Discrete Optimization for Optical Flow

Moritz Menze[1] Christian Heipke[1] Andreas Geiger[2]

[1]Institute of Photogrammetry and GeoInformation
Leibniz Universität Hannover

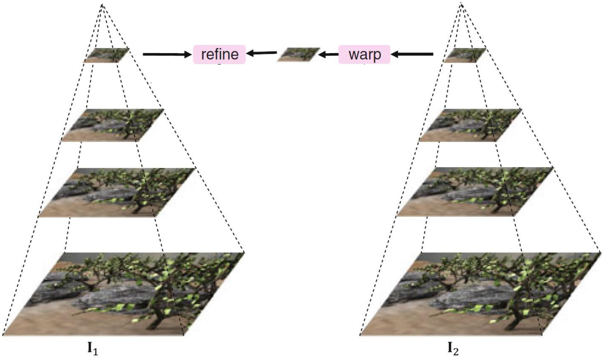
[2]Max Planck Institute for Intelligent Systems
Tübingen

7. October, 2015

Where is Sintel going?

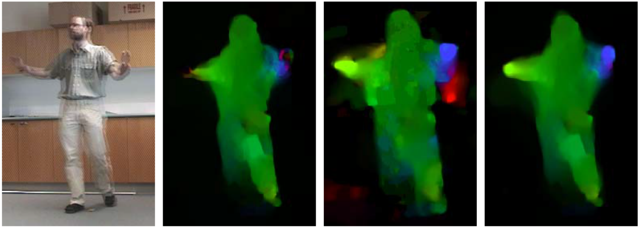
Where is Sintel going?

Related Work - Variational Methods

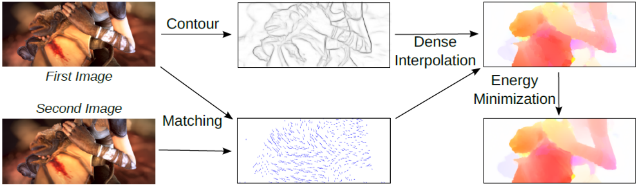


[Brox et al., ECCV 2004], [Sun et al., IJCV 2013]

Related Work - Sparse Features



[Brox et al., PAMI 2011]



[Revaud et al., CVPR 2015]

