it features a small, quickly moving ball. Using our approach, the ball is cut out properly as a result of our regularization scheme. Without this it either tends to drag its vicinity along its motion path, or a too strong regularization assigns the background flow to the whole area.

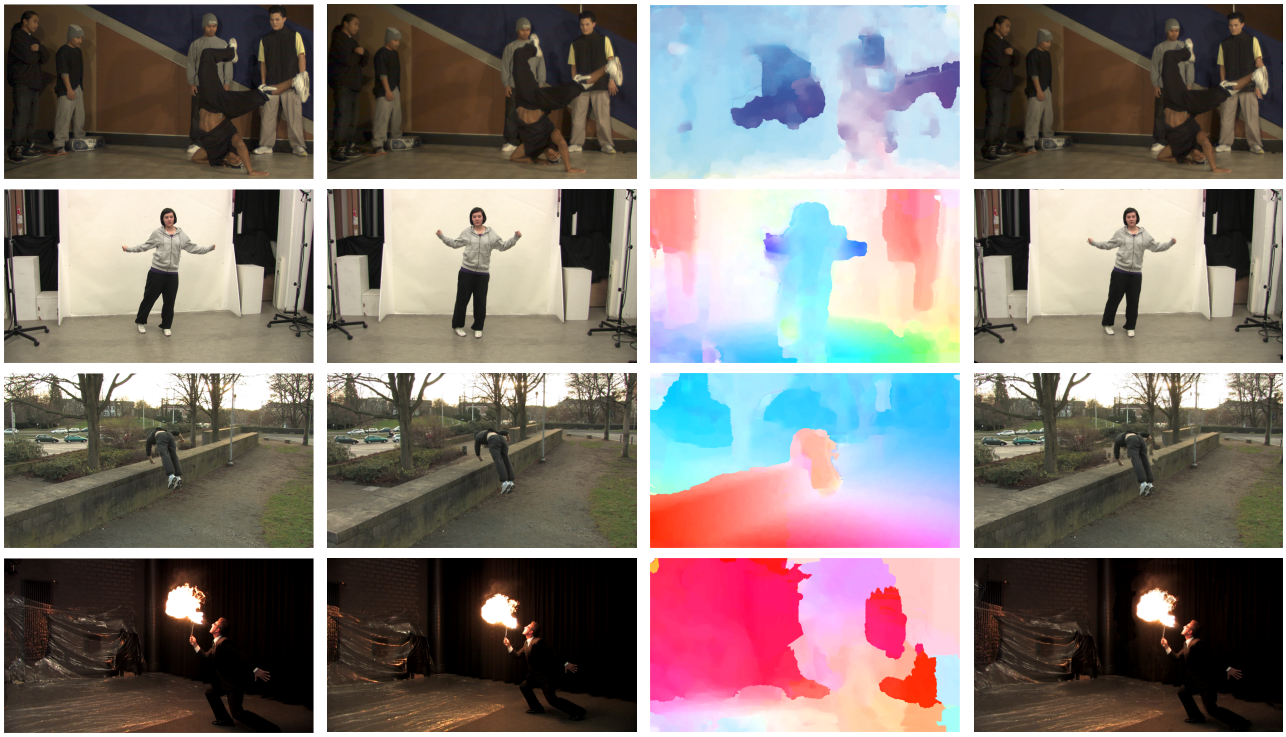
For all following scenarios, we deliberately chose image pairs with both camera and object motion. The Breakdancer scene taken from the dataset of Zitnick et al. [ZKU+04] features noisy images and fast scene motion. Still, image correspondences are successfully established. Even the shadow of the breakdancer in the background moves plausibly. The shortcomings of our simple rendering approach manifest in motion streak artifacts around the right foot of the breakdancer. Please note that in their original work, Zitnick et al. [ZKU+04] only performed stereo matching. Our algorithm does not exploit the epipolar constraint and copes with moving objects. The dancer scene looks a lot less challenging at first glance. However, the ground surface is quite demanding, since shadows and reflections of the dancer and background are visible. The Parcours scene can be interpreted as a failure case of our approach. Although large parts of the scene are matched correctly, the occluded regions around the parcours runner are not handled correctly by the Geodesic Matting. The background is too cluttered to allow a consistent local color model that separates background from foreground. The Fireball sequence shows that the algorithm copes well with illumination changes. However, the opening crack around the Fireball impairs overall rendering quality.

