Improving Dense Image Correspondence Estimation with Interactive User Guidance

Kai Ruhl, Benjamin Hell, Felix Klose, Christian Lipski,
Sören Petersen and Marcus Magnor
Computer Graphics Lab, TU Braunschweig
Muehlenpfordtstrasse 23, 38106 Braunschweig, Germany
{ruhl,hell,klose,lipski,petersen,magnor}@cg.tu-bs.de *

ABSTRACT

High quality dense image correspondence estimation between two images is an essential pre-requisite for view interpolation in visual media production. Due to the ill-posed nature of the problem, automated estimation approaches are prone to erroneous correspondences and subsequent quality degradation, e.g. in the presence of ambiguous movements that require human scene understanding to resolve. Where visually convincing results are essential, artifacts resulting from estimation errors must be repaired by hand with image editing tools. In this paper, we propose a new workflow alternative by fixing the correspondences instead of fixing the interpolated images. We combine realtime interactive correspondence display, multi-level user guidance and algorithmic subpixel precision to counteract failure cases of automated estimation algorithms. Our results show that already few interactions improve the visual quality considerably.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion*; H.5.2 [**Information Interfaces and Representation**]: User Interfaces—*Interaction styles*

Keywords

dense image correspondence estimation, optical flow, interactive, user input, view interpolation

1. INTRODUCTION

In visual media production, view interpolation is used for a multitude of purposes, from frame upsampling (purely in the temporal domain), freeze-rotate shots (purely in the spatial domain) to combinations of both.

A typical three-stage workflow consists of estimating dense image correspondences, generating interpolated views, and correcting the interpolated frames in an image editing tool.

*Video and further paper details available under:
`http://graphics.tu-bs.de/publications/ruhl2012acmmm/`

MM'12, October 29–November 2, 2012, Nara, Japan.

(a) (b) (c)

Figure 1: User-guided correspondence estimation. (a) image with user hints in green and blue (b) overlay, unsupervised run (c) overlay, user-guided run.

The accuracy of the correspondences influences the effort one has to spend on correcting the interpolated frames; the more interpolated frames are rendered, the more favorable it is to correct an error in the correspondence map instead of in all interpolated frames.

In this paper, we focus on improving the correspondence estimation step. Our proposed production workflow is to estimate dense image correspondences while leveraging user interaction, generate improved interpolated views, and thus eliminate or greatly reduce manual adjustment efforts. The more difficult the correspondence estimation (e.g. fast movements, chaotic situations with much ambiguity), the greater is the benefit gained by human guidance.

The steady rise of GPU power and the development of realtime capable optical flow GPU implementations have made interactivity feasible also for large images. Our approach is novel in being the first to explore *interactive manipulation* for dense image correspondence estimation.

1.1 Related work

Dense image correspondence estimation is an active research area of both computer graphics and computer vision.

Optical flow. The last decade has seen impressive improvements, fueled in part by quantitative evaluation benchmarks [1]. Contemporary algorithms achieve subpixel accuracy in continous space [9, 7] or focus on large displacements by sampling [3]. However, due to the ill-posed problem setting, failure cases are still frequent, e.g. visual ambiguities, violations of brightness or gradient constancy assumptions. Occlusions are also problematic because the optical flow model simply does not consider it, although e.g. Sun et al. [6] perform simultaneous layer and depth order estimation for small motions.

Flow Correction Tools. Correcting dense image correspondences manually is a recent field. While supplying priors is a common technique [2], interactive or post-estimation correction is rare. The commercial tool Ocula [8] edits stereo

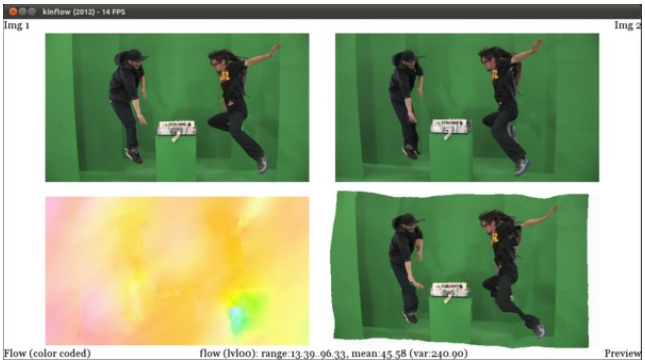


Figure 2: User interface with four views. Top row: Source and target image I_1 and I_0. Bottom row: Correspondence estimation (color-coded) and interpolation preview (warped I_1).

disparity maps after estimation. The work of Klose et al. [4] focuses on post-estimation correction of image correspondences. In contrast, we apply *interactive* correction that benefits from *ongoing* algorithmic refinement.

