Secrets of Optical Flow Estimation and Their Principles

Deqing Sun[1], **Stefan Roth**[2], and **Michael J. Black**[1]

[1]Brown University, [2]TU Darmstadt

Overview

Goals:

- What makes modern optical flow techniques accurate and why?
- How can we use such insights to improve flow techniques further?

Secrets: Quantitative analysis of current practices in optical flow estimation starting from a simple, classical formulation

Principles: Formalization of the heuristic median filtering step as an unweighted non-local term added to the original objective

Improved model: Introduce weighted non-local term that uses color, flow, and occlusion information to better preserve motion details

MATLAB code: http://www.cs.brown.edu/~dqsun/

Secrets

What makes optical flow accurate?

Classical formulation:

Standard data constancy term

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

Smoothness term: Pairwise Markov random field (MRF) with 4 neighbors

Modern implementation:

- Preprocessing: Rudin-Osher-Fatemi (ROF) structure texture decomposition [3]
- Standard incremental multi-resolution technique, 10 warping steps per level
- Graduate non-convexity (GNC) for non-quadratic penalty
- Bicubic interpolation to warp image and its derivatives
- 5-point image derivative filter
- Temporal averaging of image derivatives
- 5X5 median filtering of intermediate flow field per warping step [3]

How good is it?

	Middlebury *test* set	Avg. Rank	Avg. EPE	
Current implementation	**Classic-**Charbonnier	14.9	0.408	$\rho(x)=\sqrt{x^2+\varepsilon^2}$
	HS (Horn & Schunk)	24.6	0.501	$\rho(x)=x^2$
	Classic-Lorentzian **(BA)**	19.8	0.530	$\rho(x)=\log(1+\frac{x^2}{2\sigma^2})$
Previous implementation	**HS**	35.1	0.872	
	BA (Black& Anandan)	30.9	0.746	
Previous state of the art	**Adaptive**	11.5	0.401	
	Complementary OF	10.1	0.485	
For later comparison	**Classic-C-brightness**	N/A	0.726	

Table 1. **Models.** Average end-point error (EPE) on Middlebury *test* set by Classical formulation with different penalty functions.

What is important?

Approach: Change one property of **Classic-C** at a time, compare avg. EPE, and test statistical significance

Secrets uncovered

- **Pre-processing:** Some kind of image filtering is useful but simple gradient constancy is as good as more sophisticated texture decomposition; overfitting is more severe for brightness constancy
- **Interpolation and image derivatives:** Bicubic interpolation is slightly better than bilinear but not significantly; *spline-based bicubic interpolation* is consistently better than convolution-based (MATLAB built-in); removing temporal averaging of image derivatives, central difference filter, and 7-point derivative filter reduce accuracy, but not significantly
- **Coarse-to-fine estimation and GNC:** Pyramid downsampling factor does not matter for the convex penalty and 0.5 is fine; graduated non-convexity (GNC) helps even the convex robust Charbonnier penalty
- **Penalty function:** Less robust Charbonnier is better than Lorentzian; a slightly more robust penalty (*generalized Charbonnier* $\rho(x)=(x^2+\varepsilon^2)^a$, a = 0.45) is better still

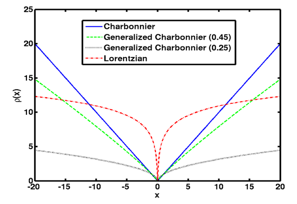


Figure 1. Different penalty functions for the spatial terms.

