EasyFlow: Increasing the Convergence Basin of Variational Image Matching with a Feature-based Cost

Jim Braux-Zin[1], Romain Dupont[1], Adrien Bartoli[2], Mohamed Tamaazousti[1]

[1]CEA, LIST, Saclay, France
[2]ISIT, Université d'Auvergne/CNRS, Clermont-Ferrand, France

Abstract: Dense motion field estimation is a key computer vision problem. Many solutions have been proposed to compute small or large displacements, narrow or wide baseline stereo disparity, or non-rigid surface registration, but a unified methodology is still lacking. We introduce a general framework that robustly combines direct and feature-based matching. The feature-based cost is built around a novel robust distance function that handles keypoints and weak features such as segments. It allows us to use putative feature matches to guide dense motion estimation out of local minima. Our framework uses a robust direct data term. It is implemented with a powerful second order regularization with external and self-occlusion reasoning. Our framework achieves state of the art performance in several cases (standard optical flow benchmarks, wide-baseline stereo and non-rigid surface registration). Our framework has a modular design that customizes to specific application needs.

1. Introduction

Image matching is essential in problems such as object tracking, motion segmentation, camera localization, 3D reconstruction and non-rigid surface registration. Most image matching techniques belong to one of two independent categories: dense matching for small deformations and sparse matching for large deformations. In the general case, dense matching is often called optical flow estimation. The majority of current methods follows the Horn and Schunk model [22]. It consists in optimizing a direct local data term (*e.g.* brightness constancy), ensuring the similarity of appearance, associated to a regularizer (*e.g.* total variation), ensuring a global coherence of the flow field. State of the art techniques use variational optimization schemes [10] with robust [45, 48, 41] and higher-order [7, 34] regularizers. However, variational optimization depends on the extent of local convexity of the data term which may only hold for sub-pixel deformations. Coarse-to-fine approaches mitigate this issue but non global minima cannot be avoided and prevent the use of such techniques for large deformations. For large deformations, the most effective approach is feature matching. It involves three steps: *detection* [20, 27, 36] where salient features are extracted, *description* [27, 4] where they are associated with a distinctive vector, and *matching* by nearest neighbor search in descriptor space. Having fewer and more discriminative candidates makes the matching process much more reliable than dense approaches. The majority of existing work concerns keypoints [20, 27, 4, 36] but other features can be used such as line segments [44, 19]. Some attempts were also made to use rich descriptors for dense matching of dissimilar images [26, 42]. However they use discrete optimization which becomes prohibitively slow as the number of motion candidates grows. To overcome this issue, SIFT-Flow [26] works on heavily subsampled

images, producing a coherent but crude displacement field, and Daisy [42] is restricted to 1D stereo estimation. Those limitations prevent one from considering these approaches as a general framework. This complementarity between accurate but local dense matches and global but sparse matches is present in the whole spectrum of image matching applications. In rigid 3D deformation, there exists dense stereo matching [21, 28, 34] for narrow baseline and sparse reconstruction via triangulation of keypoint matches (Structure-from-Motion [40]) otherwise. Large-scale dense reconstruction is usually done in two distinct steps: sparse then dense reconstruction. For example, Furukawa *et al.* [15] densify an initial sparse reconstruction by generating and filtering 3D patches using multi-view constraints. Similarly, non-rigid surface matching is split in two categories. During non-rigid *surface tracking* [16, 17], the deformation is followed over time and the frame to frame changes are small enough to make dense variational approaches viable. On the contrary, non-rigid *surface detection* [30, 31, 43] consists in the direct estimation of the potentially large deformation between a flat template and a given image. Most state of the art methods use filtered feature matches to fit a deformation model, then refined by a dense method [31] with poor performance on low texture areas.

Our goal is to overcome the dense/local features separation and take the best of both worlds to formulate a dense method with a large convergence basin. However, ensuring the convergence of a combination of sparse and dense terms is far from trivial [49, 25, 50]. In a similar fashion to Brox *et al.* [8], we jointly optimize over a direct term and a feature-based term in a variational scheme, with a novel model focusing on flexibility and robustness. In section 2, we analyze related work using features for dense registration to better justify the choices made in our method, presented in section 3. Two main contributions enlarge the convergence basin: a comprehensive occlusion handling scheme and a novel feature-based term. By grounding it int *feature distances*, we make it compatible with keypoints and line segments, and easily extensible to other local features. We use a robust estimator to implicitly filter erroneous matches and a bilinear influence function to handle sub-pixel feature locations. The genericity of our approach is showed through two applications. The proposed method is first applied to rigid matching (section 4) and then applied to non-rigid surface deformation (section 5). The benefits of our approach are demonstrated quantitatively and qualitatively. We give our conclusion and discuss future work in section 6.

2. State of the art in feature-based priors for dense image matching

We summarize the different approaches proposed in the literature to answer the two main challenges that arise when using feature-based priors in dense image matching, namely the *densification* of a sparse feature-based constraint and resistance to erroneous matches.

2.1. Densification of a sparse feature-based constraint

In order to exploit feature matches for dense image matching, one needs a process to spread the influence of sparse matches over the whole image. We call this process densification of the matches. Three approaches can be broadly identified: model-based warp fitting, discrete optimization and coarse-to-fine processing.

2.1.1. Model-based warp fitting: The most basic approach to densify a sparse constraint is the interpolation of the matches to produce a continuous field. This makes the implicit assumption of a smooth, seamless displacement field, which is verified in specific situations (*e.g.* deformable surface registration with no crease [31, 43]) but not in general. A more generic model such as a

piecewise affine displacement field can be used [49, 25]. These approaches group the matches in coherent clusters, also called layers, whose boundaries are estimated from the discontinuities in the image. If the layers are correctly estimated it is possible to densely refine them, and iterate the layer segmentation and refining steps for better results. When successful, it produces accurate displacement fields with sharp discontinuities, but it is very sensitive to the segmentation process.

2.1.2. Discrete optimization candidates: [50] collects all sparse matches to generate a global list of 2D displacements. Those are then used as candidates in a discrete optimization. This step is integrated in a standard coarse-to-fine scheme with variational refinement. This combination gives some of the most accurate results of the literature, but has drawbacks. First, it is quite slow: it needs many matches to have a representative set of candidates and the complexity of the discrete fusion algorithm (QPBO [37]) grows linearly with this number. Second, the result of the fusion depends on the order in which the candidates are processed. Third, the most significant drawback of this approach is that by reducing the matches to a set of 2D displacements, it discards all localization information and is incompatible with non-point features.

2.1.3. Coarse-to-fine densification: [8] proposes to inject feature matches (points or regions) into a variational optical flow estimation by simply adding a new sparse term in the cost function. This term constrains the estimated displacement fields in the neighborhood of each feature. The authors of [8] highlighted the convenient behaviour of coarse-to-fine warping to spread the influence of the sparse term over the whole image without degrading the accuracy of the dense term. It performs similarly to simulated annealing: at coarse resolution, the image is smoothed and the dense term is weak while features are quasi-dense and strongly constrain the optimization. At finer resolutions, the trend is reversed: features cover a tiny area of the image, preserving the accuracy of the dense term.

2.2. Resistance to erroneous matches

The feature matches considered as inputs may contain erroneous matches. Descriptor distances may not suffice to detect them due to repetitive structures or ambiguities. There are two major ways to suppress their influence: explicit and implicit filtering.

2.2.1. Explicit filtering: Parametric image matching methods are based on a deformation model [5, 23] and are often adjusted by least squares optimization, yielding a high influence to erroneous matches. They depend on a preliminary explicit filtering step to get rid of such outliers, such as RANSAC [14]. The greatest challenge of those methods is the tuning of the classifier sensitivity to balance the proportion of false positives (erroneous matches classified as inliers) and false negatives (correct matches classified as outliers).

2.2.2. Implicit filtering: Implicit filtering consists in progressively diminishing the influence of erroneous matches during the optimization without an explicit inlier/outlier classification. LDOF [8] weighs each match with a fixed confidence measure based on descriptors. However, as explained above, a proper filtering step needs other information in addition to descriptors. True implicit filtering is usually based on M-estimators, non-convex and *redescending*: their influence (derivative) first grows with the error and then vanishes. With the L^1 convex pseudo-norm used by LDOF, outliers have the same influence as inliers. The authors of [8]believe this is the main explanation of the LDOF limitation to small-baseline image pairs.

[30] uses a specific redescending estimator for non-rigid surface registration. Its selectivity is gradually increased during optimization for a behaviour similar to simulated annealing. One must distinguish this approach from LDOF, though it also behaves like simulated annealing: [30] optimizes the filtering of erroneous matches while LDOF optimizes the fusion of sparse and dense costs. The two approaches are actually complementary and we propose in section 3.2 to use both with improved results.

3. Proposed method

We propose an approach to upgrade most variational methods estimating dense displacement fields by minimizing a dense cost function defined as follows:

$$C_{\text{base}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_1} D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\mathrm{d}\boldsymbol{q}. \tag{1}$$

where $\boldsymbol{q}$ is the 2D vector of pixel coordinates in the image, $\boldsymbol{u}$ the vector of displacement associated to pixel $\boldsymbol{q}$, and D a dissimilarity term such as the intensity difference between $I_1(\boldsymbol{q})$ and $I_2(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}))$. Spatial coherence must be ensured, either with a parametric model or non-parametric regularization.

We bring two main contributions allowing us to greatly enlarge the convergence basin. First, we propose an explicit handling of occlusions, necessary when processing wide-baseline image pairs. Second, we propose a new feature-based term, compatible with non-point features (*e.g.* line segments), fractional coordinates and implicitly filtering erroneous matches. The image pair displayed in figure 3, with line segment correspondences, will be used in the next sections to illustrate the different properties of our method.

