Secrets of Optical Flow Estimation and Their Principles

Deqing Sun
Brown University

Stefan Roth
TU Darmstadt

Michael J. Black
Brown University

Abstract

The accuracy of optical flow estimation algorithms has been improving steadily as evidenced by results on the Middlebury optical flow benchmark. The typical formulation, however, has changed little since the work of Horn and Schunck. We attempt to uncover what has made recent advances possible through a thorough analysis of how the objective function, the optimization method, and modern implementation practices influence accuracy. We discover that "classical" flow formulations perform surprisingly well when combined with modern optimization and implementation techniques. Moreover, we find that while median filtering of intermediate flow fields during optimization is a key to recent performance gains, it leads to higher energy solutions. To understand the principles behind this phenomenon, we derive a new objective that formalizes the median filtering heuristic. This objective includes a non-local term that robustly integrates flow estimates over large spatial neighborhoods. By modifying this new term to include information about flow and image boundaries we develop a method that ranks at the top of the Middlebury benchmark.

1. Introduction

The field of optical flow estimation is making steady progress as evidenced by the increasing accuracy of current methods on the Middlebury optical flow benchmark [6]. After nearly 30 years of research, these methods have obtained an impressive level of reliability and accuracy [33, 34, 35, 40]. *But what has led to this progress?* The majority of today's methods strongly resemble the original formulation of Horn and Schunck (HS) [18]. They combine a data term that assumes constancy of some image property with a spatial term that models how the flow is expected to vary across the image. An objective function combining these two terms is then optimized. Given that this basic structure is unchanged since HS, what has enabled the performance gains of modern approaches?

The paper has three parts. In the first, we perform an extensive study of current optical flow methods and models. The most accurate methods on the Middlebury flow dataset make different choices about how to model the objective function, how to approximate this model to make it computationally tractable, and how to optimize it. Since most published methods change *all* of these properties at once, it can be difficult to know which choices are most important. To address this, we define a baseline algorithm that is "classical", in that it is a direct descendant of the original HS formulation, and then systematically vary the model and method using different techniques from the art. The results are surprising. We find that only a small number of key choices produce statistically significant improvements and that they can be combined into a very simple method that achieves accuracies near the state of the art. More importantly, our analysis reveals what makes current flow methods work so well.

Part two examines the *principles* behind this success. We find that one algorithmic choice produces the most significant improvements: applying a median filter to intermediate flow values during incremental estimation and warping [33, 34]. While this heuristic improves the accuracy of the recovered flow fields, it actually *increases* the energy of the objective function. This suggests that what is being optimized is actually a new and different objective. Using observations about median filtering and L1 energy minimization from Li and Osher [23], we formulate a new *non-local term* that is added to the original, classical objective. This new term goes beyond standard local (pairwise) smoothness to robustly integrate information over large spatial neighborhoods. We show that minimizing this new energy approximates the original optimization with the heuristic median filtering step. Note, however, that the new objective falls outside our definition of classical methods.

Finally, once the median filtering heuristic is formulated as a non-local term in the objective, we immediately recognize how to modify and improve it. In part three we show how information about image structure and flow boundaries can be incorporated into a weighted version of the non-local term to prevent over-smoothing across boundaries. By incorporating structure from the image, this weighted version does not suffer from some of the errors produced by median filtering. At the time of publication (March 2010), the re-

sulting approach is ranked 1st in both angular and end-point errors in the Middlebury evaluation.

In summary, the contributions of this paper are to (1) analyze current flow models and methods to understand which design choices matter; (2) formulate and compare several classical objectives descended from HS using modern methods; (3) formalize one of the key heuristics and derive a new objective function that includes a non-local term; (4) modify this new objective to produce a state-of-the-art method. In doing this, we provide a "recipe" for others studying optical flow that can guide their design choices. Finally, to enable comparison and further innovation, we provide a public MATLAB implementation [1].

2. Previous Work

It is important to separately analyze the contributions of the objective function that defines the problem (*the model*) and the optimization algorithm and implementation used to minimize it (*the method*). The HS formulation, for example, has long been thought to be highly inaccurate. Barron *et al.* [7] reported an average angular error (AAE) of ~ 30 degrees on the "Yosemite" sequence. This confounds the objective function with the particular optimization method proposed by Horn and Schunck[1]. When optimized with today's methods, the HS objective achieves surprisingly competitive results despite the expected over-smoothing and sensitivity to outliers.

Models: The global formulation of optical flow introduced by Horn and Schunck [18] relies on both brightness constancy and spatial smoothness assumptions, but suffers from the fact that the quadratic formulation is not robust to outliers. Black and Anandan [10] addressed this by replacing the quadratic error function with a robust formulation. Subsequently, many different robust functions have been explored [12, 22, 31] and it remains unclear which is best. We refer to all these spatially-discrete formulations derived from HS as "classical." We systematically explore variations in the formulation and optimization of these approaches. The surprise is that the classical model, appropriately implemented, remains very competitive.