To emphasize the high complexity of our test scenes, we also estimated optical flows for the Parcours, Dancing, and the Fireball scene with three state-of-the-art optical flow implementations [WPB10, BM11, CP11]. Fig. 7 shows the flow fields and interpolated images side-by-side in comparison to our approach. Notice that the same rendering technique was used to generate all the images. The fast GPU implementation of Werlberger et al. [WPB10] focuses on quick runtime and allows the computation of the image correspondences within several seconds, however the rendered results using this approach show severe visual artifacts. When comparing to the TV-L^1 motion estimation by Chambolle and Pock [CP11] and the large displacement optical flow by Brox and Malik [BM11], it becomes apparent that our proposed technique bears great potential. While the actual foreground objects are covered well by both algorithms as well as our implementation, challenging details in the background, e.g., the trees (Parcour scene) or the plastic foil (Fireball scene), are correctly matched only by our approach. In an additional comparison with the approach by Steinbrcker et al. [SPC09], included in the accompanying video, we made similar observations.

As a second application scenario we chose the post-production of stereoscopic footage. The processed Heidelberg sequence features repetitive patterns on the house in the background and changing illumination on the two tourists. After image rectification (Fig. 5) and disparity estimation, we applied visual effects to demonstrate the versatility of our approach. We show a synthetic depth of field effect combined with slow motion. First, we render in-between frames for the slow-motion effect using image morphing. We additionally morph disparity maps and apply a variable blur. The blur kernel size is determined per pixel by disparity.

All results were obtained on an Intel Quadcore PC with 2.66 GHz and 4 GB RAM. Depending on the maximum displacement vector, computation took be-

[1] http://graphics.tu-bs.de/projects/vvc



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

Figure 6: Image Morphing results. a) and b) two input images, c) offset vector fields from first to second image, coded in optical flow notation, d) rendered in-between image. Note that while our approach yields overall robust results, details such as the reflection of the dancer or the shadow of the breakdancer are preserved. We would like to refer to the accompanying video for further assessment.

tween 20 minutes (Heidelberg sequence) and 12 hours (Parcours sequence) per image pair.

Limitations The most severe limitation is the long run-time of our algorithm, which only allows processing single image pairs or short video clips. Until now, the implementation is completely CPU based. This is also due to the fact of high memory consumption. Since we compute two symmetric correspondence maps in parallel, memory consumption is as high as 4 GB for large displacements (e.g., the Parcours sequence which features displacements of up to 400 pixels). In order to move the computation to GPU, an even more compact data representation than our currently employed compression scheme has to be developed. Obtaining a suitable search window size is another problem, since the maximum displacement has to be known prior to the correspondence estimation. If the window size is set too small, some correspondences will not be established correctly. If it is set too high, the overall run-time will be excessively high. The last limitation of our approach is our simple rendering scheme for image morphing. Especially at occlusion borders, rendering artifacts can occur. A more elaborate rendering mechanism such as presented by Mahajan et al. [MHM+09] may be used in the future.

6 Conclusion

We presented an algorithm for robustly estimating pixel correspondences in image pairs. We showed that our approach can be used as a versatile tool for various video post-production tasks. Rendered results of challenging scenes and comparison with the State of the Art prove the robustness and accuracy of our approach. Further improvement has to be made on computational complexity and memory consumption, since the current run-times are clearly too long for many applications.

7 Acknowledgements

We would like to thank the company dongleware for providing the ”Heidelberg” stereo sequence [Ame11].

This work has been funded by the European Research Council ERC under contract No.256941 "Reality CG" and by the German Science Foundation, DFG MA 2555/1-3 and MA 2555/4-2.