3.1. Occlusions

The displacement field is defined over the whole image domain Ω_1. In other words, all pixels of I_1, even those with no correspondence in I_2 (said *occluded*) are associated to a displacement vector. Such occluded pixels must be explicitly handled to prevent them degrading the whole field. Occlusions can be separated in three classes: 1) self-occlusions (see figure 1) occur when a part of the observed scene (present in the two images) fully or partially hides another one: common instances are changes of perspective point of view and folding deformable surfaces ; 2) field of view can behave like occluding elements because the image domain is finite and some parts of the scene may be unmatchable when the point of view changes ; 3) external occlusions (figure 1) are caused by an occluding element present in only one image.

To handle occlusions, we upgrade the base cost function (1) to:

$$C^{\star}_{\text{base}}(\boldsymbol{u}, I_1, I_2) = \iint_{\Omega_1} D^{\star}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\mathrm{d}\boldsymbol{q} \tag{2}$$

where:

$$D^{\star}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) = \mathcal{P}_{\theta_s}(\boldsymbol{q}, \boldsymbol{u})\delta_{\Omega_2}(\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}))\min(\theta_e, D(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)) \tag{3}$$

where D is the base dissimilarity term, $\mathcal{P}_{\theta_s}$ is the per-pixel self-occlusion probability, δ_{Ω_2} is the indicator function of the I_2 image domain and θ_e a threshold to reduce the sensitivity to external occlusions. All those factors are explained in detail directly abelow.

Fig. 1. *Illustration of self-occlusions (green) and external occlusions (yellow).*

3.1.1. Self-occlusions: Self-occlusions are characterized by the fact that all pixels neighbouring the occluded area are matched to a single 1D boundary, called *occlusion boundary*. As a result, the self-occluded pixels are all constrained to also be matched to this occlusion boundary, and the derivative of the warp vanishes along the normal of the boundary[1]. This enables an accurate local test to detect self-occlusions: we use a slightly modified version of the method from [17], which we summarized here.

Given the warp function defined as $\mathbf{a} : \boldsymbol{q} \mapsto \boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q})$, the pixel $\boldsymbol{q}$ is self-occluded if and only if the derivative of the warp vanishes in one direction[2], *i.e.*:

$$\exists \mathbf{d} \in \mathbb{R}^2, \|\mathbf{d}\| = 1 \quad \text{such that} \quad \nabla_{\mathbf{d}} \mathbf{a}(\boldsymbol{q}) = \mathbf{0} \quad \Leftrightarrow \quad \nabla_{\mathbf{d}} \boldsymbol{u}(\boldsymbol{q}) = -\mathbf{d} \tag{4}$$

where $\nabla_{\mathbf{d}} \boldsymbol{u}(\boldsymbol{q})$ is the directional derivative of $\boldsymbol{u}$ along $\mathbf{d}$ at $\boldsymbol{q}$, which may be approximated by $\frac{\boldsymbol{u}(\boldsymbol{q}+\mathbf{d})-\boldsymbol{u}(\boldsymbol{q}-\mathbf{d})}{2}$, the central-difference-based partial derivative. The smallest squared partial derivative, written σ_0, is linked to the Jacobian $\mathbf{J}$ of the warp by the following equation:

$$\sigma_0(\boldsymbol{q}, \boldsymbol{u}) = \min_{\|\mathbf{d}\|=1} \mathbf{d}^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u}) \mathbf{d} \tag{5}$$

where:

$$\mathbf{J}(\boldsymbol{q}, \boldsymbol{u}) = \begin{pmatrix} \frac{\partial a_x(\boldsymbol{q})}{\partial x} & \frac{\partial a_x(\boldsymbol{q})}{\partial y} \\ \frac{\partial a_y(\boldsymbol{q})}{\partial x} & \frac{\partial a_y(\boldsymbol{q})}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\partial u_x(\boldsymbol{q})}{\partial x} + 1 & \frac{\partial u_x(\boldsymbol{q})}{\partial y} \\ \frac{\partial u_y(\boldsymbol{q})}{\partial x} & \frac{\partial u_y(\boldsymbol{q})}{\partial y} + 1 \end{pmatrix}. \tag{6}$$

It follows that σ_0 is the smallest singular value of $\mathbf{J}(\boldsymbol{q}, \boldsymbol{u})$, *i.e.* the smallest eigenvalue of $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u}) = \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})^\top \mathbf{J}(\boldsymbol{q}, \boldsymbol{u})$. Eigenvalues are the roots of the characteristic polynomial:

$$\begin{aligned} \sigma \text{ eigenvalue} \quad &\Leftrightarrow \quad \det\left(\mathbf{O}(\boldsymbol{q}, \boldsymbol{u}) - \sigma \mathbf{I}\right) = 0 \\ &\Leftrightarrow \quad \sigma^2 - \sigma(O_{11} + O_{22}) + O_{11}O_{22} - O_{12}^2 \quad = 0, \end{aligned} \tag{7}$$

[1]This hypothesis is verified only if the warp is smooth in the occluded area. First-orders regularizer, such as the standard Total Variation can invalidate this property and create staircasing. Our implementation in sections 4 and 5 uses second-order regularization and is thus not affected by staircasing.

[2]the direction is perpendicular to the self-occlusion boundary

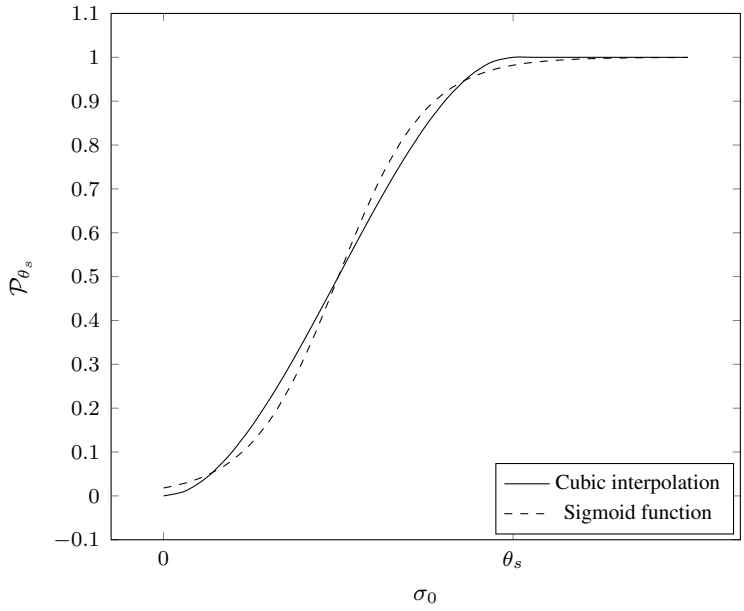


Fig. 2. *Mapping the smallest eigenvalue σ_0 of the warp's Jacobian to a non-self-occlusion probability P_{θ_s}. Our proposed cubic interpolation – equation* (10) *– has only one parameter, compared to two for the sigmoid function $\mathcal{S}'$ originally used in [17] with $\mathcal{S}'(\sigma_0) = \frac{\exp(2k(\sigma_0 - r))}{1+\exp(2k(\sigma_0 - r))}$. See section 3.1.1 for more details.*

with the following solutions:

$$\sigma = \frac{1}{2}\left(O_{11} + O_{22} \pm \sqrt{(O_{11} - O_{22})^2 + 4O_{12}^2}\right), \tag{8}$$

where O_{ij} are the coefficients of $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u})$ with the dependency on $\boldsymbol{q}$ and $\boldsymbol{u}$ hidden for readability. The eigenvalues of the $\mathbf{O}(\boldsymbol{q}, \boldsymbol{u})$ matrix are all positive with the smallest defined as:

$$\sigma_0(\boldsymbol{q}, \boldsymbol{u}) = \frac{1}{2}\left(O_{11} + O_{22} - \sqrt{(O_{11} - O_{22})^2 + 4O_{12}^2}\right). \tag{9}$$

In order to a non-self-occlusion probability from σ_0, we define a threshold θ_s and an S-shaped cubic interpolation function (see figure 2) as:

$$\mathcal{S}(x) = \begin{cases} 3x^2 - 2x^3 & \text{if } 0 \leq x \leq 1 \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

This function closely resembles a sigmoid and maps σ_0 to a value between 0 and 1, which we use to construct the non-self-occlusion probability as:

$$P(\boldsymbol{x}, \boldsymbol{u}) = \mathcal{P}_{\theta_s}(\boldsymbol{x} \text{ non-occluded} \mid \boldsymbol{u}, \theta_s) = \mathcal{S}\left(\frac{\sigma_0(\boldsymbol{x}, \boldsymbol{u})}{\theta_s}\right). \tag{11}$$

3.1.2. Image boundary: The displacement field can obviously not be estimated for pixels of I_1 which would normally lie off the domain of I_2. This can be seen as an occlusion. For small displacements, considering that all pixels outside the image domain are black can be sufficient [45], but this simplification introduces significant errors in the presence of larger displacements. In the proposed method, these occlusions are handled explicitly by suppressing the influence of the data term. We use the following indicator function:

$$\delta_{\Omega_2}(\boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q})) = \begin{cases} 1 & \text{if } \boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q}) \in \Omega_2 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

For data terms using a neighborhood (*e.g.* the CENSUS distance [52, 34]), the indicator function value is zero as soon as one pixel of the neighborhood lies outside Ω_2. This was omitted from the previous definition to maintain readability.

3.1.3. External occlusions: External occlusions are caused by an occluding element. In this case, no hypothesis can be made about the behaviour of the warp in the occluded area. As a result, we choose to consider such occluded areas as generic erroneous data, without distinction from – for example – noise and blur. We increase the robustness of the cost function to such erroneous data by truncating the data term to a threshold θ_e, similarly to [29].

