S2F: Slow-To-Fast Interpolator Flow

Yanchao Yang Stefano Soatto
UCLA Vision Lab, University of California, Los Angeles, CA 90095
{yanchao.yang, soatto}@cs.ucla.edu

Abstract

We introduce a method to compute optical flow at multiple scales of motion, without resorting to multi-resolution or combinatorial methods. It addresses the key problem of small objects moving fast, and resolves the artificial binding between how large an object is and how fast it can move before being diffused away by classical scale-space. Even with no learning, it achieves top performance on the most challenging optical flow benchmark. Moreover, the results are interpretable, and indeed we list the assumptions underlying our method explicitly. The key to our approach is the matching progression from slow to fast, as well as the choice of interpolation method, or equivalently the prior, to fill in regions where the data allows it. We use several off-the-shelf components, with relatively low sensitivity to parameter tuning. Computational cost is comparable to the state-of-the-art.

1. Introduction

Most existing optical flow algorithms struggle with *small things that move fast*, even those explicitly designed for large displacement. This phenomenon does not have a dramatic impact on the benchmarks, since the problem being with small objects makes it such algorithms are not penalized too harshly. Nevertheless, small objects are important: humans can effortlessly pick out a bee flying at a distance.

In analyzing the root causes for the failure by most algorithms to capture small things moving fast, we honed in on a fundamental problem with classical scale-space, which trades off spatial frequencies (by blurring and down-sampling images) with temporal anti-aliasing (to compute temporal derivatives). This ties the *size* of objects to the *speed* at which they can move before being blurred-away in the multi-resolution pyramid that is routinely used in multi-scale/multi-resolution stages common to most variational optical flow techniques.[1] This multi-scale structure is also common in convolutional neural network architectures, so optical flow schemes based on them are typically subject to similar failure modes.

We propose a novel scheme for multi-scale matching, where the scale-space variable is not the amount of diffusion/subsampling of spatial resolution, but instead the size of the interest region on which local matching is based, at the native resolution. Thus, like others have done before, we perform *multi-scale without multi-resolution*. The iteration is instead over the radius of the region-of-interest, whereby regions with larger and larger radii operate on smaller and smaller subsets of the image domains. Slower objects are matched first, and then faster and smaller ones, hence the name S2F.

Clearly, the prior or regularization model plays a key role in optical flow. Rather than delegating it to a dataset and a generic function approximator, we discuss the specific model assumptions made in our method, and the topology with respect to which we consider pixels to be "nearby." In other words, *we hand-engineer the prior,* almost anathema in the age of Deep Learning.

Despite the absence of any learning, our algorithm achieves top performance in the most challenging optical flow benchmark, Sintel. More importantly, we can at least try to *explain* the performance, which we do in Sect. 3. Before doing so, we relate our work to the current art in Sect. 1.1, summarize the motivations and the actual algorithm in Sect. 1.3, and describe empirical tests in Sect. 2

1.1. Related work

Optical flow has been a core concern in Computer Vision for over two decades, with too many contributions to review here. It is a building block in many low-level vision tasks, and plays a role in a large number of applications, from autonomous navigation to video post-

[1] Combinatorial matching methods are not subject to this limitation.

production, only to mention a few. The interested reader can get an overview of recent developments in [18].

The case of fast motion has been tackled head-on in many recent works on *large-displacement optical flow*, for instance [4, 17, 1, 21, 25, 7, 2, 8, 30] and references therein. Several methods are proposed, mixing sparse matching with interpolation [26, 17], a philosophy we adopt. However, to the best of our knowledge, none addresses specifically the interplay of size and motion in multi-scale processing, and proposes an iteration that increases the region-of-interest, acting on a decreasing residual domain on the image. In particular, [21] addresses matching from small to large displacements, however it follows the standard scale-space of [4], and focuses on a novel descriptor inspired by sparse coding. Also, [26] learns a regularizer from the computed flow, which however follows a standard approach to scale-space. Both significantly underperform our method on the benchmarks.

Nevertheless, many of these methods are effective at capturing the fast motion of small regions, see for instance Fig. 10 of [4]. Our work follows these trends and makes a further step to improve results on fast-moving small objects (Fig. 1); [28] addresses the problem of lost details in the coarse-to-fine matching by not completely relying on the flow propagated from the upper levels.

Some have used coarse-to-fine matching that maintains the native resolution [16, 1, 13], or other multi-scale approaches in a combinatorial setting [21, 9]. Other samples of relevant related work include [24, 27, 3, 29, 6]. None of these works, however, perform multi-scale processing quite in the manner we do: Processing smaller and smaller regions that move faster and faster.

