Combinatorial Regularization of Descriptor Matching for Optical Flow Estimation

Benjamin Drayer
drayer@cs.uni-freiburg.de
Thomas Brox
brox@cs.uni-freiburg.de

Department of Computer Science,
Centre of Biological Signalling Studies (BIOSS),
University of Freiburg, Germany

Abstract

One fundamental step in many state of the art optical flow methods is the initial estimation of reliable correspondences. It is well-established to extract and match features such as HOG to handle large displacements. We propose a combinatorial refinement of the initial matching. Optimization is done in the space of affine motion, where we regularize between neighboring points and similar regions. The evaluation on the MPI-Sintel dataset shows that the proposed method removes outliers from the initial matching and increases the number of reliable matches. The proposed refinement improves all optical flow algorithms that build upon pre-computed correspondences.

1 Introduction

Dealing with large displacements has been the main research focus in the field of optical flow estimation in recent years [4, 7, 17, 20, 21]. To estimate the large motion of small, detailed structures, a popular strategy is to pre-compute correspondences based on discriminative descriptors, such as HOG.

A reliable initial set of correspondences is an important requirement for accurate large displacement optical flow. While [4] relies on simple nearest neighbor matching to establish such correspondences, subsequent works have proposed more involved strategies to avoid wrong correspondences. The difference in performance between LDOF[4] and DeepFlow [20] is due to a more involved hierarchical matching strategy.

The contribution of the present paper is a refinement algorithm that further improves a given set of correspondences. We suggest extracting translational and affine motion hypotheses from the initial matching. This has two positive effects: First it gives us a higher order regularization and second, without increasing the computational complexity much, we increase the coverage of the apparent motion. We use the initial correspondences as an additional constraint and extend the local 4-connected neighborhood by additional, far reaching edges between homogeneous regions with similar color. An overview of the method is shown in Figure 1.

The proposed refinement is a self-contained procedure and can be employed in any matching based optical flow algorithm. We show the improvement of our matches on the challenging Sintel dataset [5], where we evaluate the proposed refinement in the framework of LDOF [4], DeepFlow [20] and EpicFlow [17].

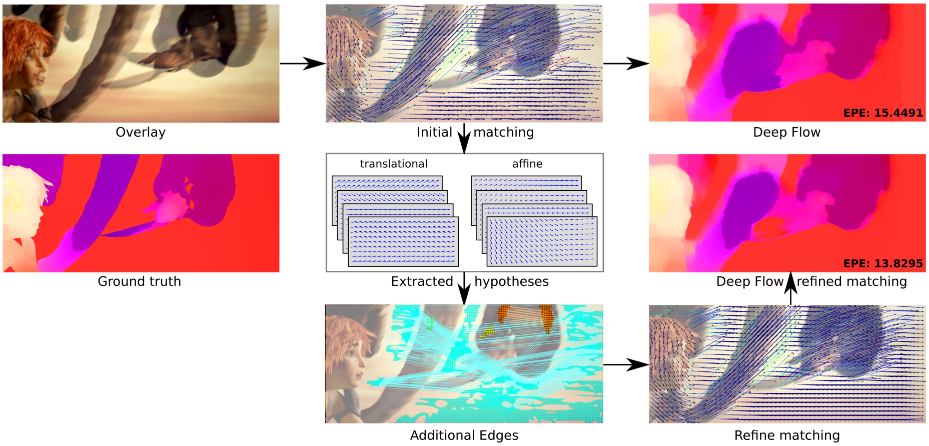


Figure 1: Outline of our method. The first row shows the overlaid input frames, the matching step (red arrows=high error, blue=low error) and the estimated optical flow using Deep Flow [20]. Starting from the initial matching, we extract translational and affine hypotheses. The 4-connected grid in the optimization/refinement step is enhanced by edges connecting similar regions. Our refinement improves at several position (note the fewer erroneous flow vectors and the denser field).

2 Related Work

Integrating an initial matching as a constraint into variational optical flow was introduced by Brox and Malik [4]. For the matching itself, they evaluated several methods: features from over-segmented regions, HOG features and geometric blur. It turned out that matching HOG features on a regular grid gives the best results. Here, a forward-backward check serves as confidence measure.

