Optical Flow

Shenlong Wang

CSC2541 Course Presentation

Feb 2, 2016

Outline

- Introduction
- Variation Models
- Feature Matching Methods
- End-to-end Learning based Methods
- Discussion

Optical Flow

- Goal: Pixel motion from Image 1 to Image 2

Image credit: KITTI

Optical Flow

- Goal: Pixel motion from Image 1 to Image 2

Image credit: KITTI

Optical Flow

- Goal: Pixel motion from Image I to Image H

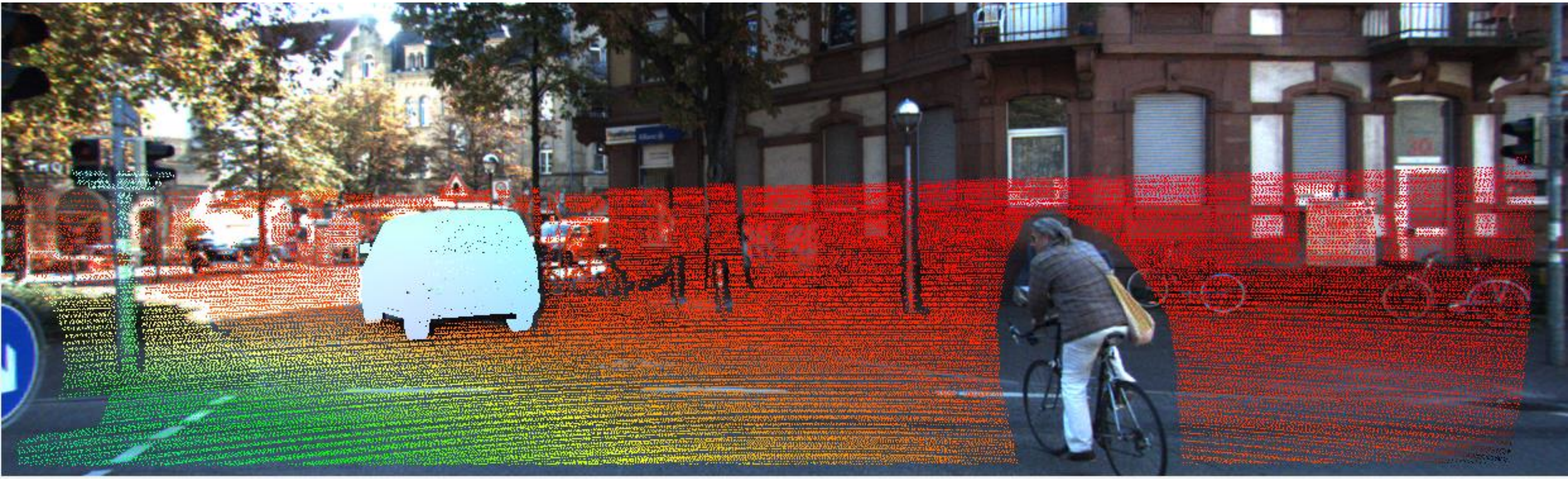


Image credit: KITTI

Example

- Goal: Pixel motion from Image I to Image H

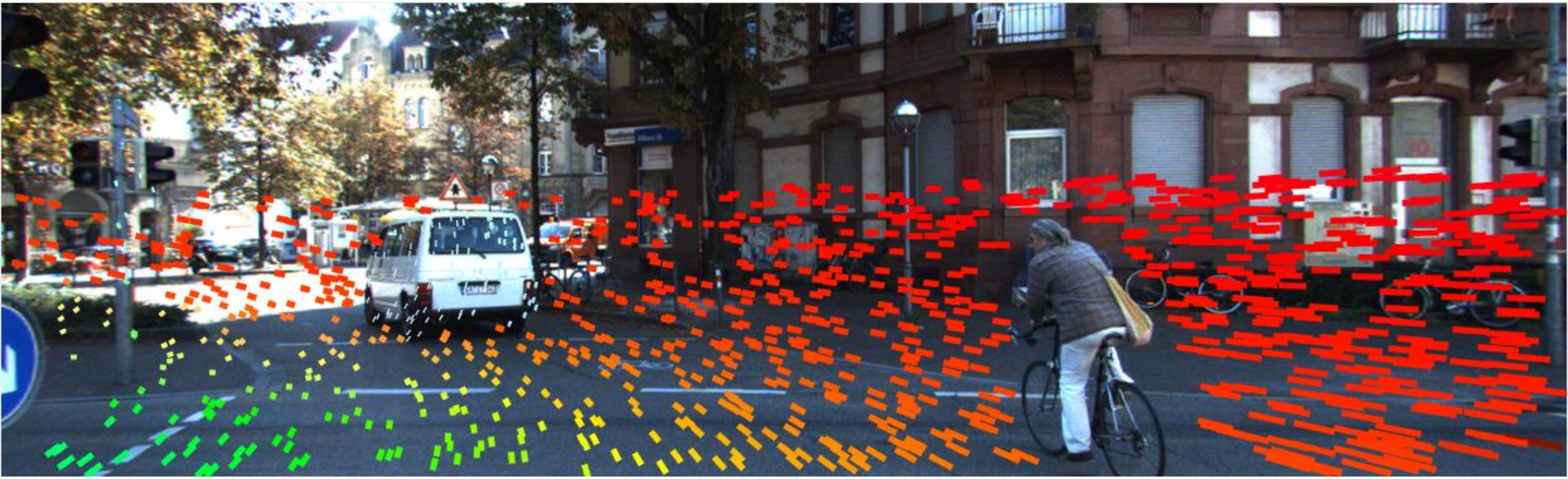


Image credit: KITTI

Why Optical Flow is Important?

- We live in a moving world

Image credit: giphy.com

Why Optical Flow is Important?

