

DeepFlow: Large displacement optical flow with deep matching

Philippe Weinzaepfel, Jérôme Revaud, Zaid Harchaoui, Cordelia Schmid

▶ **To cite this version:**

Philippe Weinzaepfel, Jérôme Revaud, Zaid Harchaoui, Cordelia Schmid. DeepFlow: Large displacement optical flow with deep matching. ICCV - IEEE International Conference on Computer Vision, Dec 2013, Sydney, Australia. pp.1385-1392, 10.1109/ICCV.2013.175 . hal-00873592

HAL Id: hal-00873592

https://inria.hal.science/hal-00873592

Submitted on 16 Oct 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DeepFlow: Large displacement optical flow with deep matching

Philippe Weinzaepfel Jerome Revaud Zaid Harchaoui Cordelia Schmid
INRIA and LJK, Grenoble, France
firstname.lastname@inria.fr

Abstract

Optical flow computation is a key component in many computer vision systems designed for tasks such as action detection or activity recognition. However, despite several major advances over the last decade, handling large displacement in optical flow remains an open problem.

Inspired by the large displacement optical flow of Brox & Malik [6], our approach, termed DeepFlow, blends a matching algorithm with a variational approach for optical flow. We propose a descriptor matching algorithm, tailored to the optical flow problem, that allows to boost performance on fast motions. The matching algorithm builds upon a multi-stage architecture with 6 layers, interleaving convolutions and max-pooling, a construction akin to deep convolutional nets. Using dense sampling, it allows to efficiently retrieve quasi-dense correspondences, and enjoys a built-in smoothing effect on descriptors matches, a valuable asset for integration into an energy minimization framework for optical flow estimation.

DeepFlow efficiently handles large displacements occurring in realistic videos, and shows competitive performance on optical flow benchmarks. Furthermore, it sets a new state-of-the-art on the MPI-Sintel dataset [8].

1. Introduction

An enormous amount of digital video content (home movies, films, surveillance tapes, TV, video games) is becoming available, along with new challenges. These include action and activity recognition from realistic videos [14, 30] and video surveillance [11]. In particular, optical flow computation is important in early stages of the video description pipeline [19, 30]. It is essential that the optical flow algorithm overcomes the many challenges that arise in realistic videos, namely: robustness to outliers (motion discontinuities, occlusions), robustness to illumination changes (with gradient constancy), ability to deal with large displacements. Efficient approaches were proposed to satisfy the first two requirements [4, 21, 31]. However, the last requirement, that is the ability to handle large displacements in optical flow, has received little attention so far [6, 33, 32, 23].

In their pioneering work [6], Brox and Malik show that a careful addition of a descriptor matching term in the variational approach allows to better handle large displacements. The main idea is to give "hints" to guide the classical variational optical flow estimation by using correspondences from sparse descriptor matching. This approach combines the advantage of region-based descriptor matching, *i.e.* the ability to estimate arbitrarily large displacement, with the strengths of variational optical flow methods.

Current descriptor matching approaches rely typically on square, rigid descriptors (*e.g.* HOGs [26]), which implicitly refers to the rigid motion hypothesis. While infinitesimal displacements fit this model well, it no longer holds for fast motions. We show that improving the descriptor matching part towards a more dense and deformable matching can lead to a significant performance gain for fast motions.

We make here a step towards bridging the gap between descriptor matching methods and current large displacement optical flow algorithms. We propose a new descriptor matching algorithm, called *deep matching*, that enjoys a built-in smoothing effect on the set of output correspondences. The outline of our approach is shown on Figure 1. We make three contributions:

- **Dense correspondences matching**: we introduce a descriptor matching algorithm, using dense sampling, that allows to retrieve dense correspondences from *single* feature correspondences with deformable patches.
- **Self-smoothed matching**: the matching algorithm works with a restricted set of feasible non-rigid warpings, which gracefully produces almost smooth dense correspondences while allowing computationally efficient comparison of non-rigid descriptors.
- **Large displacement optical flow**: our variational optical flow, DeepFlow, shows robustness to large displacements, performing equally well to Brox and Malik's approach on Middlebury dataset [2], and significantly outperforming it on the MPI-Sintel dataset [8].

This paper is organized as follows. After a review of previous works (Section 2), we start by presenting the matching algorithm in Section 3. Then, Section 4 describes our