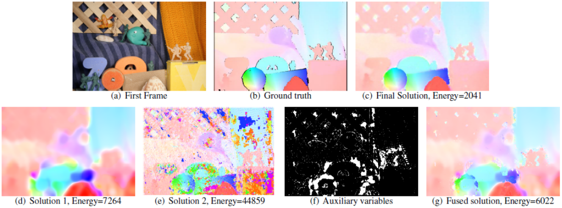
Related Work - Stereo Matching

KITTI Stereo Leaderboard

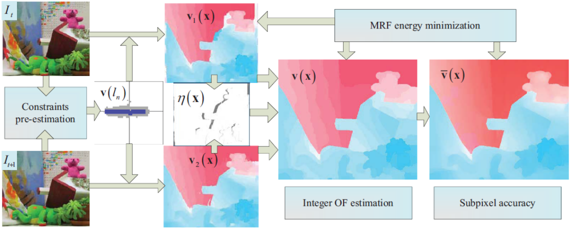
	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	MC-CNN2			2.43 %	3.63 %	0.7 px	0.9 px	100.00 %	120 s	Nvidia GTX Titan (CUDA, Lua/Torch7)	
J. Zbontar and Y. LeCun: . .											
2	Displets		code	2.47 %	3.27 %	0.7 px	0.9 px	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)	
F. Guney and A. Geiger: Displets: Resolving Stereo Ambiguities using Object Knowledge. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
3	MC-CNN			2.61 %	3.84 %	0.8 px	1.0 px	100.00 %	100 s	Nvidia GTX Titan (CUDA, Lua/Torch7)	
J. Zbontar and Y. LeCun: Computing the Stereo Matching Cost with a Convolutional Neural Network. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
4	PRSM		code	2.78 %	3.00 %	0.7 px	0.7 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.											
5	SPS-StFl			2.83 %	3.64 %	0.8 px	0.9 px	100.00 %	35 s	1 core @ 3.5 Ghz (C/C++)	
K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation. ECCV 2014.											
6	VC-SF			3.05 %	3.31 %	0.8 px	0.8 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, S. Roth and K. Schindler: View-Consistent 3D Scene Flow Estimation over Multiple Frames. Proceedings of European Conference on Computer Vision. Lecture Notes in, Computer Science 2014.											
7	Deep Embed			3.10 %	4.24 %	0.9 px	1.1 px	100.00 %	3 s	1 core @ 2.5 Ghz (C/C++)	
Z. Chen, X. Sun, Y. Yu, L. Wang and C. Huang: A Deep Visual Correspondence Embedding Model for Stereo Matching Costs. ICCV 2015.											
8	JSOSM			3.15 %	3.94 %	0.8 px	0.9 px	100.00 %	105 s	8 cores @ 2.5 Ghz (C/C++)	
Anonymous submission											
9	OSF		code	3.28 %	4.07 %	0.8 px	0.9 px	99.98 %	50 min	1 core @ 3.0 Ghz (Matlab + C/C++)	
M. Menze and A. Geiger: Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR) 2015.											
10	CoR		code	3.30 %	4.10 %	0.8 px	0.9 px	100.00 %	6 s	6 cores @ 3.3 Ghz (Matlab + C/C++)	
A. Chakrabarti, Y. Xiong, S. Gortler and T. Zickler: Low-level Vision by Consensus in a Spatial Hierarchy of Regions. CVPR 2015.											

- Naïve discretization of 2D flow space intractable
- 3 strategies enable discrete optimization for optical flow

Related Work - Discrete Optimization



[Lempitzky et al., CVPR 2008]



[Mozerov, TIP 2013]

Discrete Optical Flow

Optical flow as labeling task

$$E(\mathbf{l}) = \sum_{\mathbf{p}\in\mathcal{P}} \underbrace{\varphi_{\mathbf{p}}(l_{\mathbf{p}})}_{\text{data}} + \sum_{\mathbf{p}\sim\mathbf{q}} \underbrace{\psi_{\mathbf{p},\mathbf{q}}(l_{\mathbf{p}}, l_{\mathbf{q}})}_{\text{smoothness}}$$

- $\mathbf{p}, \mathbf{q}$: pixels
- l: discrete flow label

Discrete Optical Flow

- Robust data term based on DAISY descriptors $\mathbf{d}$

$$\varphi_{\mathbf{p}}(l_{\mathbf{p}}) = \min\left(\left\|\mathbf{d}_{\mathbf{p}} - \mathbf{d}'_{\mathbf{p}}(l_{\mathbf{p}})\right\|_1, \tau_\varphi\right)$$

- Similar flow vectors $\mathbf{f}$ are encouraged by

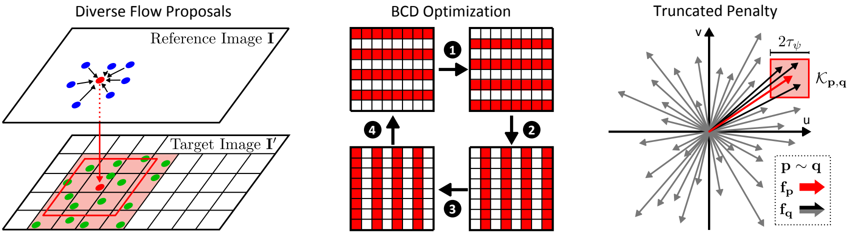
$$\psi_{\mathbf{p},\mathbf{q}}(l_{\mathbf{p}}, l_{\mathbf{q}}) = w_{\mathbf{p},\mathbf{q}} \cdot \min\left(\left\|\mathbf{f}_{\mathbf{p}}(l_{\mathbf{p}}) - \mathbf{f}_{\mathbf{q}}(l_{\mathbf{q}})\right\|_1, \tau_\psi\right)$$

  weighted by the edge strength [Dollár et al., ICCV 2013]:

$$w_{\mathbf{p},\mathbf{q}} = \exp\left(-\alpha\, \kappa^2_{\mathbf{p},\mathbf{q}}\right)$$