Our cost function is entirely hand-engineered to address known shortcomings and violations of the assumptions underlying the basic brightness constancy constraint. In part, this is because the phenomenology of correspondence is well understood, and therefore we are not compelled to learn it anew. To be fair, while phenomena like occlusions, scaling and domain deformations are well understood, the complex interaction of light and matter in non-Lambertian materials is difficult to model. This is where data-driven approaches such as [20, 22, 15, 11] have the most potential.

1.2. Summary of contributions and organization of the paper

We present *yet another algorithm for optical flow,* that focuses on the specific problem of coupling spatial and temporal statistics implicit in multi-scale/multi-resolution methods.

Our algorithm performs multi-scale inference by sequentially hypothesizing dense flow, and testing violation of the assumptions, on a shrinking domain, that is tested for increasingly large displacements.

When tested on benchmark datasets, our algorithm performs competitively. At the time of writing, it was the top performer on Sintel [5], which includes several examples of large displacement of small structures. It ranks middle-of-the-pack on Middlebury [19], which however is a very limited benchmark with only 12 image pairs, only 8 of which with ground truth. Interestingly, the only image pairs with large displacement of small objects are the four with no ground truth, which are therefore not part of the evaluation score. We also test on KITTI [12], where our approach is competitive despite no fine-tuning to the dataset being performed.

In the next section, we describe our approach in summary, then report empirical tests in Sect. 2 to show *how* it works, and in Sect. 3 we venture an explanation of *why* it works.

1.3. Rationale and underlying assumptions

Given two (grayscale) images $I_1, I_2 : D \subset \mathbb{R}^2 \to \mathbb{R}^+$, optical flow is a map $w : \mathbb{R}^2 \to \mathbb{R}^2$ defined at points $x \in D \subset \mathbb{R}^2$ implicitly by $I_1(x) = I_2(w(x)) + n(x)$, where $n(x)$ is an uninformative (white) residual. *Optical flow* is related to *motion field* (the projection of the displacement of points in space when seen in I_1 and I_2 [23]) under several assumptions on the scene around the (pre-image) point $X \in \mathbb{R}^3$ of $x \in D$, including: (i) Lambertian reflection and constant illumination, (ii) co-visibility. When (i) is violated, there is in general no relation between optical flow and motion field. When (ii) is violated (occlusion) there exists *no* transformation w mapping x in image I_1 onto a corresponding point in image I_2. When w exists, it may not be unique, *i.e.*, (iii) flow can be non-identifiable, which happens when the irradiance ("intensity") is not *sufficiently exciting* (*e.g.,* constant). This issue is usually addressed via regularization, by allowing a prior to fill in the flow from sufficiently exciting areas. A final assumption that is not necessary but common to many algorithms, is (iv) *small displacement* $w(x) \simeq x$. This allows using differential operations (regularized gradient) that facilitate variational optimization. This issue is not present in a combinatorial setting, where any large displacement is allowed, but at a prohibitive computational cost. In the variational setting, the issue is usually addressed via *multi-scale* methods, where temporal anti-aliasing is performed by spatial smoothing, through the creation of *multi-resolution* image pyramids (smoothed and sub-sampled versions of

an image [14]), where large displacements at fine-scale correspond to small displacements at coarse-scale.

Small things moving fast

There is a fundamental problem with multi-scale approaches based on classical scale-space, in that it couples *spatial* and *temporal* frequencies. In other words, it ties the *size* of objects to their allowable *speed*. This is manifested in typical failure cases with *small things moving fast* (Fig. 1). In general, the size of an object and the speed at which it moves are independent, and they should be treated as such, rather than be coupled for mathematical convenience. How then to address the spatial variability of image velocity?

Multi-scale without multi-resolution

Our approach to avoid the pitfall of multi-resolution, while addressing the intrinsically space-varying scale of motion and respecting the assumptions underlying optical flow computation, is to design a method that is multi-scale but not multi-resolution.

It operates at the native resolution, using increasingly large regions-of-interest operating on a decreasing subset of the image domain. Instead of using *spatial blurring* as the scale parameter, it uses *speed*, or magnitude of displacement. This is the key to our method, and explains the name "slow-to-fast". The next section sketches a generic implementation of our algorithm, and subsequent sections detail our choices of components and parameters.