There are many formulations beyond the classical ones that we do not consider here. Significant ones use oriented smoothness [25, 31, 33, 40], rigidity constraints [32, 33], or image segmentation [9, 21, 41, 37]. While they deserve similar careful consideration, we expect many of our conclusions to carry forward. Note that one can select among a set of models for a given sequence [4], instead of finding a "best" model for all the sequences.

Methods: Many of the implementation details that are thought to be important date back to the early days of optical flow. Current best practices include coarse-to-fine estimation to deal with large motions [8, 13], texture decomposition [32, 34] or high-order filter constancy [3, 12, 16, 22, 40] to reduce the influence of lighting changes, bicubic interpolation-based warping [22, 34], temporal averaging of image derivatives [17, 34], graduated non-convexity [11] to minimize non-convex energies [10, 31], and median filtering after each incremental estimation step to remove outliers [34].

This median filtering heuristic is of particular interest as it makes non-robust methods more robust and improves the accuracy of all methods we tested. The effect on the objective function and the underlying reason for its success have not previously been analyzed. Least median squares estimation can be used to robustly reject outliers in flow estimation [5], but previous work has focused on the data term.

Related to median filtering, and our new non-local term, is the use of bilateral filtering to prevent smoothing across motion boundaries [36]. The approach separates a variational method into two filtering update stages, and replaces the original anisotropic diffusion process with multi-cue driven bilateral filtering. As with median filtering, the bilateral filtering step changes the original energy function.

Models that are formulated with an L1 robust penalty are often coupled with specialized total variation (TV) optimization methods [39]. Here we focus on generic optimization methods that can apply to any model and find they perform as well as reported results for specialized methods.

Despite recent algorithmic advances, there is a lack of publicly available, easy to use, and accurate flow estimation software. The GPU4Vision project [2] has made a substantial effort to change this and provides executable files for several accurate methods [32, 33, 34, 35]. The dependence on the GPU and the lack of source code are limitations. We hope that our public MATLAB code will not only help in understanding the "secrets" of optical flow, but also let others exploit optical flow as a useful tool in computer vision and related fields.

3. Classical Models

We write the "classical" optical flow objective function in its spatially discrete form as

$$E(\mathbf{u},\mathbf{v}) = \sum_{i,j}\{\rho_D(I_1(i,j) - I_2(i+u_{i,j}, j+v_{i,j})) \quad (1)$$
$$+\lambda[\rho_S(u_{i,j}-u_{i+1,j}) + \rho_S(u_{i,j}-u_{i,j+1})$$
$$+\rho_S(v_{i,j}-v_{i+1,j}) + \rho_S(v_{i,j}-v_{i,j+1})]\},$$

where $\mathbf{u}$ and $\mathbf{v}$ are the horizontal and vertical components of the optical flow field to be estimated from images I_1 and I_2, λ is a regularization parameter, and ρ_D and ρ_S are the data and spatial penalty functions. We consider three different penalty functions: (1) the quadratic HS penalty $\rho(x) = x^2$;

[1] They noted that the correct way to optimize their objective is by solving a system of linear equations as is common today. This was impractical on the computers of the day so they used a heuristic method.

(2) the Charbonnier penalty $\rho(x) = \sqrt{x^2 + \epsilon^2}$ [13], a differentiable variant of the L1 norm, the most robust convex function; and (3) the Lorentzian $\rho(x) = \log(1+\frac{x^2}{2\sigma^2})$, which is a non-convex robust penalty used in [10]. Note that this classical model is related to a standard pairwise Markov random field (MRF) based on a 4-neighborhood.

In the remainder of this section we define a baseline method using several techniques from the literature. This is not the "best" method, but includes modern techniques and will be used for comparison. We only briefly describe the main choices, which are explored in more detail in the following section and the cited references, especially [30].

Quantitative results are presented throughout the remainder of the text. In all cases we report the average end-point error (EPE) on the Middlebury training and test sets, depending on the experiment. Given the extensive nature of the evaluation, only average results are presented in the main body, while the details for each individual sequence are given in [30].

3.1. Baseline methods

To gain robustness against lighting changes, we follow [34] and apply the Rudin-Osher-Fatemi (ROF) structure texture decomposition method [28] to pre-process the input sequences and linearly combine the texture and structure components (in the proportion 20:1). The parameters are set according to [34].

Optimization is performed using a standard incremental multi-resolution technique (*e.g.* [10, 13]) to estimate flow fields with large displacements. The optical flow estimated at a coarse level is used to warp the second image toward the first at the next finer level, and a flow increment is calculated between the first image and the warped second image. The standard deviation of the Gaussian anti-aliasing filter is set to be $\frac{1}{\sqrt{2d}}$, where d denotes the downsampling factor. Each level is recursively downsampled from its nearest lower level. In building the pyramid, the downsampling factor is not critical as pointed out in the next section and here we use the settings in [31], which uses a factor of 0.8 in the final stages of the optimization. We adaptively determine the number of pyramid levels so that the top level has a width or height of around 20 to 30 pixels. At each pyramid level, we perform 10 warping steps to compute the flow increment.