A hierarchical version of feature matching has been proposed by Kim *et al.* [9]. The authors use a coarse-to-fine pyramid, with a vertical regularization (spatial in the feature level) and a horizontal regularization, connecting the different levels. Revaud *et al.* [17] and Weinzaepfel *et al.* [20] introduced the so called DeepMatching. This approach works the other way around, the HOG features are matched from a fine to a coarse level, whereby at each level a convolution, followed by a max pooling and sub-sampling is done.

The PatchMatch framework of Barnes *et al.* [2] uses image patches to establish correspondences between images. The search is done in a nearest neighbor manner. Besse [3] *et al.* extended this idea with belief propagation to obtain a more regular matching.

Xu *et al.* [21], first generate a set of displacement candidates by using the SIFT matching of Lowe [13]. Spatial regularity is obtained with combinatorial optimization, the so-called fusion moves from Lempitsky *et al.* [11].

The SIFTFlow approach of Liu *et al.* [12] also uses combinatorial optimization for alignment methods in feature space. Like in Drayer and Brox [6] the method does not primarily aim for classical optical flow estimation but for dense correspondences between different object instances or scenes.

Recently, Sevilla-Lara *et al.* [18] proposed to split the gray values in different channels before smoothing and down-sampling the image. This method does not have any combinatorial components and estimates the motion of fine structures in a purely variational setting.

3 Combinatorial Refinement

The matching of LDOF [4] performs a forward-backward check to remove unreliable matches, but there is no kind of regularization. In the case of DeepMatching [20], due to the aggregation of features in the higher levels there is a sort of implicit regularization, which can handle arbitrary complex motions, but the regularization is only local and ambiguities cannot be resolved.

We propose the combinatorial refinement of a given feature matching. Our method improves the matching in several ways: (1) The refinement is computed densely (on a grid), which increases the number of matches. (2) Due to the explicit regularization, ambiguities in homogeneous regions or areas with repetitive texture are resolved. (3) Initially wrongly estimated outliers can be corrected.

We cast the refinement problem as a combinatorial energy minimization problem. The energy consists of three terms:

$$E(\mathcal{L}) = \underbrace{E_A(\mathcal{L}) + E_M(\mathcal{L})}_{\text{Data terms}} + \underbrace{E_S(\mathcal{L})}_{\text{Smoothness}} \tag{1}$$

The task is to find a labeling $\mathcal{L}$ that yields the minimal energy. The set of possible labels or hypotheses is derived from the initial matching. For each hypothesis we assign a matching cost E_M on how similar the displacement is to the initial guess (if there is one) and an appearance term E_A that measures the similarity of the matched features. We regularize on a 4-connected neighborhood extended by edges between areas with similar color; see Section 3.4 for more details. Since our method consists of several components, we give an analysis of each component's contribution. All tuning parameters are automatically optimized as described in Section 3.5, if not mentioned otherwise.

Algorithm 1: Estimate hypotheses from correspondences

```
Input : Correspondences $\mathcal{C}$, $h$
Output: Hypotheses $H$
  $\mathscr{L} \leftarrow 1$
  for $i = 1 : h$ do
      $H_i, \mathscr{L}_i \leftarrow RANSAC(\mathscr{L}, \mathcal{C})$
      $\mathscr{L} \leftarrow \mathscr{L} \setminus \mathscr{L}_i$
  $update(\mathcal{C})$
  while not converged do
      for $i = 1 : h$ do
          $H_i \leftarrow RANSAC(\mathscr{L}, \mathcal{C})$
      $\mathscr{L} \leftarrow update(H, \mathcal{C})$
```

