# Locally Affine Sparse-to-Dense Matching for Motion and Occlusion Estimation

Marius Leordeanu[1]*, Andrei Zanfir[1]*, Cristian Sminchisescu[2,1]
[1]Institute of Mathematics of the Romanian Academy
[2]Department of Mathematics, Faculty of Engineering, Lund University
marius.leordeanu@imar.ro, andrei.zanfir@imar.ro, cristian.sminchisescu@math.lth.se

## Abstract

*Estimating a dense correspondence field between successive video frames, under large displacement, is important in many visual learning and recognition tasks. We propose a novel sparse-to-dense matching method for motion field estimation and occlusion detection. As an alternative to the current coarse-to-fine approaches from the optical flow literature, we start from the higher level of sparse matching with rich appearance and geometric constraints collected over extended neighborhoods, using an occlusion aware, locally affine model. Then, we move towards the simpler, but denser classic flow field model, with an interpolation procedure that offers a natural transition between the sparse and the dense correspondence fields. We experimentally demonstrate that our appearance features and our complex geometric constraints permit the correct motion estimation even in difficult cases of large displacements and significant appearance changes. We also propose a novel classification method for occlusion detection that works in conjunction with the sparse-to-dense matching model. We validate our approach on the newly released Sintel dataset and obtain state-of-the-art results.*

## 1. Introduction

Estimating a dense correspondence field between video frames is essential for many higher-level visual tasks, such as tracking, grouping, image segmentation and object recognition. This problem, known as optical flow estimation, consists of inferring the displacements in the image caused by camera motion or the moving objects in the scene, and has been initially formulated for small, differential apparent motions.[1] The task is difficult for many reasons: **1)** finding a dense correspondence field is computationally expensive and requires sophisticated optimization methods; **2)** the presence of multiple differently moving objects in the scene yields abrupt discontinuities in the motion field and produces occlusion regions that are hard to model; **3)** when displacements are larger than the object structure, the use of efficient local image approximations is no longer valid, and matching becomes significantly more difficult; **4)** ambiguities could arise from blur and illumination changes.

Optical flow estimation is one of the first problems studied in computer vision [24, 10], and was initially formulated for Lambertian surfaces under the assumption of full visibility. Most current methods are closely related to the first mathematical models of the 1980s where brightness constancy assumptions and smooth local flow constraints were used. Such energy models are appropriate for small displacements, where Taylor image approximations hold, but have difficulties when motions are larger than the local image structure [6]. Today's methods are superior to the first approaches, but the task is far from solved, mainly in the difficult case of large displacements. Recently, there has been growing interest in integrating sparse feature matching into models that process large motions [30, 6, 32, 22], with significant improvements over traditional approaches. However, the sophistication of the sparse matching component used is still limited: in [6] authors use nearest neighbor schemes, whereas others [32, 22] rely on energy models rooted in the classic Total Variation (TV) formulation.

We make the following contributions: **1)** we propose the first affine-invariant and occlusion-aware matching algorithm to estimate a dense correspondence field between images, under potentially large displacements. We rely on a new sparse-to-dense approach starting from a complex sparse matching cost, then making the transition towards a dense motion field using a novel interpolation method. The final motion field is obtained using local refinement based on continuous optimization. **2)** We introduce a new occlusion classifier that operates in conjunction with the sparse-to-dense interpolation method, with the two modules providing mutually informative cues, to improve performance.

*The first two authors contributed equally.

[1]The is part of the general problem of *motion field estimation*, although optical cues do not always fully constrain the field, as it is well known.

**Overview of our approach:** We propose a hierarchical sparse-to-dense matching method that generalizes ideas both from the traditional coarse-to-fine variational approach and from recent methods based on sparse feature matching [4, 16, 19, 8, 6, 32, 22]. Our method consists of three main phases (Figure 1). It starts from optimizing a sparse matching energy with rich appearance unary terms and locally affine, occlusion sensitive, geometric constraints. This phase is meant to recover most of the correct motions, even in the case of large displacements. Next, using the same high-order geometric constraints, a second stage of sparse-to-dense interpolation follows, making the transition from a set of sparse matches on a grid to a dense correspondence field. Once we are in the neighborhood of the correct solution, the final output is obtained by a continuation method that uses the total variation (TV) model. After every matching stage, a classifier based on intermediate motion cues produces an occlusion probability map that is then passed to the next matching level. At the end, all cues obtained from the different stages of correspondence field estimation are fed into the occlusion classifier that produces the final occlusion map. Through this synergy between matching and occlusion estimation we obtain both improved optical flow and occlusion maps. The stages of our approach are:

1. **Sparse Matching:** Initialize a discrete set of candidate matches for each point on a sparse grid (in the first image) using dense kNN matching (in the second image) based on local feature descriptors. Use the output of a boundary detector and cues from the initial sparse matching model to infer a first occlusion probability map. Then discretely optimize a sparse matching cost function using both unary, data terms (from local features), and geometric relationships based on a locally affine model with occlusion constraints. Use the improved matches to refine the occlusion map.
2. **Sparse-to-Dense Interpolation:** Fix the sparse matches from previous step. Then apply the same occlusion sensitive geometric model to obtain an accurate sparse-to-dense interpolation of matches.
3. **Dense matching refinement:** use a TV model with continuous flow optimization to obtain the final dense correspondence field. Use matching cues computed from all stages to obtain the final occlusion map.