At each warping step, we linearize the data term, which involves computing terms of the type $\frac{\partial}{\partial x} I_2(i+u^k_{i,j}, j+v^k_{i,j})$, where $\partial/\partial x$ denotes the partial derivative in the horizontal direction, u^k and v^k denote the current flow estimate at iteration k. As suggested in [34], we compute the derivatives of the second image using the 5-point derivative filter $\frac{1}{12}[-1\ 8\ 0\ -8\ 1]$, and warp the second image and its derivatives toward the first using the current flow estimate by bicubic interpolation. We then compute the spatial derivatives of the first image, average with the warped derivatives of the second image (*c.f.* [17]), and use this in place of $\frac{\partial I_2}{\partial x}$. For pixels moving out of the image boundaries, we set both their corresponding temporal and spatial derivatives to zero. After each warping step, we apply a 5×5 median filter to the newly computed flow field to remove outliers [34].

For the Charbonnier (**Classic-C**) and Lorentzian (**Classic-L**) penalty function, we use a graduated non-convexity (GNC) scheme [11] as described in [31] that linearly combines a quadratic objective with a robust objective in varying proportions, from fully quadratic to fully robust. Unlike [31], a single regularization weight λ is used for both the quadratic and the robust objective functions.

	Avg. Rank	Avg. EPE
Classic-C	14.9	0.408
HS	24.6	0.501
Classic-L	19.8	0.530
HS [31]	35.1	0.872
BA (Classic-L) [31]	30.9	0.746
Adaptive [33]	11.5	0.401
Complementary OF [40]	10.1	0.485

Table 1. **Models.** Average rank and end-point error (EPE) on the Middlebury *test set* using different penalty functions. Two current methods are included for comparison.

3.2. Baseline results

The regularization parameter λ is selected among a set of candidate values to achieve the best average end-point error (EPE) on the Middlebury training set. For the Charbonnier penalty function, the candidate set is $[1,\ 3,\ 5,\ 8,\ 10]$ and 5 is optimal. The Charbonnier penalty uses $\epsilon = 0.001$ for both the data and the spatial term in Eq. (1). The Lorentzian uses $\sigma = 1.5$ for the data term, and $\sigma = 0.03$ for the spatial term. These parameters are fixed throughout the experiments, except where mentioned.

Table 1 summarizes the EPE results of the basic model with three different penalty functions on the Middlebury test set, along with the two top performers at the time of publication (considering only published papers). The classic formulations with two non-quadratic penalty functions (**Classic-C**) and (**Classic-L**) achieve competitive results despite their simplicity. The baseline optimization of **HS** and BA (**Classic-L**) results in significantly better accuracy than previously reported for these models [31]. Note that the analysis also holds for the training set (Table 2).

At the time of publication, **Classic-C** ranks 13$^{\text{th}}$ in average EPE and 15$^{\text{th}}$ in AAE in the Middlebury benchmark despite its simplicity, and it serves as the baseline below. It is worth noting that the spatially discrete MRF formulation taken here is competitive with variational methods such as [33]. Moreover, our baseline implementation of **HS** has a lower average EPE than many more sophisticated methods.

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
HS	0.384	**1**	**0.0078**
Classic-L	0.319	**1**	**0.0078**
Classic-C-brightness	0.288	0	0.9453
HS-brightness	0.387	**1**	**0.0078**
Classic-L-brightness	0.325	0	0.2969
Gradient	0.305	0	0.4609

Table 2. **Pre-Processing.** Average end-point error (EPE) on the Middlebury *training set* for the baseline method (**Classic-C**) using different pre-processing techniques. Significance is always with respect to **Classic-C**.

4. Secrets Explored

We evaluate a range of variations from the baseline approach that have appeared in the literature, in order to illuminate which may be of importance. This analysis is performed on the Middlebury training set by changing only *one property at a time*. Statistical significance is determined using a Wilcoxon signed rank test between each modified method and the baseline **Classic-C**; a p value less than 0.05 indicates a significant difference.

Pre-Processing. For each method, we optimize the regularization parameter λ for the training sequences and report the results in Table 2. The baseline uses a non-linear prefiltering of the images to reduce the influence of illumination changes [34]. Table 2 shows the effect of removing this and using a standard brightness constancy model (***-brightness**). **Classic-C-brightness** actually achieves lower EPE on the training set than **Classic-C** but significantly lower accuracy on the test set: **Classic-C-brightness** = 0.726, **HS-brightness** = 0.759, and **Classic-L-brightness** = 0.603 – see Table 1 for comparison. This disparity suggests overfitting is more severe for the brightness constancy assumption. **Gradient** only imposes constancy of the gradient vector at each pixel as proposed in [12] (*i.e.* it robustly penalizes Euclidean distance between image gradients) and has similar performance in both training and test sets (*c.f.* Table 8). See [30] for results of more alternatives.

Secrets: Some form of image filtering is useful but simple derivative constancy is nearly as good as the more sophisticated texture decomposition method.