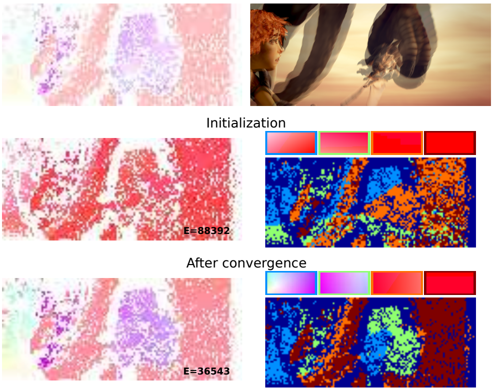


Figure 2: **Left:** Pseudocode for hypotheses generation. **Right:** Computing $h = 4$ hypotheses from the initial matching (top row). The initial estimation is biased by the dominant motion of the matching (middle row). The alternating optimization decreases the error and reconstructs the given correspondences better. The right column shows the best hypothesis for each point.

3.1 Hypotheses

Rather than instance based translational hypotheses, we propose using hypotheses based on affine motion models. This is beneficial for two reasons. First, affine object motion will not induce cost in the regularizer. Second, it increases the space of possible displacements in a very effective way and allows us to generate hypotheses in areas without initial correspondences.

A good set of hypotheses should well cover most of the initial correspondences $\mathcal{C}$ while being small. A set of translational hypotheses is implicitly given by the initial matching. We extend the translation hypotheses with a set of affine motion hypotheses $\{H_i\}$, where

$$H_i = \begin{pmatrix} a_1 & a_2 & d_x \\ a_3 & a_4 & d_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

In contrast to [7, 8], we use an alternating optimization scheme to estimate the affine hypotheses and do not update them at any later step. We fix the number of hypotheses $h = 15$ and focus on minimizing the error to the given correspondences:

$$\max_{\mathscr{L},H} \sum_{\mathbf{x}} f(\mathscr{L},H,\mathcal{C},\mathbf{x})$$
$$f(\mathscr{L},H,\mathcal{C},\mathbf{x}) = \begin{cases} 1 & \text{if } \left|\left(H_{\mathscr{L}(\mathbf{x})}\mathbf{x} - \mathbf{x}\right) - \mathcal{C}(\mathbf{x})\right| < \varepsilon \\ 0 & \text{else} \end{cases}, \tag{3}$$

where $\varepsilon = 1$ is the threshold that separates good hypotheses from bad ones. We solve the problem by alternatingly optimizing the labeling $\mathscr{L}$ and the hypotheses H. To optimizing for $\mathcal{L}$, the correspondence is assigned to the hypothesis it fits best:

$$\mathscr{L}(\mathbf{x}) = \min_i |(H_i\mathbf{x} - \mathbf{x}) - \mathcal{C}(\mathbf{x})|. \tag{4}$$

Fixing $\mathscr{L}$, we use RANSAC to improve the hypotheses. We repeat these two steps until $\mathscr{L}$ and H converge. The initial solution is computed greedily. We set $\mathscr{L}$ to one (consider the whole matching) and estimate the affine matrix. All correspondences that are further away than ε are considered for the estimation of the next matrix. We proceed in this fashion until we have an initial guess for all hypotheses. The algorithm with an accompanying example is shown in Figure 2.

3.2 Data term

The data term consists of two terms. The appearance term E_A measures the similarity in descriptor space. Additionally, we penalize the deviation from the initial correspondences with the matching term E_M. The former one consists of the cosine distance between the features $\mathcal{F}_A$ and $\mathcal{F}_B$:

$$E_A(\mathcal{L}) = \sum_{\mathbf{x}} -\rho_\alpha \left(\frac{\langle \mathcal{F}_A(\mathbf{x}), \mathcal{F}_B(H_{\mathcal{L}(\mathbf{x})}\mathbf{x}) \rangle}{\|\mathcal{F}_A(\mathbf{x})\|_2 \cdot \|\mathcal{F}_B(H_{\mathcal{L}(\mathbf{x})}\mathbf{x})\|_2} \right) \cdot \sigma(\mathbf{x}) \tag{5}$$

Due to its invariance against multiplicative changes, lightning changes due to shadows are handled well. In case of HOG features, we gain extra invariance against additive changes.