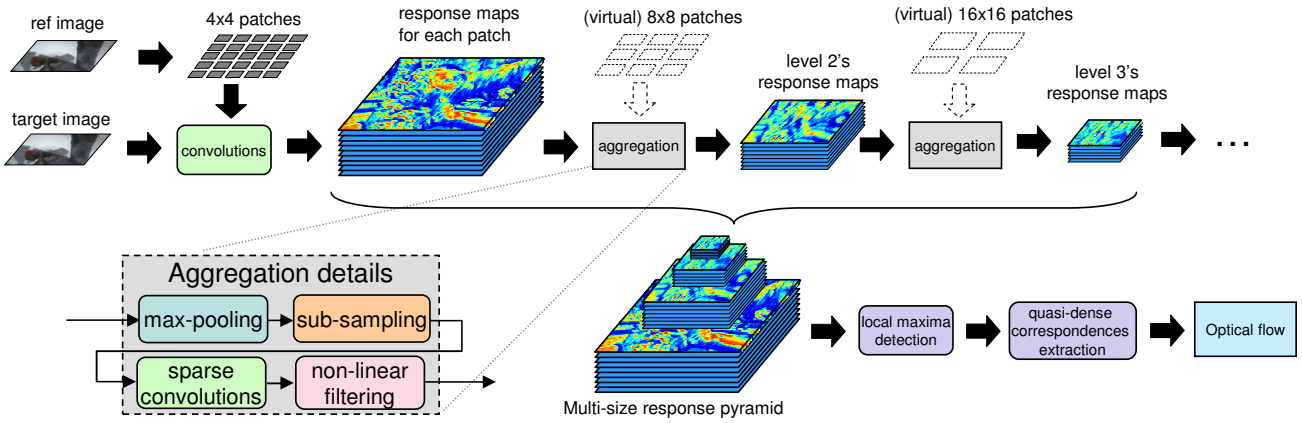


Figure 1. Outline of DeepFlow.

variational optical flow approach termed DeepFlow. Finally, we present experimental results in Section 5.

2. Related work

Large displacement in optical flow estimation. Variational methods are the state-of-the-art family of methods for optical flow estimation. Since the pioneering work of Horn and Schunck [1], research has focused on alleviating the drawbacks of this method. A series of improvements were proposed over the years [4, 31, 7, 21, 2, 25, 29]. Brox *et al*. [5] combine most of them into a variational approach. Energy minimization is performed by solving the Euler-Lagrange equations, then reducing the problem to solving a sequence of large and structured linear systems.

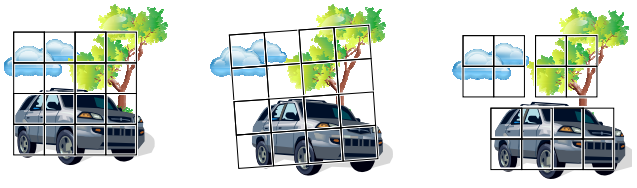
To handle large displacements, a descriptor matching component is incorporated in the variational approach in [6]. One major drawback of this method is that local descriptors are reliable only at salient locations and are locally rigid. Adding a matching component challenges the energy formulation as it could deteriorate performance at small displacement locations. Indeed, matching can give false matches, ambiguous matches, and has low precision (a pixel). In a different context, namely scene correspondence, descriptors or small patches were used in SIFT-flow [17] and PatchMatch [3] algorithms. Xu *et al*. [33] integrate matching of SIFT [26] and PatchMatch [3] to refine the flow initialization at each level. Excellent results were obtained, yet at the cost of expensive fusion steps. Leordeanu *et al*. [16] propose to extend sparse matching, with locally affine constraint, to dense matching before using a total variation algorithm to refine the flow estimation. We present here a computationally efficient, yet competitive approach for large displacement optical flow using a deep convolutional matching procedure.

Descriptor matching. Image matching consists of two steps: extraction of local descriptors and matching them. Initial image descriptors were extracted at sparse, scale-invariant or affine-invariant image locations [26, 20]. For the purpose of optical flow estimation, recent work showed that dense descriptor sampling improves performance [27, 6, 17]. In all cases, descriptors are extracted in rigid (generally square) local frames. Matching descriptors is then generally reduced to a nearest-neighbor problem [26, 3, 6]. Methods such as reciprocal nearest-neighbors allow to prune lots of false matches, but as a side effect also eliminate correct matches in weakly to moderately textured regions. We show here that (i) extraction of descriptors in non-rigid frames and (ii) dense matching in all image regions, yields a competitive approach, with a significant performance boost on MPI-Sintel [8] and KITTI [10] datasets.

Non-rigid matching. Our proposed matching algorithm, called *deep matching*, is strongly inspired by non-rigid 2D warping and deep convolutional networks [15, 28, 12]. It also bears similarity with non-rigid matching approaches developed in different contexts. In [9], Ecker and Ullman proposed a similar pipeline to ours (albeit more complex) to measure the similarity of small images. However, their method lacks a way of merging correspondences belonging to objects with contradictory motions (*e.g*., on different focal planes). In a different context, Wills *et al*. [32] estimated optical flow by robustly fitting smooth parametric models (homography and splines) to local descriptor matches. In contrast, our approach is non-parametric and model-free. More recently, Kim *et al*. [13] proposed a hierarchical matching to obtain dense correspondences, but their method works in a coarse-to-fine (top-down) fashion, whereas deep matching works bottom-up. In addition, their method requires inexact inference using loopy belief propagation.

3. Deep Matching

In this section, we present the matching algorithm, termed deep matching, and discuss its main features. The matching algorithm builds upon a multi-stage architecture



(a) (b) (b')

Figure 2. Illustration of moving quadrant similarity (a quadrant is a quarter of a SIFT patch, *i.e.* a group of 2×2 cells). (a) Reference image; (b) target image with optimal standard SIFT matching; (b') target image with optimal *moving quadrant* SIFT matching.

with about 6 layers (depending on the image size), interleaving convolutions and max-pooling, a construction akin to deep convolutional nets [15].

3.1. Insights on the approach

The SIFT descriptor [26] is a histogram of gradient orientations with 4×4 spatial cells, yielding a 128-dimensional real vector $H \in \mathbb{R}^{128}$. Now, let us split the SIFT patch into 4 so-called quadrants, as in Figure 2.(b'): we have $H = [H^1\, H^2\, H^3\, H^4]$ with $H^s \in \mathbb{R}^{32}$.