Coarse-to-fine estimation and GNC. We vary the number of warping steps per pyramid level and find that **3 warping steps** gives similar results as using 10 (Table 3). For the GNC scheme, [31] uses a downsampling factor of 0.8 for non-convex optimization. A downsampling factor of 0.5 (**Down-**0.5), however, has nearly identical performance

Removing the GNC step for the Charbonnier penalty function (**w/o GNC**) results in higher EPE on most sequences and higher energy on all sequences (Table 4). This suggests that the GNC method is helpful even for the convex Charbonnier penalty function due to the nonlinearity of the data term.

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
3 **warping steps**	0.304	0	0.9688
Down-0.5	0.298	0	1.0000
w/o GNC	0.354	0	0.1094
Bilinear	0.302	0	0.1016
w/o TAVG	0.306	0	0.1562
Central derivative filter	0.300	0	0.7266
7-point derivative filter [13]	0.302	0	0.3125
Bicubic-II	0.290	**1**	**0.0391**
GC-0.45 ($\lambda = 3$)	0.292	**1**	**0.0156**
GC-0.25 ($\lambda = 0.7$)	0.298	0	1.0000
MF 3×3	0.305	0	0.1016
MF 7×7	0.305	0	0.5625
$2\times$ **MF**	0.300	0	1.0000
$5\times$ **MF**	0.305	0	0.6875
w/o MF	0.352	**1**	**0.0078**
Classic++	0.285	**1**	**0.0078**

Table 3. **Model and Methods.** Average end-point error (EPE) on the Middlebury *training set* for the baseline method (**Classic-C**) using different algorithm and modeling choices.

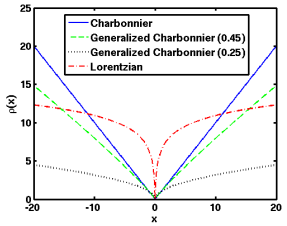


Figure 1. Different penalty functions for the spatial terms: Charbonnier ($\epsilon = 0.001$), generalized Charbonnier ($a = 0.45$ and $a = 0.25$), and Lorentzian ($\sigma = 0.03$).

Secrets: The downsampling factor does not matter when using a convex penalty; a standard factor of 0.5 is fine. Some form of GNC is useful even for a convex robust penalty like Charbonnier because of the nonlinear data term.

Interpolation method and derivatives. We find that bicubic interpolation is more accurate than bilinear (Table 3, **Bilinear**), as already reported in previous work [34]. Removing temporal averaging of the gradients (**w/o TAVG**), using **Central difference filters**, or using a 7**-point derivative filter** [13] all reduce accuracy compared to the baseline, but not significantly. The MATLAB built-in function *interp2* is based on cubic convolution approximation [20]. The spline-based interpolation scheme [26] is consistently better (**Bicubic-II**). See [30] for more discussions.

Secrets: Use spline-based bicubic interpolation with a 5-point filter. Temporal averaging of the derivatives is probably worthwhile for a small computational expense.

Penalty functions. We find that the convex Charbonnier penalty performs better than the more robust, non-convex Lorentzian on both the training and test sets. One reason might be that non-convex functions are more difficult to optimize, causing the optimization scheme to find a poor local

(a) **With** median filtering (b) **Without** median filtering

Figure 2. Estimated flow fields on sequence "RubberWhale" using **Classic-C** with and without (**w/o MF**) the median filtering step. Color coding as in [6]. (a) (**w/ MF**) energy 502, 387 and (b) (**w/o MF**) energy 449, 290. The median filtering step helps reach a solution free from outliers but with a higher energy.

optimum. We investigate a generalized Charbonnier penalty function $\rho(x) = (x^2 + \epsilon^2)^a$ that is equal to the Charbonnier penalty when $a = 0.5$, and non-convex when $a < 0.5$ (see Figure 1). We optimize the regularization parameter λ again. We find a slightly non-convex penalty with $a = 0.45$ (**GC-**0.45) performs consistently better than the Charbonnier penalty, whereas more non-convex penalties (**GC-**0.25 with $a = 0.25$) show no improvement.

Secrets: The less-robust Charbonnier is preferable to the Lorentzian and a slightly non-convex penalty function (**GC-**0.45) is better still.

Median filtering. The baseline 5×5 median filter (**MF** 5×5) is better than both **MF** 3×3 [34] and **MF** 7×7 but the difference is not significant (Table 3). When we perform 5×5 median filtering twice ($2\times$ **MF**) or five times ($5\times$ **MF**) per warping step, the results are worse. Finally, removing the median filtering step (**w/o MF**) makes the computed flow significantly less accurate with larger outliers as shown in Table 3 and Figure 2.

Secrets: Median filtering the intermediate flow results once after every warping iteration is the single most important secret; 5×5 is a good filter size.

4.1. Best Practices

Combining the analysis above into a single approach means modifying the baseline to use the slightly non-convex generalized Charbonnier and the spline-based bicubic interpolation. This leads to a statistically significant improvement over the baseline (Table 3, **Classic++**). This method is directly descended from **HS** and **BA**, yet updated with the current best optimization practices known to us. This simple method ranks 9th in EPE and 12th in AAE on the Middlebury test set.

5. Models Underlying Median Filtering

Our analysis reveals the practical importance of median filtering during optimization to *denoise* the flow field. We ask whether there is a *principle* underlying this heuristic?