The robust function $\rho_\alpha(x) = \max(x, \alpha)$ prevents hypotheses with poor similarity from being matched. This filtering step is especially important in occluded regions. The matching of features tends to work better in structurally rich areas than in homogeneous regions. Thus, we weight the appearance term by the amount of local structure $\sigma(\mathbf{x})$, which we compute as the second eigenvalue of the structure tensor.

The valuable information of the initial matching contributes the other part of the data term:

$$E_M(\mathcal{L}) = \sum_{\mathbf{x}} \min(\|H(\mathcal{L}(\mathbf{x}))\mathbf{x} - \mathcal{M}(\mathbf{x})\|_2, \theta) \cdot C(\mathbf{x}) \tag{6}$$

Here θ marks the trade off between trusting the initial correspondence and allowing for outliers. $C(\mathbf{x})$ is a confidence function from the initial matching that rates the quality of the match or just acts as an indicator function whether a correspondence is available at $\mathbf{x}$.

3.3 Regularization

There are multiple possibilities on how to regularize affine motion. A straightforward method is to directly compare the affine parameters [16], which comes with the drawback that changes in parameter space are not proportional to the respective flow vectors. Hornacek *et al.* [7] directly regularize in the space of flow vectors. While they only deal with spatially neighboring points, in our case the edges $\mathcal{E}$ may connect arbitrary points. This is why we follow [8] and use their so-called flow continuity term

$$E_S(\mathcal{L}) = \sum_{(\mathbf{x_1},\mathbf{x_2})\in\mathcal{E}} \omega(\mathbf{x_1},\mathbf{x_2}) \cdot d\left(H_{\mathcal{L}(\mathbf{x_1})}, H_{\mathcal{L}(\mathbf{x_2})}, \frac{\mathbf{x_1}+\mathbf{x_2}}{2}\right), \tag{7}$$

where the measure between two hypotheses is defined as:

$$d(H_1, H_2, \mathbf{x}) = \min(|H_1\mathbf{x} - H_2\mathbf{x}|_1, \beta). \tag{8}$$

One very important aspect in terms of optimization is that this specific regularizer is a metric and thus the occurring binary optimization problems are sub-modular problems.

Since motion edges are strongly correlated with image edges, it is a common procedure [1, 14, 19, 21] to weight the smoothness term with the gradient magnitude in the image. We adapt this idea to the transitions of features between neighboring points:

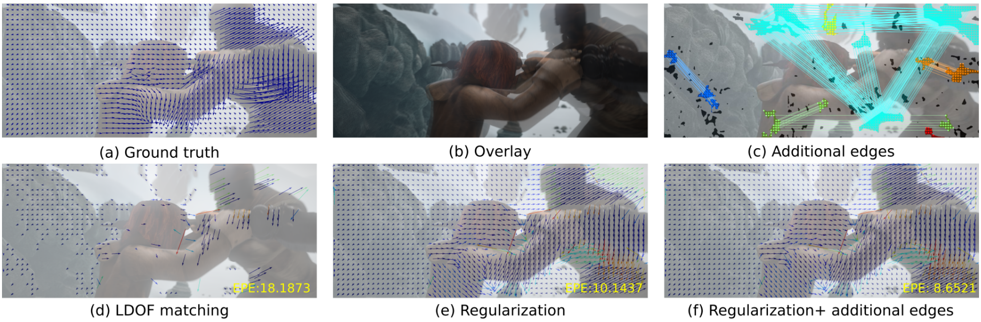
$$\omega(\mathbf{x_1},\mathbf{x_2}) = \lambda \cdot \exp - \frac{\|\mathcal{F}_A(\mathbf{x_1}) - \mathcal{F}_A(\mathbf{x_2})\|_2}{\nu}, \tag{9}$$

where λ is the global weighting parameter, controlling the smoothness and ν adjusts the features magnitude.

3.4 Additional Edges

It is well-established to use a 4- or 8-connected neighborhood for discrete regularization. This concept falls short in enclosed homogeneous areas without sufficient characteristics that can be put into correspondence. Therefore, we propose to extend the set of edges by connecting homogeneous regions which likely belong together.