We want to match a source descriptor with a target descriptor. Rather than keeping the fixed 4×4 grid for both descriptors, we propose to optimize the positions of the 4 quadrants of the target descriptor H' so as to maximize $\text{sim}(H, Q(p)) = \sum_{s=1}^{4} \max_{p_s} H_s^T Q(p_s)$, where $Q(p) \in \mathbb{R}^{32}$ is the descriptor of a single quadrant extracted at position p. Assuming that each of the four quadrants can move independently (within some extent), the similarity can be estimated efficiently, yielding a coarse non-rigid matching. When applied recursively, this strategy allows for fine non-rigid matching with explicit pixel-wise correspondences.

3.2. Deep matching as 2D warping

For the sake of clarity, we describe the 1D warping case. The extension to the 2D case is straightforward, see [28, 12] for details on 2D warping. Consider two 1D sequences of descriptors, called *reference* $\mathbf{P} = \{\mathbf{P}_i\}_{i=0}^{L-1}$ and *target* $\mathbf{P}' = \{\mathbf{P}'_i\}_{i=0}^{L-1}$. The optimal warping between them is defined by the function $w^* : \{0 \dots L-1\} \to \{0 \dots L-1\}$ that maximizes the sum of similarities between their elements:

$$S(w^*) = \max_{w \in W} S(w) = \max_{w \in W} \sum_i \text{sim}\left(\mathbf{P}(i), \mathbf{P}'(w(i))\right) \tag{1}$$

where $w(i)$ returns the position of element i in $\mathbf{P}'$. In practice, we use the non-negative cosine similarity between pixel gradients as the similarity in Equation (1). The set of *feasible warpings* W is defined recursively so that (i) finding the optimal warping w^* is computationally efficient and (ii) warping is tolerant to moderate deformations.

Efficient computation of response maps. Let us denote a subsequence of $\mathbf{P}$ of size $N \leqslant L$ and centered in δ as $\mathbf{P}[\delta, N] = \{\mathbf{P}(i)\}_{i=\delta-\frac{N}{2}}^{\delta+\frac{N}{2}-1}$. A sub-warping from $\mathbf{P}[\delta, N]$ to $\mathbf{P}'[T, N]$ is denoted as $w_{N,\delta\to T}$. The key idea of deep matching is that we assume the displacements of the left-hand half and the right-hand half, that is resp. $\mathbf{P}[\delta - \frac{N}{4}, \frac{N}{2}]$ and $\mathbf{P}[\delta + \frac{N}{4}, \frac{N}{2}]$ of any subsequence $\mathbf{P}[\delta, N]$ to be both independent and limited, with respect to the displacement of $\mathbf{P}[\delta, N]$ and proportionally to N; see Figure 3. For the sake of clarity, define the short-hand notations $S_{\text{left}}(\frac{N}{2}, t) := S\left(w^*_{\frac{N}{2},\delta-\frac{N}{4}\to T-\frac{N}{4}+t}\right)$ and $S_{\text{right}}(\frac{N}{2}, t) := S\left(w^*_{\frac{N}{2},\delta+\frac{N}{4}\to T+\frac{N}{4}+t}\right)$ resp., and $S(N) = S(w^*_{N,\delta\to T})$. Then, Equation (1) rewrites as a recursion in N:

$$S(N) = \max_{t\in\left[-\frac{N}{8},\frac{N}{8}\right]} S_{\text{left}}\left(\frac{N}{2}, t\right) + \max_{t\in\left[-\frac{N}{8},\frac{N}{8}\right]} S_{\text{right}}\left(\frac{N}{2}, t\right). \tag{2}$$

Note that this formula implicitly defines the set of feasible warpings W, without enforcing monotonicity nor continuity, in contrast to [28, 12] – thus making the problem much easier. Indeed, it allows for an efficient computation of both $S(w^*)$ and w^* using dynamic programming. At $N = 4$, we stop the recursion and, assuming rigid subsequences $\mathbf{P}[\cdot, 4]$ and $\mathbf{P}'[\cdot, 4]$, we directly compute the score of Equation (1) using convolutions. We call the values in $\{S(w^*_{N,\delta\to T})\}_T$, for fixed δ, N and varying T, the *response map* of the reference subsequence on the target sequence. We observe that response maps obtained at higher N vary slowly with T, so we incorporate a sub-sampling step of factor 2 in the recursion. This compensates for the larger max-pooling area in Equation (2). See Algorithm 1 for a summary of the approach.

This procedure produces a pyramid of response maps, see Figure 1. Some are shown in Figure 4 for $N \in \{4, 8, 16\}$. Local maxima in the response maps correspond to good matches of corresponding local image patches. To obtain dense correspondences between any matched patches (*i.e.* at local maxima), it suffices to recover the path of response values that generated this maximum using Equation (2).

Max-pooling with rectification. In contrast to most algorithms for optical flow [5, 31], our algorithm works in a bottom-up fashion. The algorithm starts at a fine level, and moves up to coarser levels (larger patches), which are built as an aggregation of responses of smaller patches.

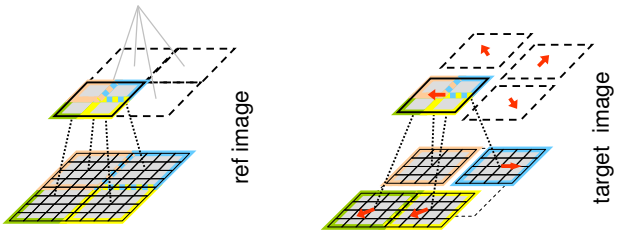


Figure 3. *Left:* patch hierarchy in a reference image; *right:* one possible displacement of corresponding patches in a target image.