- Recognize actions in video

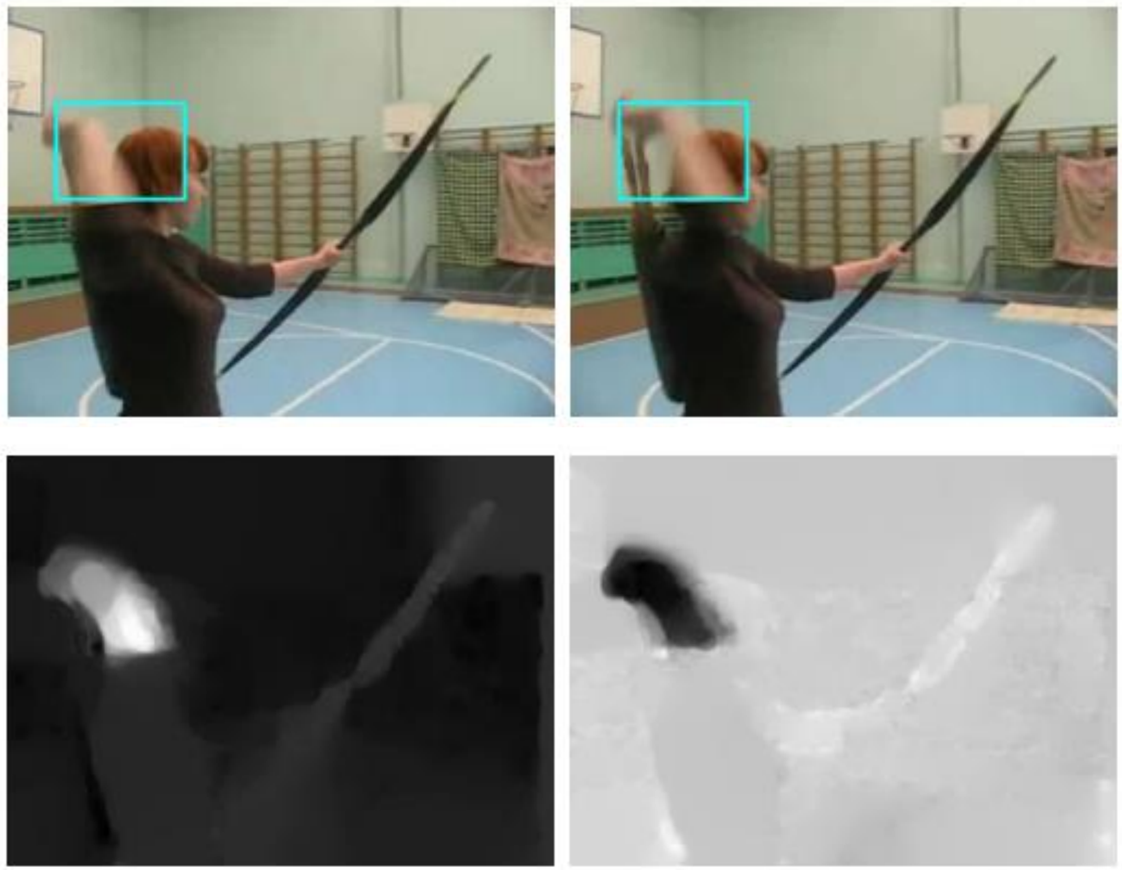
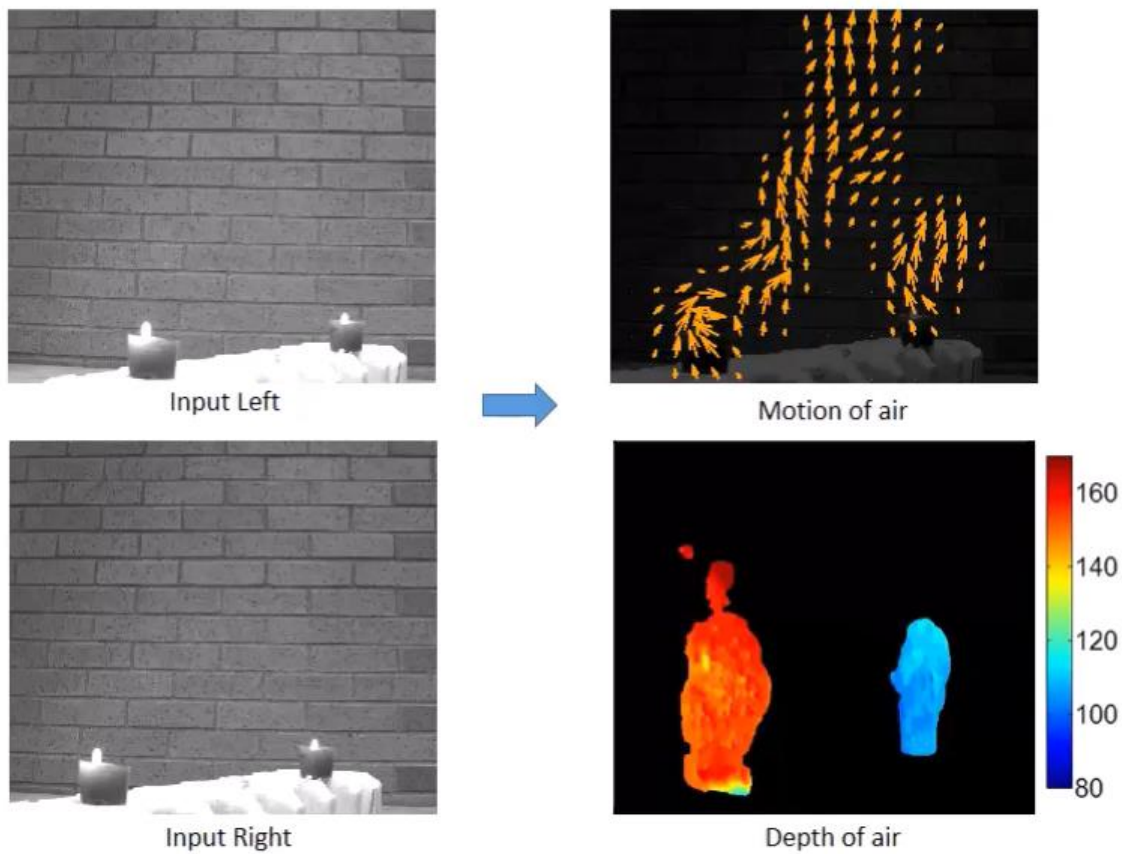


Image credit: Simonyan et al.