**Discussion:** Phase 1 of our approach is related to both graph matching algorithms [4, 16, 19, 8] that use sparse local features and second or higher order geometric relationships, and recent optical flow methods that used descriptor matching [6, 32, 22]. This stage aims to minimize the relative motion error even when the displacements are generally large, so to bring the solution in the neighborhood of the correct one. The next sparse-to-dense interpolation phase ensures a natural transition towards a dense motion field. At the last stage we perform continuous optimization of a variational

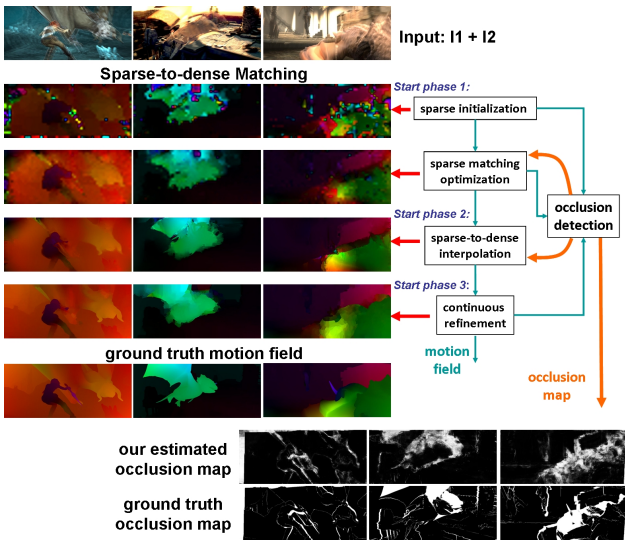


Figure 1. Left: motion fields estimated at different stages of our method. Right: flowchart of our approach. Sparse matching results are resized to original image size for display.

model for accurate localization. The different phases complement each other: phase 1 is not sensitive to small displacements, since it uses sparse matches and descriptors extracted over relatively large neighborhoods; phase 3 uses a TV model and is most appropriate for small motions; phase 2 provides a transition between 1 and 3. Our approach is different from large displacement optical flow methods that append an energy term based on sparse matching to the variational framework [6]. It is also different from methods that use descriptor matching only to find candidate flows, but then optimize an energy model with local smoothness constraints [32]. The SIFT-flow energy model [22] uses rich descriptors [23], but a local flow spatial regularization term. Our method is closer to the approach of [4] that first applies sparse graph matching formulated as IQP, then uses thin plate splines for dense interpolation. Differently from that work, we efficiently match thousands of features and preserve boundaries, as discussed shortly.

## 2. Mathematical Formulation

Before we detail our main sparse-to-dense algorithm, we first define the sparse matching task. Given two images $I_1$ and $I_2$ of the same scene, the sparse matching problem consists of finding the correspondences in $I_2$ for $N$ points located on a uniform grid in $I_1$. Finding the correct matches is equivalent to solving for the sparse displacement field–a $2N \times 1$ vector $\mathbf{w}$, such that the displacement of feature $i$ is $[w_{ix}, w_{iy}]$. Sparse matching is then formulated as the optimization of a cost function that is a linear combination of a data term and a spatial energy term:

$$E(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_S(\mathbf{w}). \quad (1)$$

Here $\lambda$ weighs differently the spatial term $E_S(\mathbf{w})$ vs. the data term $E_D(\mathbf{w})$. We want to find $\mathbf{w}$ that minimizes $E(\mathbf{w})$.

**Data term:** For the unary data term we use descriptors computed over relatively large regions ($51 \times 51$ patches in $436 \times 1024$ images) in order to better cope with large displacements, motion and defocus blur, and illumination variations. Our data term $E_D(\mathbf{w}) = \sum_{\mathbf{i}} \psi_{D_i}(\mathbf{w}_i)$ is the sum of appearance errors, where $\mathbf{w}_i = [w_{ix}, w_{iy}]$ is the displacement of feature $i$ and $\psi_{D_i}(\mathbf{w}_i)$ is a linear combination of Euclidean distances computed for color and boundary descriptors: $\psi_{D_i}(\mathbf{w}_i) = \|\mathbf{d}_c^{(1)}(\mathbf{p}_i) - \mathbf{d}_c^{(2)}(\mathbf{p}_i + \mathbf{w}_i)\| + \beta\|\mathbf{d}_b^{(1)}(\mathbf{p}_i) - \mathbf{d}_b^{(2)}(\mathbf{p}_i + \mathbf{w}_i)\|$. Here, $\mathbf{p}_i$ denotes the $x, y$ location of feature $i$, and $\mathbf{d}_c^{(k)}(\mathbf{p})$ and $\mathbf{d}_b^{(k)}(\mathbf{p})$ are the color and boundary descriptors from image $I_k$ at location $\mathbf{p}$. Using edge information is not uncommon to optical flow estimation methods [6, 32]. However, in our case, instead of using local, pixel-wise gradients, we compute boundary descriptors over larger regions with a more accurate boundary detector [18]. Note that our formulation is general enough to work with different appearance descriptors.

**Spatial term:** Points that are likely to be on the same object surface and are close to each other are also likely to have a similar displacement. The motions of such points within a certain neighborhood are expected to closely follow an affine transformation. To model this, we first define a geometric neighborhood system over the grid locations (see Figure 2), and link neighboring points $(i, j)$ using an edge strength function $e_{ij}$–meant to measure the probability that two points lie on the same object surface and obey a similar affine transformation from $I_1$ to $I_2$. We model $e_{ij} \in [0, 1]$ by an exponential that considers the pairwise distance $d_{ij}$ between the points $(i, j)$, the average occlusion probability $o_{ij}$ on the straight segment between $i$ and $j$ and the intervening maximum boundary response $b_{ij}$ along the same segment: $e_{ij} = \exp(-\mathbf{a}^\top \mathbf{g}_{ij})$, where $\mathbf{g}_{ij} = [d_{ij}, o_{ij}, b_{ij}]$ and $\mathbf{a}$ is a vector of positive weights; $o_{ij}$ and $b_{ij}$ are closely related to the intervening contours used for computing soft Ncuts segmentations in natural boundary detection [2]. Thus, $e_{ij}$ represent segmentation cues, softly separating points from different objects.