Identifying the homogeneous regions is the first step in doing so. We classify a pixel as structureless, if the smaller eigenvalue of its structure tensor is below a threshold (in practice



(a) Ground truth (b) Overlay (c) Additional edges

(d) LDOF matching (e) Regularization (f) Regularization+ additional edges

Figure 3: Benefit of additional edges. When matching the first frame to the second (b), we compute regions with little texture, highlighted as black and colored areas in (c). Noise and small regions are discarded (marked as black). The remaining regions are connected if their appearance is similar. Connected regions have the same color. The bottom row shows the improvement from the initial LDOF [4] correspondences (d), over the refined matches (e) to and the refinement with the extended set of edges (f). The arrows are colored from blue to red (small to high error).

we used 0.153). Color histograms D_i in HSV-space describe the appearance of the individual connected components. In order to obtain an expressive descriptor we ignore small regions with less then 400 pixels. The distance between the normalized histograms is

$$d_D(D_i,D_j) = \left|D_i - D_j\right|. \tag{10}$$

We only connect two regions $\mathcal{A} = \{\mathbf{a}_i\}$ and $\mathcal{B} = \{\mathbf{b}_j\}$, if $d_D < 1.3$. Connecting all points of $\mathcal{A}$ with all points of $\mathcal{B}$ is computational expensive, when it comes to minimizing the energy. Connecting only the centers of the regions on the other hand has only little effect on the regularity of the solution. We strike a balance, by determining a matrix M that maps the maximal number of points from $\mathcal{A}$ to $\mathcal{B}$:

$$\begin{gathered}\max_{M} \sum_{\mathbf{a}_i \in \mathcal{A}} m(\mathbf{a}_i, \mathcal{B}, M) \\ m(\mathbf{a}, \mathcal{B}, M) = \begin{cases} 1 & \text{if } \exists \mathbf{b}_j \in \mathcal{B} : M\mathbf{a} - \mathbf{b}_j = 0 \\ 0 & \text{else} \end{cases}.\end{gathered} \tag{11}$$

We restrict the optimization of M to translations and variations in scale. In this way we preserve the relative ordering of the points, while solving the equation can be done quickly with a multi scale convolution.

The newly introduced edges might connect regions not belonging together, therefore they are weighted proportional to the regions' appearance similarity:

$$\overline{\omega(\mathbf{x_1},\mathbf{x_2})} = \omega(\mathbf{x_1},\mathbf{x_2}) \cdot \frac{1}{d_D(D_i,D_j)}. \tag{12}$$

The benefit and computation of these additional edges is illustrated in Figure 3.

	4-connected	affine hypotheses	additional edges	affine & edges
EPE	5.484	5.397	5.4127	**5.303**

Table 1: Analysis of the different steps in the refinement on the whole grid. Starting from a 4-connected neighborhood with translational hypotheses, the affine hypotheses and the additional edges improve the matching. Best results are obtained by combining both.

	Ldof	Ldof+R	Deep	Deep+R
EPE	3.4627	**3.4537**	3.5073	**3.1757**
Points	$2.14 \cdot 10^3$	$\mathbf{6.19 \cdot 10^3}$	$5.87 \cdot 10^3$	$\mathbf{6.56 \cdot 10^3}$

Table 2: Evaluation of the refinement of the LDOF (Left) and DeepMatching (Right) matches on the final pass of the Sintel training set. Regarding the very same points, the refinement of LDOF gives only a very slight improvement, but overall it triples the number of correspondences. The improvement on the initial point set with DeepMatching is almost 0.4 and in average, we establish additional 690 matches for each frame.

3.5 Optimization

For minimizing the sub-modular energy, we use the Fast_PD solver of Komodakis and Tziritas [10]. We optimized the parameters $(\alpha = 0.9, \beta = 65, \theta = 37.341, \lambda = 0.00001, \nu = 2.1)$ that are directly involved in the MRF. Starting from several initial seed guesses, we optimized using the downhill-simplex algorithm of Nelder and Mead [15]. The optimization is done on a subset of the final sintel training sequences and contains 50 pairs of images.