Why Optical Flow is Important?

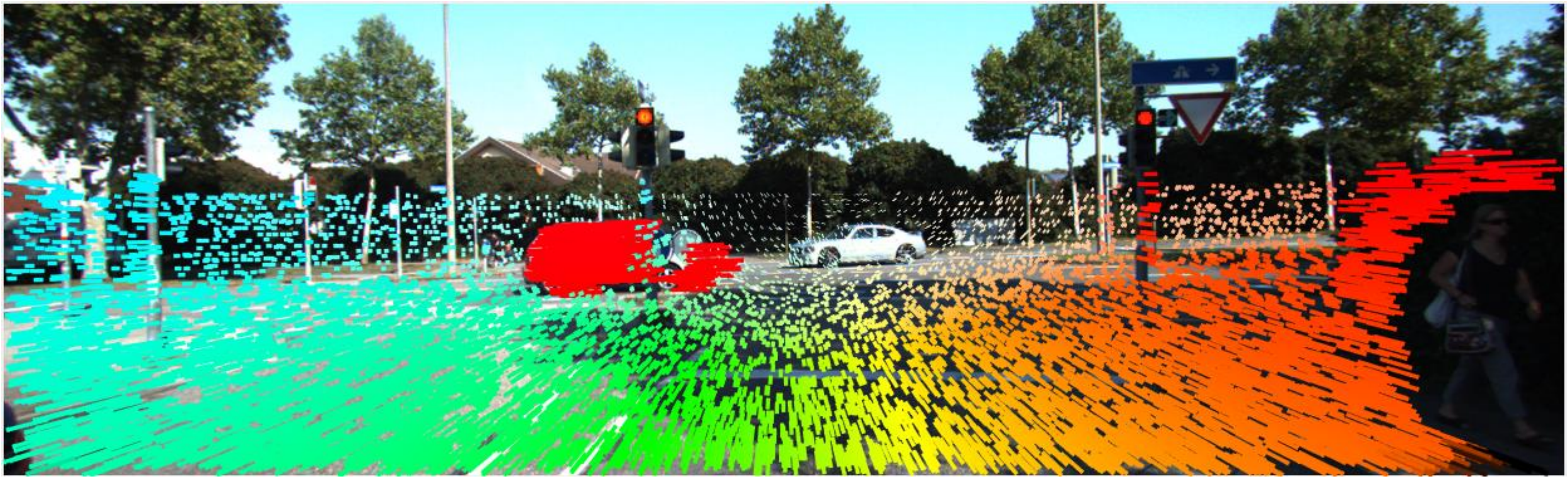
- Velocity/depth of imperceptible air motion



Video credit: Xue et al.

Optical Flow for Autonomous Driving

- Tracking motion of objects



Optical Flow for Autonomous Driving

- Tracking motion of objects

Image credit: Geiger et al.

Optical Flow for Autonomous Driving

- Estimate the motion of the car itself

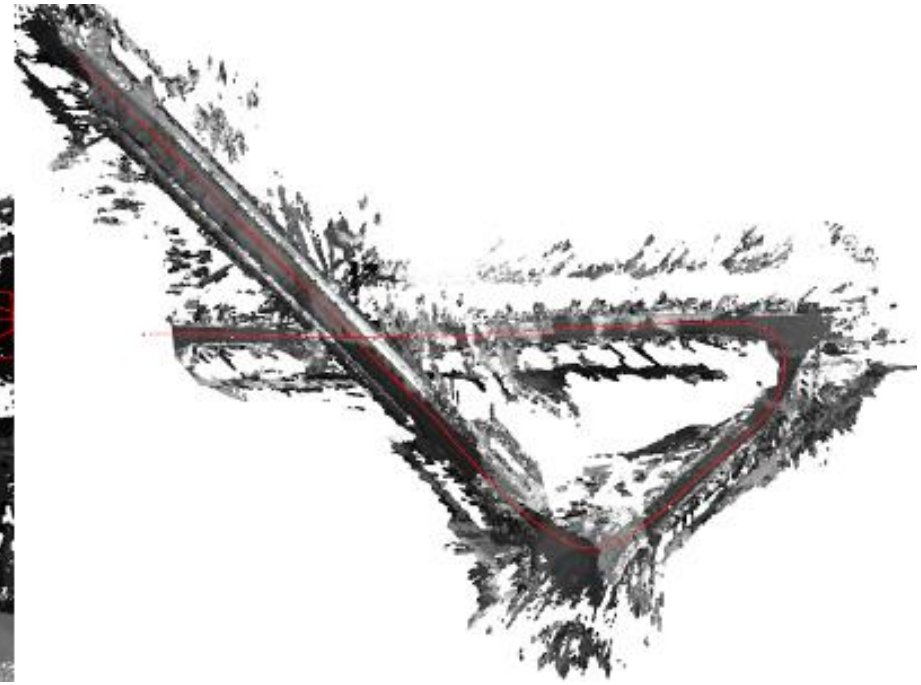
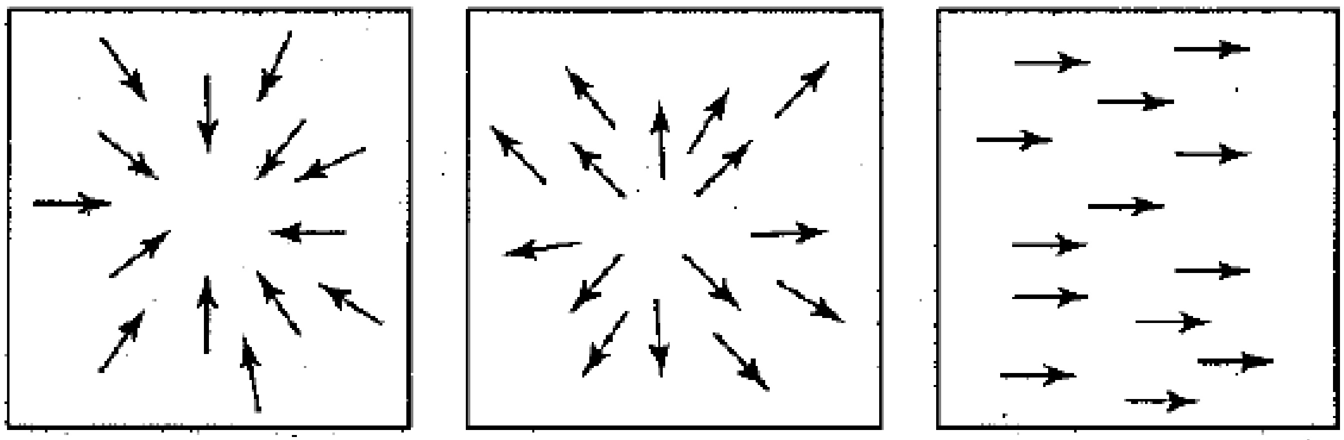