In our model, each feature $i$ has a neighborhood $\mathcal{N}_i$, with $N_i$ neighbors (minimum 21 in our implementation), to which it connects with strength $e_{ij}$, for $j \in \mathcal{N}_i$. We use the locations of $i$'s neighbors, $\mathbf{p}_{N_i}^{(1)}$ in $I_1$, their current estimated destinations in $I_2$, $\mathbf{p}_{N_i}^{(2)}$, and their membership weights, to estimate a motion model that *predicts* the destination $\mathbf{p}_i^{(2)}$ of the current point $i$ in $I_2$: $\tilde{\mathbf{p}}_i^{(2)} = T_{\mathcal{N}_i}(\mathbf{p}_i^{(1)})$; $T_{\mathcal{N}_i}$ is potentially *different* for each $i$ - the transformation is not rigid and varies, if necessary, over the grid; as we show next, it is also discontinuity preserving through the pairwise weights $e_{ij}$. The prediction error between the current $\mathbf{p}_i^{(2)}(\mathbf{w}_i) = \mathbf{p}_i^{(1)} + \mathbf{w}_i$ and the estimated $\tilde{\mathbf{p}}_i^{(2)}$ forms the basis of our spatial energy term:

$$E_S(\mathbf{w}) = \sum_i \|\mathbf{p}_i^{(2)}(\mathbf{w}_i) - T_{\mathcal{N}_i}(\mathbf{p}_i^{(1)})\|^2. \quad (2)$$

The use of the squared error, corresponding to Gaussian noise, preserves, in our case, flow discontinuities near occlusions and image boundaries, as the neighbors used to estimate $T_{\mathcal{N}_i}$ participate with weights $e_{ij}$, which are functions of relative occlusion and boundary cues. Note that the general form of our spatial energy term from Eq. 2 is of order $N_i$. To estimate $T_{\mathcal{N}_i}$, for each point $i$, we would generally need the displacements of all its $N_i$ neighbors. Every point connects to all its neighbors and will be involved in $N_i$ transformation estimates. This makes the problem intractable, in general. However, when $T_{\mathcal{N}_i}$ is a local affine transformation, Eq. 2 reduces to a second order energy that can be efficiently optimized (see next).

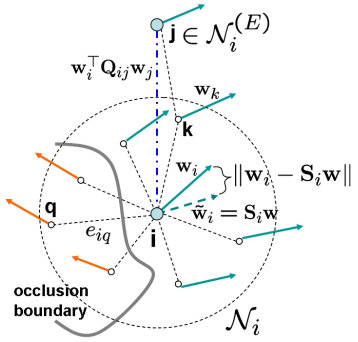


Figure 2. A feature $i$ is connected to its neighbor $q$ in $\mathcal{N}_i$ with strength $e_{iq}$ - a function of distance, and intervening boundary and occlusion information. The current motions of neighbors in $\mathcal{N}_i$ are used to predict the motion $\tilde{\mathbf{w}}_j$ at point $i$, through an affine mapping $\mathbf{S}_i$. Our novel quadratic affine error model effectively produces an extended neighborhood system $\mathcal{N}_i^{(E)}$ where pairs of *connected* points $(i, j)$ contribute to the total error as $\mathbf{w}_i^\top \mathbf{Q}_{ij} \mathbf{w}_j$.

## 2.1. Locally Affine Spatial Model

For each point $i$, we estimate its affine transformation $T_{\mathcal{N}_i} = (\mathbf{A}_i, \mathbf{t}_i)$ from neighboring motions by weighted least squares, with weights $e_{ij}$, $\forall j \in \mathcal{N}_i$. Let $2N_i \times 1$ $\mathbf{p}_{N_i} = \mathbf{p}_{N_i}^{(2)}$ be the estimated positions of its neighbors in $I_2$ and $\mathbf{M}$ the Moore-Penrose pseudoinverse of the least squares problem. Note that $\mathbf{M}$ is fixed for each $i$ and can be pre-computed before optimization. It depends only on the locations of the neighbors in $I_1$, $\mathbf{p}_{N_i}^{(1)}$, and their weights. The least squares solution $\mathbf{M}\mathbf{p}_{N_i}$ gives the estimated $(\mathbf{A_i}, \mathbf{t_i})$:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{m}_1^\top \mathbf{p}_{N_i} & \mathbf{m}_2^\top \mathbf{p}_{N_i} \\ \mathbf{m}_3^\top \mathbf{p}_{N_i} & \mathbf{m}_4^\top \mathbf{p}_{N_i} \end{bmatrix} \qquad \mathbf{t}_i = \begin{bmatrix} \mathbf{m}_5^\top \mathbf{p}_{N_i} \\ \mathbf{m}_6^\top \mathbf{p}_{N_i} \end{bmatrix}.$$

Here $\mathbf{m}_k^\top$ denotes the k-th row of $\mathbf{M}$. Using $\mathbf{p}_i^{(1)} = [x_i, y_i]$, the predicted end position $\tilde{\mathbf{p}}_i^{(2)}$ of $i$ in image $I_2$ is:

$$\begin{aligned}\tilde{\mathbf{p}}_i^{(2)} &= T_{\mathcal{N}_i}(\mathbf{p}_i^{(1)}) = \mathbf{A}_i\mathbf{p}_i^{(1)} + \mathbf{t}_i \qquad (3)\\ &= \begin{bmatrix} x_i\mathbf{m}_1^\top + y_i\mathbf{m}_2^\top + \mathbf{m}_5^\top \\ x_i\mathbf{m}_3^\top + y_i\mathbf{m}_4^\top + \mathbf{m}_6^\top \end{bmatrix} \mathbf{p}_{N_i} \\ &= \overbrace{\left[\begin{array}{c|c|c} 0 & \begin{array}{c} x_i\mathbf{m}_1^\top + y_i\mathbf{m}_2^\top + \mathbf{m}_5^\top \\ x_i\mathbf{m}_3^\top + y_i\mathbf{m}_4^\top + \mathbf{m}_6^\top \end{array} & 0 \end{array}\right]}^{\mathbf{S}_i} \mathbf{p}^{(2)}.\end{aligned}$$

Here $\mathbf{p}^{(2)}$ are the estimated final destinations of all points in $I_2$ and $\mathbf{S}_i$ is a $2 \times N$ matrix with padded zero columns corresponding to elements in $\mathbf{p}^{(2)}$ not belonging to points in $\mathcal{N}_i$. Note that $\mathbf{S}_i$ does not depend on the unknown displacements (or final positions $\mathbf{p}^{(2)}$) which we want to solve for. This fact, together with the next property, are essential for efficient optimization (§3).