Sketch of S2F-IF

Call $\phi(x; w, I_1, I_2)$ the point-wise cost function used by any baseline optical flow algorithm, for instance $\phi(x; \hat{w}, I_1, I_2) = |I_1(x) - I_2(\hat{w}(x))|$, where we may omit some of the arguments when obvious from the context. Then:

1. Choose an initial radius $r > 0$;

2. Use a *baseline optical flow* algorithm to compute putative forward $\hat{w}$ and backward $\hat{w}^{-1}$ displacements; point-wise residual ρ, where $\hat{w} = \arg\min_w \int_D \phi(x; w, I_1, I_2)dx$, $\rho(x) = \phi(x; \hat{w})$, and $\hat{w}^{-1} = \arg\min_w \int_D \phi(x; w, I_2, I_1)dx$. Also compute forward-backward (f-b) compatibility $b(x) \doteq \|I_{2\times 2} - \hat{w} \circ \hat{w}^{-1}(x)\|$.

   Test violations of (i) and (ii) using the residual $\rho(x)$ and f-b compatibility $b(x)$ respectively, aggregated on a region/window $\mathcal{B}(r)$ with radius r, using a conservative threshold.

   This leaves a (typically sparse) set of points $\mathcal{D} = \{x_i\}_{i=1}^{N(r)}$, and yields their (by assumption, typically small) displacements $w_i = w(x_i)$.

3. *Interpolate* the sparse matches to fill unmatched regions $D\backslash\mathcal{D}$ that violated (i)-(iv), based on a choice of prior/regularizer, leading again to a dense field $\tilde{w}$ and point-wise residual $\tilde{\rho}(x) = \phi(x; \tilde{w})$. Given flow at each point, check f-b compatibility after warping; large residuals are considered occlusions (violations of (ii)).

4. Optionally partition I_1 into piecewise constant regions $\{S_j\}_{j=1}^{M}$ (*super-pixels*), to facilitate computation, and expand $\mathcal{D}$ to include simply-connected regions with small residual $S_j \cap \chi(\tilde{\rho} < \epsilon_r)$.

5. Mask the matched regions $\mathcal{D}$ from the images, $I_1 \leftarrow I_1 \cdot \chi(D\backslash\mathcal{D})$, and similarly for the warped $I_2 \circ \tilde{w}$, where the dot indicates point-wise multiplication (matched regions are now black).

6. $r \leftarrow r + \delta$, and go to step 2. We use $\delta \geq 1$ pixels, and terminate when r reaches the size of the image, or no more matches could be found.

Several comments are now in order:

- We choose $r = 5, 8$ pixels in (1.) for KITTI and Sintel respectively as in [1]; we use [1] as a baseline optical flow in (2.), and the *census transform* to test compatibility with (i)-(ii). We reject points that fail either the residual ($\epsilon_r = 30$) or the f-b test($\epsilon_c = 1, 5$). We choose [17] for interpolation in 3, and [10] for superpixelization. Finally, we use $\delta = 1, 2$ pixels for the scale increment.

- Step 2 implements a conservative *sparse matching* procedure for regions of size r, that leads to a set of sparse matches. Our choice [1] can be replaced by any other conservative sparse matching.

- The matched region $\mathcal{D}$ typically grows monotonically, so the procedure either terminates with a non-empty unmatched set, if no further matches could be found, or each pixel is matched $\mathcal{D} = D$.

- In theory, the process should be terminated before each pixel is matched, as displacement is not defined in occluded region. In practice, all pixels are typically matched, exploiting the regularizer imposed by the interpolation step.

- The first regions of the scene to be matched are the ones that are (i) Lambertian, with (ii) sufficiently exciting radiance, are (iii) co-visible, and (iv) moving slowly. As iterations progress, smaller and smaller regions that are moving faster and faster are matched. For this reason, we call this scheme *Slow-To-Fast* (S2F) Interpolator Flow (IF), as the final solution is influenced heavily by the prior.

- The crucial characteristic of the algorithm above, which is responsible for edging the state-of-the-art, is its lossless multi-scale nature, that is the search at multiple scales of motion, without changing the resolution of the images.

- The algorithm is relatively insensitive to the choice of component algorithms at each step, although the most crucial is the choice of interpolation, which we discussed at in Sect. 3.2

2. Experiments

2.1. Qualitative results

Fig. 1 illustrates the key characteristic of our method in comparison to most alternate methods, which we choose to represent with a close-to state-of-the-art baseline [18]. Small objects that move fast are diffused away by scale-space by the time their displacement becomes small enough for a variational optical flow algorithm to resolve. Modifying spatial frequencies (smoothing and down-sampling) to achieve temporal anti-aliasing (to enable approximation of temporal derivatives with first differences) ties the *size* of objects with their *speed*, in ways that are detrimental. Our approach treats them as independent, thus enabling us to capture their motion. It should be mentioned that combinatorial search-base schemes are not subject to this limitation, but suffer from prohibitive computational complexity.