Image credit: Geiger et al.

How does it generate?

- Motion of the object + Motion of the camera



Zoom out **Zoom in** **Pan right to left**

Image credit: S. Seitz.

Motion Field

- The motion field is the projection of the 3D scene motion into the image.

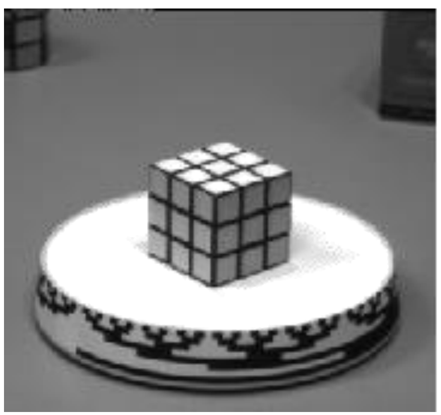
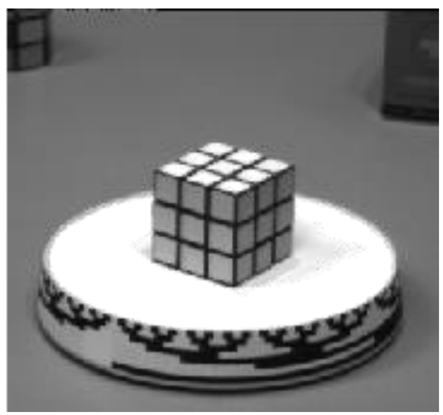
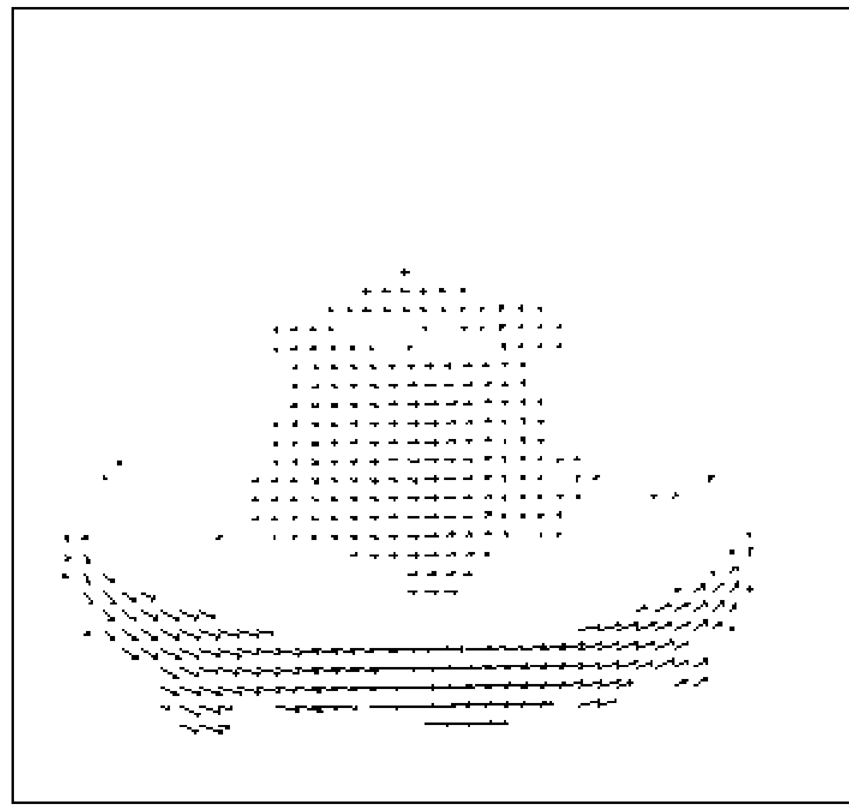




Image credit: S. Seitz.

Motion Field

- The motion field is the projection of the 3D scene motion into the image.
  - **P**(t) is a moving 3D point
  - Velocity of scene point: **V** = d**P**/dt
  - **p**(t) = ($x(t)$,$y(t)$) is the projection of **P** in the image
  - Apparent velocity **v** in the image: given by components $v_x = dx/dt$ and $v_y = dy/dt$
  - These components are known as the *motion field* of the image