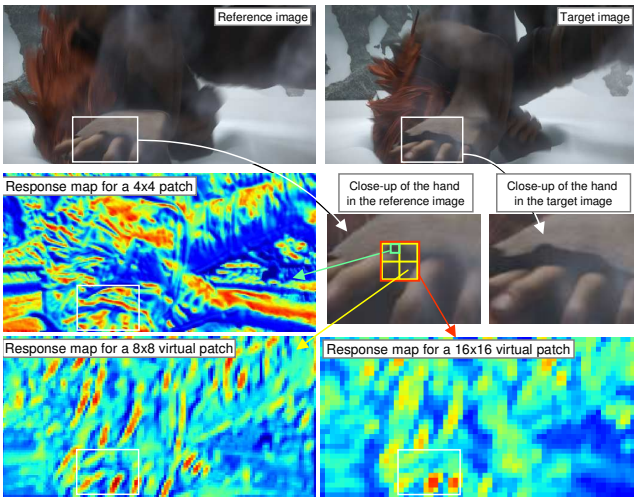


Figure 4. Response maps of reference patches on the target image at various levels of the pyramid. *Middle-left*: response map of a 4x4 patch. *Bottom-right*: the response map of a 16x16 patch is obtained from aggregating responses of children 8x8 patches (*bottom-left*), themselves obtained from 4x4 patches. The map of the 16x16 patch is clearly more discriminative than previous ones despite the change in appearance of the region.

Algorithm 1 Computing the response maps of every patch of the reference image to the target image (1D version).

Input: $\mathbf{P}$, $\mathbf{P}'$ are 1D sequences to match
Set $N \leftarrow 4$, $L \leftarrow |\mathbf{P}|$, $L' \leftarrow |\mathbf{P}'|$
For $\delta \in [2, L-2]$ with step 4 **do**
  For $T \in [0, L'-1]$ **do**
    Compute the initial response map (convolutions):
    $S\left(w^*_{4,\delta\to T}\right) = \sum_{i=-2}^{1} \mathrm{sim}(P_{\delta+i}, P'_{T+i})$
While $N < L$ **do**
  $N \leftarrow 2N$
  max-pool the response maps (max in Eq. 2)
  subsample the response maps by a factor 2
  For $\delta \in \left[\frac{N}{2}, L - \frac{N}{2}\right]$ with step N **do**
    compute the response map $\left\{S\left(w^*_{N,\delta\to T}\right)\right\}_T$ (Eq. 2)
    apply the non-linear filtering (Eq. 3)
Return the response maps $\left\{S\left(w^*_{N,\delta\to T}\right)\right\}_T$ for all δ, N

In order to better propagate responses after each level, we use a power transform after max-pooling and subsampling steps [18, 15]:

$$S'\left(w^*_{N,\delta\to T}\right) = S\left(w^*_{N,\delta\to T}\right)^{\lambda}. \quad (3)$$

Merging dense correspondences. A single maxima in the response maps is unlikely to explain by itself the full set of pixel-wise correspondences between the two images. The combination of several maxima, corresponding to different patch positions and sizes, better explains the global flow.

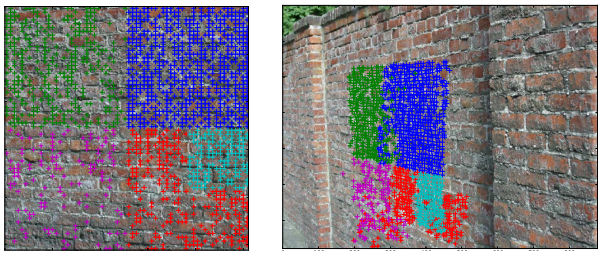


Figure 5. Matching result between two images with repetitive textures. Each color refers to correspondences obtained from a different maximum in the response maps.

Therefore, we merge the correspondences extracted from all local maxima. We weight each correspondence according to three factors: weight $= \mathrm{sim}(\mathbf{P}_4, \mathbf{P}'_4) \cdot l \cdot S(w^*)$ (similarity of the concerned atomic 4×4 patches, the level of the maximum in the pyramid and the value of the maximum). We use the pyramid level to favor correspondences originating from larger patches: they must be more reliable, since they are more discriminative (Figure 4). We found this simple heuristic produces good results in practice.

Finally, we retain the best correspondence (in terms of its weight) in every 4×4 non-overlapping block in both images. The final set of correspondences is the intersection of the retained correspondences in both images. An illustration of the final correspondences extracted for a pair of images is shown in Figure 5.

Proposition 1: *Finding the optimal matching score among all feasible non-rigid warpings in W for all square patches of sizes in $\{4, 8, 16, \ldots\}$ from the first image at all locations in the second image can be done with complexity $O(LL')$, where L and L' are the number of pixels of the two images.*

The most expensive part of our algorithm lies in the computation of the first level convolutions, see [22] for a proof.