Fig. 2 illustrates the various stages of evolution of our algorithm, corresponding to the sketch in Sect. 1.3.

Fig. 3 shows the evolution of the matched domain, which typically shrinks monotonically to encompass the entire image domain, with the last, unmatched region filled in by the regularizer.

2.2. Benchmark comparisons

Fig. 4 shows representative samples for the benchmarks used. The Middlebury dataset [19] comprises 12 pairs of images of mostly static man-made scenes seen under a short baseline. There are few small objects, and none moves fast in the only 8 ground-truthed pairs. The only pairs showing large displacement of small objects

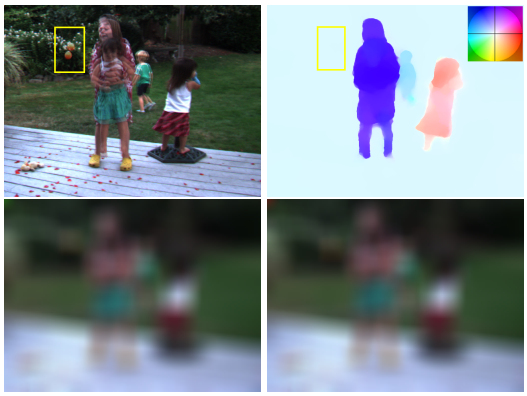

Figure 1. *Small things moving fast (top-left) two images from the Middlebury dataset (shown superimposed) with the fast-moving ball highlighted, are a classic failure mode of multi-resolution optical flow (top-right; the inset* color wheel *shows the map from color to image displacement). Small objects disappear at coarse resolution, where large motions are computed (bottom row), and are never recovered in a differential-based variational scheme [18] (top-right).*

Method	Avg. Rank	Method	Avg. Rank
CPM-Flow	53.7	EpicFlow	57.4
DeepFlow2	54.0	FlowNetS	80.4
S2F-IF	**38.6**	FlowFields	41.2

Table 1. Average endpoint error on Middlebury for the top-performing algorithms on Sintel. Full ranking can be accessed directly on the Middlebury flow page `http://vision.middlebury.edu/flow/eval/`.

are the 4 with no ground truth, including the one shown in Fig. 4, which are unfortunately not included in the evaluation. Our algorithm estimates flow more accurately on these sequences. In overall performance, our method ranks in the middle-of-the-pack on this dataset. As a sanity check, we use the Middlebury dataset to compare against the algorithms that report top performance on Sintel, which is a larger dataset showing a wider variety of motions, including large displacement of small objects. The results in Table 1, show our algorithms comparing favorably. The fact that top performers on Sintel are different from top performers on Middlebury suggests that one of the datasets, or both, are easily overfit. Middlebury only has 12 image pairs, only 8 of which with ground truth, none of them with large displacement.

A better benchmark is the KITTI dataset [12], which consists of outdoor driving sequences, with sparse ground truth. Quantitative comparisons with competing algorithms is shown in Table 2. We use default param-

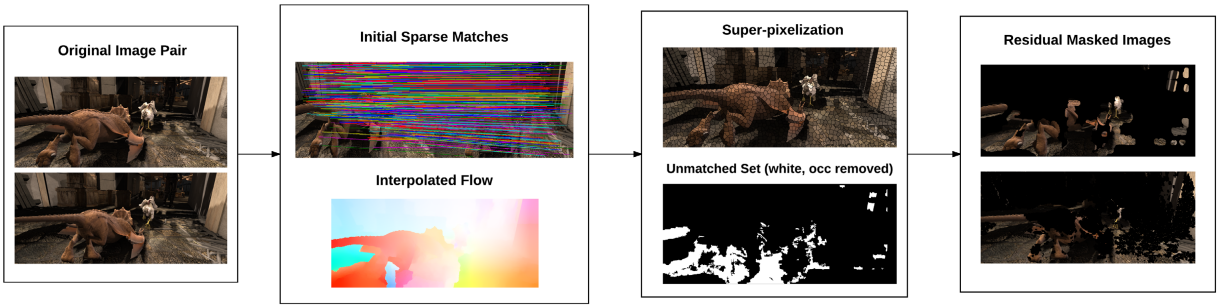

Figure 2. *Visualization of the stages of our algorithm: Original images (left), initial sparse matches (middle-left, step 2), interpolated flow (middle-left, step 3), super-pixelization (middle-right, step 4), matched set (middle-right, step 5) and residual masked image (right) after the first iteration.*