Median filtering: Median filtering the intermediate flow field is the single most important secret; 5x5 is a good filter size

Wilcoxon signed rank test between each **variant** and baseline **Classic-C**

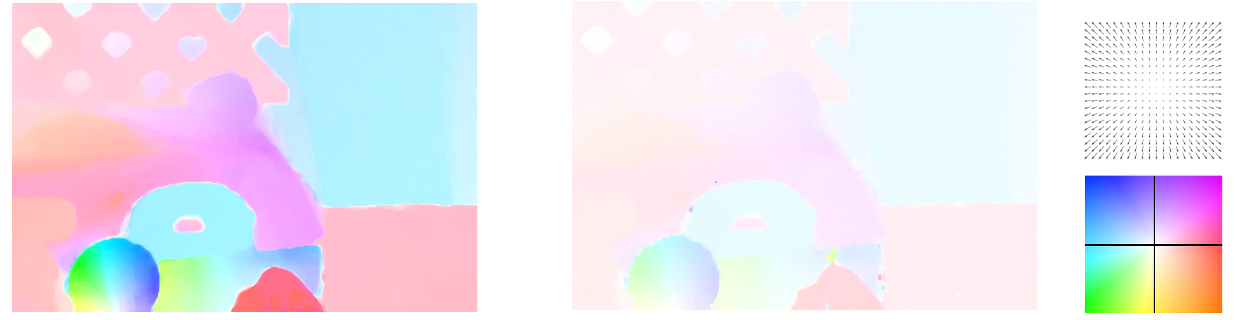
	Middlebury *training* set	Avg. EPE	signif.	*p*-value
Baseline	Classic-C	0.298	-	-
Preprocessing	Brightness constancy	0.288	0	0.9453
	Gradient constancy	0.305	0	0.4609
Interpolation and image derivatives	Bilinear interpolation	0.302	0	0.1016
	Central difference filter	0.300	0	0.7266
	7-point derivative filter	0.302	0	0.3125
	Spline-based bicubic interpolation	0.290	1	0.0391 ☺
	No temporal average of derivatives	0.306	0	0.1562
Coarse-to-fine estimation and GNC	Downsampling factor 0.5	0.298	0	1.0000
	3 warping steps per level	0.304	0	0.9688
	No graduated non-convexity (GNC)	0.354	0	0.1094
Penalty function	Generalized Charbonnier-0.45	0.292	1	0.0156 ☺
	Generalized Charbonnier-0.25	0.298	0	1.0000
Median filtering	Median filter size 3 X 3	0.305	0	0.1016
	Median filter size 7 X 7	0.305	0	0.5625
	Median filtering twice	0.300	0	1.0000
	No median filtering	0.352	1	0.0078 ☹
Best practices	Classic++	0.285	1	0.0078 ☺

Table 2. **Models and Methods.** Average end-point error (EPE) on Middlebury *training* set for **Classic-C** and its variants.

Best practices (Classic++): Modify baseline Classic-C to use the slightly non-convex generalized Charbonnier and spline-based bicubic interpolation. This method is directly descended from HS and BA, yet updated with current best practices known to us. It ranks 8th in EPE on the public Middlebury test set.

Principles

Median filtering leads to lower EPE, but higher energy solutions!



With MF: Energy **502,387**, EPE 0.093 **Without MF:** Energy **449,290**, EPE 0.113

Figure 2. Estimated flow fields on "RubberWhale" by **Classic-C**.

What is being minimized?

Observation: Median filtering can be posed as L1 energy minimization [1]. Replace median filter with minimization of this objective function:

$$E(\hat{\mathbf{u}}) = \lambda_2\,\|\mathbf{u}-\hat{\mathbf{u}}\|^2 + \sum_{i,j}\sum_{(i',j')\in N_{i,j}} |\hat{u}_{i,j}-\hat{u}_{i',j'}|$$

New objective function: Non-local term robustly integrates information over a **large spatial neighborhood**

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

Classical formulation

$$+\lambda_2(\|\mathbf{u}-\hat{\mathbf{u}}\|^2 + \|\mathbf{v}-\hat{\mathbf{v}}\|^2)$$

Coupling term

$$+\sum_{i,j}\sum_{(i',j')\in N_{i,j}} \lambda_3(|\hat{u}_{i,j}-\hat{u}_{i',j'}| + |\hat{v}_{i,j}-\hat{v}_{i',j'}|)$$

Non-local term

5X5 neighborhood

Optimization: Alternate optimization between coupled classical and non-local terms

Alternating optimization **(Classic-C-A)** leads to similar performance

Middlebury ***training*** set	Avg. EPE	Significance	*p*-value
Classic-C	0.298	-	-
Classic-C-A	0.305	0	0.8125

Table 3. Average EPE on *training* set for the new objective with alternating optimization.