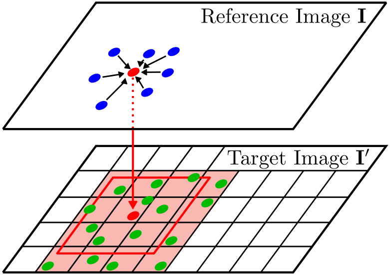
Strategies

Three strategies for efficient inference:



1. Diverse Flow Proposals

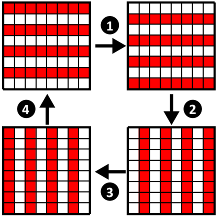
- Efficient search structure
- 300 nearest neighbors
- 200 proposals from neighboring pixels



- Reduction of proposals form 250.000 to 500 per pixel

2. Block Coordinate Descent

- Inference via Block Coordinate Descent (BCD)[1]

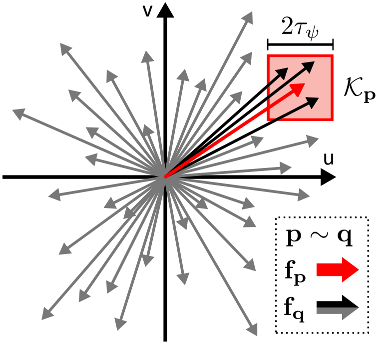


- Alternating optimization of image rows and columns
- Sub-problems solved optimally via dynamic programming

[1] Chen and Koltun, CVPR 2014

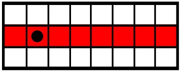
3. Truncated Pairwise Potential

- Evaluation of only a few combinations



3. Truncated Pairwise Potential

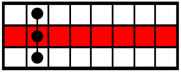
Naïve Dynamic Programming:



$$\mathbf{C}(x, l) = \varphi_{(x,y)}(l)$$

3. Truncated Pairwise Potential

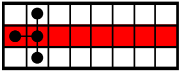
Naïve Dynamic Programming:



$$\begin{aligned}\mathbf{C}(x, l) = {} & \varphi_{(x,y)}(l) \\ & + \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) \\ & + \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1})\end{aligned}$$

3. Truncated Pairwise Potential

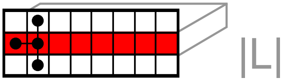
Naïve Dynamic Programming:



$$
\begin{aligned}
\mathbf{C}(x, l) = \varphi_{(x,y)}(l) & \\
&+ \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) \\
&+ \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1}) \\
+ \min_{0 \leq k < L} \big(\psi_{(x,y),(x-1,y)}(l, k) + \mathbf{C}(x-1, k)\big) &
\end{aligned}
$$

3. Truncated Pairwise Potential

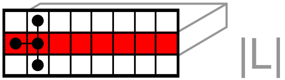
Naïve Dynamic Programming:



$$\begin{aligned}\mathbf{C}(x, l) = \varphi_{(x,y)}(l) & \\ + \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) & \\ + \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1}) & \\ + \min_{0 \le k < L} \left(\psi_{(x,y),(x-1,y)}(l, k) + \mathbf{C}(x-1, k)\right) &\end{aligned}$$

3. Truncated Pairwise Potential

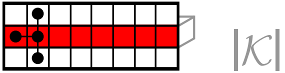
Efficient Dynamic Programming:



$$\mathbf{C}(x, l) = \varphi_{(x,y)}(l) + \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) + \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1})$$

3. Truncated Pairwise Potential

Efficient Dynamic Programming:

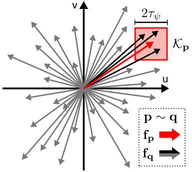


$$\mathbf{C}(x,l) = \varphi_{(x,y)}(l) + \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) + \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1})$$

$$+ \min\left(\min_{k \in \mathcal{K}_{(x,y),(x-1,y),l}} \left(\psi_{(x,y),(x-1,y)}(l,k) + \mathbf{C}(x-1,k)\right), c\right)$$

3. Truncated Pairwise Potential

Efficient Dynamic Programming:



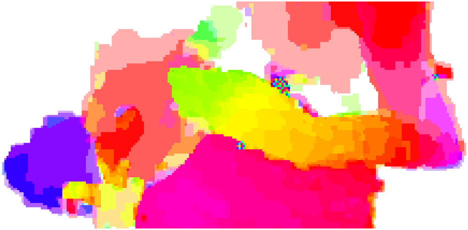
$$\begin{aligned}\mathbf{C}(x, l) = \varphi_{(x,y)}(l) \\ + \psi_{(x,y),(x,y-1)}(l, l^*_{x,y-1}) \\ + \psi_{(x,y),(x,y+1)}(l, l^*_{x,y+1})\end{aligned}$$

$$+ \min\left(\min_{k \in \mathcal{K}_{(x,y),(x-1,y),l}} \left(\psi_{(x,y),(x-1,y)}(l, k) + \mathbf{C}(x-1, k)\right), c\right)$$

with: $$c = \min_{0 \le k < L} \left(w_{(x,y),(x-1,y)}\tau_\varphi + \mathbf{C}(x-1, k)\right)$$

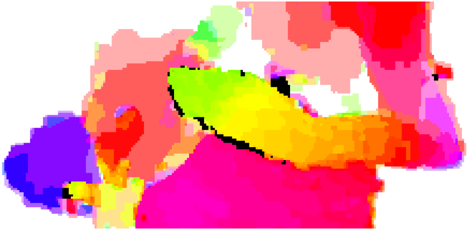
Post-Processing

- Inference yields integral flow



Post-Processing

- Inference yields integral flow
- Forward-backward consistency check



Post-Processing

- Inference yields integral flow
- Forward-backward consistency check
- Interpolation & subpixel refinement
  - Epicflow [Revaud, CVPR 2015]

Quantitative Evaluation - MPI Sintel

	EPE (px)		
	All	Noc	Occ
Flow Fields [Bailer et al., ICCV 2015]	**5.810**	**2.621**	31.799
Ours+EpicFlow	6.077	2.937	**31.685**
DM+EpicFlow [Revaud et al., CVPR 2015]	6.285	3.060	32.564
TF+OFM [Kennedy et al., EMMCVPR 2015]	6.727	3.388	33.929
Deep+R [Drayer et al., BMVC 2015]	6.769	2.996	37.494

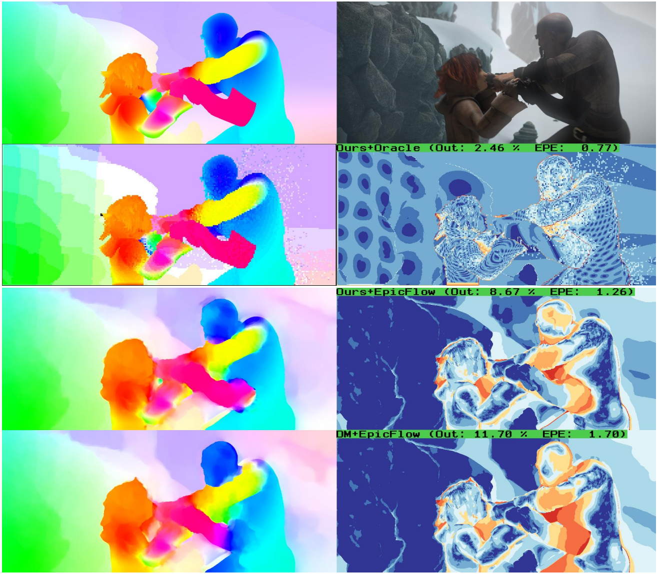
MPI Sintel final (48 methods listed online)