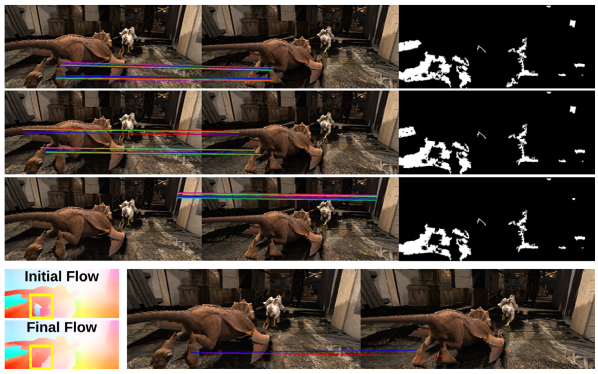

Figure 3. *Matched regions as the iteration evolves from the first (top row) to the last (bottom rows). The unmatched region (white) shrinks in size, until it converges to regions that are compatible with the hypotheses, but where there is no unique match (third row). On these, the regularizer has license to fill in (bottom), where we highlight details on the legs of the dinosaur, where the overall procedure corrects initial matching errors of the baseline flow algorithm.*

Method	Out-Noc	Out-All	Avg-Noc	Avg-All
CPM-Flow	5.79 %	13.70 %	1.3 px	3.2 px
EpicFlow	7.88 %	17.08 %	1.5 px	3.8 px
DeepFlow2	6.61 %	17.35 %	1.4 px	5.3 px
FlowNetS	37.05 %	44.49 %	5.0 px	9.1 px
FlowFields	5.77 %	14.01 %	1.4 px	3.5 px
S2F-IF	6.20 %	15.68 %	1.4 px	3.5 px
FlowField-	6.49 %	15.94 %	1.5 px	3.9 px

Table 2. Comparison on the KITTI dataset. Our method uses as a component `FlowField-` for flow computation. As expected, it improves its performance. The same algorithm, however, fine-tuned to the dataset (indicted as `FlowFields`, for which no parameters are disclosed) further improves performance. We do not fine-tune ours, and simply report our performance with the same tuning for all datasets. Out-Noc indicates the percentage of pixels with error larger than 3 pixels in non-occluded regions, whereas Out-All indicates percentage of outliers among all pixels. Avg denotes the average end-point error, again for non-occluded, or all pixels.

eters, not fine-tuned for the dataset, and show competitive performance. As expected, we outperform the baseline flow algorithm we use as a component, shown as the last line on the table as `FlowField-`. It should be noticed that the same algorithm has been fine-tuned to the KITTI dataset by the authors, shown on the table as `FlowFields`, with a considerable improvement in performance, suggesting that this dataset can also be overfit. Since the parameters chosen for the test are not disclosed, we use the same parameters of the baseline as released, with no fine-tuning for the dataset. We feel that this test is more representative than reporting the best score with different parameters for each dataset.

Again, we use the same settings as in [1] on the Sintel dataset [5], which is a synthetic one, but challenging in that it includes fast motion, motion blur, and has precise ground truth. We report the performance in the official benchmark in Table 3, with our algorithm exhibiting top performance in overall end-point error at the time of writing.

These results illustrate the benefit in specifically handling multi-scale phenomena without sacrificing resolution and confusing spatial statistics with temporal ones. Several representative sample results are shown in the Supplementary Material, and the up-to-date ranking on the benchmark can be verified on the Sintel website `http://sintel.is.tue.mpg.de/results`.

The next section gives more details on our choice of component methods for the generic algorithm described in Sect. 1.3.

Figure 4. *Representative samples from various datasets: Middlebury (row 1), KITTI (rows 2, 3, 4), SINTEL (rows 5,6). We compare the component flow [1] (FlowFields), with ours (S2F). More examples are shown in the Supplementary Material; Small objects moving fast are highlighted in the yellow box.*

3. Technical Details

The basic algorithm was described in Sect. 1.3, and consists of sparse matching, followed by interpolation, followed by testing for violation of the hypotheses, where the iteration is with respect to a growing radius for the region of interest, which operates on smaller and smaller residual unmatched portion of the image domain.