3.6 Matching Score

As a post processing step, we assign scores to the estimated correspondences. First, we do a forward-backward check and score the matches by the inverse of the distance. Second the color combined with the structure tensor comes in as weight (as in [20]). In practice, this scoring scheme works well with LDOF and DeepFlow.

4 Results

4.1 Matching

We evaluated the combinatorial refinement on the final pass of the Sintel training dataset. Both, the affine hypotheses and the additional edges improved the matching; see Table 1.

	Matching	Refinement	Optical Flow	Total
LDOF	22.367	13.148	26.011	61.526
DeepFlow	126.721	11.871	40.737	179.329
EpicFlow			4.251	142.843

Table 3: The runtime analysis (in seconds) of the different components shows that the additional cost of the proposed refinement is low in relation to the other parts.

Method	EPE	EPE noc	EPE occ	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
Ldof [4]	6.026	4.713	13.693	7.200	4.895	4.099	2.134	5.195	18.348
Ldof+R	5.616	4.333	13.234	6.528	4.325	3.812	2.283	4.970	16.634
Deep [20]	4.022	2.668	11.889	5.618	2.752	1.727	1.902	4.619	14.214
Deep+R	3.852	2.457	11.962	5.466	2.575	1.475	1.824	4.533	14.027
Epic [17]	3.566	2.489	9.838	5.238	2.570	1.534	1.828	4.179	13.728
Epic+R	3.497	2.366	9.996	5.052	2.460	1.481	1.826	4.180	13.568

Table 4: Comparison of the different optical flow methods with (+R) and without our regularization on the final pass of the Sintel training dataset. Throughout the methods, we get better results with the refined matches.

Method	EPE	EPE noc	EPE occ	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
Deep [20]	7.212	3.336	38.781	5.650	3.144	2.208	1.284	4.107	44.118
Deep+R	6.769	2.996	37.494	5.182	2.770	2.064	1.157	3.837	41.687

Table 5: The proposed regularization (+R) improves the DeepFlow, by almost 0.5 on the final pass of the Sintel test dataset.

By combining the two, we obtained an additional benefit, since regularization over a large spatial distance cannot be handled in the translational space.

Directly comparing the respective matching and its refinement is difficult due the different set of correspondences. Therefore, we performed two comparisons, one on the very same points and the other one on the joint set of correspondences (Table 2).

In the case of LDOFmatching, the refinement performs equally well on the same points. The small initial density makes it hard for the refinement to filter out wrong correspondences. However, we find many additional correspondences. Averaged over the dataset, the number of correspondences triples.

Using Deep Matching, we get a denser set of initial matches. Here, the proposed refinement improves the matching by almost 0.4 EPE. Regarding the number of matches, we found 690 additional correspondences per frame.

4.2 Optical Flow

For LDOF we observed the largest gain; see Table 4. Here the quality of the initial matches did not change much (see Table 2), so we explain this by the large increase of correspondences.

The increased number of matches and the lower EPE of the refined DeepMatches (see Table 2) resulted in a better optical flow estimation for both DeepFlow and EpicFlow (see Table 4). Using the very same matches, we observed a stronger gain for DeepFlow. We put this down to the fact that EpicFlow has already some kind of regularization. Despite this redundancy, the results indicate that our proposed regularization is complementary (compare Figure 5). For all of the methods, we report an improvement on the final pass of the Sintel training set.

Finally, we evaluated our proposed refinement on the test set. We chose the currently most popular algorithm, DeepFlow and achieved an improvement of almost 0.5 EPE; see Table 5.