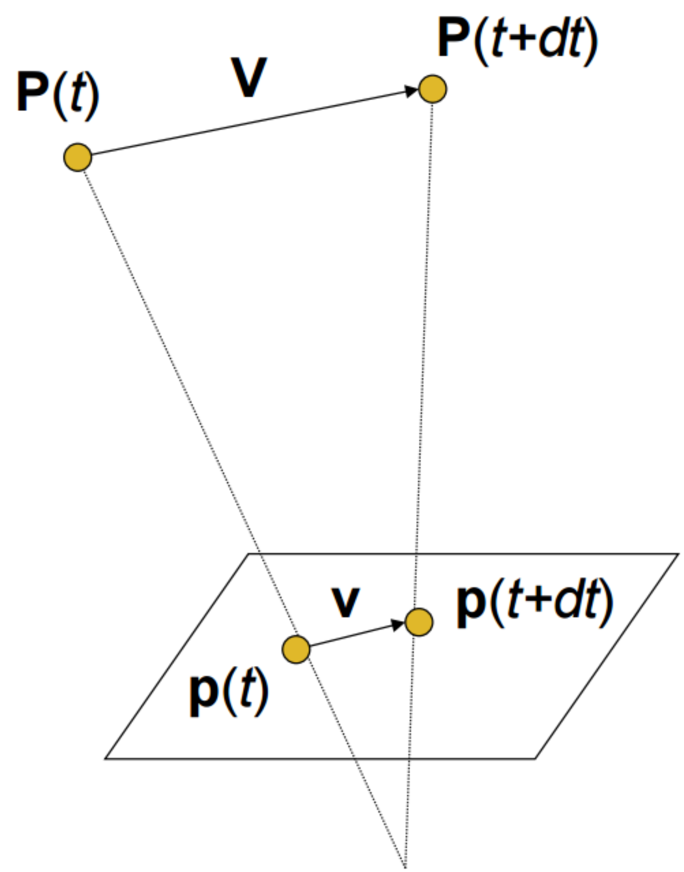


Image credit: S. Seitz.

Why Optical Flow is Difficult?

- Illumination change
- Scale change
- Large Displacement
- Occlusion
- Transparent and reflective
- Repetitive structure
- Aperture problem
- Small objects

Image credit: KITTI

Why Optical Flow is Difficult?

- Illumination change
- Scale change
- Large Displacement
- Occlusion
- Transparent and reflective
- Repetitive structure
- Aperture problem
- Small objects

Image credit: Sintel

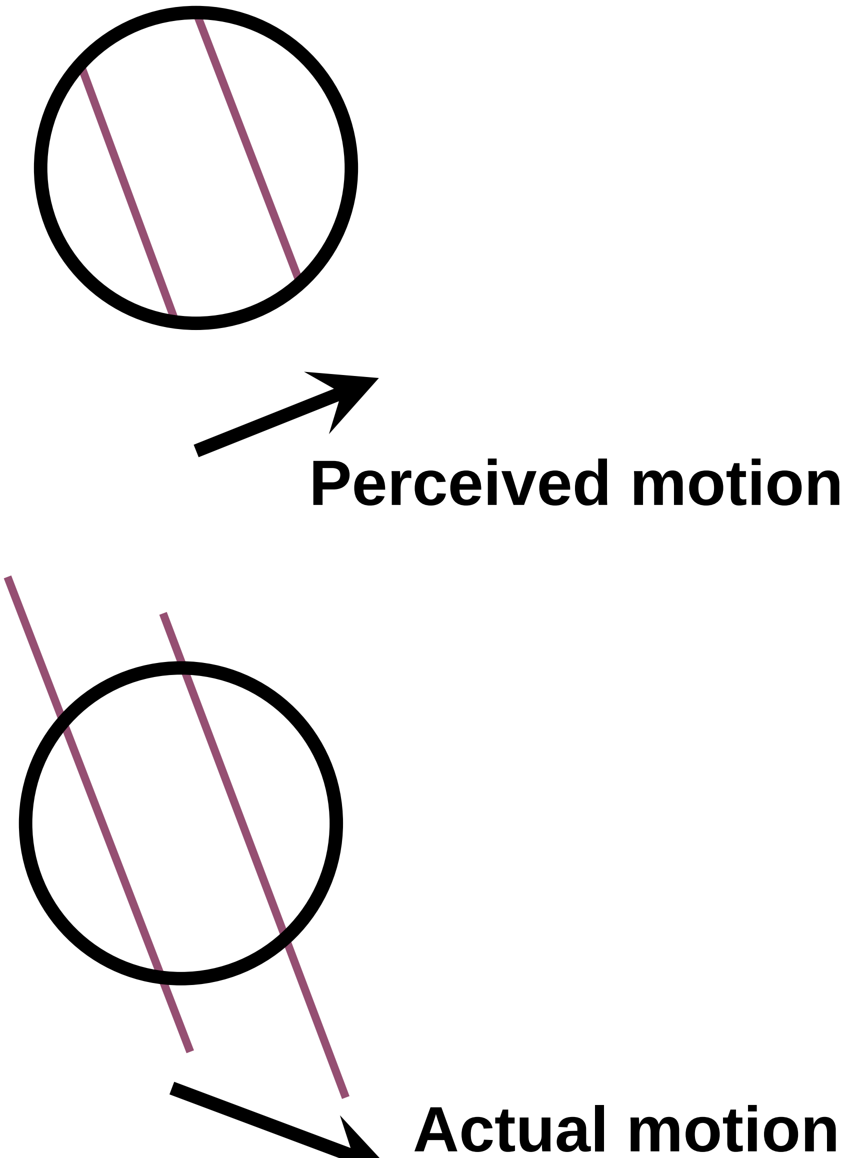
Why Optical Flow is Difficult?

- Illumination change
- Scale change
- Large Displacement
- Occlusion
- Transparent and reflective
- Repetitive structure
- Aperture problem
- Small objects

Image credit: Sintel

Why Optical Flow is Difficult?