3.1. Sparse matching

Step 2 of our algorithm results in a sparse set of regions being matched over short displacements. This is not because we actively seek for sparse matches with small displacement. On the contrary, we start with a dense flow, specifically [1], but then conservatively reject all regions that fail hypotheses (i)-(ii) based on residual or f-b compatibility. This naturally results in a sparse set, because sufficient excitation conditions (which are tested through f-b compatibility) require large gradients in two independent directions, which is typically only satisfied on a sparse subset of the image domain. Conceptually, any other sparse matching would do, and the algorithm is not very sensitive to the choice of method for this step, which we therefore do not further discuss.

Method	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
FlowFields	5.810	2.621	31.799	4.851	2.232	**1.682**	1.157	3.739	33.890
FlowFields+	5.707	2.684	30.356	**4.691**	**2.117**	1.793	1.131	**3.330**	34.167
SPM-BPv2	5.812	2.754	30.743	4.736	2.255	1.933	**1.048**	3.468	35.118
FullFlow	5.895	2.838	30.793	4.905	2.506	1.913	1.136	3.373	35.592
CPM-Flow	5.960	2.990	30.177	5.038	2.419	2.143	1.155	3.755	35.136
EpicFlow	6.285	3.060	32.564	5.205	2.611	2.216	1.135	3.727	38.021
DeepFlow2	6.928	3.093	38.166	5.207	2.819	2.144	1.182	3.859	42.854
S2F-IF	**5.417**	**2.549**	**28.795**	4.745	2.198	1.712	1.157	3.468	**31.262**

Table 3. Comparison on the Sintel dataset. Refer to http://sintel.is.tue.mpg.de/results for details and for links to the various methods listed on the left. EPE stands for end-point error, among all, matched, and unmatched pixels (second through third column). dX-Y stands for error restricted to pixels between X and Y of objects boundaries, thus discounting error at occluded regions. sX-Y stands for pixels with displacements between X and Y pixels. Our method is competitive on all counts, and shines for large displacements, as expected.

3.2. Interpolation

The algorithm is sensitive to the choice of prior, which in our case corresponds to the choice of interpolation algorithm. To describe and motivate our choice, let $x, y \in D \subset \mathbb{R}^2$ be two points on the pixel lattice, with distance $d(x, y)$ for some choice of norm. We are interested in inferring the value of the displacement $w(x)$ at x from observations performed at y. We assume a parametric form for the likelihood function

$$p_\theta(w(x)|y) = \mathcal{N}\left(Ax + b; \Sigma(x, y)\right). \tag{1}$$

whereby the displacement w at x is a Gaussian random vector having as mean an affine deformation, depends on y, of the point x, with an uncertainty

$$\Sigma(x, y) = \beta^2 \exp\left(d(x, y)\right) I_{2\times 2} \tag{2}$$

that grows exponentially with the distance of the observation point. The parameters $\theta = \{A, b\}$ can be inferred via maximum-likelihood, given a sample $\mathcal{D} = \{x_i, w_i\}_{i=1}^N$, where $w_i = w(x_i)$, as

$$\begin{aligned} \hat{A}, \hat{b} &= \arg\max_\theta \prod_{i=1}^N p_\theta(w_i|x) \\ &= \arg\max_{A,b} \prod_{i=1}^N \mathcal{N}(Ax_i + b; \Sigma(x_i, x)) \\ &= \arg\min_{A,b} \sum_{i=1}^N \frac{\|w_i - Ax_i - b\|_2^2}{\beta^2 \exp\left(d(x_i, x)\right)} \end{aligned} \tag{3}$$

leaving β as a tuning parameter. This is essentially the locally-weighted (LA) estimator in Eq. (2) of [17]. Note that $p_\theta(w(x)|x) = \mathcal{N}(Ax + b; \beta^2 I_{2\times 2})$ and the parameters θ (which are the sufficient statistics of the dataset $\mathcal{D}$ for the displacement $w(x)$) are a function of the location x. We make this explicit by writing $\theta = \{A(x), b(x)\}$. A point-estimate, for instance the conditional mean, of the displacement can be obtained at each point x,

$$w(x) = A(x)x + b(x). \tag{4}$$

This approach follows [17] to avoid solving a variational optimization problem with explicit regularization, which is instead implicit in the finite-dimensional class of transformations (affine) and the finite data sample $\mathcal{D}$. The behavior of this interpolation method hinges critically on the choice of distance d in (3), which we describe next.

3.3. Topology

The distance between two points $d(x, y)$ can be based on the topology of the image domain, for instance $d_2(x, y) = \|x - y\|_2$, where nearby pixels are considered close, or the topology of the image range, for instance $d_I(x, y) = \|I(x) - I(y)\|$, where pixels with similar intensity are considered close. Ideally, we would like to use the topology of the *scene*, and consider points $x, y \in D$ close if the distance between their pre-images (back-projection) onto the scene $X, Y \in \mathbb{R}^3$ is close. This would be a geodesic distance, assuming the scene to be multiply-connected and piecewise smooth, infinite if X, Y are on different connected components.