References

[ADPS07] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez, *Symmetrical dense optical flow estimation with occlusions detection*, IJCV **75** (2007), no. 3, 371–385, ISSN 1573-1405.

[Ame11] Meinolf Amekudzi, *Dongleware website*, 2011, /www.dongleware.de.

[BM11] T. Brox and J. Malik, *Large displacement optical flow: descriptor matching in variational motion estimation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **33** (2011), no. 3, 500–513, ISSN 0162-8828.

[BN92] Thaddeus Beier and Shawn Neely, *Feature-based image metamorphosis*, SIGGRAPH'92 Proceedings of the 19th annual conference on computer graphics and interactive techniques **26** (1992), no. 2, 35–42, ISSN 0097-8930.

[BS09] Xue Bai and Guillermo Sapiro, *Geodesic Matting: A Framework for Fast Interactive Image and Video Segmentation and Matting*, IJCV **82** (2009), no. 2, 113–132, ISSN 0920-5691.

[BSL+07] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski, *A Database and Evaluation Methodology for Optical Flow*, IEEE 11th International Conference on Computer Vision ICCV (Colorado Springs, USA), IEEE Computer Society, 2007, pp. 1–8.

[CP11] Antonin Chambolle and Thomas Pock, *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging*, J. Math. Imaging Vis. **40** (2011), no. 1, 120–145, ISSN 0924-9907.

[CW93] Shenchang Eric Chen and Lance Williams, *View interpolation for image synthesis*, Proc. of ACM SIGGRAPH'93 (New York), ACM Press/ACM SIGGRAPH, 1993, pp. 279–288, ISBN 0-89791-601-8.

[FD07] Brendan J. Frey and Delbert Dueck, *Clustering by Passing Messages Between Data Points*, Science **315** (2007), no. 5814, 972–976, ISSN 1095-9203.

[FI08] A. Fusiello and L. Irsara, *Quasi-euclidean Uncalibrated Epipolar Rectification*, 19th International Conference on Pattern Recognition ICPR 2008 (Tampa, FL, USA), 2008, pp. 1–4, ISBN 978-1-4244-4420-5.

[FP06] P. F. Felzenszwalb and D. P.Huttenlocher, *Efficient Belief Propagation for Early Vision*, IJCV, vol. 70, 2006, pp. 41–54.

[GGC11] Scott Grauer-Gray and John Cavazos, *Optimizing and auto-tuning belief propagation on the GPU*, Proceedings of the 23rd international conference on Languages and compilers for parallel computing (Berlin, Heidelberg), LCPC'10, Springer-Verlag, 2011, pp. 121–135, ISBN 978-3-642-19594-5.

[Hal08] Lucy Hallpike, *The Role of Ocula in Stereo Post Production*, 2008, .

[Kru56] Joseph B. Kruskal, *On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem*, Proceedings of the American Mathematical Society **7** (1956), no. 1, 48–50, ISSN 0002-9939.

[LLB+10] Christian Lipski, Christian Linz, Kai Berger, Anita Sellent, and Marcus Magnor, *Virtual Video Camera: Image-Based Viewpoint Navigation Through Space and Time*, Computer Graphics Forum **29** (2010), no. 8, 2555–2568, ISSN 1467-8659.

[Low99] David G. Lowe, *Object Recognition from Local Scale-Invariant Features*, Proceedings of the International Conference on Computer Vision (Colorado Springs, USA), ICCV '99, vol. 2, IEEE Computer

Society, 1999, pp. 1150–1157, ISBN 0-7695-0164-8.

[Low04] David G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, Int. J. Comput. Vision **60** (2004), no. 2, 91–110, ISSN 1573-1405.

[LTWS11] Ruxandra Lasowski, Art Tevs, Michael Wand, and Hans-Peter Seidel, *Wavelet Belief Propagation for Large Scale Inference Problems*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011) (Colorado Springs, USA), IEEE Computer Society, 2011, pp. 1921–1928, ISBN 978-1-4577-0394-2.