Table 4. Screen shot of the Middlebury optical flow benchmark (June 2010).

Average endpoint error	avg. rank	Army (Hidden texture) all	Army disc	Army untext	Mequon (Hidden texture) all	Mequon disc	Mequon untext	Schefflera (Hidden texture) all	Schefflera disc	Schefflera untext	Wooden (Hidden texture) all	Wooden disc	Wooden untext
Classic+NL [38]	5.8	0.08 1	0.23 1	0.07 2	0.22 8	0.74 9	0.18 10	0.29 7	0.65 7	0.19 7	0.15 1	0.73 3	0.09 1
MDP-Flow [30]	6.4	0.09 2	0.25 3	0.08 5	0.19 2	0.54 2	0.18 10	0.24 1	0.55 3	0.20 9	0.16 4	0.91 8	0.09 1
NL-TV-NCC [28]	7.4	0.10 7	0.26 6	0.08 5	0.22 8	0.72 8	0.15 3	0.35 9	0.85 9	0.16 1	0.15 1	0.70 2	0.09 1
Layer+dense [37]	8.1	0.09 2	0.28 12	0.08 5	0.22 8	0.74 9	0.19 13	0.25 4	0.58 4	0.21 11	0.17 6	0.92 10	0.09 1
Complementary OF [24]	9.1	0.10 7	0.26 6	0.09 10	0.20 6	0.70 6	0.14 2	0.35 9	0.85 9	0.16 1	0.19 9	1.05 14	0.10 7
Adaptive [23]	10.2	0.09 2	0.26 6	0.06 1	0.23 13	0.78 12	0.18 10	0.54 20	1.19 22	0.21 11	0.18 7	0.91 8	0.10 7
Adapt-Window [36]	11.5	0.10 7	0.24 2	0.09 10	0.19 2	0.59 3	0.15 3	0.27 5	0.64 5	0.17 4	0.18 7	0.82 5	0.11 11
Classic++ [39]	12.2	0.09 2	0.25 3	0.07 2	0.23 13	0.78 12	0.19 13	0.43 12	1.00 14	0.22 14	0.20 12	1.11 17	0.10 7

Average endpoint error	Grove (Synthetic) all	Grove disc	Grove untext	Urban (Synthetic) all	Urban disc	Urban untext	Yosemite (Synthetic) all	Yosemite disc	Yosemite untext	Teddy (Stereo) all	Teddy disc	Teddy untext
Classic+NL [38]	0.64 1	0.93 1	0.47 1	0.52 9	1.12 3	0.33 6	0.16 21	0.13 6	0.29 27	0.49 1	0.98 1	0.74 5
MDP-Flow [30]	0.74 2	1.06 2	0.61 5	0.46 5	1.02 2	0.35 8	0.12 7	0.14 11	0.17 7	0.78 17	1.68 20	0.97 17
NL-TV-NCC [28]	0.79 5	1.16 7	0.51 3	0.78 17	1.38 8	0.48 14	0.16 21	0.15 16	0.26 19	0.55 3	1.16 3	0.55 1
Layer+dense [37]	0.87 9	1.17 8	0.94 21	0.35 1	0.95 1	0.31 4	0.16 21	0.13 6	0.28 25	0.55 3	1.17 4	0.79 6
Complementary OF [24]	0.87 9	1.25 10	0.71 8	1.46 26	1.61 13	0.73 22	0.11 5	0.12 4	0.21 13	0.60 5	1.37 8	0.80 9
Adaptive [23]	0.88 12	1.25 10	0.73 13	0.50 8	1.28 6	0.31 4	0.14 14	0.16 21	0.22 15	0.65 8	1.37 8	0.79 6
Adapt-Window [36]	0.74 2	1.07 3	0.56 4	1.78 35	1.73 18	0.95 30	0.22 33	0.16 21	0.45 36	0.70 11	1.28 5	0.88 13
Classic++ [39]	0.87 9	1.30 13	0.66 6	0.47 6	1.62 15	0.33 6	0.17 25	0.14 11	0.32 31	0.79 19	1.64 17	0.92 16