3.2. Proposed feature-based data term

We introduce a new feature-based data term. We adopt the coarse-to-fine densification (section 2.1.3) and an implicit filtering of erroneous matches with the Geman-McClure M-estimator: $\Psi_\sigma(x) = \frac{x^2}{x^2+\sigma}$ though any other M-estimator could be used. We also propose additional contributions to handle non-point features and non-integer coordinates, more specifically applied to the handling of line segment matches.

Given a set $\mathcal{F} = \left\{(\mathbf{f}_1^{(1)}, \mathbf{f}_2^{(1)}), \dots (\mathbf{f}_1^{(n)}, \mathbf{f}_2^{(n)})\right\}$ of n feature matches, our feature-based data term is:

$$C_{\text{feat.}}(\boldsymbol{u}, \mathcal{F}) = \iint_{\Omega_1} \sum_{(\mathbf{f}_1,\mathbf{f}_2)\in\mathcal{F}} F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_1, \mathbf{f}_2)\ \mathrm{d}\boldsymbol{q}, \tag{13}$$

where:

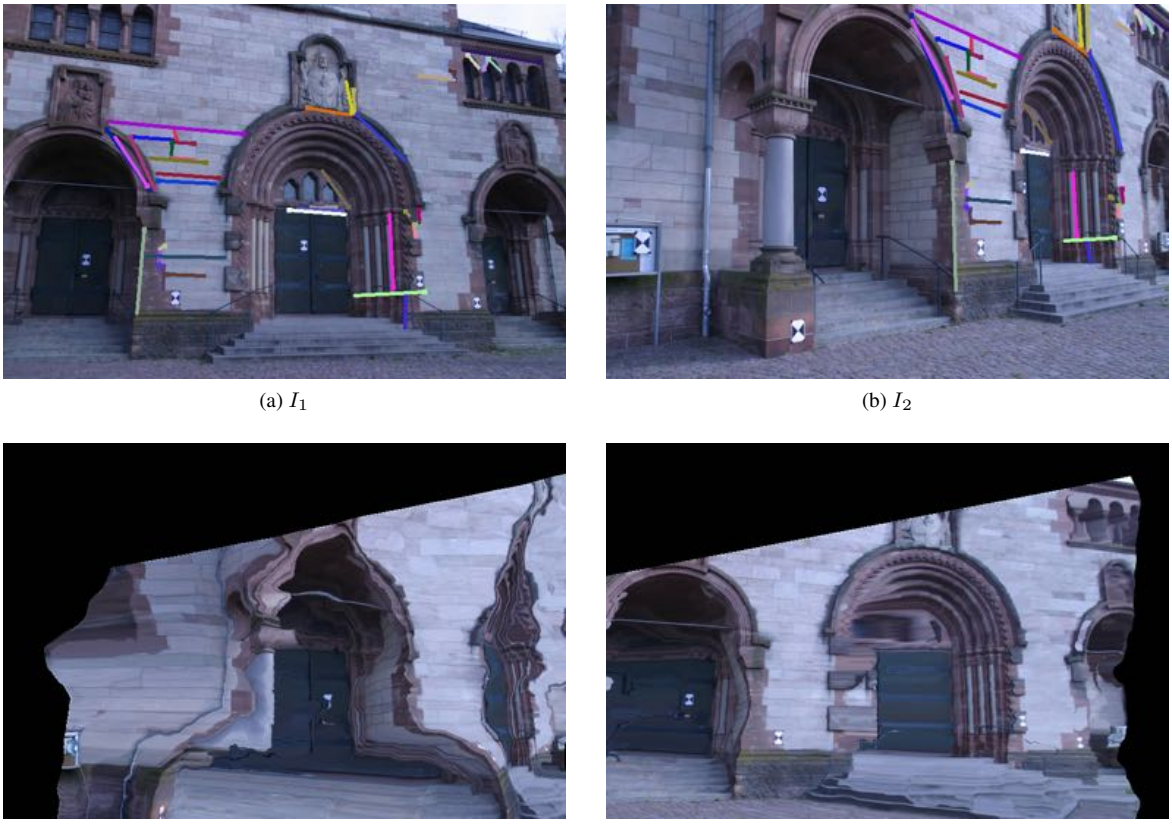
$$F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_1, \mathbf{f}_2) = \rho(\boldsymbol{q}, \mathbf{f}_1)\Psi_\sigma(\Delta(\boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q}), \mathbf{f}_2)) \tag{14}$$

is the per-feature data term. The influence function ρ and the point-primitive distance Δ are explained in the following paragraphs. Figure 3 illustrates qualitatively the expected gains for large displacement estimation.

3.2.1. Point-primitive distance: A challenge not addressed in the literature is the handling of non-point features. Indeed, given a feature $\mathbf{f}_1$ of I_1 matched to a feature $\mathbf{f}_2$ in I_2, most methods [8, 50] start by extracting the associated displacement $\mathbf{f}_1 - \mathbf{f}_2$, which is uniquely defined for points only. We adopt a more generic approach by considering an abstract point-feature distance $\Delta(\boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q}), \mathbf{f}_2)$ between the matches $\boldsymbol{q}+\boldsymbol{u}(\boldsymbol{q})$ of the pixels belonging to $\mathbf{f}_1$ and the corresponding $\mathbf{f}_2$ feature. We here give two distances – for points and segments – represented in figure 4.

The appropriate distance for points is the standard Euclidean distance. Given a point $\mathbf{f} = \boldsymbol{q}_f$:

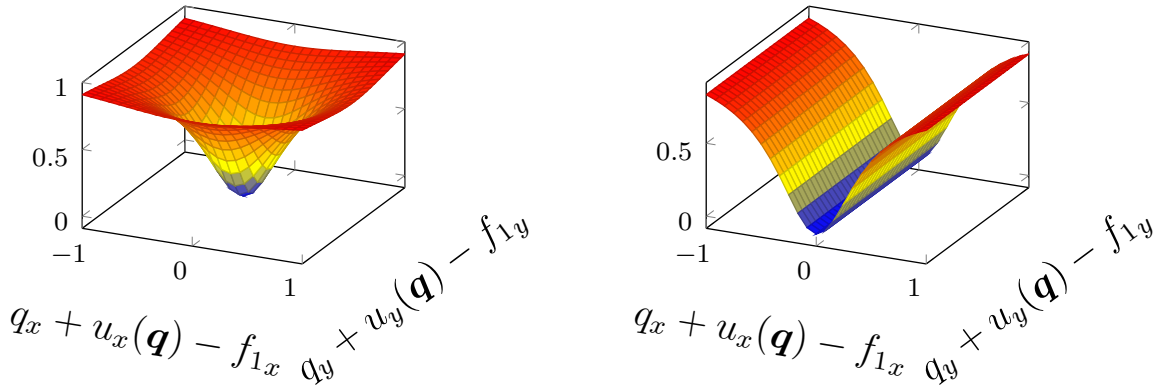
$$\Delta_{\text{point}}(\boldsymbol{q}, \mathbf{f}) = \|\boldsymbol{q} - \boldsymbol{q}_f\|. \tag{15}$$



(a) I_1 (b) I_2

(c) without sparse correspondences (d) with sparse correspondences

Fig. 3. *Top: two views of a rigid scene, extracted from the dataset [42], with line segment matches [44] (note that there is no mismatch). Bottom: I_2 image warped to I_1 using the estimated displacement field: (c) without and (d) with our feature-based term (see section 3.2).*



(a) Points

(b) Line segments

Fig. 4. *Point-feature distance with the Geman McClure estimator:* $\Psi_\sigma \circ \Delta$. *Plots represent the distances for a pixel* $\boldsymbol{q} = (q_x, q_y) \in \Omega_1$, *with a displacement* $\boldsymbol{u}(\boldsymbol{q}) = (u_x(\boldsymbol{q}), u_y(\boldsymbol{q}))$ *and two matched features* $(\mathbf{f}_0, \mathbf{f}_1)$.

Line segment matches, on the other hand, must lie on the same line but no point match (*e.g.* of the end points of the segments) is guaranteed. Because of occlusions and perspective deformations, predictible point matches are indeed rarely observed in practice. Thus, the best distance is the orthogonal point-line distance, constraining only one direction. Given a line segment defined by its end points $\mathbf{f} = (\boldsymbol{q}_{f_b}, \boldsymbol{q}_{f_e})$:

$$\Delta_{\text{segment}}(\boldsymbol{q}, \mathbf{f}) = \frac{\|(\boldsymbol{q}_{f_e} - \boldsymbol{q}_{f_b}) \times (\boldsymbol{q} - \boldsymbol{q}_{f_b})\|}{\|\boldsymbol{q}_{f_e} - \boldsymbol{q}_{f_b}\|}. \tag{16}$$

3.2.2. Bilinear spread function: Most features are localized with sub-pixel accuracy. The function ρ spreads the influence of features to the nearest neighboring pixels with bilinear weights. Given a feature $\mathbf{f}$ located at $\boldsymbol{q}_\mathbf{f} = \boldsymbol{q}_{0_\mathbf{f}} + \boldsymbol{dq}$ where $\boldsymbol{q}_{0_\mathbf{f}}$ is the non-fractional part of $\boldsymbol{q}_f$, we define $\overline{\boldsymbol{dq}} = (1,1)^T - \boldsymbol{dq}$ and the intermediate influence function ρ'_i for the four neighbors:

$$\begin{aligned} \rho'\left(\boldsymbol{q}_{0_\mathbf{f}}, \mathbf{f}\right) &= \overline{\boldsymbol{dq}}_x \overline{\boldsymbol{dq}}_y \quad \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (1,0)^T, \mathbf{f}\right) = \boldsymbol{dq}_x \overline{\boldsymbol{dq}}_y \\ \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (0,1)^T, \mathbf{f}\right) &= \overline{\boldsymbol{dq}}_x \boldsymbol{dq} \quad \rho'\left(\boldsymbol{q}_{0_\mathbf{f}} + (1,1)^T, \mathbf{f}\right) = \boldsymbol{dq}_x \boldsymbol{dq}_y. \end{aligned} \tag{17}$$

Line segments are first discretized into a set of points spaced by 1 pixel and the influence of each point is calculated as above. The resulting influence map is an anti-aliased representation of the segment. However, this would give a disproportionate weight to matches of long segments which are not necessarily more trustworthy. This issue is solved by normalizing the influence by the length of the segment, so that the sum of the weights is 1 for every feature. The corresponding equation is then, given $l(\mathbf{f})$ the length in pixels of the feature $\mathbf{f}$:

$$\rho(\boldsymbol{q}, \mathbf{f}) = \frac{\rho'(\boldsymbol{q}, \mathbf{f})}{l(\mathbf{f})} \quad \forall \boldsymbol{q} \in \Omega_1. \tag{18}$$

This formulation can be generalized by considering that points have a 1-pixel length. Different features, point or line segments, can then be mixed seamlessly.