[LYT+08] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman, *SIFT Flow: Dense Correspondence across Different Scenes*, ECCV '08: Proceedings of the 10th European Conference on Computer Vision, 2008, pp. 28–42, ISBN 978-3-540-88689-1.

[MHM+09] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur, *Moving Gradients: A Path-Based Method for Plausible Image Interpolation*, ACM Trans. Graph. **28** (2009), no. 3, ISSN 0730-0301, Article No. 42.

[MLSM10] Benjamin Meyer, Christian Lipski, Björn Scholz, and Marcus Magnor, *Real-time Free-Viewpoint Navigation from Compressed Multi-Video Recordings*, 3DPVT 2010, 2010.

[MSMP08] Benjamin Meyer, Timo Stich, Marcus Magnor, and Marc Pollefeys, *Subframe Temporal Alignment of Non-Stationary Cameras*, Proc. British Machine Vision Conference BMVC '08, 2008, ISBN 978-1-901725-36-0.

[SLAM08] Timo Stich, Christian Linz, Georgia Albuquerque, and Marcus Magnor, *View and Time Interpolation in Image Space*, Computer Graphics Forum **27** (2008), no. 7, 1781–1787, ISSN 1467-8659.

[SLW+11] Timo Stich, Christian Linz, Christian Wallraven, Douglas Cunningham, and Marcus Magnor, *Perception-motivated interpolation of image sequences*, ACM Trans. Appl. Percept. **8** (2011), no. 2, 1–25, ISSN 1544-3558, Article no. 11.

[SPC09] F. Steinbrücker, T. Pock, and D. Cremers, *Large Displacement Optical Flow Computation without Warping*, IEEE International Conference on Computer Vision (ICCV) (Colorado Springs, USA), IEEE Computer Society, 2009, pp. 1609–1614, ISBN 978-1-4244-4420-5.

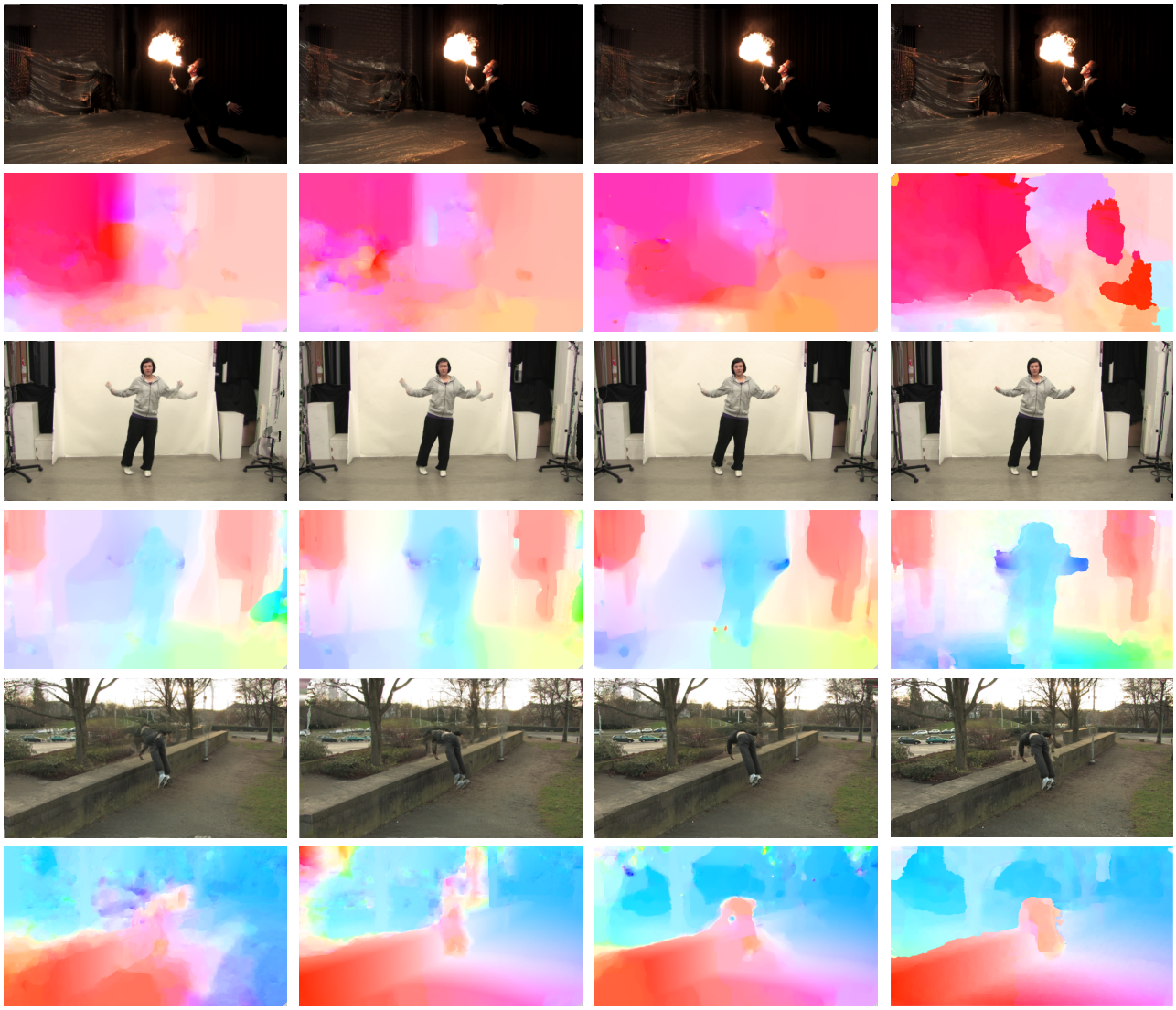
[SZJ09] B. M. Smith, Li Zhang, and Hailin Jin, *Stereo matching with nonparametric smoothness priors in feature space*, Computer Vision and Pattern Recognition, IEEE Computer Society Conference on (2009), 485–492.