Since we do not have a model of the scene, we use a proxy, whereby the distance between two points on the same connected component X, Y is the distance between their projections $x = \pi(X), y = \pi(Y)$ on the image, whereas the distance between points on different connected components adds a term proportional to their depth differential relative to the distance from the camera.

While we do not know their depth, disconnected components result in occlusion regions with area propor-

tional to the relative depth differential, where the optical flow residual $\phi(x) = \min_w \|I_1(x) - I_2(w(x))\|$ is generally large. Therefore, we can take the path-integral of optical flow residual as a proxy of the geodesic distance:

$$d_w(x,y) \doteq \min_{\gamma} \int_{\gamma_{x\to y}} \phi(z)dz \tag{5}$$

where $\gamma_{x\to y}$ is any path from x to y.

We can also assume that objects are smoothly colored, and therefore large intensity changes can be attributed to points being on different objects. Clearly this is not always the case, as smooth objects can have sharp material transitions, but nevertheless one can restrict the topology to simply connected components of the piecewise smooth albedo, and define d_I as

$$d_I(x,y) \doteq \min_{\gamma} \int_{\gamma_{x\to y}} |\nabla I(z)|dz \tag{6}$$

and similarly bypass the minimization by using a cordal distance. Various product distances, and various approximations to the geodesic, can be derived, for instance those in [17]. We use (6) in our algorithm.

3.4. Hypotheses (i)-(iv) testing

The key to our algorithm is the multi-scale iteration, starting from large regions that move slowly, eventually matching small regions that move fast. At each iteration, hypotheses of (i) Lambertian reflection and constant illumination, and (ii) co-visibilty (large residual) are tested conservatively relative to a fixed radius of the region of interest. Furthermore, backward-forward compatibility tests (iii) sufficient excitation; where failed, the regularizer (which in our case is implicit in the interpolation scheme) has license to take over.

While it would be desirable to have an integrated Bayesian framework where the thresholds are automatically determined by competing hypotheses, in practice these stages boil down to threshold selection. Importantly, the algorithm is not extremely sensitive to choice of thresholds. For reproducibility purposes, all parameters are reported in the Supplementary Material, and our implementation can be found at: `http://vision.ucla.edu/s2f.html`.

3.5. Computational cost

The computational cost of our algorithm is essentially dictated by the choice of components. Run-time depends on the complexity of the motion, since the length of our iteration is data-dependent. On average, it takes about 1m per pair of frames in Sintel, where images are of size 1024×436, on a commodity 4-core 3.1GHz desktop. We have observed convergence in as little as 20s, and as long as 2m. This includes all component elements of our pipeline.

On smaller images, for instance Middlebury's, (300×400), our algorithm runs in about 15s/pair of frames.

On KITTI, that has 400×1234 pixels per image, our algorithm runs, on average, at 1.5m per pair of frames.

4. Discussion

Our algorithm uses off-the-shelf components (sparse matching, interpolation, superpixelization), but in a manner that allows us to break free of the limitations of classical scale-space, that ties spatial frequency degradation to temporal anti-aliasing. Instead, we iterate matching over larger and larger domains of interest, on smaller and smaller regions at the native resolution. This is the key, together with a choice of regularizer designed to respect the phenomenology of correspondence, including occlusions, domain deformations, and relatively rudimentary illumination changes.

Acknowledgments

Research supported by ONR N00014-15-1-2261 and AFOSR FA9550-15-1-0229.

References

[1] Bailer, Christian and Taetz, Bertram and Stricker, Didier. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4015–4023, 2015. 2, 3, 5, 6

[2] Bao, Linchao and Yang, Qingxiong and Jin, Hailin. Fast edge-preserving patchmatch for large displacement optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3534–3541, 2014. 2

[3] Braux-Zin, Jim and Dupont, Romain and Bartoli, Adrien. A general dense image matching framework combining direct and feature-based costs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 185–192, 2013. 2

[4] Brox, Thomas and Bregler, Christoph and Malik, Jitendra. Large displacement optical flow. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 41–48. IEEE, 2009. 2

[5] Butler, Daniel J and Wulff, Jonas and Stanley, Garrett B and Black, Michael J. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, October 2012. 2, 5

[6] vCech, Jan and Sanchez-Riera, Jordi and Horaud, Radu. Scene flow estimation by growing correspondence seeds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3129–3136. IEEE, 2011. 2