Improved model

Problem: Unweighted non-local term (median filtering) destroys fine structures that differ from the majority of neighbors

Solution: Weight neighbors adaptively according to color, flow, and occlusion state information

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

15X15 neighborhood

Color vector in LAB space [4]

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

Occlusion state determined by flow divergence and data constancy error [2]

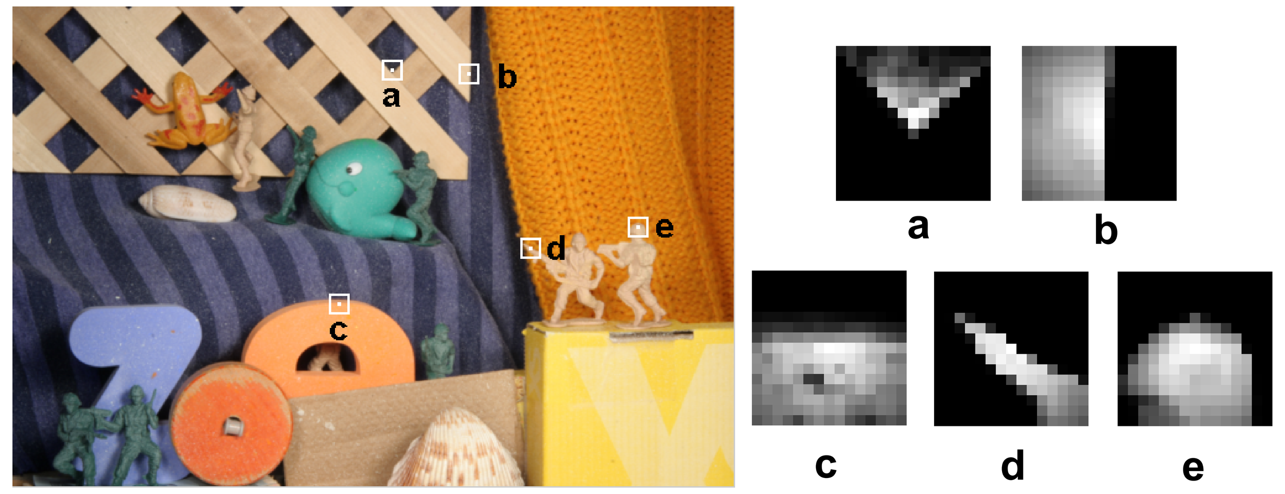


Figure 3. Neighbor weights of the proposed non-local term at different positions in the "Army" sequence.

Classic++ Classic+NL

Figure 4. Median filtering over-smooths the rifle in the "Army" sequence while the proposed non-local term preserves the details.

Conclusions

- Classical formulations competitive with current practices
- Median filtering is key to accuracy, but increases energy
- Formalize median filtering as a non-local term that integrates information over a large spatial neighborhood
- Weighting neighbors adaptively preserves motion details
- MATLAB code: http://www.cs.brown.edu/~dqsun/

References

[1] Li and Osher. *A new median formula with applications to PDE based denoising*. Commun. Math. Sci., 7(3):741–753, 2009.

[2] Sand and Teller. *Particle video: Long-range motion estimation using point trajectories*. IJCV, 2008.

[3] Wedal et al. *An improved algorithm for TV-L1 optical flow*. Dagstuhl Motion Workshop, 2008.

[4] Yoon and Kweon. *Adaptive support-weight approach for correspondence search*. PAMI, 2006.

***Acknowledgments.** DS and MJB were supported by a gift from Intel Corp. and NSF CRCNS award IIS-0904875.