**Property** 1**: a)** $\mathbf{S}_i\mathbf{p}^{(1)} = \mathbf{p}_i^{(1)}$, where $\mathbf{p}^{(1)}$ are the initial grid locations of all points. **b)** $\mathbf{S}_i\mathbf{w} = \tilde{\mathbf{w}}_i$, where $\tilde{\mathbf{w}}_i$ is the estimated motion of point $i$.

**Proof:** a). Note that $\mathbf{S}_i\mathbf{p}^{(1)}$ is the estimated $\tilde{\mathbf{p}}_i$ at location $i$ if the final positions of all points in image $I_2$ are the same as the initial positions $\mathbf{p}^{(2)} = \mathbf{p}^{(1)}$. Since the points do not move, the affine model will be the identity transformation, so $\tilde{\mathbf{p}}_i = \mathbf{p}_i^{(1)}$. **b)**. We use the previous result from **1.a)** to show that: $\tilde{\mathbf{p}}_i = \mathbf{S}_i\mathbf{p}^{(2)} = \mathbf{S}_i(\mathbf{p}^{(1)} + \mathbf{w}) = \mathbf{p}_i^{(1)} + \mathbf{S}_i\mathbf{w}$. Then $\tilde{\mathbf{w}}_i = \tilde{\mathbf{p}}_i - \mathbf{p}_i^{(1)} = \mathbf{S}_i\mathbf{w} + \mathbf{p}_i^{(1)} - \mathbf{p}_i^{(1)} = \mathbf{S}_i\mathbf{w}$. □

Using Property 1, Eq. 2 can now be written as:

$$\begin{aligned}E_S(\mathbf{w}) &= \sum_i \|\mathbf{p}_i^{(2)} - \mathbf{S}_i\mathbf{p}^{(2)}\|^2 = \sum_i \|\mathbf{w}_i - \mathbf{S}_i\mathbf{w}\|^2 \\ &= \sum_i (\mathbf{w}_i - \mathbf{S}_i\mathbf{w})^\top(\mathbf{w}_i - \mathbf{S}_i\mathbf{w}) \\ &= \mathbf{w}^\top(\mathbf{I} - 2\mathbf{S}_{1\ldots n} + \sum_i \mathbf{S}_i^\top\mathbf{S}_i)\mathbf{w} \\ &= \mathbf{w}^\top\mathbf{S}\mathbf{w}, \qquad (4)\end{aligned}$$

where $\mathbf{S}_{1\ldots n} = \left[\begin{array}{c|c|c} \mathbf{S}_1^\top & \cdots & \mathbf{S}_n^\top \end{array}\right]^\top$. To simplify calculations we make matrix $\mathbf{S}$ symmetric $\mathbf{S} \leftarrow \frac{1}{2}(\mathbf{S}^\top + \mathbf{S})$ without changing the energy.

**Discussion:** $\mathbf{S}$ implicitly induces a different, extended neighborhood system in the spatial energy, as follows: for any two points $(i, j)$ with a common neighbor in the original neighborhood system, that is $\exists k\ s.t.\ k \in \mathcal{N}_i \wedge k \in \mathcal{N}_j$, it is relatively easy to show that their corresponding $2 \times 2$ sub-matrix $\mathbf{Q}_{ij} \leftarrow \mathbf{S}_{(ix,iy);(jx,jy)}$ in Eq. 4 has, in general, nonzero elements, thus the two points $(i, j)$ will influence each other and contribute to the total energy with $\mathbf{w}_i^\top\mathbf{Q}_{ij}\mathbf{w}_j$. Let $\mathcal{N}_i^{(E)}$ be the extended neighborhood of $i$, including such points $j$ that influence $i$ (Figure 2). The neighbors in $\mathcal{N}_i^{(E)}$ are used in our efficient optimization with sequential updates (§3).

As mentioned previously our geometric constraints are affine invariant - stated in the next property:

**Property 2:** Let $\mathbf{w}_A$ be the motion field corresponding to some global affine transformation $A$ over all points in $I_1$. Then $E_S(\mathbf{w}_\mathbf{A}) = 0$ regardless of the geometric neighborhood system chosen and the weights on edges.

**Proof:** Follows immediately from the fact that each individual motion will perfectly agree with the estimated affine transformation. Regardless of the neighbors chosen and their soft memberships, the estimated affine model will always be the same as the global model.□

This property has interesting implications. The spatial error term will be zero for transformations such as rotation or scaling, regardless of the scaling factor or the rotation angle, as our model is affine invariant. This is radically different and more powerful than both variational optical flow, as well as graph matching approaches based on second or third-order constraints, which are not affine invariant. Moreover, the use of boundary and occlusion information could extend the class of zero spatial error displacement fields to piecewise affine transformations, since it could decouple the graph. Soft boundary information has a gradual effect in allowing a degree of abrupt motion discontinuity.

Other formulations related to our sparse matching model include [20, 13, 14], with [20] being the only affine invariant model, but without occlusion. Inspired by non-linear embedding techniques [20] propose an LP formulation for locally affine invariant matching, which cannot handle a very large number of features efficiently, however (see §3 for numerical details). Our sparse matching approach offers a locally affine formulation based on only pairwise, second-order relationships, allowing efficient optimization (§3).