3.2.3. Comparison with the state of the art: We stress the differences and the original contributions of the proposed method compared to the two closest approaches [30] and [8] from the literature.

[30] also uses an implicit filtering of erroneous matches, in the context of non-rigid surface registration. It relies on a purely feature-based matching, which is fast but discards much of the image information. In our method, the feature-based term only guides the dense estimation out of local minima and all the image information is exploited for optimal accuracy. Moreover, [30] uses a custom redescending estimator whose derivative is zero above a given threshold[3] potentially preventing convergence if initialized too far from the solution. Lastly, the use of coarse-to-fine processing allows us to use an M-estimator with a constant selectivity, while [30] must adjust it manually during the optimization, adding parameters to the algorithm.

We borrow the idea of multi-resolution densification from [8], but significantly upgrade their approach to overcome its limitations. First, we propose the use of a redescending robust estimator to allow the implicit filtering of erroneous matches, while [8] relies on descriptor matching scores. Those scores are inherently unreliable because they are based only on local data and can be fooled *e.g.* by repetitive patterns while our implicit filtering aims for a globally coherent motion. Moreover, only depending on the matches' positions has practical advantages: it is easier to change features and descriptors, and to thus reuse existing public implementations.

Finally, non-point features had received little attention in the image matching literature. We believe our formulation that handles line segments to be novel and to allow the use of more complex features such as areas, and curves. The flexibility and pertinence of our approach is validated by two implementations: a generic one with a non-parametric model and one dedicated to non-rigid surface registration with a parametric model.

4. Implementation and results with a non-parametric model

Our non-parametric implementation (see algorithm 1), conceived to be as generic as possible, is based on a dense ternary Census data term [52, 34] and a second-order Total Generalized Variation regularization [7, 34]:

$$C_{\text{non-param}}(\boldsymbol{u}, I_1, I_2) = \lambda \iint_{\Omega_I} D_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)\mathrm{d}\boldsymbol{q} + \mu C_{\text{feat.}}(\boldsymbol{u}, \mathcal{F}) + R_{\text{TGV}^2}(\boldsymbol{u}, \alpha_0, \alpha_1), \quad (19)$$

where $\lambda \in \mathbb{R}$ and $\mu \in \mathbb{R}$ are the respective weights of the dense and feature-based data terms.

4.1. Implementation details

4.1.1. Optimization: The algorithm from [10], relying on a primal-dual optimization with preconditioning [32], is used for optimizing the cost function (19). The practical implementation details of this algorithm are given in [6, 34]. This algorithm is iterative, we denote as i the number of iterations.

4.1.2. Linearization: The optimization algorithm expects a convex cost function while image-based terms are not. Convexity can however be assumed for small displacements, up to about

[3] The estimator of [30] is defined by: $\Psi_r(x) = \begin{cases} \frac{3(r^2-x^2)}{4r^3} & \text{if } x^2 < r^2 \\ 0 & \text{otherwise.} \end{cases}$

Data: Images I_1 and I_2, feature matches $\mathcal{F}$, dense data term D_{Census}
Result: Displacement field $\boldsymbol{u} : \Omega_1 \mapsto \mathbb{R}^2$

foreach *multi-resolution level* **do**
  for w *iterations* **do** /*registration of I_2 to I_1*/
    foreach *pixel* $\boldsymbol{q} \in \Omega_1$ **do**
      `/* external occlusions (section 3.1.3) */`
      $D^\star_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow \min(D_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2), \theta_e)$;
      `/* image borders (section 3.1.2) */`
      $D^\star_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow 0$ if $\boldsymbol{q} + \boldsymbol{u}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \notin \Omega_2$;
      `/* self-occlusions (section 3.1.1) */`
      $D^\star_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) \leftarrow \mathcal{P}_{\theta_s}(\boldsymbol{q}, \boldsymbol{u}) \cdot D^\star_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2)$;
      `/* summation of the feature-based term (section 3.2) */`
      $D_{\text{all}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2, \mathcal{F}) \leftarrow D^\star_{\text{Census}}(\boldsymbol{q}, \boldsymbol{u}, I_1, I_2) + \sum_{(\mathbf{f}_1, \mathbf{f}_2) \in \mathcal{F}} F(\boldsymbol{q}, \boldsymbol{u}, \mathbf{f}_1, \mathbf{f}_2)$;
      linearization of D_{all} wrt. $\boldsymbol{u}$;
    end
    for i *iterations* **do** convex optimization and regularization TGV^2;
    upscaling of the displacement field to next resolution level ;
  end
end

Algorithm 1: Summary of the non-parametric implementation.

one pixel. In practice, the data term is linearized at before each warp w and each update of the displacement field is restricted to a radius r to preserve the validity of the approximation. Like [47], we initialize r to 1 and divide it by 1.20 at each iteration to prevent oscillations.

4.1.3. Multi-resolution: The warps are repeated at each level of the image pyramid, from coarse to fine resolution. The original images are subsampled by a factor $s \in [0.5, 1[$. To ensure a maximum depth of this pyramid, we adopt the anisotropic scaling of [41]: the scaling factor is reduced for the smallest image dimension so that the coarsest level is a 4×4 square. All down- and upscaling use linear interpolation.

4.1.4. Parameter set: This method has been developed to be robust and generic. Thus a great care has been taken to propose a sensible set of parameters, listed in table 1. Only two parameters, λ and θ_s, need to be modified along the diverse proposed experiments.

The iteration numbers w and i, as well as the scaling factor s are taken from [34]. The 3×3 Census data term is preferred for a better robustness to distortions. We first learnt the parameters α_0, λ, θ_e and θ_s without the feature-based term to reduce the parameter space. Then we learnt only σ and μ with SIFT [27] matches. For each subset of parameters we repeated the following process:

1. particle swarm [12] global optimization of the average endpoint error on the 20 first image pairs with ground truth from the KITTI dataset, scaled by a 0.3 factor to speed up the evaluation,
2. rounding of the parameters to one significant digit to prevent overfitting,
3. evaluation on various sequences with ground truth, at full resolution.

Table 1 Parameter set for the matching method with non-parametric model.

	Small displacements	**Stereo**	**Non-rigid**
Regularization	TGV^2	—	—
α_0	1	—	—
α_1	1	—	—
Dense term	Census 3×3	—	—
λ	6	30	1
θ_e	0.5	—	—
θ_s	0.2	0.5	—
Features	SIFT/ASIFT	Line segments	SURF
μ	1	—	—
σ	0.2	—	—
Optimization	Chambolle-Pock [10]	—	—
s	0.8	—	—
w	20	—	—
i	40	—	—

The symbol — is used when the parameters are identical to the first column.

4.2. Experiments

Various experiments were made, first on small displacement datasets to quantitatively evaluate the gain brought by each of our contributions, then on more difficult datasets to demonstrate the widening of the convergence basin enabled by our approach.

4.2.1. Small displacements: We call small displacements problems the ones that can be reliably estimated by standard variational methods. This definition is broader than the one from LDOF [8]. Indeed, even if LDOF remains a reference in term of robustness, current variational methods have similar or superior performance. Our contributions aim at widening the convergence basin while preserving the accuracy. However, small displacement datasets are still useful to validate the choice of the dense data term and regularization. We will also check that the feature-based data term does not degrade the performance even when convergence is possible without it.

Three datasets with partially hidden ground truth are prominently used to evaluate optical flow algorithms:

- Middlebury [2] contains image pairs with tiny displacements, mostly piecewise constant, in controlled lighting environments ; it is mainly used, at the time of writing, to evaluate segmentation-based approaches ;
- KITTI [18] is a large dataset that covers optical flow, stereo and odometry. It was acquired in real conditions from an instrumented vehicle ; the diversity of contexts (urban, peri-urban, countryside), lighting conditions and displacement magnitudes makes it challenging, unforgiving to non-robust approaches ;
- Sintel [9] is a synthetic dataset generated from the realistic animated movie Sintel, showing a wide variety of motion and important perturbations such as blur, fog, smoke, snow and dirt.

Table 2 displays a detailed evaluation of the different components involved in the proposed algorithm. The mean error, in pixels, is measured over all the training datasets (with ground truth) of Middlebury and on the 40 first pairs of KITTI. To obtain a representative sample of the Sintel dataset (23 sequences of about 70 images), two consecutive images are randomly chosen in each sequence. For each dataset, we compare Total Variation (TV) and Total Generalized Variation (TGV^2), as well as the dense data terms absolute-differences and Census. Occlusion handling is

Table 2 Average error in pixels of our method on the Middlebury (top), KITTI (middle) and Sintel (bottom) datasets with different data terms: Absolute Differences (AD) and Census 3×3, and different regularizations: Total Variation (TV) and Total Generalized Variation (TGV). The dots mean "components of the preceding column".