2. USER INTERFACE

Our interactive tool assists a visual artist in the correspondence estimation process (Fig. 2). The top row displays the source and target images, the bottom row shows the current flow field estimation (color-coded) together with a preview of the warped source image. This allows a quality assessment, where the warped source image should ideally match the target image.

In order to formulate appropriate user actions, we first identify issues with a purely automated estimation and then define user guidance operations to ameliorate them.

2.1 Problem formulation

While contemporary optical flow algorithms achieve impressive results for a wide range of input images, there are reccuring errors that have the common trait of being readily noticable to a human observer, but hard to address computationally. The most common errors include:

(**E1**) long displacement in wrong direction or magnitude, caused by ambiguities such as several similar objects. In classical coarse-to-fine warping approaches (as opposed to dedicated large displacement algorithms), objects that are smaller than their movements also disappear in the image pyramid downsampling.

(**E2**) long or short displacement in wrong direction or magnitude, caused by violation of the brightness constancy assumption, e.g. glossy objects look different in the two respective images.

(**E3**) continuity where there should be a discontinuity, caused by uncertainty on where to split the flow. Anisotropic flow [7] is a partial remedy, but cannot naturally distinguish between object boundaries and boundaries within an object, e.g. hand and wall vs. hand and sleeve: both have distinct colors but only one is an object boundary.

(**E4**) discontinuity where none should exist. If forced to diverge, algorithms cannot make scene-based preferences about the location of the discontinuity.

Problems (**E1**) and (**E2**) relate to displacement decisions, and (**E3**) and (**E4**) relate to discontinuity decisions.

2.2 User guidance operations

While (**E1**) has been addressed by algorithms with a local displacement sampling of more than one pixel [5, 3], ambiguities like several similar objects will still cause confusion. (**E2**) will hold true for all algorithms that rely on the brightness constancy assumption or a variant thereof. We address this issue by:

(**A1**) A user defined offset prior $p_{\text{off}}(\mathbf{x}) = (\text{offset}_x, \text{offset}_y)^T$ for a circular region of the source image I_1 to the target image I_0, specified as prolonged mouse clicks into I_1 and I_0, shown in Fig. 1. In case of a brightness constancy violation (**E2**), a sufficiently large area around the violation must be chosen. Because no solution can be found for the violated pixels, the surrounding region must enforce a common motion direction.

Problems (**E3**) and (**E4**) are for the most part segmentation problems, as flow field discontinuities often relate to object boundaries. A layered representation requiring at least 3 input images has been addressed by Sun et al. [6]. We address this by:

(**A2**) A user defined local data weighting $p_{\text{data}}(\mathbf{x})$ which increases or decreases regularization (enforcement of smoothness) on a circular area, specified as prolonged mouse clicks into I_1. Decreased regularization will allow discontinuities in the flow field, addressing (**E3**). Increased regularization will hold the region together, providing a remedy for (**E4**).

All operations are applied while the algorithm is paused between iterations. With the user interface always showing the current estimation, the user pauses upon visual identification of a mismatch, applies the guidance, and resumes into repeating the current iteration, which refines the input.

In line with our objective to enhance rather than supplant existing optical flow algorithms, additional global parameter tuning can also be performed as in the automated case.