3.3. Analysis of deep matching

Multi-size patches and repetitive textures. We consider patches of different sizes (all 2^n sizes of patches), in contrast to other optical flow methods relying on descriptor matches. This is a key feature of our approach when dealing with repetitive textures. As one moves up to coarser levels, the matching problem gets easier. Larger patches get more credit, and our method can correctly match repetitive patterns. Figures 5-6 illustrate this property.

Quasi-dense correspondences. Our method retrieves dense correspondences from every matched patch (*i.e.* local maximum), even in weakly textured areas; this is in contrast to single correspondences obtained when matching pairs of descriptors (*e.g.* SIFT). Figure 6 illustrates this characteristic. Quantitative assessment, by comparing the density of matches obtained from several matching schemes, is given in Section 5.

Non-rigid deformations. Our matching algorithm is able

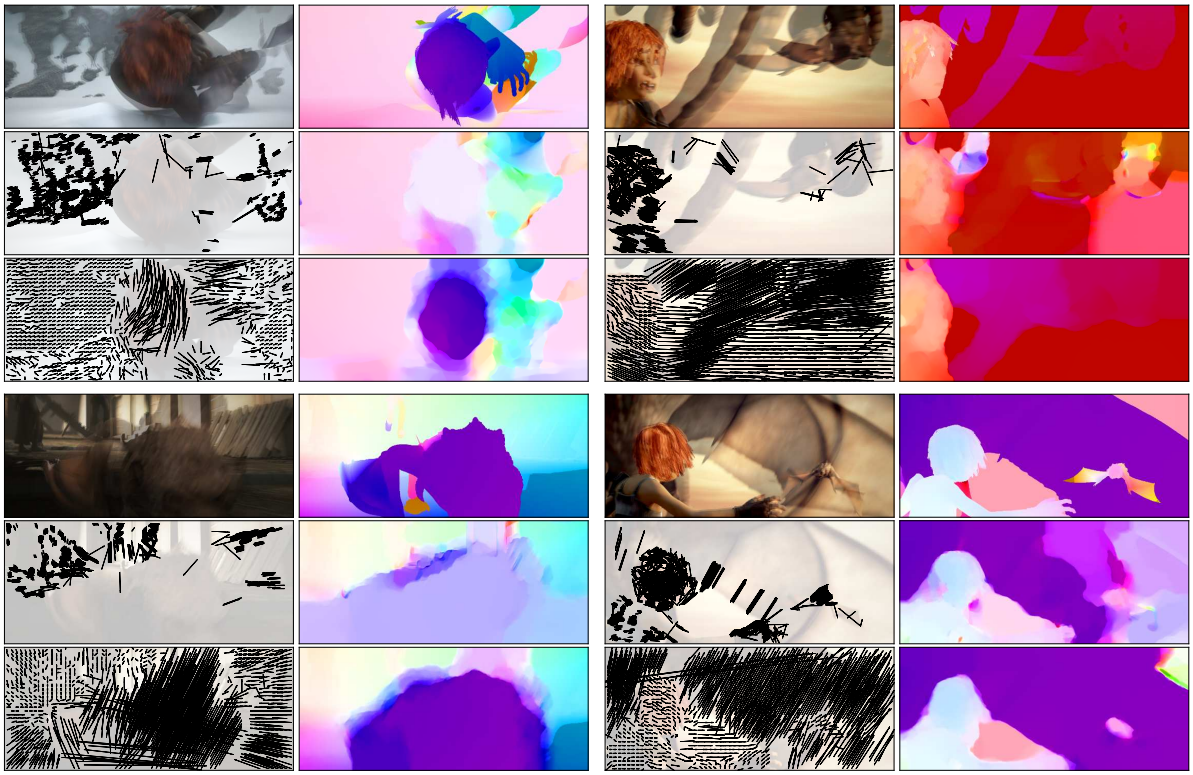


Figure 6. Sample results from the MPI-Sintel dataset [8]. (For each 3×2 block) *From top to bottom:* mean of the two frames and ground-truth flow; dense HOG matching and flow computed with Brox and Malik [6] executable; our matches and flow.

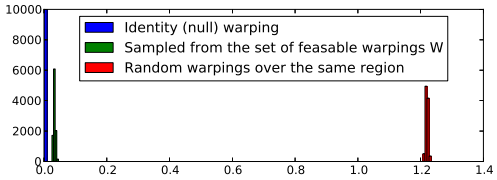


Figure 7. Histogram of the per-pixel averaged smoothness (computed from Eq. 6) of 10,000 warpings randomly sampled from the set of feasible warpings W_{128} and the set of random warpings over the same region. The identity warping has a smoothness of 0.

to deal with various sources of image deformations: object-induced or camera-induced. The set of feasible warpings, see Equation (2), theoretically allows to deal with a scaling factor in the range $[\frac{1}{2}, \frac{3}{2}]$ and rotations roughly in the range $[-30^o, 30^o]$ (although at the 4×4 patch level, the warping model is purely translational); see [22] for a proof. Furthermore, the matching process enjoys a built-in "post-smoothing". Indeed, feasible warpings cannot be too "far" from the identity warping. Figure 7 illustrates this, by comparing the smoothness of warpings sampled from W_{128} (*i.e.* from and to a 128×128 pixels region) with random warpings. Clearly, they are different by orders of magnitude. Figure 6 also shows that output match fields are locally smooth.