## 3. Sparse-to-Dense Matching Algorithm

Since the error cannot be negative, $\mathbf{S}$ is semi-positive definite, giving a convex spatial term. However, due to the data term that could *a priori* have almost any shape, the overall cost $E_S(\mathbf{w})$ is generally non-convex. Therefore we can only hope for efficient local optimization. As shown next, it turns out that the quadratic form of the spatial term in Eq. 4 allows us to perform an efficient optimization of the sparse matching, by sequential updates. Our approach is similar to Iterated Conditional Modes [5] and the L2QP method [17] for MAP labeling problems in MRFs - this matching task could be seen as a *labeling* problem since each node $i$ has a finite list of candidate assignments.

Let us consider the individual update of the displacement $\mathbf{w}_i$ at node $i$. Let $\mathbf{w}_0$ be a $2N \times 1$ vector with 0's at the two $x, y$ positions in $\mathbf{w}$ corresponding to $i$, and equal to $\mathbf{w}$ everywhere else. Let $\mathbf{l}_i = 2\sum_{j \in \mathcal{N}_i^{(E)}} \mathbf{Q}_{ij}\mathbf{w}_j$, with $\mathbf{Q}_{ij}$, the

sub-matrix of $\mathbf{S}$ defined earlier. Then the total energy is:

$$\begin{aligned} E(\mathbf{w}) &= \sum_{j \neq i} \psi_{D_j}(\mathbf{w_j}) + \psi_{D_i}(\mathbf{w_i}) \\ &\quad + \lambda \mathbf{w}_0^\top \mathbf{S} \mathbf{w}_0 + \mathbf{l}_i^\top \mathbf{w}_i + \mathbf{w}_i^\top \mathbf{Q}_{ii} \mathbf{w}_i \\ &= K_0 + \underbrace{\psi_{D_i}(\mathbf{w_i}) + \mathbf{l}_i^\top \mathbf{w}_i + \mathbf{w}_i^\top \mathbf{Q}_{ii} \mathbf{w}_i}_{K_i}. \quad (5) \end{aligned}$$

The energy is the sum of a constant term $K_0$, independent of $\mathbf{w}_i$ and $K_i$, which is a function of $\mathbf{w}_i$. For each $i$, $K_i$ can be computed with minimal memory and computational cost, as both $\mathbf{l_i}$ and $\mathbf{Q_i}$ use only information from neighbors in $\mathcal{N}_i^{(E)}$. This leads to our Algorithm 1 based on sequential updates. Lists $L_i$ contain the initial matches. During each iteration, sites $i$ are traversed in a certain order. At step S1 the optimum $\mathbf{w}_i^{(S)}$ of the local quadratic geometric term is found in closed-form. At step S2 the list of candidate matches for the current site is temporarily augmented with its current match (if not already included) and the continuous solution $\mathbf{w}_i^{(S)}$. From the augmented list the minimizer $\mathbf{w}_i^*$ of $K_i$ is found (S3) and the current $\mathbf{w}_i$ is updated (S4). Eq. 5 guarantees that energy is lowered after every iteration. In practice this sequential optimization procedure brings about $30\%$ improvement over the kNN initialization. Since the total cost is non-negative, bounded below, and the energy is monotonically decreasing, the sparse matching method is guaranteed to converge. Then, the sparse matches become input to an interpolation procedure that produces a dense field $\mathbf{w}^{(d)}$.

**Sparse-to-Dense Interpolation:** we fix the sparse displacements $\mathbf{w}$ as anchors, and, for each point $i$ at dense locations, we define a similar neighborhood system, with neighbors taken from the set of sparse anchors. Having fixed the sparse matches, we can afford to drop the data term and use only the spatial constraints to predict the dense motions based on the same locally affine model with occlusion and boundary constraints. It is straightforward to show that the spatial, dense energy model, can be globally optimized by simply setting the value of the dense field $\mathbf{w}^{(d)}$ to the predicted field $\tilde{\mathbf{w}}^{(d)}$, for each site $i$: $\mathbf{w}_i^{(d)} \leftarrow T_{\mathcal{N}_i}(\mathbf{p}_i) - \mathbf{p}_i$. As presented at the beginning, the dense output of this interpolation step becomes input to a TV-based continuation method, for dense refinement.

**Numerical Considerations:** The computational complexity of each sequential update is $O(N_i^{(E)})$, for a total cost, for all sites, per iteration, of $O(NN_i^{(E)})$. This has low, linear complexity in the number of features for a fixed number of neighbors $N_i^{(E)}$ per feature. The number of iterations in practice is between 5 to 10, which makes our sparse matching algorithm very efficient (30 sec). The low computational cost is possible only due to the use of sequential, local updates, which, in turn, are made possible by our novel quadratic spatial term (Eq. 4), with pairwise constraints. In contrast, the $L1$ norm formulations of [20] results in an $LP$ relaxation with $N^2$ variables and $N^2$ constraints. In our case ($N \approx 1500$) variables would be in the order of millions, which would make that approach computationally prohibitive. Different from the discrete optimization of [20], we use a mixed continuous-discrete scheme, with new continuous candidates $\mathbf{w}_i^{(S)}$ proposed as optima of the local spatial quadratic term, for each sequential update.