Method on Middlebury	TV + AD	TGV^2 + AD	TV + Census	...+ threshold	TGV^2 + Census	...+ threshold	TV + AD + SIFT
error (pix)	0.6657	0.533	0.4154	0.4100	0.5864	0.5167	0.4196

Method on KITTI	TV + AD	TGV^2 + AD	TGV^2 + Census	...+ threshold	...+ self-occlusions	...+ SIFT
error (pix)	11.885	2.5395	1.5998	1.5353	1.4846	1.3686

Method on Sintel	TV + AD	TGV^2 + AD	TGV^2 + Census	...+ threshold	...+ self-occ	...+ RGB	...+ L*a*b*	...+ SIFT	...+ ASIFT
error (pix)	9.54	8.05	11.29	7.10	6.78	6.87	6.47	6.40	5.49

also evaluated: the threshold for external occlusions (section 3.1.3) and self-occlusion weighting (section 3.1.1). Color images also allow us to evaluate different color spaces.

The Middlebury dataset comprises mostly piecewise constant displacement fields and is thus *a priori* more suited to Total Variation regularization. However, the results of table 2 can appear incoherent at first. This is due in part to the fact that in this dataset displacement magnitudes are very small and the errors are thus not very significant, but also to the behaviour of the data terms. The discrete Census distance produces sharper cost variations than absolute difference, that are better handled by second-order regularization.

On the more realistic datasets, one can see that the combination of TGV^2 and thresholded Census gives the best results. The great influence of the threshold on the Sintel dataset is due to the presence of numerous perturbations (blur, fog, smoke...): the discrete nature of the Census descriptor can produce a high gradient in the affected areas and degrade the estimated displacement field. Thresholding limits the influence of outliers. Moreover the L*a*b* color space, constructed such that the Euclidean distance is a good indicator of colors, similarly to the human eye, is unsurprisingly the best candidate for color images.

Our feature-based data term is also evaluated. SIFT matches improve the accuracy of the small displacements of the Middlebury dataset. However, the more realistic KITTI dataset presents more local minima that can be avoided with features. The Sintel dataset also has local minima, but the difficult conditions make the SIFT matches not numerous enough and not reliable enough, so we opt for the more robust ASIFT [51] features. The improvements are then very significant, as displayed in the quantitative results and illustrated by figure 5.

Lastly, we compare in table 3 our method to the state of the art on the KITTI (with SIFT matches) and Sintel (with ASIFT matches) public benchmarks. Our algorithm ranks among the

Fig. 5. *Demonstration of the improvement from the feature-based data term on an image pair from the* `market_5` *sequence of the Sintel dataset. The large motion and the high level of blur make the estimation difficult. One should note that erroneous ASIFT matches, clearly visible, do not degrade the estimation.*

Table 3 Rankings on public optical flow benchmarks (November 2015).

	Sintel		KITTI		
Method	final[a]	clean[a]	pure[b]	processing time[c]	Avg-All [d]
EpicFlow [35]	1	1	6	15s	3.8 px
TriFlowFused (anon.)	2	2	18	350s	-
DeepFlow [46]	3	3	7	17s	5.8 px
IVANN (anon.)	4	4	14	1073s	-
Our method *(EasyFlow)*	**7**	**8**	**3**	**10+5s**	4.5 px

[a] The Sintel benchmark consists in two versions of the dataset: with (*final*) and without (*clean*) perturbations such as blur, fog and smoke.
[b] At the time of writing, the 6 top results from KITTI use additional information: stereo pairs, epipolar or multi-views constraints. We call *pure* the methods computing an unconstrained optical flow, like ours.
[c] Our processing time is split into sparse and dense matching.
[d] Average disparity / end-point error in total (when publicly available).

top ones on both benchmarks[4]. One should note that at the time of writing, the 6 first algorithms on KITTI took advantage of additional information: stereo pairs, epipolar or multi-view constraints, and cannot be directly compared to ours.

DeepFlow [46] is the only non-anonymous method with higher accuracy on the Sintel dataset. Their optimization scheme is very similar to LDOF but they use a novel method to get high-quality semi-dense matches. Moreover they optimize the parameter set for each dataset while we use the same for all evaluations. Even with this overfitting, the less-textured images and the limits of first-order regularization makes it inferior to our method.

4.2.2. Feature-based term: We experimentally observe the property of the feature-based term introduced in section 3.2. We consider an image pair (figure 6) generated by a synthetic 180 degrees rotation, preventing convergence of all feature-less coarse-to-fine based methods. To study the influence of features we use a set of perfect synthetic matches on a regular grid.

We start with figure 6b where we measure the average error with regard to the weight μ of the feature-based term with 256 perfect feature matches. A sharp transition appears clearly, confirming the hypothesis that feature matches are needed to enable convergence on this image pair. The

[4] In Sintel dataset (`http://sintel.is.tue.mpg.de/results`), our algorithm was named "AnyFlow" instead of "EasyFlow".

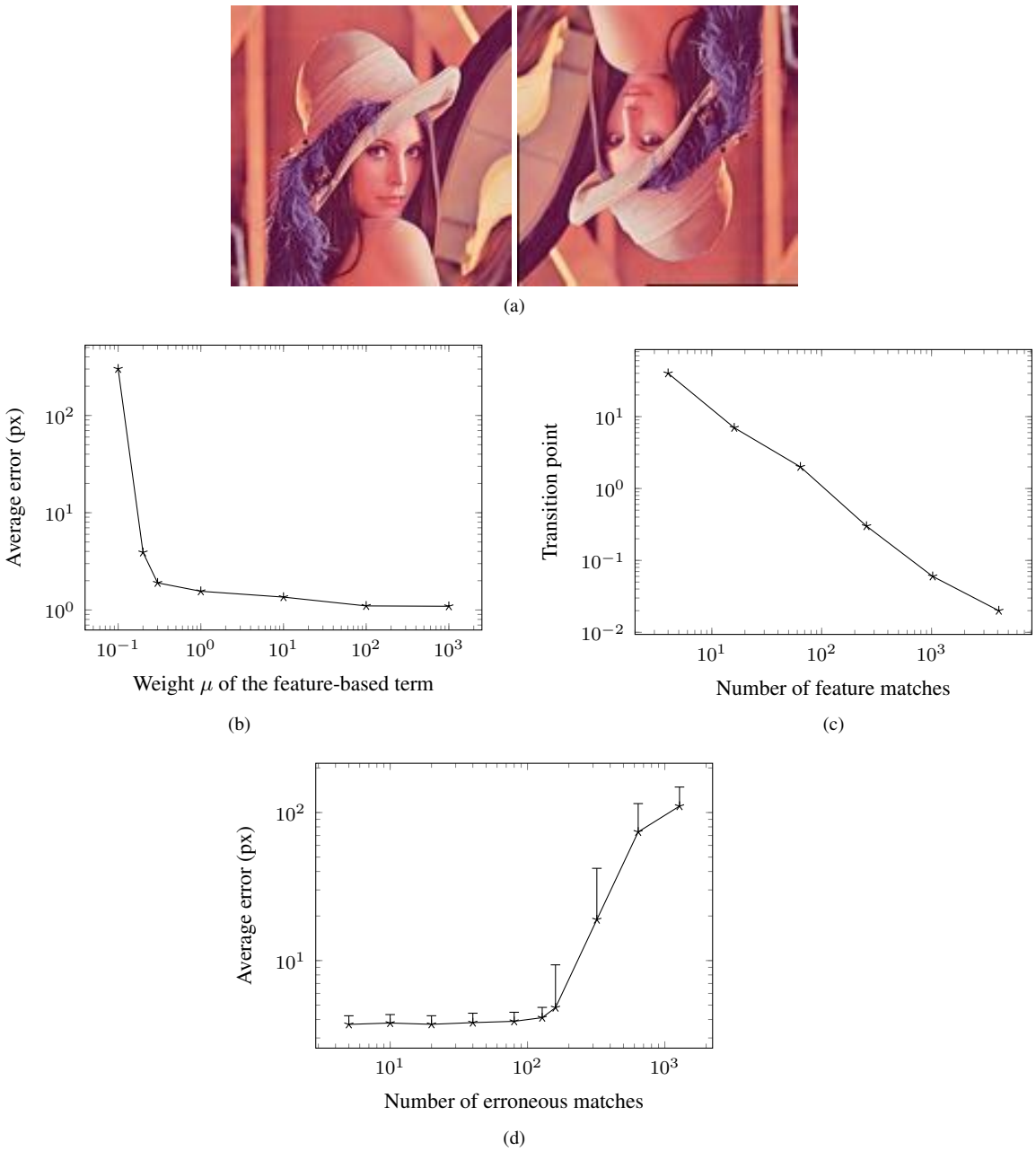


Fig. 6. *(a) The* Lena *pair, with a 180° rotation. (b) Influence of the weight of the feature-based term on the Lena image pair, with a regular grid of 256 perfect matches. We observe a sharp phase transition. (c) Evolution of the transition point with regard to the number of feature matches. The transition point is defined as the minimal weight μ enabling convergence. The two axes use a logarithmic scale. (d) Influence of erroneous matches. Erroneous matches are gradually added to 256 perfect matches. Points on the plot represent the average error over 100 runs and the error bars the standard deviation. The error is the average 2d euclidean distance between estimated matches and ground truth. Occluded pixels in the ground truth are ignored.*

sharpness of this transition demonstrates that the feature-based term guides the optimization out of local minima but have a limited influence on the end accuracy once convergence is assured. The influence of the number of features is evaluated in figure 6c: the minimal value of the weight μ to reach convergence is inversely proportional to the number of feature matches. This means that the global influence of features is proportional to the covered area (number of pixels) and explains the behavior of the coarse-to-fine densification (section 2.1.3).

We also measure the strength of the Geman-McClure implicit filtering in figure 6d. The average error is measured by gradually introducing random erroneous feature matches to 256 perfect ones on a regular grid. We conclude that the estimator is able to filter out about 200 erroneous matches without significant degradation, *i.e.* a rate of 44%.