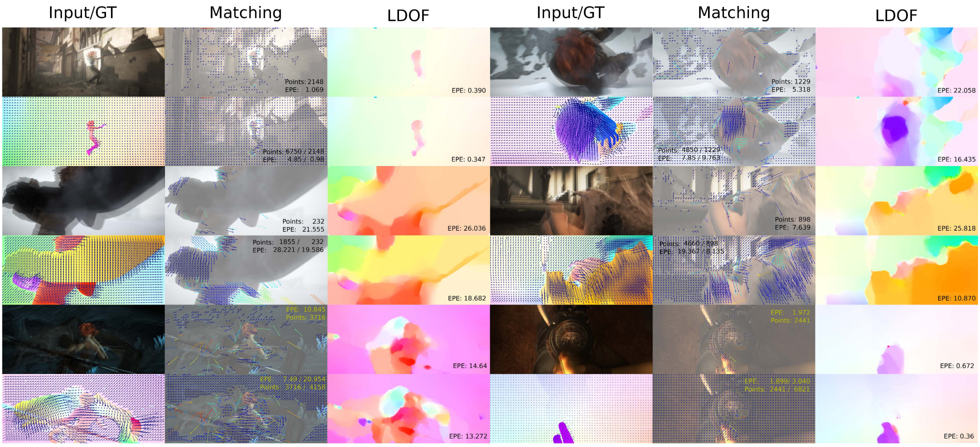


Figure 4: The upper line of each example shows the overlay of the input frames, the initial matching and the results of LDOF. The bottom line contains, groundtruth, our refined matching and the respective optical flow.

The additional time for the refinement is relative low in relation to the other parts of the respective algorithm. In Table 3, a detailed analysis for LDOF, DeepFlow and EpicFlow is given. The runtimes are measured on an Intel®Xeon®E5630 (2.53 GHz) with 12Gb RAM.

5 Conclusion

We have presented a novel regularization strategy to improve the initial feature matching in several state of the art methods (LDOF, DeepFlow and EpicFlow). Analysis on the MPI-Sintel benchmark shows that we improve both, the quality (EPE) and the quantity of the correspondences. As a consequence, the respective optical flow algorithms become more accurate.

Acknowledgements

This study was supported by the Excellence Initiative of the German Federal and State Governments (EXC 294) and by the ERC Starting Grant VIDEOLEARN.

References

[1] L. Alvarez, J. Esclarín, M. Lefébure, and J. Sánchez. A PDE model for computing the Optical Flow. In *Proc. XVI Congreso de Ecuaciones Diferenciales y Aplicaciones*, pages 1349–1356, September 1999.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28 (3):24:1–24:11, 2009.