Quantitative Evaluation - MPI Sintel

	EPE (px)		
	All	Noc	Occ
Ours+EpicFlow	**3.567**	1.108	**23.626**
FlowFields [Bailer et al., ICCV 2015]	3.748	**1.056**	25.700
DM+EpicFlow [Revaud et al., CVPR 2015]	4.115	1.360	26.595
PH-Flow [Yang et al., CVPR 2015]	4.388	1.714	26.202
AggregFlow [Fortun et al., ARXIV 2014]	4.754	1.694	29.685

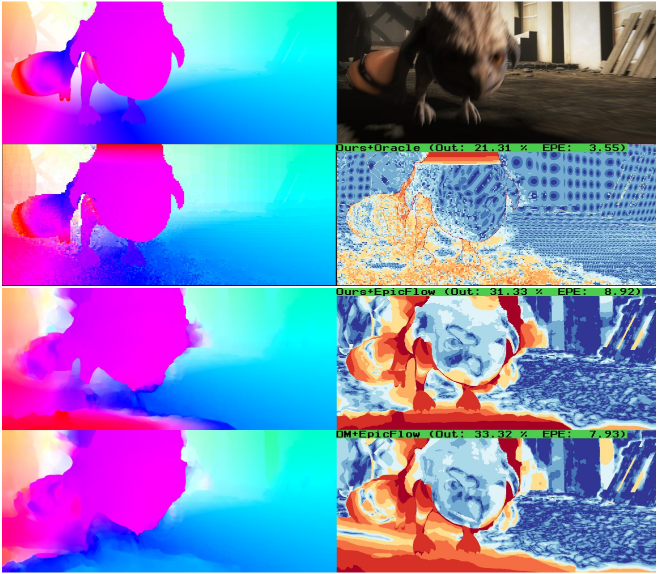
MPI Sintel clean (48 methods listed online)

Qualitative Results - MPI Sintel



Qualitative Results - MPI Sintel

Qualitative Results - MPI Sintel



Quantitative Evaluation - KITTI

	Outliers (%)		EPE (px)	
	Noc	All	Noc	All
PH-Flow [Yang et al., CVPR 2015]	**5.76**	**10.57**	**1.3**	2.9
FlowFields [Bailer et al., ICCV 2015]	5.77	14.01	1.4	3.5
NLTGV-SC [Ranftl et al., ECCV 2014]	5.93	11.96	1.6	3.8
DDS-DF [Wei et al., 3DV 2014]	6.03	13.08	1.6	4.2
TGV2ADCSIFT [Braux-Zin et al., ICCV 2013]	6.20	15.15	1.5	4.5
Ours+EpicFlow	6.23	16.63	**1.3**	3.6
AnyFlow [Submitted to PAMI]	6.37	15.80	1.5	4.3
BTF-ILLUM [Demetz et al., ECCV 2014]	6.52	11.03	1.5	**2.8**
CRT-TGV [Submitted to IJCV]	6.71	12.09	2.0	3.9
⋮	⋮	⋮	⋮	⋮
DM+EpicFlow [Revaud et al., CVPR 2015]	7.88	17.08	1.5	3.8

KITTI 3 px (69 methods listed online)

Quantitative Evaluation - KITTI

	Outliers (%)		EPE (px)	
	Noc	All	Noc	All
Ours+EpicFlow	**3.89**	12.46	**1.3**	3.6
PH-Flow [Yang et al., CVPR 2015]	3.93	**7.72**	**1.3**	2.9
FlowFields [Bailer et al., ICCV 2015]	3.95	10.21	1.4	3.5
DDS-DF [Wei et al., 3DV 2014]	4.41	10.41	1.6	4.2
NLTGV-SC [Ranftl et al., ECCV 2014]	4.50	9.42	1.6	3.8
AnyFlow [Submitted to PAMI]	4.51	12.55	1.5	4.3
TGV2ADCSIFT [Braux-Zin et al., ICCV 2013]	4.60	12.17	1.5	4.5
BTF-ILLUM [Demetz et al., ECCV 2014]	4.64	8.11	1.5	**2.8**
CRT-TGV [Submitted to IJCV]	5.01	8.97	2.0	3.9
⋮	⋮	⋮	⋮	⋮
DM+EpicFlow [Revaud et al., CVPR 2015]	5.36	12.86	1.5	3.8