3. ALGORITHM ENHANCEMENT

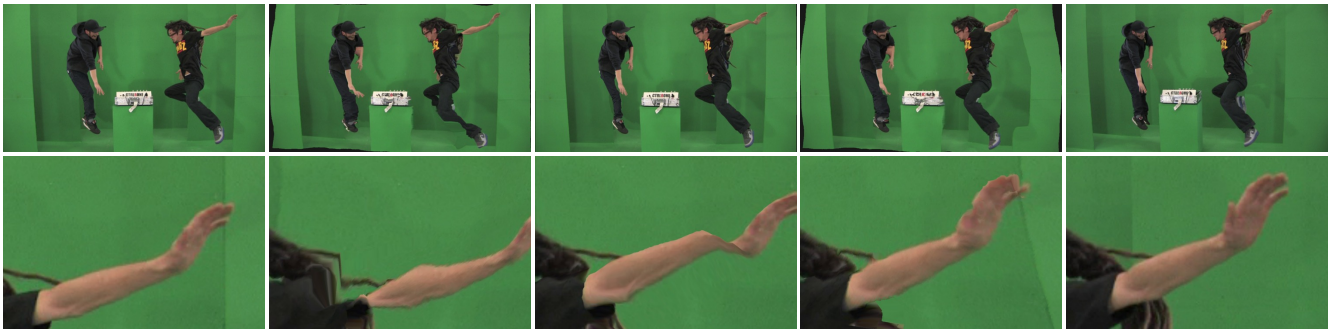
We take the TV-L1 optical flow algorithm by Zach et al. [9] as basis for our work. Following their notation, we assume two images I_0, I_1 with coordinates $\mathbf{x} = (x, y)^T$, and strive to attain the backward flow $\mathbf{u} = (u_x, u_y)$ such that $I_1(\mathbf{x} + \mathbf{u}(\mathbf{x})) = I_0(\mathbf{x})$, where $I_{\{0,1\}}(\mathbf{x})$ is a brightness value and $\mathbf{u}(\mathbf{x})$ a two-dimensional displacement vector. The employed coarse-to-fine pyramid has levels $L \in [0..n]$, with 0 the finest and n the coarsest image resolution. To integrate the user priors, we first analyze relevant parts of the algorithm and then describe (**A1**) and (**A2**).

3.1 Optical flow assessment

The TV-L1 approach is an energy minimization with a coupling term that allows alternating optimizations of the data and smoothness terms, and is thereby well suited for visual analysis. The overall energy to be minimized (eq. 12 in [9]) is defined as:

$$E = \int_\Omega |\nabla \mathbf{u}| + \frac{1}{2\theta}(\mathbf{u} - \mathbf{v})^2 + \lambda |\rho(\mathbf{v})| d\mathbf{x} \qquad (1)$$

Both $\mathbf{u}$ and $\mathbf{v}$ represent the correspondences to be estimated. The regularizer $|\nabla \mathbf{u}|$ enforces smoothness of the flow field, the residual $|\rho(\mathbf{v})|$ enforces adherence to the brightness constancy (data term) and λ is a weight relating data and smoothness term. The coupling term $\frac{1}{2\theta}(\mathbf{u} - \mathbf{v})^2$ penalizes deviations of $\mathbf{u}$ and $\mathbf{v}$, allowing the algorithm to perform



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

Figure 3: Jump scene from a movie production. (a) source image I_1 (b) unguided TV-L1 (c) large displacement optical flow [3] (d) user guided TV-L1 (e) target image I_0. Both automated algorithms have partial failure cases, which are remedied with mid-level user interactions and subsequent algorithmic refinement.

alternate updates to $\mathbf{u}$ and $\mathbf{v}$ (eq. 13 and 15 in [9]). After convergence, $\mathbf{u}$ is equal or very close to $\mathbf{v}$.

Considering the data term in more detail, the residual ρ is defined as the difference between the warped source image I_1 and the target image I_0. In order to make the function locally convex, a first order Taylor expansion is applied:

$$\begin{aligned}\rho(\mathbf{u}) &= I_1(\mathbf{x}+\mathbf{u}) - I_0(\mathbf{x}) \\ &\approx I_1(\mathbf{x}+\mathbf{u_0}) + \langle \nabla I_1(\mathbf{x}), \mathbf{u}-\mathbf{u_0}\rangle - I_0(\mathbf{x}) \quad (2)\end{aligned}$$

For this, the flow $\mathbf{u}$ is subdivided into a fixed part $\mathbf{u_0}$ and a differentiable part $\mathbf{u}-\mathbf{u_0}$ which is optimized pointwise along ∇I_1. Since Taylor expansion is only valid for small distances, a coarse-to-fine warping scheme is employed where $\mathbf{u_0}$ is the upsampled flow from a coarser level.