4. DeepFlow

We now present our variational optical flow, termed DeepFlow, that blends the deep matching algorithm into an energy minimization framework. We use a similar approach to Brox and Malik [6]. We make three additions: (i) we incorporate the deep matching algorithm as matching component, (ii) we add a normalization in the data term to downweight the impact of locations with high spatial image derivatives, (iii) we use a different weight at each level to downweight the matching term at finer scales. Let $I_1, I_2 : \Omega \rightarrow \mathbb{R}^c$ be two consecutive images defined on Ω with c channels. The goal is to estimate the flow $\boldsymbol{w} = (u, v)^\top : \Omega \rightarrow \mathbb{R}^2$. We assume that the images are already smoothed using a Gaussian filter of standard deviation σ. The energy we optimize is a weighted sum of a data term E_D, a smoothness term E_S and a matching term E_M:

$$E(\boldsymbol{w}) = \int_\Omega E_D + \alpha E_S + \beta E_M \boldsymbol{dx} \qquad (4)$$

For the three terms, we use a robust penalizer $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$ which has shown excellent results [25].

Data term. Our data term is a separate penalization of the color and gradient constancy assumptions with a normalization factor as in [34]. We start from the optical flow

constraint assuming the brightness constancy: $(\nabla_3^\top I)\boldsymbol{w} = 0$ with $\nabla_3 = (\partial x, \partial y, \partial t)^\top$ the spatio-temporal gradient. A basic way to build a data term is to penalize it, *i.e.* $E_D = \Psi(\boldsymbol{w}^\top \boldsymbol{J}_0 \boldsymbol{w})$ with $\boldsymbol{J}_0$ the tensor defined by $\boldsymbol{J}_0 = (\nabla_3 I)(\nabla_3^\top I)$. As highlighted by Zimmer *et al.* [34], such a data term overweights locations with high spatial image derivatives. We normalize it by the norm of the spatial derivatives plus a small factor to avoid division by zero and to reduce a bit the influence in tiny gradient locations [34]. Let $\bar{\boldsymbol{J}}_0$ be the normalized tensor $\bar{\boldsymbol{J}}_0 = \theta_0 \boldsymbol{J}_0$ with $\theta_0 = (\|\nabla_2 I\|^2 + \zeta^2)^{-1}$. We set $\zeta = 0.1$ in the following. To deal with color images, we consider the tensor defined for a channel i denoted by upper indices $\bar{\boldsymbol{J}}_0^i$ and we penalize the sum over channels: $\Psi(\sum_{i=1}^c \boldsymbol{w}^\top \bar{\boldsymbol{J}}_0^i \boldsymbol{w})$. We consider images in the RGB colorspace.

We separately penalize the gradient constancy assumption [7]. Let I_x and I_y be the derivatives of the images with respect to the x and y axis respectively. Let $\bar{\boldsymbol{J}}_{xy}^i$ be the tensor for the channel i including the normalization $\bar{\boldsymbol{J}}_{xy}^i = (\nabla_3 I_x^i)(\nabla_3^\top I_x^i)/(\|\nabla_2 I_x^i\|^2 + \zeta^2) + (\nabla_3 I_y^i)(\nabla_3^\top I_y^i)/(\|\nabla_2 I_y^i\|^2 + \zeta^2)$. The data term is the sum of two terms, balanced by two weights δ and γ:

$$E_D = \delta \Psi \left(\sum_{i=1}^{c} \boldsymbol{w}^\top \bar{\boldsymbol{J}}_0^i \boldsymbol{w} \right) + \gamma \Psi \left(\sum_{i=1}^{c} \boldsymbol{w}^\top \bar{\boldsymbol{J}}_{xy}^i \boldsymbol{w} \right) \quad (5)$$

Smoothness term. Our smoothness term is a robust penalization of the gradient flow norm:

$$E_S = \Psi \left(\|\nabla u\|^2 + \|\nabla v\|^2 \right). \quad (6)$$

Matching term. The matching term encourages the flow estimation to be similar to a precomputed vector field $\boldsymbol{w}'$. To this end, we penalize the difference between $\boldsymbol{w}$ and $\boldsymbol{w}'$, using a robust penalizer Ψ. Since the matching is not totally dense, we add a binary term $c(\boldsymbol{x})$ which is equal to 1 if and only if a match is available at $\boldsymbol{x}$.

We also multiply each matching penalization by a weight $\phi(\boldsymbol{x})$, which must be low in flat regions or when matches look false. To this end, we call $\tilde{\lambda}(\boldsymbol{x})$ the minimum eigenvalue of the autocorrelation matrix multiplied by 10. We also compute $\Delta(\boldsymbol{x}) = \sum_{i=1}^c |I_1^i(\boldsymbol{x}) - I_2^i(\boldsymbol{x} - \boldsymbol{w}'(\boldsymbol{x}))| + |\nabla I_1^i(\boldsymbol{x}) - \nabla I_2^i(\boldsymbol{x} - \boldsymbol{w}'(\boldsymbol{x}))|$. We then compute the score ϕ as a Gaussian kernel on Δ weighted by $\tilde{\lambda}$ with a parameter σ_M, experimentally set to $\sigma_M = 50$. More precisely, we define $\phi(\boldsymbol{x})$ at each point $\boldsymbol{x}$ with a match $\boldsymbol{w}'(\boldsymbol{x})$ as: $\phi(\boldsymbol{x}) = \sqrt{\tilde{\lambda}(\boldsymbol{x})}/(\sigma_M \sqrt{2\pi}) \exp(-\Delta(\boldsymbol{x})/2\sigma_M)$. The matching term is $E_M = c\phi\Psi(\|\boldsymbol{w} - \boldsymbol{w}'\|^2)$.