[7] Chang, Haw-Shiuan and Wang, Yu-Chiang Frank. Superpixel-based large displacement optical flow. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 3835–3839. IEEE, 2013. 2

[8] Chen, Zhuoyuan and Jin, Hailin and Lin, Zhe and Cohen, Scott and Wu, Ying. Large displacement optical flow from nearest neighbor fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2443–2450, 2013. 2

[9] Dekel, Tali and Oron, Shaul and Rubinstein, Michael and Avidan, Shai and Freeman, William T. Best-buddies similarity for robust template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2021–2029. IEEE, 2015. 2

[10] Dollár, Piotr and Zitnick, C Lawrence. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence*, 37(8), pages 1558–1570. IEEE, 2015. 3

[11] Fischer, Philipp and Dosovitskiy, Alexey and Ilg, Eddy and Häusser, Philip and Hazırbaş, Caner and Golkov, Vladimir and van der Smagt, Patrick and Cremers, Daniel and Brox, Thomas. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 2

[12] Geiger, Andreas and Lenz, Philip and Stiller, Christoph and Urtasun, Raquel. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11), 1231–1237, Sage Publications Sage UK: London, England, 2013. 2, 4

[13] Hu, Yinlin and Song, Rui and Li, Yunsong. Efficient coarse-to-fine patchmatch for large displacement optical flow. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5704–5712, 2016 2

[14] Lindeberg, Tony. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013. 3

[15] Mac Aodha, Oisin and Humayun, Ahmad and Pollefeys, Marc and Brostow, Gabriel J. Learning a confidence measure for optical flow. *IEEE transactions on pattern analysis and machine intelligence*, 35(5), pages 1107–1120. IEEE, 2013. 2

[16] Revaud, Jerome and Weinzaepfel, Philippe and Harchaoui, Zaid and Schmid, Cordelia. Deepmatching: Hierarchical deformable dense matching. *International Journal of Computer Vision*, 120(3), pages 300–323. Springer, 2016. 2

[17] Revaud, Jerome and Weinzaepfel, Philippe and Harchaoui, Zaid and Schmid, Cordelia. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015. 2, 3, 7, 8

[18] Sun, Deqing and Roth, Stefan and Black, Michael J. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2), pages 115–137. Springer, 2014. 2, 4

[19] Szeliski, Richard and Zabih, Ramin and Scharstein, Daniel and Veksler, Olga and Kolmogorov, Vladimir and Agarwala, Aseem and Tappen, Marshall and Rother, Carsten. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6), pages 1068–1080. IEEE, 2008. 2, 4

[20] Thewlis, James and Zheng, Shuai and Torr, Philip HS and Vedaldi, Andrea. Fully-trainable deep matching. *arXiv preprint arXiv:1609.03532*, 2016. 2

[21] Timofte, Radu and Van Gool, Luc. Sparse flow: Sparse matching for small to large displacement optical flow. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 1100–1106. IEEE, 2015. 2

[22] Tran, Du and Bourdev, Lubomir and Fergus, Rob and Torresani, Lorenzo and Paluri, Manohar. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 2

[23] Verri, Alessandro and Poggio, Tomaso. Motion field and optical flow: Qualitative properties. *IEEE Transactions on pattern analysis and machine intelligence*, 11(5), pages 490–498. IEEE, 1989. 2

[24] Wedel, Andreas and Cremers, Daniel. Optical flow estimation. In *Stereo Scene Flow for 3D Motion Analysis*, pages 5–34. Springer, 2011. 2

[25] Weinzaepfel, Philippe and Revaud, Jerome and Harchaoui, Zaid and Schmid, Cordelia. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013. 2

[26] Wulff, Jonas and Black, Michael J. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 120–130. IEEE, 2015. 2

[27] Xu, Li and Dai, Zhenlong and Jia, Jiaya. Scale invariant optical flow. In *Computer Vision–ECCV 2012*, pages 385–399. Springer, 2012. 2

[28] Xu, Li and Jia, Jiaya and Matsushita, Yasuyuki. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9), pages 1744–1757. IEEE 2012. 2

[29] Yang, Jiaolong and Li, Hongdong. Dense, accurate optical flow estimation with piecewise parametric model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1019–1027, 2015. 2

[30] Sevilla-Lara, Laura and Sun, Deqing and Learned-Miller, Erik G and Black, Michael J Optical flow estimation with channel constancy. *European Conference on Computer Vision*, pages 423–438. Springer, 2014. 2