The smoothness term $|\nabla \mathbf{u}|$ is already a convex function, so no further modification is required.

With these definitions in mind, we formulate operations (**A1**) and (**A2**) in more detail.

3.2 User defined correspondence

We address a user defined offset in the smoothness update step (eq. 13 in [9]). Given $p_{\text{off}}(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^2$ at pyramid level L, we integrate the prior at the level initialization stage by replacing $\mathbf{u_0}(\mathbf{x})$ locally with $p_{\text{off}}(\mathbf{x})$. We omit the replacement when $\mathbf{u_0}(\mathbf{x})$ and $p_{\text{off}}(\mathbf{x})$ are already suffiently close (within 1 pixel distance). As a consequence, on a higher-resolution level the closeness requirement is tighter.

The user-defined motion is guaranteed to be correct even in ambiguous cases (**E1**) but not subpixel precise. We therefore propagate $p_{\text{off}}(\mathbf{x})$ to levels L to $L+m$, where m defaults to 10, and resume the estimation at level $L+m$. Particularly on levels smaller than L, the optimization of $\mathbf{u}-\mathbf{u_0}$ determines the final subpixel precise placement.

The introduced offset is by nature a sharp discontinuity in the flow field, that the smoothness optimization step will try to erase. We therefore locally replace the L1 norm $|\nabla \mathbf{u}|$ with the more robust Huber-L1 norm from Werlberger et al. [7] $|D^{\frac{1}{2}}\nabla \mathbf{u}|_\epsilon$ which penalizes quadratically for motions smaller than ϵ and linearly for larger motions. $D^{\frac{1}{2}}$ is a 2x2 matrix that linearly weights $\nabla \mathbf{u}_x$ and $\nabla \mathbf{u}_y$ with respect to the image gradient ∇I_1, and influences the update step for the smoothness term (eq. 15 in [7], eq. 10 in [9]):

$$\mathbf{p}_d^{n+1} = \frac{\mathbf{p}_d^n + \tau(D^{\frac{1}{2}}\nabla \mathbf{u}_d^{n+1} - \epsilon \mathbf{p}_d^n)}{\max(1, |\mathbf{p}_d^n + \tau(D^{\frac{1}{2}}\nabla \mathbf{u}_d^{n+1} - \epsilon \mathbf{p}_d^n)|)} \quad (3)$$

As in [7], we set $D^{\frac{1}{2}} = \exp(-\alpha|\nabla I_1|^\beta)\vec{n}\vec{n}^T + \vec{n}^\perp \vec{n}^{\perp T}$ with $\vec{n} = \frac{\nabla I_1}{|\nabla I_1|}$, and $\vec{n}^\perp$ a unit vector perpendicular to $\vec{n}$, with defaults $\alpha = 3$ and $\beta = 0.5$. This has the effect that the large discontinuity that has to occur due to the defined offset is preferably around image gradients in I_1, which often coincides with boundaries in user selected objects. Even if this is not the case, the approach will still work, but boundary regions will not be as well defined.

Violations of the brightness constancy assumption (**E2**) cannot be resolved by ρ since it is not possible to guess the "correct" color of e.g. a specularity. In this case, the region must be chosen large enough so that the smoothness term will enforce compliance to surrounding displacements.

3.3 User defined regularization

We address a user defined data weight in the data update step (eq. 15 in [9]). Given $p_{\text{data}}(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ on pyramid level L, we locally set $\lambda := \lambda * \exp(\eta p_{\text{data}}(\mathbf{x}))$, $\eta = 0.5$, for all regions defined in p_{data}, thereby influencing the thresholding step:

$$\mathbf{v} = \mathbf{u} + = \begin{cases} +\lambda\theta\nabla I_1 & \text{if } \rho(\mathbf{u}) < -\lambda\theta|\nabla I_1|^2 \\ -\lambda\theta\nabla I_1 & \text{if } \rho(\mathbf{u}) > +\lambda\theta|\nabla I_1|^2 \\ -\rho(\mathbf{u})\frac{\nabla I_1}{|\nabla I_1|^2} & \text{otherwise} \end{cases} \quad (4)$$

The effect is that the thresholding step assumes a smaller or larger valid range $\pm\lambda\theta|\nabla I_1|^2$ along which to follow the image gradient according to the residual ρ.