**Algorithm 1** Sparse-to-Dense Matching

Initialize lists $L_i$ of candidate assignments for each point $i$ using kNN.
Use 1-NN to initialize $\mathbf{w} \leftarrow \text{argmin}_{\mathbf{w}} E_D(\mathbf{w})$
**Sparse Matching**:
**repeat**
  **for all** sites $i$ **do**
    S1: $\mathbf{w}_i^{(S)} \leftarrow \text{argmin}_{\mathbf{u} \in \mathbb{R}^2} \mathbf{l_i}^\top \mathbf{u} + \mathbf{u}^\top \mathbf{Q_{ii}} \mathbf{u}$
    S2: $L \leftarrow L_i \cup \{\mathbf{w}_i^{(S)}, \mathbf{w}_i\}$
    S3: $\mathbf{w}_i^* \leftarrow \text{argmin}_{\mathbf{u} \in L} \psi_{D_i}(\mathbf{u}) + \mathbf{l}_i^\top \mathbf{u} + \mathbf{u}^\top \mathbf{Q}_{ii} \mathbf{u}$
    S4: Update $\mathbf{w}_i \leftarrow \mathbf{w}_i^*$
  **end for**
**until** convergence
**Sparse-to-Dense Interpolation**:
For each dense point $i$ set $\mathbf{w}_i^{(d)} \leftarrow T_{\mathcal{N}_i}(\mathbf{p}_i^{(1)}) - \mathbf{p}_i^{(1)}$
**Dense matching refinement**:
TV optimization: $\mathbf{w}_{opt}^{(d)} \leftarrow \text{argmin}_{\mathbf{w}} E_{\mathbf{TV}}(\mathbf{w})$, starting from $\mathbf{w}^{(d)}$
**return** $\mathbf{w}_{opt}^{(d)}$

## 4. Occlusion Detection

As objects move in the scene, background regions near their boundaries become occluded (or unoccluded) from one frame to the next. The size and shapes of these regions of occlusion depend mainly on the magnitude and velocity of the relative motion and 3D geometric scene layout. Occlusion detection is a notoriously difficult problem [26, 29, 9, 25, 11, 3], important in computer vision for 3D scene understanding, segmentation and recognition. The true motion field inside occlusion regions is hard to predict and current motion models are not yet powerful enough to accomplish this task, even though there are many promising recent approaches. Many occlusion estimation methods, including ours, take advantage of the fact that these optical flow energy models often fail in the areas of occlusion [27, 1, 11, 12]. In our work, we consider model fitting errors as well as conflicting matching outputs, where several points from one image land on the same point in the other image. Also, occlusion regions are often around discontinuities in the image or in the matching field, which turns out to be an important cue for occlusion region detection.

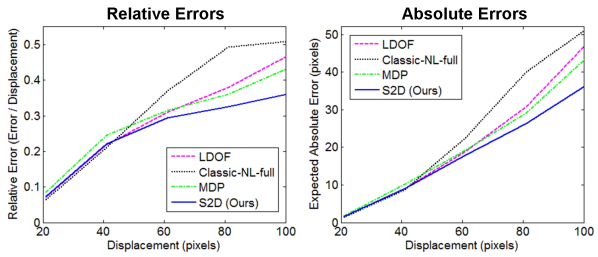


Figure 4. Relative and absolute average motion field errors on the 996 image pairs of Sintel-TrainTest set, that we used for testing, between our method and three other optical flow algorithms: MDP [32], LDOF [6] and Classic-NL-full [28]. Note the superior performance of our method in cases of large displacement.

We treat the occlusion estimation task as a binary classification problem [21, 15]. We compute cues based on the ideas mentioned above and train a different Random Forest classifier for each matching phase, depending on the cues available at that stage. We use the MATLAB function $treebagger$. It is worth mentioning here the most important features (out of 20) from the final occlusion estimation stage, automatically ranked by $treebagger$. We divide the most important cues in four categories, depending on the type of information they use. **1) Cues from matching initialization stage:** the most relevant feature in this category was the confidence of initial nearest neighbor matching–the number of neighbors close enough (within a fixed threshold) to the first nearest neighbor. Regions that are occluded cannot be found in the second image, so the size of this list is expected to be very small in their case. **2) Cues from the sparse matching stage:** we considered the different deviations of the output from the sparse matching model–errors in the data and spatial terms as well as errors in the affine model least squares fitting. The deviation from the affinity model assumption was the most relevant, in this category, for estimating occlusion regions. **3) Dense matching cues:** the most powerful feature computed from the final motion field is the number of points from $I_1$ matched to the same point in $I_2$. Those points in $I_1$ are very likely to be in fact occluded in $I_2$, since they cannot land on the same point in $I_2$. **4) Motion boundary features**: once we have the final motion field we computed features from the continuous output of Gb [18] using the $x, y$ motions as input layers.

## 5. Experiments

We performed experiments on the new MPI Sintel dataset and benchmark [7], consisting of a test set with 552 image pairs (12 folders) and 1040 training image pairs (23 folders). The ground truth used in the evaluation is the actual motion field in both visible and occluded (unmatched) areas. The dataset contains long photo-realistic video sequences from the animated movie Sintel with extremely difficult cases of very large displacements (often around 40 pixels or more), motion and defocus blur, specular reflections and strong atmospheric effects. We divided the official Sintel training set in two sets. The first one, Sintel-Train46, consists of 46 image pairs, with two pairs selected from the middle of each training folder. We used it for training and parameter validation. The second set, Sintel-TrainTest, contains the rest of the 994 training pairs, which we used for testing, along the official benchmark Sintel Test set. In Table 1 we present comparative results on Sintel-TrainTest. The end point errors (EPE) are averages over all pixels (occluded or not) for all 994 Sintel-TrainTest image pairs. In Figure 4 we plot the EPE of different methods vs. the displacement magnitude. Note our superior performance, especially for large displacements.