4.2.3. Wide-baseline stereo: We focus now on the most important gain enabled by our approach: the significant widening of the convergence basin allows one to considerably extend the applicability domain of variational methods. In particular, we describe here its application to wide-baseline stereo.

The dense registration of two stereo images is a critical step of 3D reconstruction, and wide-baselines are preferred for two reasons. First, the wider the baseline is, the smaller the depth uncertainties are. Moreover, a wide baseline means that fewer images are needed to cover the same area, potentially saving storage and processing time.

However, registration is often much harder with a wide baseline. Indeed the occurrence of perspective distortion and light variations usually make photometric descriptors fail. Moreover, large image areas are impossible to match because they actually appear in only one image. Lastly, wide-baseline images are often not taken at the same time, and the larger the time difference is, the more chances there are that the static scene hypothesis is defeated.

Thus, few methods have been proposed in the literature, with the best results obtained by [42] with the DAISY descriptor. This descriptor is still photometric but covers a large image area, and is engineered for robustness and efficient dense computation. A discrete optimization (with an explicit labeling for occlusions) allows the author to estimate accurate depth maps with significantly wide baselines.

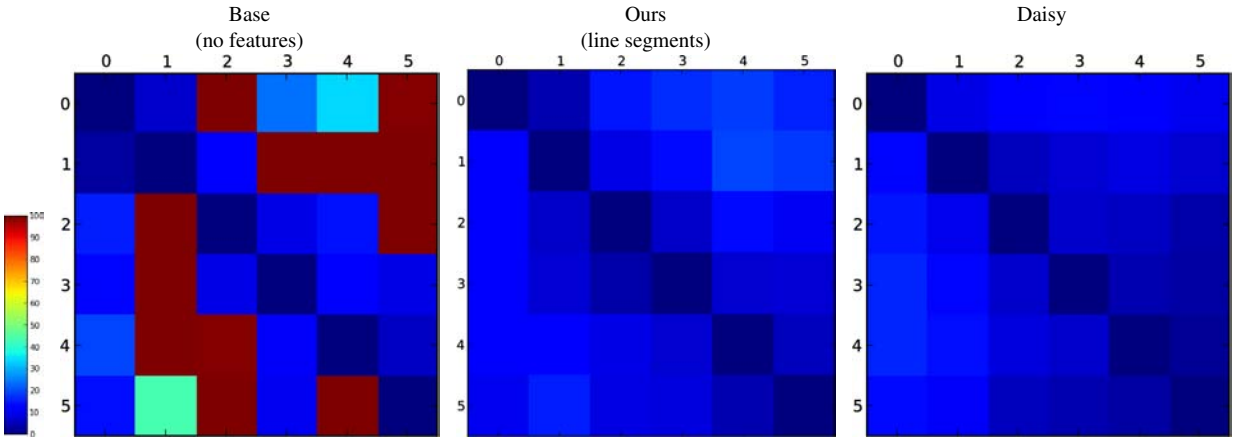
We try to reproduce their results on the `herzjesu` dataset by using line segment matches [44], robust to perspective distortions and lighting changes. We keep the Census data term but increase the threshold $\theta_s = 0.5$ (see table 1), slightly overestimating self-occlusions. This has the effect of reducing the influence of the Census data term on heavily slanted surfaces where it would be unreliable. This modification allows us to increase the overall dense data term weight to $\lambda = 30$ because the scenes considered are well-textured with no perturbations. In order to output a depth map from our optical flow algorithm we project the displacement vectors onto the epipolar lines along the optimization, then consider the displacement field as a set of matches that we can triangulate.

Table 4 quantitatively compares the base method (without using features), the proposed method and DAISY [42]. The results demonstrate that our feature-based term greatly enlarges the convergence basin, dividing the average error by almost five[5]. Moreover, our results are within 2% of DAISY's, proving that our approach can make standard generic methods in par with specific ones: DAISY optimizes over a discrete set of disparities and can only operate in one dimension while our method is still a variational optical flow estimation, free from those restrictions.

Two qualitative examples, with no available ground truth, further illustrate the capabilities of

[5]The error is defined as the amount of erroneous depth estimations, where a depth is considered as erroneous if the error with regard to the ground truth is greater than 5% of the total depth range, see Table 4.

Table 4 Amount of erroneous depth estimations on the `herzjesu` dataset. Like [42], we consider a depth estimate to be erroneous if the error with regard to the ground truth is greater than 5% of the total depth range. The first line is a color representation of the error for each image pair. The second line displays the average error over all pairs.



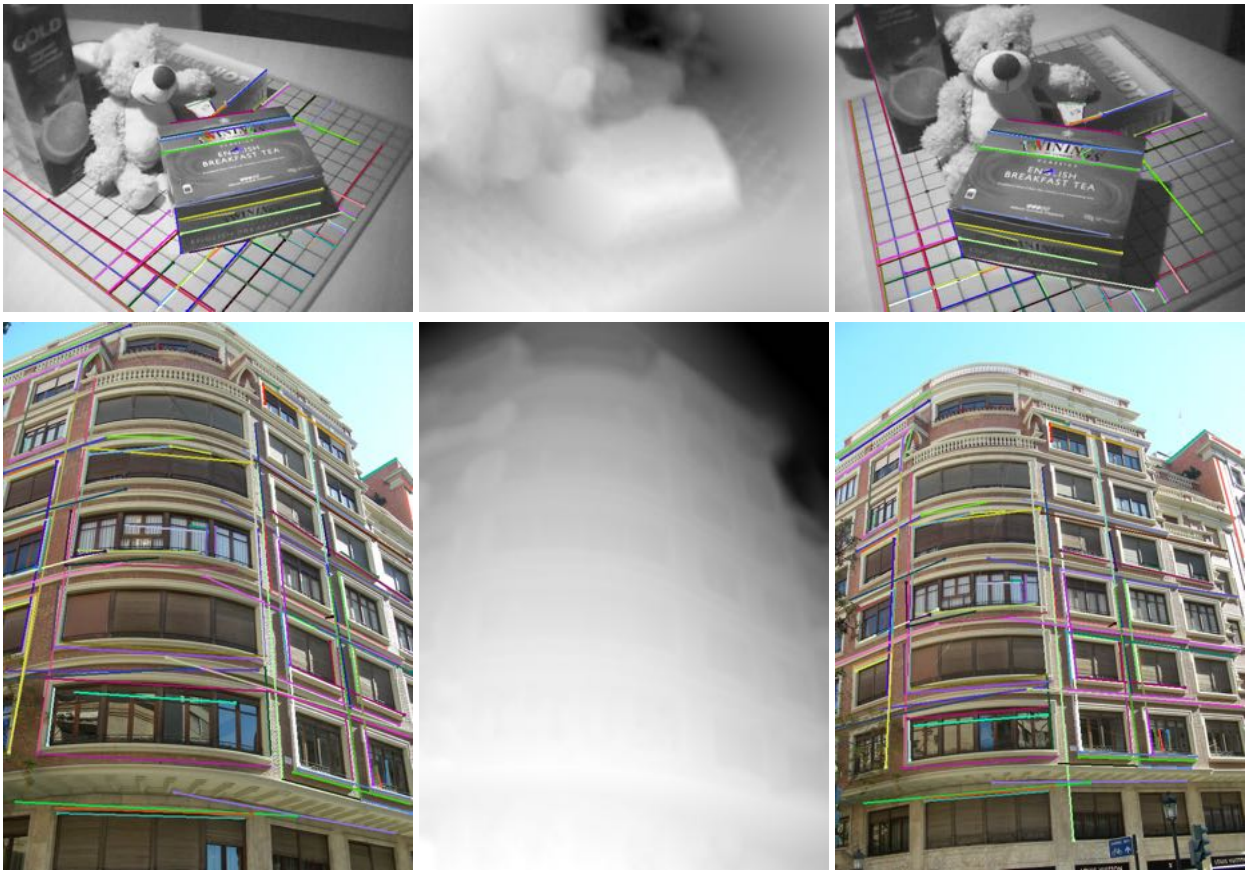


Fig. 7. *Depth maps estimated with our method. From left to right: reference image, depth map, second image. 81 and 111 line segment matches have respectively been used.*

our method in figure 7.

4.2.4. Non-rigid surface registration: Parametric model-based methods are usually more suited to the non-rigid surface registration problem (see section 5), but it constitutes a good challenge to prove the non-parametric approach generic through some qualitative samples in figure 8. Our results are compared to FBDSD [31], state of the art among the feature-based methods. We use the public C++ implementation from [1] reusing the same SURF [4] as input of our algorithm for a fair comparison.

The deformation is estimated by using the planar template as the reference image I_1 because the displacement field is then defined over the whole image domain. The displacement field is then inverted and applied to a colored grid, superimposed over the image of the deformed surface. The dense data term is subject to numerous local minima in the presence of such large non-rigid deformations, so we reduce its influence with $\lambda = 1$ (see table 1). To better observe the influence of the dense term on the accuracy, we do one estimation while suppressing its influence with $\lambda = 0$.

Results show that our feature-based term allows one to upgrade standard variational methods for non-rigid registration. The deformations are well estimated and self-occlusions correctly handled. Even without the dense data term, our results are better than FBDSD [31] for the two first image pairs, demonstrating the effectiveness of implicit filtering. The dense data term still brings significant gains. Some defects can still be observed but are mostly caused by the regularization. Section 5 presents quantitative results for a dedicated method with a parametric deformation model and a well-suited regularization.

5. Implementation and results with a parametric model for non-rigid surface registration

In order to further demonstrate the generic aspect of our approach, we implement our feature-based term into the parametric model-based non-rigid surface registration method from [17]. The displacement field uses *Free-Form Deformation* [23] as deformation model where the parameters are the displacements D of control points:

$$\boldsymbol{u}: \quad (\boldsymbol{q}, D) \mapsto FFD(\boldsymbol{q}, D) - \boldsymbol{q} \tag{20}$$

where FFD is defined as in [23].