4. RESULTS

Since visual authenticity is our main objective, we perform a visual assessment of the warped source image I_1 instead of using a metric such as average endpoint and angular error.

Fig. 3 shows a green-screen example from a movie production. Both TV-L1 and LDOF (large displacement optical flow [3]) do not match arm and leg of the right actor correctly. A series of six manual offset priors on level 10 allows our interactive algorithm to find an improved solution, which is then automatically refined on level 9 and upwards.

Fig. 4 shows two frames from the Middlebury [1] backyard sequence. To simulate faster movements like in real-world examples, we employ a frame skip of 3. Both TV-L1 and LDOF do not match the ball correctly. One manually set offset region on level 10 solves the issue. Further problems of automated TV-L1, e.g. the green skirt, the older girl's

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

Figure 4: Backyard scene with 3 frames difference, from the Middlebury [1] data set. (a) source image I_1 (b) unguided TV-L1 (c) large displacement optical flow [3] (d) user guided TV-L1 (e) target image I_0. Both automated algorithms are unable to match the ball properly. A single user operation fixes the issue.

left face side or the boy's leg, are resolved by smoothness priors or offsets.

In both examples, the interaction time has been less than 30 seconds, with manual offsets as main input type.

5. DISCUSSION

Unsurprisingly, the visual quality of interpolated images improves with additional user input. The main question from a productivity perspective is whether the time spent in flow interactivity makes up for the time saved in final-frame editing. In our experience, interaction times are short, usually a few seconds, since only approximate inputs are needed. This makes the approach valuable for any number of interpolated frames.

Editing on an intermediate pyramid level is recommended as user input is in most cases imprecise; the remaining levels then refine the details of the priors globally.

The total runtime depends on the optical flow. On a Nvidia GTX590, our implementation takes around 33 seconds for 720p footage, while reaching editable levels already after 5 seconds. User guidance adds a few to tens of seconds.

We found that the general effectiveness of user inputs depends drastically on the edited scene. Editing rapidly changing fine structures such as hair are best left in image space, whereas object displacement can be handled well in correspondence space.

6. CONCLUSION

We presented a novel approach for user-guided dense image correspondence estimation. Our tool visualizes flow formation, and integrates user-defined correspondences and local smoothness priors to affect the emerging flow field.

The approach improves on the good results of automated algorithms, and allows user assistance in failure cases or regions of failure. Even if input is imprecise, our subsequent estimation compensates for it, reducing interaction times. Global parameter tuning efforts are also reduced because the results become immediately visible after adjustment.

Our implementation is based on a comparatively pure optical flow formulation. By integrating the interactive approach, similar results can be expected for more complex and sophisticated state-of-the art algorithms. An extension to more than two images (either a video stream or multi-view setting) will also prove to be beneficial by allowing edit operations that span several frames.

7. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement no. 256941, Reality CG, and from the German Science Foundation DFG MA 2555/1-3. Additional thanks go to Zach et al. [9] and Brox et al. [3] for openly providing the TV-L1 source code and LDOF binaries, respectively.

8. REFERENCES

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[2] T. Beier and S. Neely. Feature-based image metamorphosis. In *Proc. 19th Computer Graphics and Interactive Techniques*, SIGGRAPH, pages 35–42, 1992.

[3] T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48, 2009.

[4] F. Klose, K. Ruhl, C. Lipski, and M. Magnor. Flowlab - an interactive tool for editing dense image correspondences. In *Proc. European Conference on Visual Media Production*, pages 1–8, 2011.

[5] F. Steinbruecker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *ICCV*, pages 1609–1614, 2009.

[6] D. Sun, E. Sudderth, and M. Black. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. *Advances in Neural Information Processing Systems*, 2010.

[7] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, volume 34, pages 1–11, 2009.

[8] L. Wilkes. The role of ocula in stereo post production. *The Foundry, Whitepaper*, 2009.

[9] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Pattern recognition: 29th DAGM symposium*, volume 29, pages 214–223, 2007.