One interesting observation is that flow fields obtained with median filtering have substantially *higher* energy than those without (Table 4 and Figure 2). If the median filter is helping to optimize the objective, it should lead to lower energies. Higher energies and more accurate estimates suggest that incorporating median filtering changes the objective function being optimized.

The insight that follows from this is that the median filtering heuristic is related to the minimization of an objective function that differs from the classical one. In particular the optimization of Eq. (1), with interleaved median filtering, approximately minimizes

$$\begin{aligned} E_A(\mathbf{u},\mathbf{v},\hat{\mathbf{u}},\hat{\mathbf{v}}) = & \qquad (2)\\ & \sum_{i,j}\Big\{\rho_D(I_1(i,j) - I_2(i+u_{i,j}, j+v_{i,j})) \\ & \quad +\lambda[\rho_S(u_{i,j}-u_{i+1,j}) + \rho_S(u_{i,j}-u_{i,j+1}) + \\ & \quad \rho_S(v_{i,j}-v_{i+1,j}) + \rho_S(v_{i,j}-v_{i,j+1})]\Big\} \\ & +\lambda_2(||\mathbf{u}-\hat{\mathbf{u}}||^2 + ||\mathbf{v}-\hat{\mathbf{v}}||^2) \\ & +\sum_{i,j}\sum_{(i',j')\in N_{i,j}} \lambda_3(|\hat{u}_{i,j}-\hat{u}_{i',j'}| + |\hat{v}_{i,j}-\hat{v}_{i',j'}|), \end{aligned}$$

where $\hat{\mathbf{u}}$ and $\hat{\mathbf{v}}$ denote an auxiliary flow field, $N_{i,j}$ is the set of neighbors of pixel (i,j) in a possibly large area and λ_2 and λ_3 are scalar weights. The term in braces is the same as the flow energy from Eq. (1), while the last term is new. This *non-local* term [14, 15] imposes a particular smoothness assumption within a specified region of the auxiliary flow field $\hat{\mathbf{u}}, \hat{\mathbf{v}}$[2]. Here we take this term to be a 5×5 rectangular region to match the size of the median filter in **Classic-C**. A third (coupling) term encourages $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ and $\mathbf{u}, \mathbf{v}$ to be the same (*c.f.* [33, 39]).

The connection to median filtering (as a denoising method) derives from the fact that there is a direct relationship between the median and L1 minimization. Consider a simplified version of Eq. (2) with just the coupling and non-local terms, where $E(\hat{\mathbf{u}}) =$

$$\lambda_2||\mathbf{u}-\hat{\mathbf{u}}||^2 + \sum_{i,j}\sum_{(i',j')\in N_{i,j}} \lambda_3|\hat{u}_{i,j}-\hat{u}_{i',j'}|. \qquad (3)$$

While minimizing this is similar to median filtering $\mathbf{u}$, there are two differences. First, the non-local term minimizes the L1 distance between the central value and all flow values in its neighborhood except itself. Second, Eq. (3) incorporates information about the data term through the coupling equation; median filtering the flow ignores the data term.

The formal connection between Eq. (3) and median filtering[3] is provided by Li and Osher [23] who show that min-

[2] Bruhn *et al.* [13] also integrated information over a local region in a global method but did so for the data term.

[3] Hsiao *et al.* [19] established the connection in a slightly different way.

	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Classic-C	0.589	0.748	0.866	0.502	1.816	2.317	1.126	1.424
w/o GNC	0.593	0.750	0.870	0.506	1.845	2.518	1.142	1.465
w/o MF	0.517	0.701	0.668	0.449	1.418	1.830	1.066	1.395

Table 4. Eq. (1) energy ($\times 10^6$) for the optical flow fields computed on the Middlebury *training set*. Note that **Classic-C** uses graduated non-convexity (GNC), which reduces the energy, and median filtering, which increases it.

imizing Eq. (3) is related to a different median computation

$$\hat{u}_{i,j}^{(k+1)} = \text{median}(\text{Neighbors}^{(k)} \cup \text{Data}) \tag{4}$$

where $\text{Neighbors}^{(k)} = \{\hat{u}_{i',j'}^{(k)}\}$ for $(i',j') \in N_{i,j}$ and $\hat{\mathbf{u}}^{(0)} = \mathbf{u}$ as well as

$$\text{Data} = \{u_{i,j}, u_{i,j} \pm \tfrac{\lambda_3}{\lambda_2}, u_{i,j} \pm \tfrac{2\lambda_3}{\lambda_2} \cdots, u_{i,j} \pm \tfrac{|N_{i,j}|\lambda_3}{2\lambda_2}\},$$

where $|N_{i,j}|$ denotes the (even) number of neighbors of (i,j). Note that the set of "data" values is balanced with an equal number of elements on either side of the value $u_{i,j}$ and that information about the data term is included through $u_{i,j}$. Repeated application of Eq. (4) converges rapidly [23].