Table 1. Average end point motion errors (EPE) over all dense locations for the 996 image pairs from Sintel-TrainTest.

| Alg. | S2D (Ours) | MDP [32] | LDOF [6] | NL-full [28] |
|---|---|---|---|---|
| EPE | **5.45** | 5.96 | 6.20 | 6.30 |

Table 2. Final results on Sintel optical flow benchmark; $epe$ stands for end point errors, $epe_m$ are errors for visible points, and $epe_o$ are errors for occluded points. The last three columns show average errors for different motion magnitudes (in pixels).

| Alg. | $epe$ | $epe_m$ | $epe_o$ | 0-10 | 10-40 | 40+ |
|---|---|---|---|---|---|---|
| S2D | **7.87** | **3.92** | **40.1** | 1.17 | 4.69 | **48.78** |
| [32] | 8.45 | 4.15 | 43.43 | 1.42 | 5.45 | 50.51 |
| [6] | 9.12 | 5.04 | 42.34 | 1.49 | 4.84 | 57.30 |
| [28]-full | 9.16 | 4.81 | 44.51 | 1.11 | **4.50** | 60.29 |
| [10] | 9.61 | 5.42 | 43.73 | 1.88 | 5.34 | 58.27 |
| [28]++ | 9.96 | 5.41 | 47.00 | 1.40 | 5.10 | 64.14 |
| [28]-fast | 10.09 | 5.66 | 46.15 | **1.09** | 4.67 | 67.80 |
| [31] | 11.93 | 7.32 | 49.37 | 1.16 | 7.97 | 74.80 |

We also tested our method on the official Sintel benchmark and obtained state-of-the-art results (see Table 2), outperforming the other published methods especially on large displacements (larger than 40 pixels). The experiments show that our method is generally superior to [32, 6] for displacements both large and small, while having a larger small-motion error than TV-based methods [28].

Without continuous refinement, our method has an average EPE error of 9.4 on Sintel benchmark, superior to [10], [28]-fast and [28]++. This proves the usefulness of the first two phases in our hierarchical approach. We also obtained competitive results on the Middlebury dataset, on which S2D-Matching ranked better than [6, 10, 31] and [28]++.

**Experiments on occlusion detection:** In Table 3 we present quantitative comparisons of our occlusion classifier versus two state-of-the-art optical flow methods with occlusion estimation. We also present representative results in Figure 3. Our occlusion maps are raw, pixel-wise classifi-

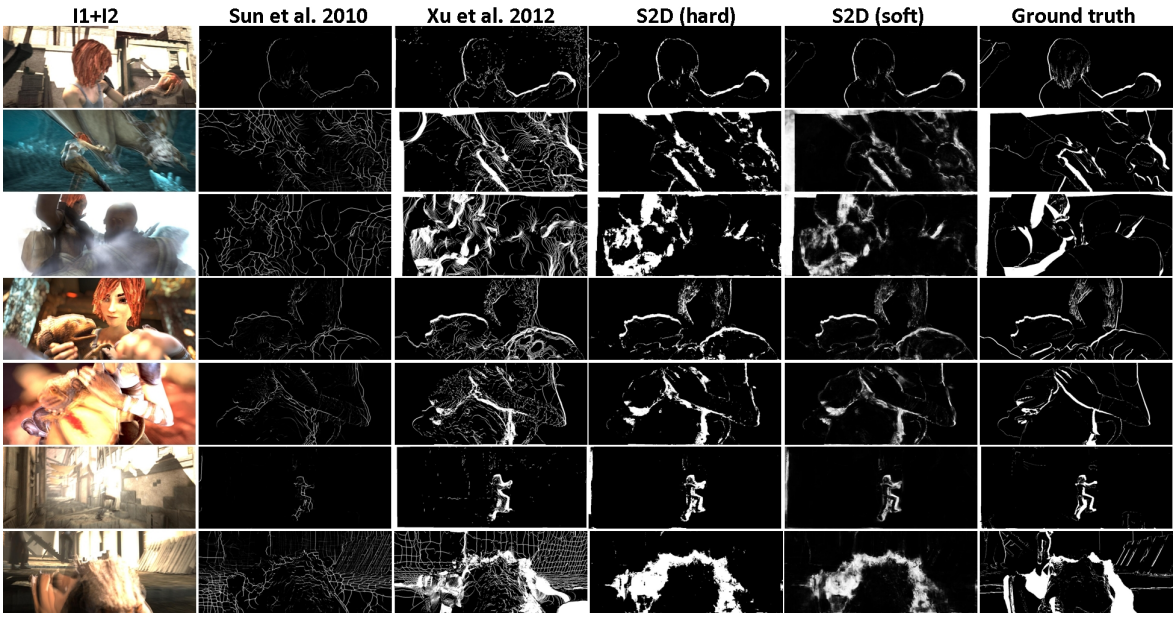


Figure 3. Occlusion estimation representative results. Our method is better suited to find large occluded regions.

cations, without any post-processing or (MRF) smoothing. Most methods published for occlusion detection produce contours, not large occluded regions.

**Implementation aspects:** For sparse matching, we sample points on a grid every 18 pixels for a total of 1425 features per Sintel image pair. They are patches on 3 Lab color channels and 1 channel for Gb's soft-boundary map[18]. During sparse matching initialization, we use $51 \times 51$ pixels patches, and $19 \times 19$ patches during optimization. The number of neighbors $N_i$ per point $i$ in its geometric neighborhood system $\mathcal{N}_i$ is minimum 21 (neighborhood radius of 45 pixels). The neighborhood size is adaptively increased, if necessary, until at least half of the neighbors considered are visible according to our occlusion map (maximum 60 neighbors allowed). The extended neighborhood $\mathbf{N}_i^{(E)}$ has twice the radius and four times the number of neighbors. This is very large compared to other current models. For the last step of continuous TV optimization we used the publicly available code for Classic-NL-fast [28]. Note that, by itself, Classic-NL-fast is ranked 7-th on the Sintel final benchmark, with an error that is on average 25% larger than when used, as a refinement step, in our sparse-to-dense method.

**Computation time:** Our method's total runtime on a Sintel image pair, in Matlab (3.2 GHz) is 38 mins. Runtimes per module: sparse initialization (4 mins), sparse optimization (0.5 mins), sparse-to-dense interpolation (13 mins), occlusion detection (12 mins), continuous refinement (8 mins).