The cost function to minimize is:

$$C_{\text{GB}}(\mathcal{D}, I_1, I_2) = \underbrace{\lambda \iint_{\Omega_{I_1}} C^{\star}_{\text{AD}}(\boldsymbol{u}(\boldsymbol{q}, D), I_1, I_2)\mathrm{d}\boldsymbol{q} + \lambda_s C_{\text{shrinker}}{}^{\ddagger}(D) + R_{\text{bend.}}(D)}_{\text{[17]}} + \mu\, C_{\text{feat.}}(\boldsymbol{u}_D(\boldsymbol{q}), \mathbf{f}) \tag{21}$$

$R_{\text{bend.}}$ is the bending energy of the displacement field, defined by the equation:

$$R_{\text{bend.}}(\mathbf{a}) = \iint_{\Omega_{I_1}} \left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial xy}\right) + \left(\frac{\partial^2 \mathbf{a}(\boldsymbol{q})}{\partial y^2}\right)^2 \mathrm{d}\boldsymbol{q} \tag{22}$$

and computable directly from the field D [33].

[‡]The FFD deformation model can attain aberrant 2D configurations called *folds*. [17] add a dedicated *shrinker* term to penalize those.

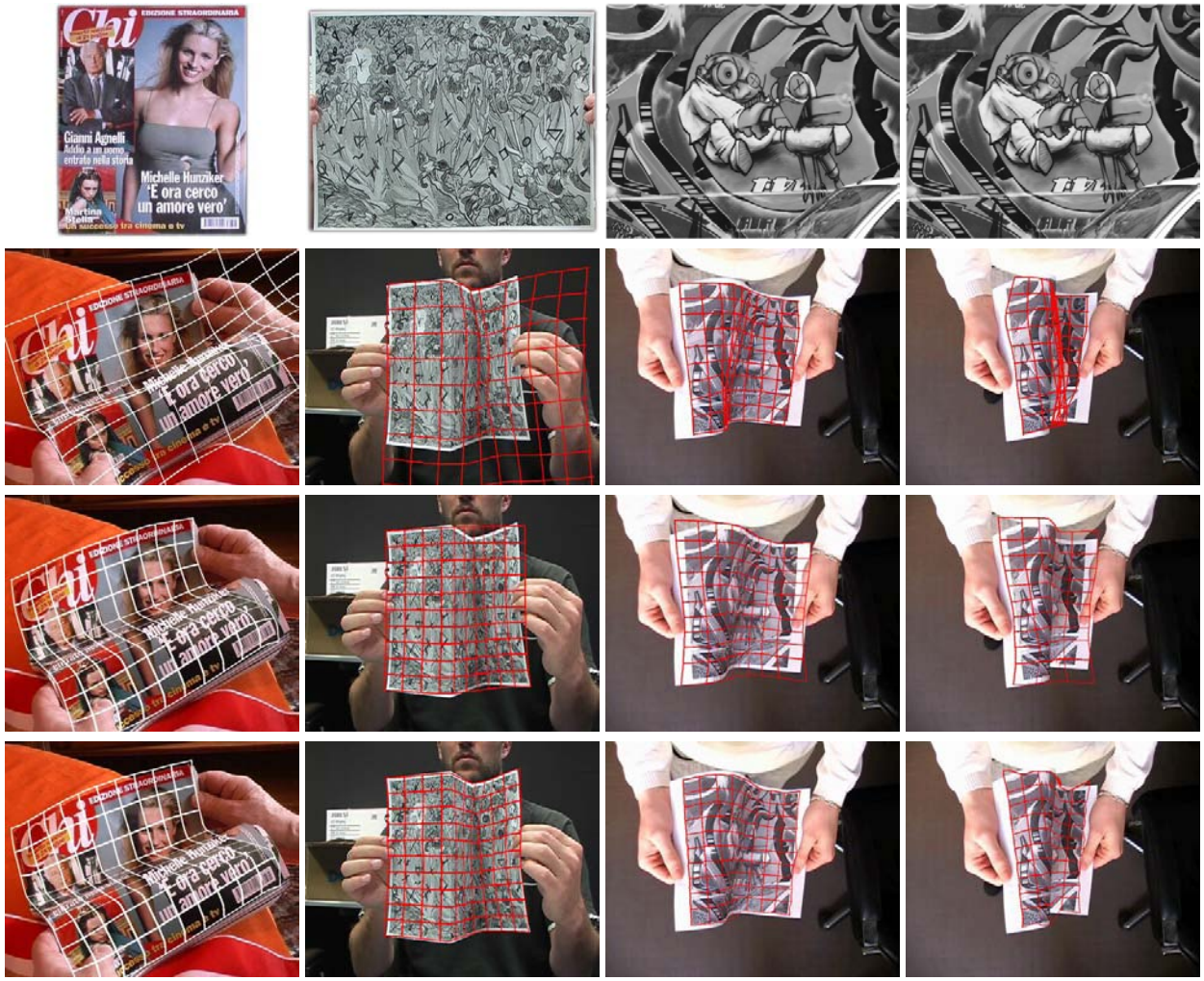


Fig. 8. *Non-rigid surface registration with a non-parametric model. From top to bottom: 1) planar templates, 2) FBDSD results [31], 3) our results without dense data term ($\lambda = 0$), 4) our results with $\lambda = 1$, and 5) the number of SURF matches used. From left to right, the image pairs come from the following publications: [13, 39, 17].*

5.1. Implementation details

Our feature-based term is integrated into a Matlab implementation from [17] with:

- Gauss-Newton optimization,
- 6 multi-resolution levels with an isotropic scaling factor $s = 0.5$,
- control points on a grid separated by a step $\varepsilon = 5$ pixels,
- the following weights: $\lambda = 2 \cdot 10^{-4}$, $\lambda_s = 20$ and $\mu = 0.16$ (paragraph 4.1.4 describes the method used to fix the different weights).

5.2. Experiments

The modified method with our feature-based data-term is compared to the original method [17], and to two feature-based methods with explicit feature filtering [31, 43]. These methods use a model to explicitly remove feature matches considered to be outliers. An FFD deformation model is then fitted by least squares optimization to the remaining matches. Qualitative results in figure 9 show our approach to be competitive.

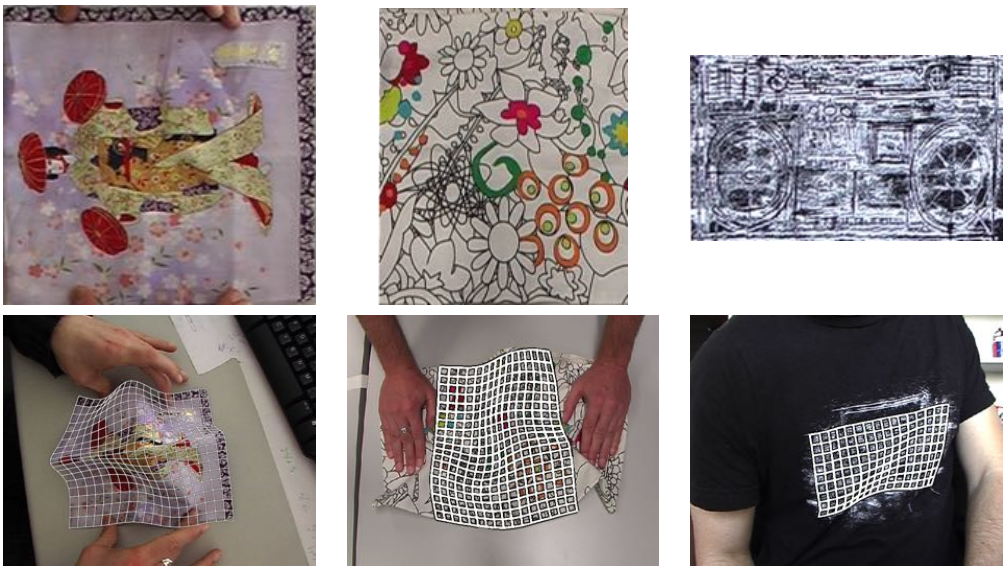
5.3. Real sequence

No public datasets with ground truth displaying sufficiently large non-rigid deformations were found at the time of writing. To enable a quantitative evaluation of the method, we thus use the `Graffiti` sequence from [17], and use their tracking (frame-to-frame) results as ground truth. Then, using a set of SIFT matches, we compare our approach to the feature-based ones from [43, RANSAC], and [31, FBDSD], and a variant of the latter, refined at the end with [17, FBDSD+P]. Estimations are computed between the first image of the sequence and directly with each of the other frames, with no tracking (the displacement field is initialized to zero for each pair). The results displayed on figure 10, clearly show that our upgrade does not bring any significant degradation for small deformations and outperform by far the other methods for the most challenging deformations.

6. Conclusion

We presented a generic approach to upgrade any variational image registration method and greatly enlarge its convergence basin through the addition of a feature-based term and an explicit handling of occlusions. A robust estimator enables implicit filtering of the feature matches and shields the end result from the influence of mismatches. The feature-based term also significantly improves the robustness by avoiding local minima while preserving the accuracy of the underlying dense variational method. To our knowledge, no other method is at the time of writing able to obtain top results on public optical flow benchmarks while being suited without any modification to wide-baseline matching and image registration with large non-rigid deformations.