Observe that, as λ_3/λ_2 increases, the weighted data values on either side of $u_{i,j}$ move away from the values of Neighbors and cancel each other out. As this happens, Eq. (4) approximates the median at the first iteration

$$\hat{u}_{i,j}^{(1)} \approx \text{median}(\text{Neighbors}^{(0)} \cup \{u_{i,j}\}). \tag{5}$$

Eq. (2) thus combines the original objective with an approximation to the median, the influence of which is controlled by λ_3/λ_2. Note in practice the weight λ_2 on the coupling term is usually small or is steadily increased from small values [34, 39]. We optimize the new objective (2) by alternately minimizing

$$\begin{aligned} E_O(\mathbf{u},\mathbf{v}) = \sum_{i,j} & \rho_D(I_1(i,j) - I_2(i+u_{i,j}, j+v_{i,j})) \\ & + \lambda[\rho_S(u_{i,j} - u_{i+1,j}) + \rho_S(u_{i,j} - u_{i,j+1}) \\ & + \rho_S(v_{i,j} - v_{i+1,j}) + \rho_S(v_{i,j} - v_{i,j+1})] \\ & + \lambda_2(||\mathbf{u} - \hat{\mathbf{u}}||^2 + ||\mathbf{v} - \hat{\mathbf{v}}||^2) \end{aligned} \tag{6}$$

and

$$\begin{aligned} E_M(\hat{\mathbf{u}},\hat{\mathbf{v}}) = & \lambda_2(||\mathbf{u} - \hat{\mathbf{u}}||^2 + ||\mathbf{v} - \hat{\mathbf{v}}||^2) \\ & + \sum_{i,j} \sum_{(i',j') \in N_{i,j}} \lambda_3(|\hat{u}_{i,j} - \hat{u}_{i',j'}| + |\hat{v}_{i,j} - \hat{v}_{i',j'}|). \end{aligned} \tag{7}$$

Note that an alternative formulation would drop the coupling term and impose the non-local term directly on $\mathbf{u}$ and $\mathbf{v}$. We find that optimization of the coupled set of equations is superior in terms of EPE performance.

The alternating optimization strategy first holds $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ fixed and minimizes Eq. (6) w.r.t. $\mathbf{u}, \mathbf{v}$. Then, with $\mathbf{u}, \mathbf{v}$ fixed, we minimize Eq. (7) w.r.t. $\hat{\mathbf{u}}, \hat{\mathbf{v}}$. Note that Eqs. (3) and (7) can be minimized by repeated application of Eq. (4); we use this approach with 5 iterations. We perform 10 steps of alternating optimizations at every pyramid level and change λ_2 logarithmically from 10^{-4} to 10^2. During the first and second GNC stages, we set $\mathbf{u}, \mathbf{v}$ to be $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ after every warping step (this step helps reach solutions with lower energy and EPE [30]). In the end, we take $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ as the final flow field estimate. The other parameters are $\lambda = 5, \lambda_3 = 1$.

	Avg. EPE	significance	p-value
Classic-C	0.298	—	—
Classic-C-A	0.305	0	0.8125

Table 5. Average end-point error (EPE) on the Middlebury *training set* is shown for the new model with alternating optimization (**Classic-C-A**).

Alternatingly optimizing this new objective function (**Classic-C-A**) leads to similar results as the baseline **Classic-C** (Table 5). We also compare the energy of these solutions using the new objective and find the alternating optimization produces the lowest energy solutions, as shown in Table 6. To do so, we set both the flow field $\mathbf{u}, \mathbf{v}$ and the auxiliary flow field $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ to be the same in Eq. (2).

In summary, we show that the heuristic median filtering step in **Classic-C** can now be viewed as energy minimization of a new objective with a non-local term. The explicit formulation emphasizes the value of robustly integrating information over large neighborhoods and enables the improved model described below.

6. Improved Model

By formalizing the median filtering heuristic as an explicit objective function, we can find ways to improve it. While median filtering in a large neighborhood has advantages as we have seen, it also has problems. A neighborhood centered on a corner or thin structure is dominated by the surround and computing the median results in oversmoothing as illustrated in Figure 3(a).

Examining the non-local term suggests a solution. For a given pixel, if we know which other pixels in the area belong to the same surface, we can weight them more highly. The modification to the objective function is achieved by introducing a weight into the non-local term [14, 15]:

$$\sum_{i,j} \sum_{(i',j') \in N_{i,j}} w_{i,j,i',j'}(|\hat{u}_{i,j} - \hat{u}_{i',j'}| + |\hat{v}_{i,j} - \hat{v}_{i',j'}|), \tag{8}$$

where $w_{i,j,i',j'}$ represents how likely pixel i', j' is to belong to the same surface as i, j.

	Venus	Dimetrodon	Hydrangea	RubberWhale	Grove2	Grove3	Urban2	Urban3
Classic-C	0.817	0.903	1.202	0.674	2.166	3.144	1.954	2.153
Classic-C w/o MF	0.886	0.945	1.299	0.725	2.315	3.513	2.234	2.712
Classic-C-A	0.784	0.889	1.139	0.666	2.064	2.976	1.922	2.049

Table 6. Eq. (2) energy ($\times 10^6$) for the computed flow fields on the Middlebury *training set*. The alternating optimization strategy (**Classic-C-A**) produces the lowest energy solutions.