Table 3. Occlusion detection results on 996 Sintel-TrainTest image pairs. For each method we present the maximum F-measure evaluated w.r.t. the ground truth occlusion regions available. We trained our classifier on a different set of 46 image pairs (Sintel-Train46).

| Alg. | S2D (Ours) | [32] | [28] |
|---|---|---|---|
| F-measure | **0.57** | 0.48 | 0.18 |

## 6. Conclusions

We proposed an efficient sparse-to-dense matching method for motion and occlusion estimation, using locally affine and occlusion aware constraints. Differently from the coarse-to-fine continuation approaches, we start from a sparse matching stage based on complex geometric relationships, and move towards a dense correspondence field based on a TV optical flow energy model. This natural decomposition of the dense matching problem produces state of the art results on the most extensive motion field benchmark to date. We also propose a novel occlusion classifier, with competitive performance, which works in good synergy with matching. Future work will investigate improved models with tightly integrated matching and occlusion reasoning components.

**Acknowledgements:**
This work was supported in part by CNCS-UEFICSDI under CT-ERC-2012-1 and PNII PCE-2012-4-0581.

## References

[1] L. Alvarez, R. Deriche, T. Papadopoulo, and J. Sánchez. Symmetrical dense optical flow estimation with occlusions

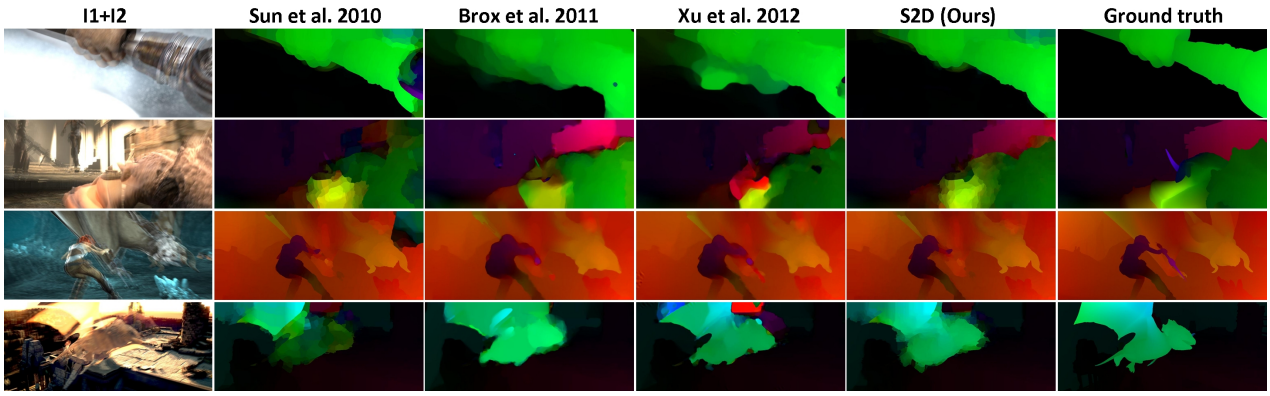


Figure 5. Motion field estimation results. Our algorithm can handle large displacements (in the sparse matching phase) while preserving motion details by TV continuous refinement at the last stage.

detection. In *ECCV*, 2002.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5), 2011.

[3] A. Ayvaci, M. Raptis, and S. Soatto. Sparse occlusion detection with optical flow. *IJCV*, 97(3):322–338, 2012.

[4] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005.

[5] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, 48(5):259–302, 1986.

[6] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *PAMI*, 33(3):500–513, 2011.

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

[8] O. Duchenne, F. Bach, I. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. In *CVPR*, 2009.

[9] X. He and A. Yuille. Occlusion boundary detection using pseudo-depth. In *ECCV*, 2010.

[10] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3), 1981.

[11] A. Humayun, O. M. Aodha, and G. Brostow. Learning to find occlusion regions. In *CVPR*, 2011.

[12] S. Ince and J. Konrad. Occlusion-aware optical flow estimation. *Image Processing*, 17(8):1443–1451, 2008.

[13] H. Jiang, T. Tian, and S. Sclaroff. Scale and rotation invariant matching using linearly augmented trees. In *CVPR*, 2011.

[14] H. Jiang and S. Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, 2009.

[15] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, 2001.

[16] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV*, 2005.

[17] M. Leordeanu and M. Hebert. Efficient map approximation for dense energy functions. In *International Conference on Machine Learning*, 2006.

[18] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *ECCV*, 2012.

[19] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Semi-supervised learning and optimization for hypergraph matching. In *ICCV*, 2011.

[20] H. Li, E. Kim, X. Huang, and L. He. Object matching with a locally affine-invariant constraint. In *CVPR*, 2010.

[21] K. Lim, A. Das, and M. Chong. Estimation of occlusion and dense motion fields in a bidirectional bayesian framework. *PAMI*, 24(5):712–718, 2002.

[22] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *TPAMI*, 33(5):978–994, 2011.

[23] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, Kerkyra, Greece, 1999.

[24] B. Lucas and T. Kanade. An iterative image resistration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[25] M. Sargin, L. Bertelli, B. Manjunath, and K. Rose. Probabilistic occlusion boundary detection on spatio-temporal lattices. In *ICCV*, 2009.

[26] A. Stein and M. Hebert. Occlusion boundaries from motion: Low-level detection and mid-level reasoning. *IJCV*, 82(3), 2009.

[27] C. Strecha, R. Fransens, and L. V. Gool. A probabilistic approach to large displacement optical flow and occlusion detection. In *Stat. Meth. in Video Proc.*, 2004.

[28] D. Sun, S. Roth, and M. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[29] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[30] J. Weber and J. Malik. Robust computation of optical flow in a multiscale differential framework. *IJCV*, 14:67–81, 1995.

[31] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, 2009.

[32] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *TPAMI*, 34(9), 2012.