Minimization. This energy functional is non-convex and non-linear. To solve it, we use the framework of Brox *et al.* [5]. An incremental coarse-to-fine warping strategy is used with a downsampling factor $\eta = 0.95$. The remaining equations are still non-linear due to the robust penalizers. We apply 5 inner fixed point iterations where the non-linear weights and the flow increments are iteratively updated while fixing the other. To approximate the solution of the linear system, we use 25 iterations of the Successive Over Relaxation (SOR) method.

To downweight the matching term on fine scales, we use a different weight β^k at each level as proposed by Stoll *et al.* [24]. We set $\beta^k = \beta(k/k_{\max})^b$ where b is a parameter of the flow, k the current level of computation and k_{max} the number of the coarsest level.

5. Experiments

In this section, we evaluate the deep matching and DeepFlow on three challenging datasets. We compare several matching methods and show how our matching algorithm significantly improves the flow performance. For an evaluation of the parameters of matching and flow, we refer to an extended version [22].

The Middlebury dataset [2] has been extensively used for evaluating optical flow methods. It contains complex motions, but the displacements are small. Less than 3% of the pixels have a displacement over 20 pixels, and none goes over 25 pixels (training set).

The MPI-Sintel dataset [8] is a challenging flow evaluation benchmark. It contains long sequences with large motions and specular reflections. In the training set, more than 17.5% of the pixels have a motion over 20 pixels, approximately 10% over 40 pixels. We use throughout all our experiments the "final" version, containing rendering effects such as motion blur, defocus blur and atmospheric effects, see Figure 6. For our experiments, we split the original training set into a "small" training set (20%) and a validation set (80%). "EPE all" measures the average endpoint error over all pixels, s10-40 only over pixels with a speed between 10 and 40 pixels (similarly for s0-10 and s40+).

The KITTI dataset [10] contains real-world sequences taken from a driving platform. It includes non-lambertian surfaces, different lighting conditions, a large variety of materials and large displacements. More than 16% of the pixels have motion over 20 pixels.

5.1. Comparison of matching algorithms

We compare our matches to those obtained from several state-of-the-art algorithms: KLT tracks [1], sparse SIFT matching [26] and dense HOG matching used in LDOF [6], called HOG-NN in the following. Comparing different matching algorithms is delicate, as they produce matches possibly at different locations. We propose the following setup for a fair comparison. We assign to each point of a fixed grid with a spacing of 16 pixels the nearest neighbor match. We compute the density as the percentage of points with at least one match in a neighborhood of 15 pixels. We compute also precision as the percentage of those matches with an error below 10 pixels.

Matching method	Precision	Density
Deep matching	92.07%	**80.35**%
HOG-NN	92.49 %	40.06%
SIFT-NN	**93.89** %	16.35%
KLT	91.25%	35.61%

Table 1. Evaluation of the matching on the MPI-Sintel validation set. The density is the percentage of points from a fixed grid with at least one match in the neighborhood. The precision represents the ratio of correct matches.

Table 1 presents a comparison of the four methods on the MPI-Sintel validation set. Our deep matching method significantly outperforms the other methods in terms of density, for a similar precision. This is because KLT, SIFT and HOG matches appear only in highly textured regions. On the contrary, our method covers most of the image area and covers large motions better, see Figure 6.

5.2. Impact of the matches on the flow

To precisely evaluate the importance of the matching part in the flow estimation, we compare results obtained without and with deep matching. We also experiment with KLT, SIFT-NN and HOG-NN matches. For all cases, we carefully optimize the flow parameters independently on the "small" training set of MPI-Sintel. We employ a gradient descent strategy with multiple initializations followed by a small grid search. For HOG-NN, the weights ϕ are set to $(d_2 - d_1)/d_1$ as in [6] that measures the uniqueness of the match, where d_i is the distance to the ith nearest neighbor descriptor.

Matching input	EPE all	s0-10	s10-40	s40+
Deep matching	**4.422**	**0.712**	5.092	**29.229**
HOG-NN	5.273	0.764	**4.972**	37.858
SIFT-NN	5.444	0.846	5.313	38.283
KLT	5.513	0.820	5.304	39.197
No match	5.538	0.786	5.229	39.862

Table 2. Comparison of the endpoint error on the validation set of MPI-Sintel when changing the input matches.

Table 2 shows the average endpoint error, averaged over all pixels and over regions with large displacements for the MPI-Sintel validation set. KLT, SIFT-NN and HOG-NN slightly improve the performance for fast motion, between 1 and 2 pixels. Deep matching outperforms them especially for large displacements, for which the error is reduced by 10 pixels on average. This demonstrates that the estimation of the flow greatly benefits from our dense matches. Figure 6 displays a comparison of our flow with LDOF [6]. Clearly, the motion of many difficult regions is better captured with the help of dense matching. This is especially important in weakly textured regions, see top-right example of Figure 6.

5.3. Results on MPI-Sintel

Table 3 compares our method to state-of-the-art algorithms on the MPI-Sintel test set; parameters are optimized on the "small" training set. Our method outperforms current algorithms, especially for s40+. We refer to the webpage for complete results including the "clean" version.