(a) (b)

Figure 3. Median filtering over-smoothes the rifle in the "Army" sequence, while the proposed weighted non-local term preserves the detail. Results of (a) **Classic++** (b) **Classic+NL**.

Of course, we do not know $w_{i,j,i',j'}$, but can approximate it. We draw ideas from [29, 36, 38] to define the weights according to their spatial distance, their color-value distance, and their occlusion state as $w_{i,j,i',j'} \propto$

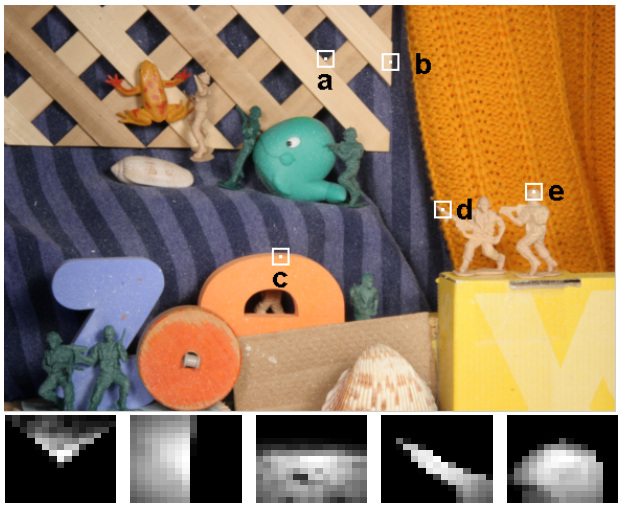
$$\exp\left\{-\frac{|i-i'|^2+|j-j'|^2}{2\sigma_1^2}-\frac{|\mathbf{I}(i,j)-\mathbf{I}(i',j')|^2}{2\sigma_2^2}\right\}\frac{o(i',j')}{o(i,j)}, \quad (9)$$

where the occlusion variable $o(i,j)$ is calculated using Eq. (22) in [29], $\mathbf{I}(i,j)$ is the color vector in the Lab space, and $\sigma_1 = 7, \sigma_2 = 7$. Examples of such weights are shown for several 15×15 neighborhoods in Figure 4; bright values indicate higher weights. Note the neighborhood labeled **d**, corresponding to the rifle. Since pixels on the rifle are in the minority, an unweighted median would oversmooth. The weighted term instead robustly estimates the motion using values on the rifle. A closely related piece of work is [27], which uses the intervening contour to define affinities among neighboring pixels for the local Lucas and Kanade [24] method. However it only uses this scheme to estimate motion for sparse points and then interpolates the dense flow field.

We approximately solve Eq. (9) for $\hat{\mathbf{u}}, \hat{\mathbf{v}}$ as the following weighted median problem

$$\min_{\hat{u}_{i,j}} \sum_{(i',j')\in N_{i,j}\cup\{i,j\}} w_{i,j,i',j'}|\hat{u}_{i,j}-u_{i',j'}|, \quad (10)$$

using the formula (3.13) in [23] for all the pixels (**Classic+NL-Full**). Note if all the weights are equal, the solution is just the median. In practice, we can adopt a fast version (**Classic+NL**) without performance loss. Given a current estimate of the flow, we detect motion boundaries using a Sobel edge detector and dilate these edges with a 5×5 mask to obtain flow boundary regions. In these regions we use the weighting in Eq. (9) in a 15×15 neighborhood. In the non-boundary regions, we use equal weights in a 5×5 neighoborhood to compute the median.



(a) (b) (c) (d) (e)

Figure 4. Neighbor weights of the proposed weighted non-local term at different positions in the "Army" sequence.

	Avg. EPE	significance	p-value
Classic+NL	0.221	—	—
Classic+NL-Full	0.222	0	0.8203

Table 7. Average end-point error (EPE) on the Middlebury *training set* is shown for the fast and full versions of the improved model.

	Avg. Rank	**Avg. EPE**
Classic++	13.4	0.406
Classic++Gradient	15.1	0.430
Classic+NL	6.2	0.319
Classic+NL-Full	6.6	0.316

Table 8. Average end-point error (EPE) on the Middlebury *test set* for the **Classic++** model with two different preprocessing techniques and its improved model.

Tables 7 and 8 show that the weighted non-local term (**Classic+NL**) improves the accuracy on both the training and the test sets. Note that the fine detail of the "rifle" is preserved in Figure 3(b). At the time of publication, **Classic+NL** ranks 1st in both AAE and EPE in the Middlebury evaluation and has the lowest average AAE and EPE among all listed algorithms. The running time on the test "Urban" sequence is about 70 minutes for **Classic+NL-Full** and about 16 miniutes for **Classic+NL** in MATLAB.