- Illumination change
- Scale change
- Large Displacement
- Occlusion
- Transparent and reflective
- Repetitive structure
- Aperture problem
- Small objects

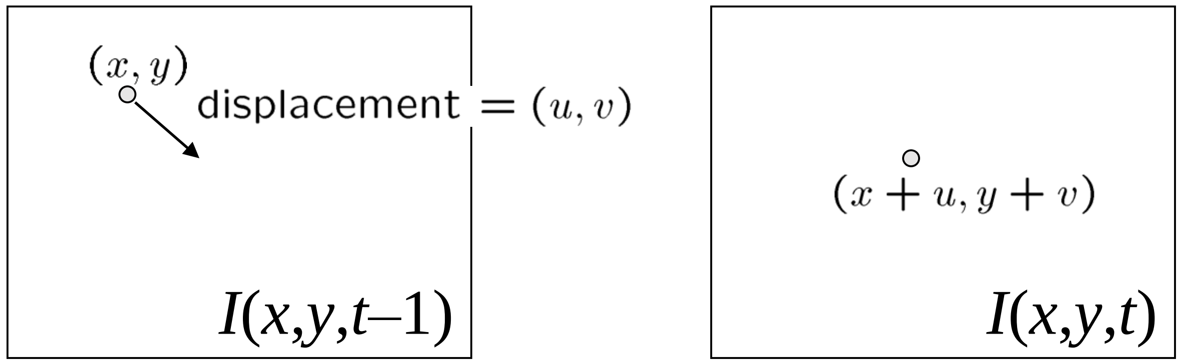


http://en.wikipedia.org/wiki/Barberpole_illusion

Key Assumptions

- Consistency: Corresponding points look similar
- Small motion: Points do not move very far
- Smoothness: Motion is locally smooth and consistent

Color Consistency



- Brightness Constancy Equation:

$$I(x, y, t-1) = I(x+u(x,y), y+v(x,y), t)$$

Can be written as:

shorthand: $I_x = \frac{\partial I}{\partial x}$

$$I(x, y, t-1) \approx I(x, y, t) + I_x \cdot u(x, y) + I_y \cdot v(x, y)$$

So, $I_x \cdot u + I_y \cdot v + I_t \approx 0$

Quiz1: How do we get that?

Quiz2: When the approx. is good?

Horn–Schunck method

- So our data term is:

$$E_{\text{data}} = \sum_{x,y} (I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y) + I_t(x,y))^2$$

- And we expect motion should be smooth:

$$E_{\text{regularization}} = \lambda \sum_{x,y} (\|\nabla u(x,y)\|^2 + \|\nabla v(x,y)\|^2)$$

- Can be solved by Euler-Lagragian Equation:

$$u^{k+1} = \overline{u}^k - \frac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \qquad v^{k+1} = \overline{v}^k - \frac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}$$

Variation models

- Essentially design continuous optimization model:

$$\min_{\mathbf{u},\mathbf{v}} E_{\mathrm{data}}(\mathbf{u},\mathbf{v}) + \lambda E_{\mathrm{regularization}}(\mathbf{u},\mathbf{v})$$

- But it will generate over-smooth the result:

Image credit: Liu et al.

Variation models

- Essentially design continuous optimization model:

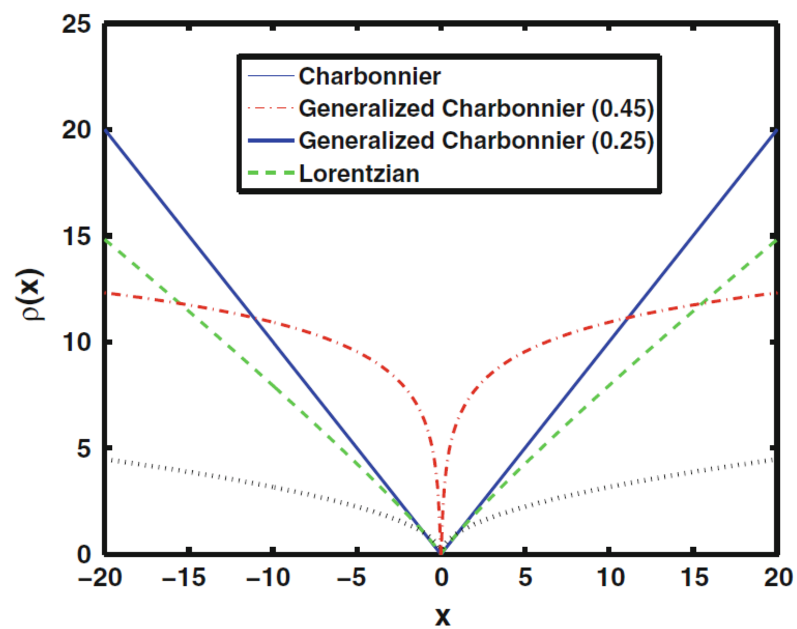
$$\min_{\mathbf{u},\mathbf{v}} E_{\text{data}}(\mathbf{u},\mathbf{v}) + \lambda E_{\text{regularization}}(\mathbf{u},\mathbf{v})$$

- So people try different smoothness penalty

$$\rho(x) = x^2$$

$$\rho(x) = \log(1 + \frac{x^2}{2\sigma^2})$$

$$\rho(x) = \sqrt{x^2 + \epsilon^2}$$



Quiz3: Why some prior works better?

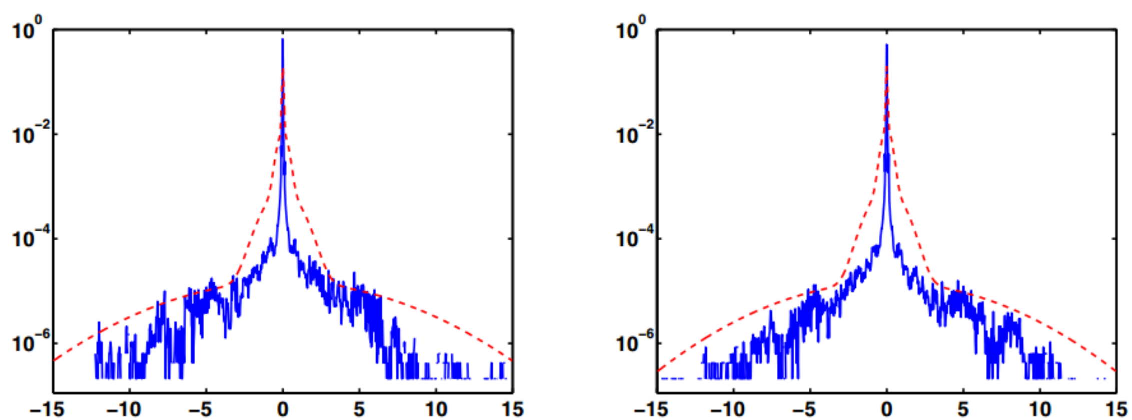
Image credit: Sun et al.