Method	EPE all	s0-10	s10-40	s40+	Time
DeepFlow	**7.212**	1.284	**4.107**	**44.118**	**19**
S2D-Matching [16]	7.872	1.172	4.695	48.782	~2000
MDP-Flow2 [33]	8.445	1.420	5.449	50.507	709
LDOF [6]	9.116	1.485	4.839	57.296	30
Classic+NL [25]	9.153	**1.113**	4.496	60.291	301

Table 3. Results on the "Final" version of the MPI-Sintel test set. The reported time is for one processor core @3.6GHz in seconds.

Timings. Our matching algorithm takes approximately 2 seconds per frame pair[1] while the flow computation takes around 17 seconds using one CPU core. The total time to compute the flow is thus below 20 seconds. See Table 3 for a comparison with other approaches. All timings are obtained by running the online available code on the same processor core with the exception of [16], where we report timings obtained from the authors.

5.4. Results on Middlebury

We optimize the parameters on the Middlebury training set by minimizing the average angular error with the same strategy as for MPI-Sintel. We find weights quasi-zero for the matching term due to the absence of large displacements. Table 4 compares our results on the test set to a few state-of-the-art methods. Our mean rank based on the endpoint error is 45.9 at the time of publication. Note that a small difference in one sequence can lead to a huge difference in ranking. We can clearly observe that our matching algorithm does not improve the motion estimation in the context of small displacements.

Method	AEE	AAE
DeepFlow	0.42	4.22
MDP-Flow2 [33]	**0.25**	**2.45**
LDOF [6]	0.56	4.55
Classic+NL [25]	0.32	2.90

Table 4. Results on Middlebury. Average endpoint error (AEE) and angular error (AAE) of a few methods over the test sequences.

5.5. Results on KITTI

Table 5 summarizes the main results on the KITTI benchmark (see official website for complete results), when optimizing the parameters on the KITTI training set. AEE is the average endpoint error over all pixels while AEE-Noc only considers non-occluded areas. Out-Noc3, respectively Out3, is the percentage of pixels with an endpoint error over 3 pixels for non-occluded areas, resp. all pixels. DeepFlow is competitive with the other methods. Note that the learned

[1] Note that we resize images to 256×128 pixels in all our experiments before computing the deep matching.

Method	AEE	AEE-Noc	Out3	Out-Noc3
DeepFlow	5.8	**1.5**	17.93%	7.49%
Data-Flow [29]	**5.5**	1.9	**14.85%**	**7.47%**
LDOF [6]	12.4	5.5	31.31%	21.86%
Classic+NL [25]	7.2	2.8	20.66%	10.60%

Table 5. Results on KITTI test set. See text for details.

parameters on KITTI and MPI-Sintel are close. In particular, running the experiments with the same parameters as MPI-Sintel decreases AEE-Noc by only 0.1 pixels. This shows that our method does not suffer from overfitting.

Acknowledgments. This work was supported by the European integrated project AXES, the MSR/INRIA joint project, the LabEx PERSYVAL-Lab (ANR-11-LABX-0025), and the ERC advanced grant ALLEGRO.

References

[1] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 2004. 2, 6

[2] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *IJCV*, 2011. 1, 2, 6

[3] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010. 2

[4] M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 1996. 1, 2

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 2, 3, 6

[6] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. PAMI*, 2011. 1, 2, 5, 6, 7, 8

[7] A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *ICCV*, 2005. 2, 6

[8] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 1, 2, 5, 6

[9] A. Ecker and S. Ullman. A hierarchical non-parametric method for capturing non-rigid deformations. *Image and Vision Computing*, 2009. 2

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 2013. 2, 6

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. PAMI*, 2007. 1

[12] D. Keysers, T. Deselaers, C. Gollan, and H. Ney. Deformation models for image recognition. *IEEE Trans. PAMI*, 2007. 2, 3

[13] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *CVPR*, 2013. 2

[14] I. Laptev and P. Pérez. Retrieving actions in movies. In *ICCV*, 2007. 1

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998. 2, 3, 4

[16] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *ICCV*, 2013. 2, 7

[17] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Trans. PAMI*, 2011. 2

[18] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, 1990. 4

[19] P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Work.*, 2009. 1

[20] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 2005. 2

[21] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optic flow computation with theoretically justified warping. *IJCV*, 2006. 1, 2

[22] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Deep matching and its application to large displacement optical flow. Technical report, 2013. 4, 5, 6

[23] F. Steinbrucker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. In *ICCV*, 2009. 1

[24] M. Stoll, S. Volz, and A. Bruhn. Adaptive integration of feature matches into variational optical flow methods. In *ACCV*, 2012. 6

[25] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010. 2, 5, 7, 8

[26] R. Szeliski. *Computer Vision: Algorithms and Applications*. 2010. 1, 2, 3, 6

[27] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. *CVPR*, 2008. 2

[28] S. Uchida and H. Sakoe. A monotonic and continuous two-dimensional warping based on dynamic programming. In *ICPR*, 1998. 2, 3

[29] C. Vogel, S. Roth, and K. Schindler. An evaluation of data costs for optical flow. In *GCPR*, 2013. 2, 8

[30] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 2013. 1

[31] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, 2009. 1, 2, 3

[32] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *IJCV*, 2006. 1, 2

[33] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. PAMI*, 2012. 1, 2, 7

[34] H. Zimmer, A. Bruhn, and J. Weickert. Optic flow in harmony. *IJCV*, 2011. 5, 6