KITTI 5 px (69 methods listed online)

Conclusions

- Three strategies to reduce computational complexity
- Enable discrete optimization for optical flow
- State-of-the-art performance with sub-pixel refinement

Outlook

- Integrate multiple frames
- Reason about multiple scales
- Include semantic knowledge

Discrete Optimization for Optical Flow

Thank you!

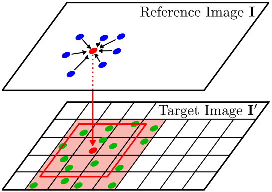
Project page:
www.cvlibs.net/projects/discrete_flow

Runtimes

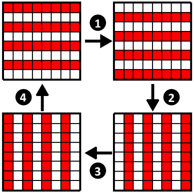
Percentage Runtime:

40% | 35%

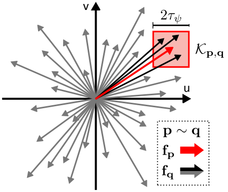
Diverse Flow Proposals



BCD Optimization



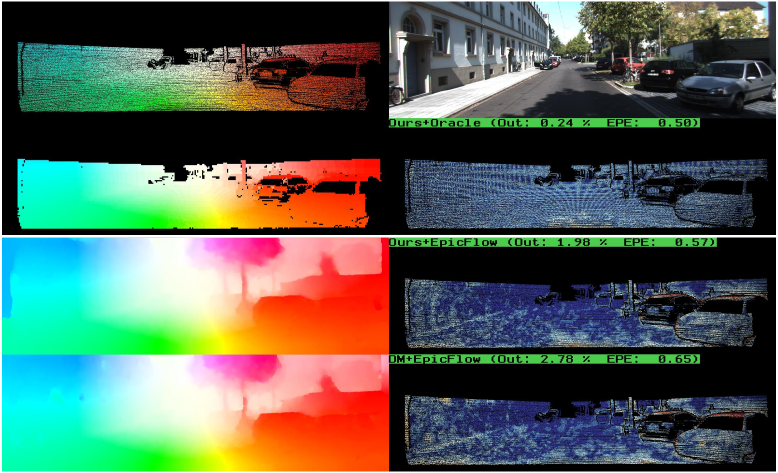
Truncated Penalty



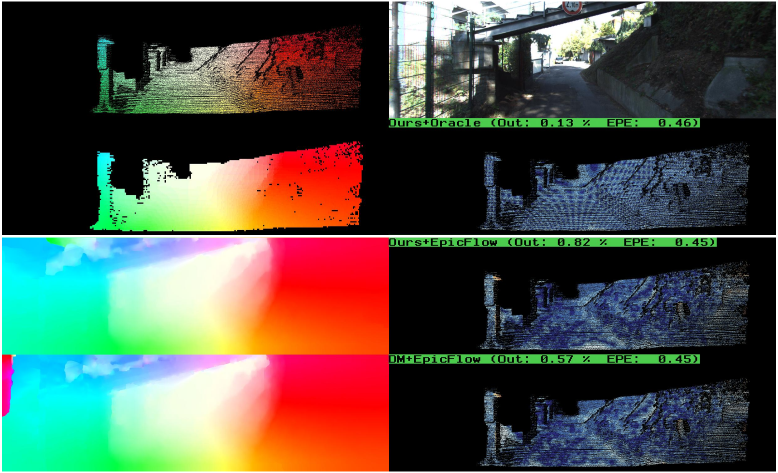
(remaining 25% for descriptor extraction and post-processing)

- Total runtime ∼3 minutes

Qualitative Results - KITTI



Qualitative Results - KITTI



Qualitative Results - KITTI