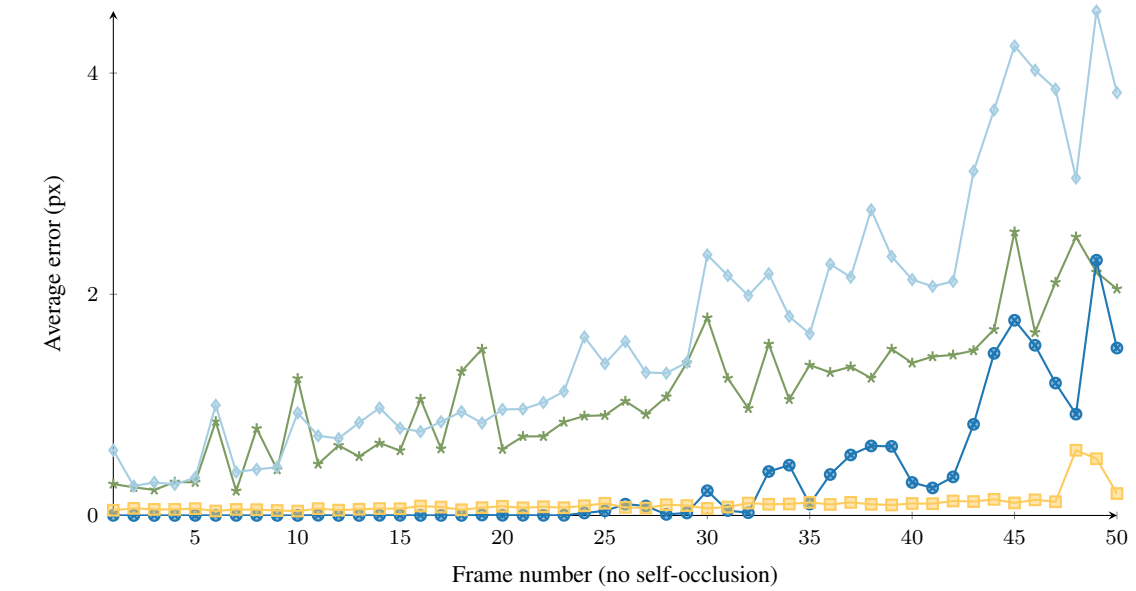
Each contribution has been conceived with the constraint to keep it as generic as possible: our method supports model-based or non-parametric base methods, points or line segments features (extensible to more feature types), and the occlusion handling mechanism has been validated for rigid scenes with perspective deformations as well as non rigid surfaces. Moreover, the coupling between components has been kept at a bare minimum, allowing to easily change the combination of data terms, regularization, feature types and optimization method. One illustration of this

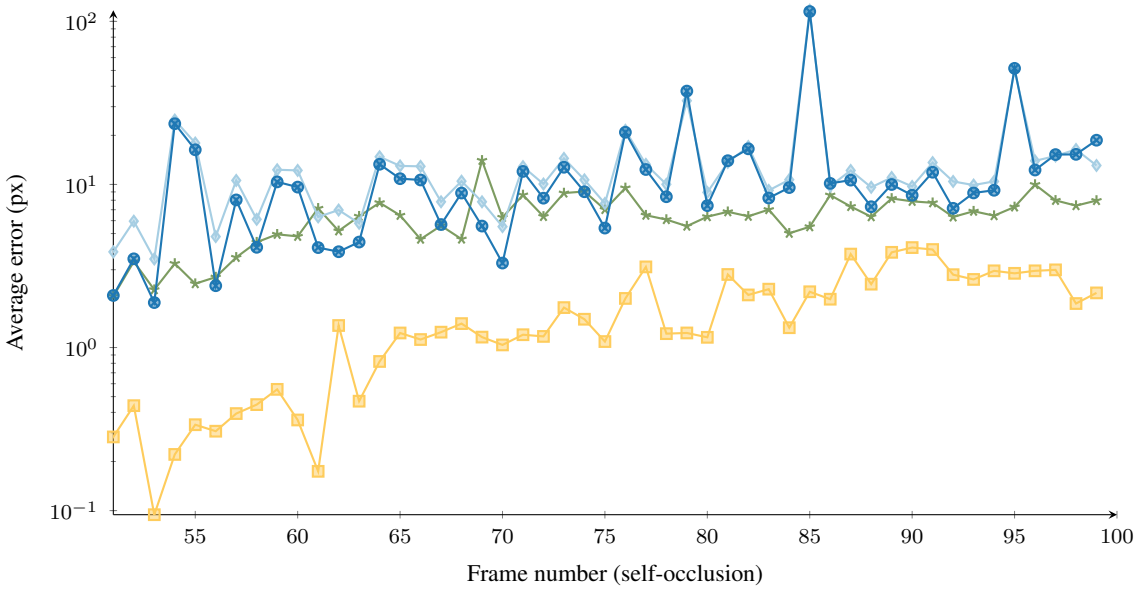


(a)

(b)

Fig. 9. *(a) Qualitative results on the [39, 38] dataset. First line: planar template, second line: estimated deformation. (b) Qualitative results on a particularly challenging image pair from the* `toys` *dataset [13]. From left to right: planar template, deformation estimated with FBDSD [31], deformation estimated with our method and the self-occlusion probability* $\mathcal{P}_{\theta_s}$ *(in white).*





Method	average error with no self-occlusions	with self-occlusions
RANSAC [43]	1.07	6.22
FBDSD [31]	1.59	14.06
FBDSD+P [31]	0.29	12.74
proposed method	0.10	1.60

Fig. 10. *Evolution of the FFD control points error. The ground truth was obtained by frame-to-frame tracking with the method from [17].*

flexibility is the fact that we were able to demonstrate our approach on two totally different implementations. For all these reasons, we call our approach *EasyFlow*.

We believe that the presented contributions can switch the scope of dense registration methods from domain-specific to cross-domain improvement of shared components. Future works will involve trying out new combinations, for example the semi-dense high-quality features from [46, *DeepFlow*], as well as curve or surface matches or more recent approaches based on convolutional neural network such as *FlowNet* [11]. The most limiting issue remaining is the handling of occlusion boundaries, that may be beneficial to explicitly detect [24]. Semi-random matching techniques from the *PatchMatch* [3] family could further reduce the influence of local minima. Another interesting way to explore is new applications enabled by the enlarged convergence basin, such as matching between different scenes like *SIFT-Flow* [26].

Acknowledgments

This research has received funding from the EU's FP7 ERC research grant 307483 FLEXABLE.

7. References

[1] P. Fernández Alcantarilla and A. Bartoli. Deformable 3d reconstruction with an object database. In *British Machine Vision Conference (BMVC)*, pages 1–12, 2012.

[2] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 2011.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), 2009.

[4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, 2006.

[5] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[6] K. Bredies. Recovering piecewise smooth multichannel images by minimization of convex functionals with total generalized variation penalty. *SFB Report*, 6, 2012.

[7] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 2010.

[8] T Brox and J Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

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

[10] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 2011.

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazrba, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks, Dec 2015.

[12] R. C Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Congress on Evolutionary Computation*, volume 1, pages 84–88. IEEE, 2000.

[13] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision (ECCV)*, pages 40–54. Springer, 2004.

[14] M. A Fischler and R. C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[15] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, 2010.

[16] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 2013.

[17] V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[18] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

[19] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. LSD: a Line Segment Detector. *Image Processing On Line*, 2:35–55, 2012.

[20] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, 1988.

[21] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 807–814, 2005.

[22] BKP K P Horn and B G Schunck. Determining optical flow. *Artificial Intelligence*, 1981.

[23] S. Lee, G. Wolberg, K. Chwa, and S. Shin. Image metamorphosis with scattered feature constraints. *Visualization and Computer Graphics*, 1996.

[24] M. Leordeanu, R. Sukthankar, and C. Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *European Conference on Computer Vision (ECCV)*, pages 516–529. Springer, 2012.

[25] M. Leordeanu, A. Zanfir, and C. Sminchisescu. Locally affine sparse-to-dense matching for motion and occlusion estimation. In *International Conference on Computer Vision (ICCV)*, 2013.

[26] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[27] D. G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[28] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *Third ICCV Workshop on GPUs for Computer Vision*, 2011.

[29] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences (SIIMS)*, 2014. Preprint.

[30] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, 2008.

[31] D. Pizarro and A. Bartoli. Feature-based deformable surface detection with self-occlusion reasoning. *International Journal of Computer Vision*, 2012.

[32] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *International Conference on Computer Vision (ICCV)*, pages 1762–1769, 2011.

[33] M. Prasad and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1345–1354. IEEE, 2006.

[34] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *Intelligent Vehicles Symposium (IV)*, 2012.

[35] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow. In *CVPR 2015 - IEEE Conference on Computer Vision & Pattern Recognition*, Boston, United States, June 2015.

[36] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 2010.

[37] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[38] M. Salzmann and P. Fua. Reconstructing sharply folding surfaces: A convex formulation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1054–1061. IEEE, 2009.

[39] M. Salzmann, R. Hartley, and P.d Fua. Convex optimization for deformable surface 3-d tracking. In *International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.

[40] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):835–846, 2006.

[41] D. Sun, S. Roth, and M. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[42] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[43] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S Brown, and D. Suter. In defence of RANSAC for outlier rejection in deformable registration. In *European Conference on Computer Vision (ECCV)*, pages 274–287. Springer, 2012.

[44] L. Wang, U. Neumann, and S. You. Wide-baseline image matching using line signatures. In *International Conference on Computer Vision (ICCV)*, 2009.

[45] A Wedel, T Pock, C Zach, H Bischof, and D. Cremers. An improved algorithm for TV-L 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, 2009.

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

[47] M. Werlberger. *Convex Approaches for High Performance Video Processing*. PhD thesis, Institute for Computer Graphics and Vision, Graz University of Technology, Graz, Austria, June, 2012.

[48] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and Horst Bischof. Anisotropic Huber-L1 optical flow. In *British Machine Vision Conference (BMVC)*, 2009.

[49] J. Wills, S. Agarwal, and S. Belongie. A feature-based approach for dense segmentation and estimation of large disparity motion. *International Journal of Computer Vision*, 2006.

[50] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2010.

[51] G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011.

[52] R Zabih and J Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision (ECCV)*, 1994.