[WPB10] Manuel Werlberger, Thomas Pock, and Horst Bischof, *Motion Estimation with Non-Local Total Variation Regularization*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (Colorado Springs, USA), IEEE Computer Society, 2010, pp. 2464–2471, ISBN 978-1-4244-6984-0.

[ZKU+04] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon A. J. Winder, and Richard Szeliski, *High-quality video view interpolation using a layered representation*, ACM Trans. Graph. **23** (2004), no. 3, 600–608, ISSN 0730-0301.

Citation

Christian Lipski, Christian Linz, Thomas Neumann, Markus Wacker, and Marcus Magnor, *High Resolution Image Correspondences for Video Post-Production*, Journal of Virtual Reality and Broadcasting 9(2012), no. 8, December 2012, urn:nbn:de:0009-6-35543, ISSN 1860-2037.



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

Figure 7: Comparison of image interpolation results using state-of-the-art optical flow algorithms. Results are shown in columns, computed using a) the GPU implementation by Werlberger et al. [WPB10], b) TV-L^1 motion estimation by Chambolle and Pock [CP11], c) large displacement optical flow by Brox and Malik [BM11], and d) our approach. From top to bottom, the columns of the figure show an interpolated image and below the color-coded correspondence field. Errors in the estimated flow field typically show up as ghosting artifacts. Only our approach is able to correctly estimate the motion of the background in the fire scene (row 1 and 2) and the large arm motion of the dancer (row 3 and 4). In the parcour scene (row 5 and 6), our approach fails at the borders of the dancer, but correctly estimates correspondences of the trees in the background and manages to find the huge displacement of the wall due to change of camera perspective.

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

Figure 8: Results for the stereoscopic Heidelberg scene. a) Original image (left camera image), b) disparity, c) synthetic depth of field, d) fog and e) baseline editing. Our approach can be used for various tasks in stereoscopic post-production. To view the stereoscopic image in 3D please use cyan-red anaglyph glasses.