Variation models

- Essentially design continuous optimization model:

$$\min_{\mathbf{u},\mathbf{v}} E_{\text{data}}(\mathbf{u},\mathbf{v}) + \lambda E_{\text{regularization}}(\mathbf{u},\mathbf{v})$$

- So people try different smoothness penalty



(a) $\partial_x \mathbf{u}$, $\kappa = 420.5$ (b) $\partial_y \mathbf{u}$, $\kappa = 527.3$

Image credit: Sun et al.

Two tricks in Sun et al.

- Coarse-to-fine: handle large displacement
- Median filtering: “de-noise” intermediate result

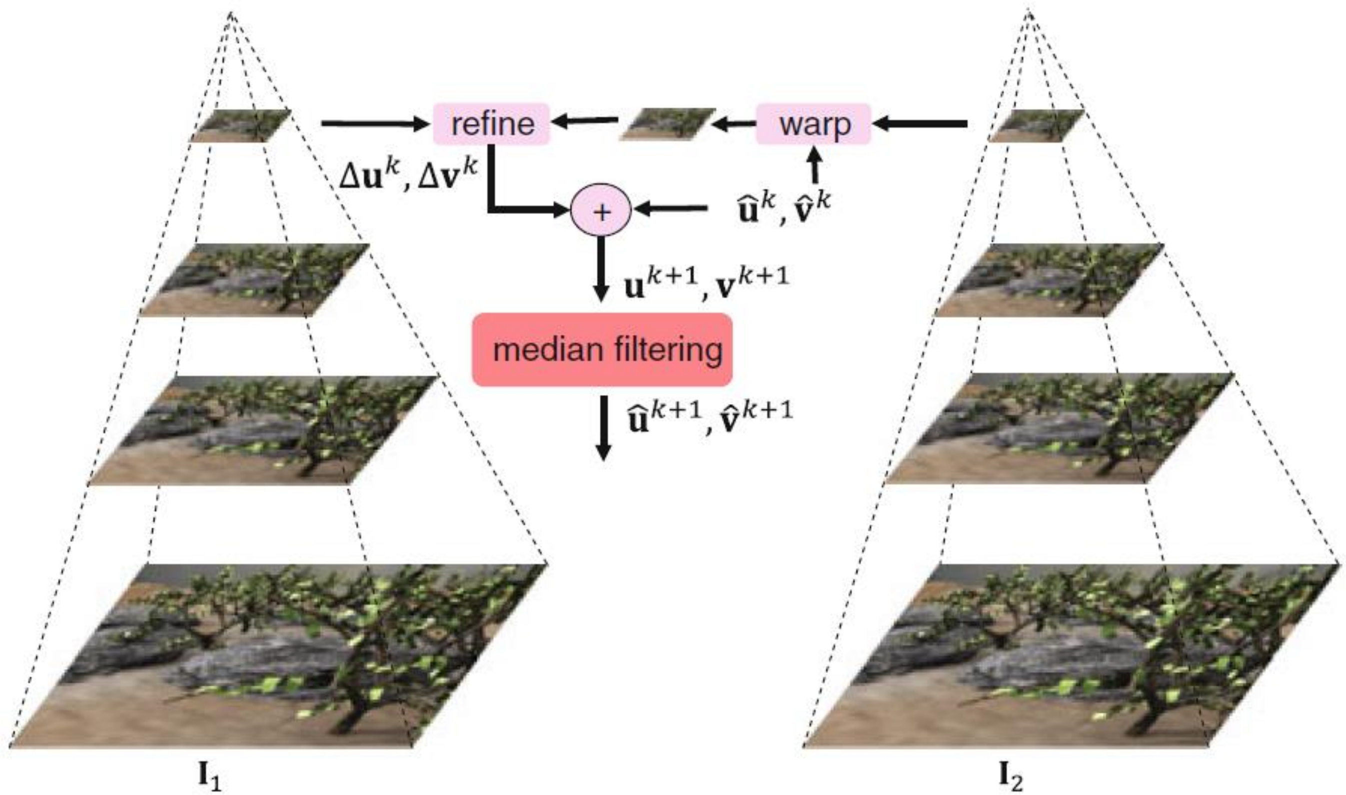


Image credit: Sun et al.

Two tricks in Sun et al.