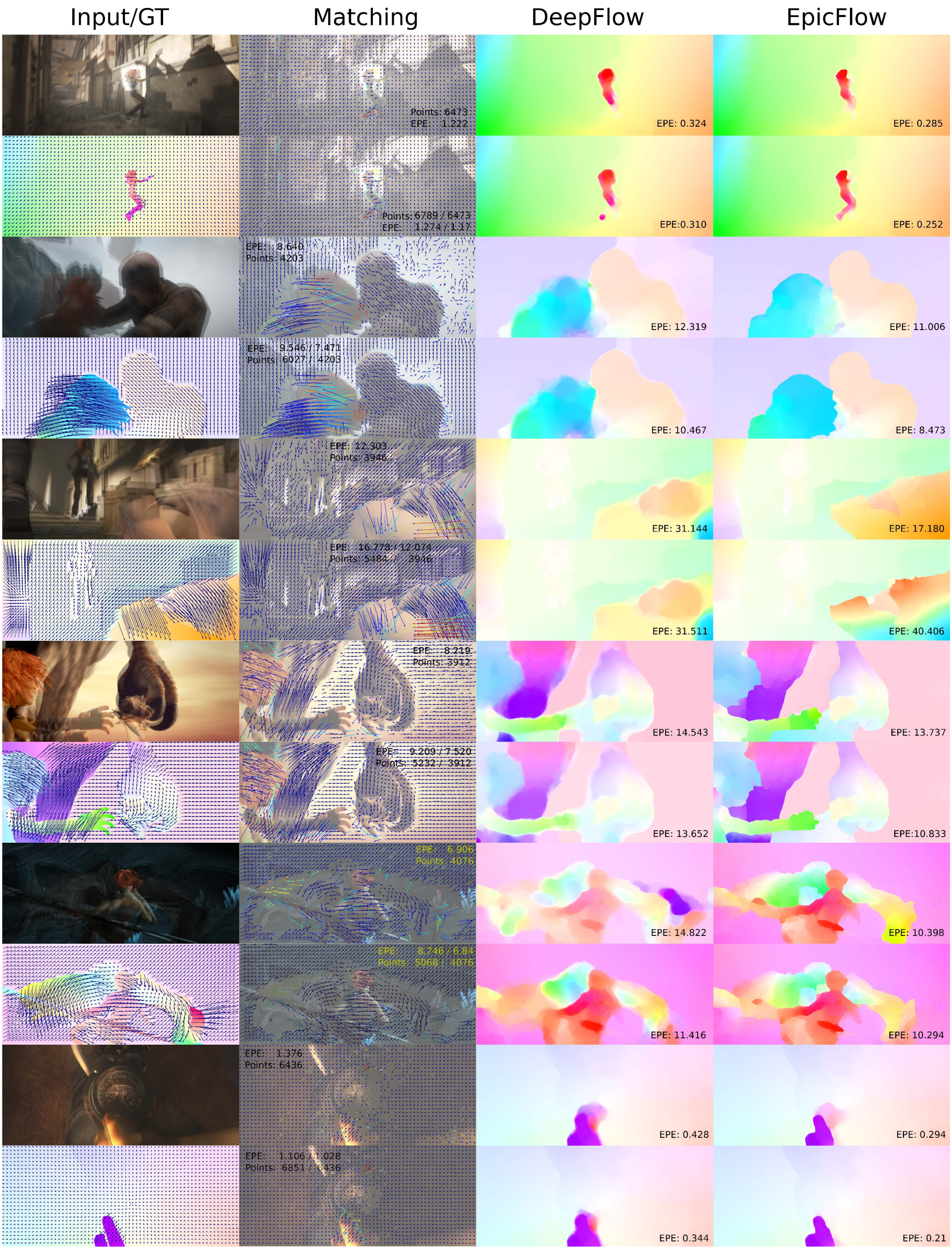


Figure 5: Qualitative results for the regularization in DeepFlow and EpicFlow. Each example shows an overlay of the frames (top left) and the groundtruth (bottom left). The top row shows DeepMatching, DeepFlow and EpicFlow, our refinement is compared in the bottom row.

[3] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. In *British Machine Vision Conference (BMVC)*, 2012.

[4] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625. Springer-Verlag, 2012.

[6] B. Drayer and T. Brox. Distances based on non-rigid alignment for comparison of different object instances. In *Pattern Recognition (GCPR 2013)*, volume 8142 of *LNCS*, pages 215–224. Springer, 2013.

[7] M. Hornacek, F. Besse, J. Kautz, A.W. Fitzgibbon, and C. Rother. Highly overparameterized optical flow using patchmatch belief propagation. In *European Conf. on Computer Vision (ECCV)*, pages 220–234, 2014.

[8] Y. Jiaolong and L. Hongdong. Dense, accurate optical flow estimation with piecewise parametric model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[9] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2307–2314. IEEE, 2013.

[10] N. Komodakis and G. Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1436–1453, August 2007.

[11] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, June 2008.

[12] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman. Sift flow: Dense correspondence across different scenes. In *European Conf. on Computer Vision (ECCV)*, pages 28–42. Springer-Verlag, 2008.

[13] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[14] H. H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986.

[15] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[16] T. Nir, A.M. Bruckstein, and R. Kimmel. Over-parameterized variational optical flow. *International Journal of Computer Vision*, 1976.

[17] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] L. Sevilla-Lara, D. Sun, E.G. Learned-Miller, and M.J. Black. Optical flow estimation with channel constancy. In *European Conf. on Computer Vision (ECCV)*, volume 8689 of *Lecture Notes in Computer Science*, pages 423–438. Springer International Publishing, 2014.

[19] A. Wedel, D. Cremers, T. Pock, and H. Bischof. Structure- and motion-adaptive regularization for high accuracy optic flow. In *IEEE International Conference on Computer Vision (ICCV)*, 2009.

[20] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1385–1392. IEEE, 2013.

[21] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1744–1757, September 2012.