7. Conclusions

Implemented using modern practices, classical optical flow formulations produce competitive results on the Mid-

dlebury training and test sets. To understand the "secrets" that help such basic formulations work well, we quantitatively studied various aspects of flow approaches from the literature, including their implementation details. Among the good practices, we found that using median filtering to denoise the flow after every warping step is key to improving accuracy, but that it increases the energy of the final result. Exploiting connections between median filtering and L1-based denoising, we showed that algorithms relying on a median filtering step are approximately optimizing a different objective that regularizes flow over a large spatial neighborhood. This principle enables us to design and optimize improved models that weight the neighbors adaptively in an extended image region. At the time of publication (March 2010), the resulting algorithm ranks 1st in both angular and end-point errors in the Middlebury evaluation. The MATLAB code is publicly available [1].

How far can the 2-frame classical methods be pushed? Our sense is that they are likely to improve incrementally for several years to come, but that the big gains will come from methods that go beyond the classical formulation to reason more explicitly about surfaces and boundaries and how they move over time.

Acknowledgments. DS and MJB were supported by a gift from Intel Corp. and NSF CRCNS award IIS-0904875. We thank the reviewers for constructive comments, especially the connection between our original "area" term and non-local regularization, P. Yadollahpour for his early work on implementing the HS method, S. Zuffi for suggesting the color version of the non-local term, T. Brox, A. Wedel, and M. Werlberger for clarifying details about their papers, and D. Scharstein for maintaining the online optical flow benchmark.

References

[1] http://www.cs.brown.edu/people/dqsun/
[2] http://gpu4vision.icg.tugraz.at/
[3] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.
[4] O. M. Aodha, G. Brostow, and M. Pollefeys. Segmenting video into classes of algorithm-suitability. In *CVPR*, 2010.
[5] A. Bab-Hadiashar and D. Suter. Robust optic flow computation. *IJCV*, 29(1):59–77, 1998.
[6] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. In *ICCV*, 2007.
[7] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt. Performance of optical flow techniques. *IJCV*, 92:236–242, 1994.
[8] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, pages 237–252, 1992.
[9] M. Black and A. Jepson. Estimating optical-flow in segmented images using variable-order parametric models with local deformations. *PAMI*, 18(10):972–986, 1996.
[10] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 63:75–104, 1996.
[11] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, Massachusetts, 1987.
[12] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, pages 25–36, 2004.
[13] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: combining local and global optic flow methods. *IJCV*, 61(3):211–231, 2005.
[14] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *CVPR*, 2005.
[15] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7:1005–1028, 2008.
[16] F. Glaer, G. Reynolds, and P. Anandan. Scene matching by hierarchical correlation. In *CVPR*, pages 432–441, 1983.
[17] B. Horn. *Robot Vision*. MIT Press, 1986.
[18] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 16:185–203, Aug. 1981.
[19] I. Hsiao, A. Rangarajan, and G. Gindi. A new convex edge-preserving median prior with applications to tomography. *IEEE TMI*, 22(5):580–585, 2003.
[20] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE TASSP*, 29:1153–1160, 1981.
[21] C. Lei and Y.-H. Yang. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *ICCV*, 2009.
[22] V. Lempitsky, S. Roth, and C. Rother. FusionFlow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008.
[23] Y. Li and S. Osher. A new median formula with applications to PDE based denoising. *Commun. Math. Sci.*, 7(3):741–753, 2009.
[24] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.
[25] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *PAMI*, 8(5):565–593, 1986.
[26] W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical Recipes in C++: the art of scientific computing*. Cambridge University Press, New York, NY, USA, 2002.
[27] X. Ren. Local grouping for optical flow. In *CVPR*, 2008.
[28] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.
[29] P. Sand and S. Teller. Particle video: Long-range motion estimation using point trajectories. *IJCV*, 80(1):72–91, 2008.
[30] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. Technical report, Brown-CS-10-03, 2010.
[31] D. Sun, S. Roth, J. Lewis, and M. J. Black. Learning optical flow. In *ECCV*, volume 3, pages 83–97, 2008.
[32] A. Wedel, T. Pock, J. Braun, U. Franke, and D. Cremers. Duality TV-L1 flow with fundamental matrix prior. In *IVCNZ*, 2008.
[33] A. Wedel, T. Pock, and D. Cremers. Structure- and motion-adaptive regularization for high accuracy optic flow. In *ICCV*, 2009.
[34] A. Wedel, T. Pock, C. Zach, D. Cremers, and H. Bischof. An improved algorithm for TV-L1 optical flow. In *Dagstuhl Motion Workshop*, 2008.
[35] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. In *BMVC*, 2009.
[36] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi. Bilateral filtering-based optical flow estimation with occlusion detection. In *ECCV*, pages I: 211–224, 2006.
[37] L. Xu, J. Chen, and J. Jia. A segmentation based variational model for accurate optical flow estimation. In *ECCV*, pages 671–684, 2008.
[38] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *PAMI*, 28(4):650–656, 2006.
[39] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *DAGM*, 2007.
[40] H. Zimmer, A. Bruhn, J. Weickert, L. Valgaerts, A. Salgado, B. Rosenhahn, and H.-P. Seidel. Complementary optic flow. In *EMMCVPR*, 2009.
[41] C. Zitnick, N. Jojic, and S. B. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, pages 1308–1315, 2005.