- Coarse-to-fine
- Median filtering

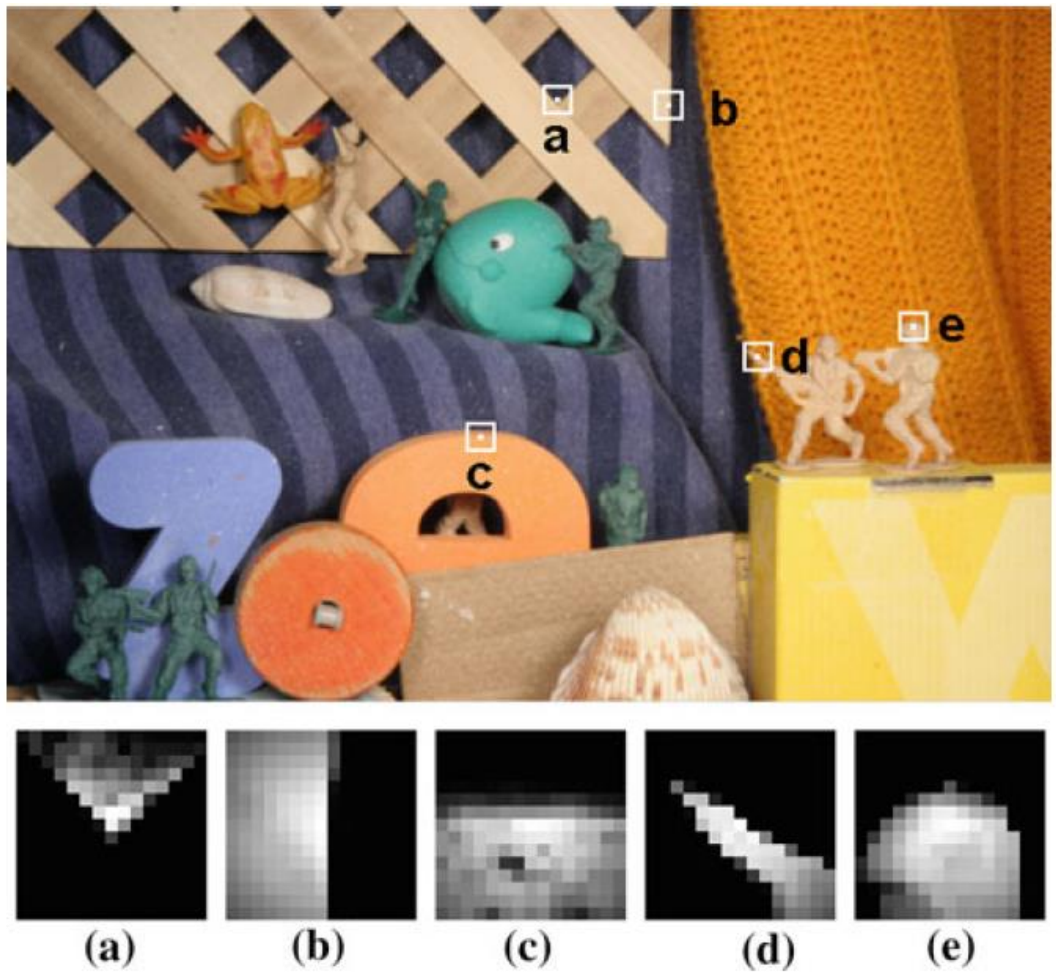
$$E = E_{\mathrm{HS}} + \gamma \underbrace{\sum_{x,y} \sum_{(x',y' \in \mathcal{N}(x,y))} (|u(x,y) - u(x',y')| + |v(x,y) - v(x',y')|)}_{\text{L1 distance of u,v between neighboring pixels}}$$

- Total variation (TV-L1) / Nonlocal TV-L1
- Although convex, minimization is not trivial
- Many PhD students (>100) have suffered from this
- Sun et al. used auxiliary variable for minimization
- Latest best method is called Primal-dual method

C Zach 2007 A Chambolle and T Pock 2011

Anisotropic weight

- Weighting the penalty by color/spatial distance



Results

Quantitative Results

- Flow metric:
  - Outlier percentage (> 3 pixel)
  - End point error

Method	Out-Noc (%)	Out-All (%)	Avg-Noc (pixel)	Avg-All (pixel)
HS	19.92	28.86	5.8	11.7
Classic+NL	24.64	33.35	9.0	16.4
HSP	14.77	24.08	4.0	9.0
Classic+NL-FastP	12.42	22.27	3.2	7.8
Classic+NLP	10.60	20.66	2.8	7.2
Classic++P	10.16	20.29	2.6	7.1

Summary of Variation Models

- Data term + smoothness term
- MAP Inference for continuous Markov random field
- Some tricks help
- Choosing a better smoothness term
- Further extensions:
  - Adopting state-of-the-art optimizers
  - Learning high-order smoothness regularization

What we haven't covered?

- Data term!

$$E_{\text{data}} = \sum_{x,y} (I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y) + I_t(x,y))^2$$

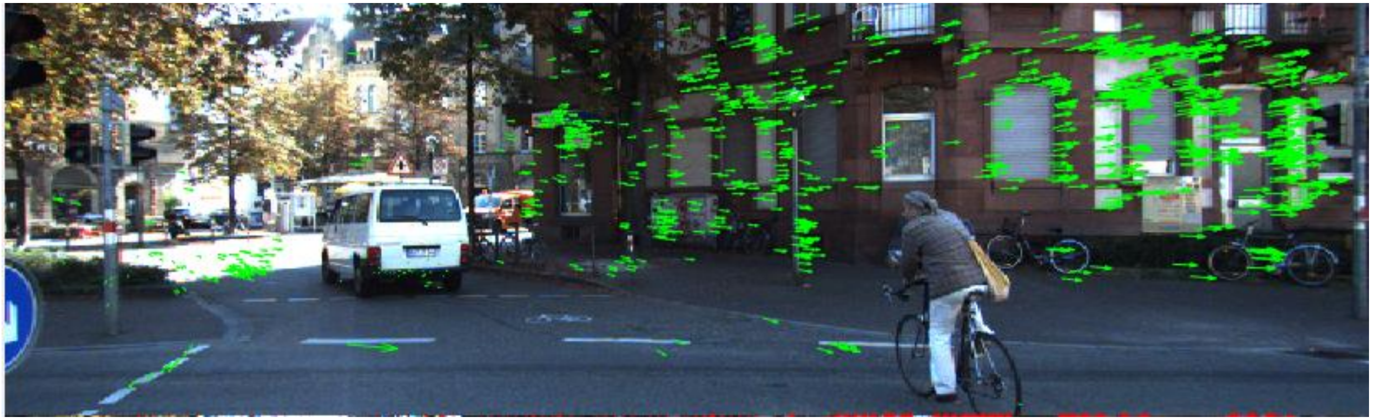
- Underlying assumption:
  - Gaussian observation noise
  - Color consistent across images
  - None of them are perfect
  - Try to warp the image with GT flow and compute the empirical distribution of the errors
- We also need a robust data term

LDOF (Brox & Malik)

- We might need robust data term:

$$\rho(x) = \sqrt{x^2 + \epsilon^2}$$

- Features should be more invariant than color:
  - HOG, SIFT, DAISY, Census, Walsh-Hamardard
- Maybe track sparse features?
  - No need to work on ambiguous regions (smooth, line)
  - We track sparse matching and propagate.



LDOF (Brox & Malik)

- Global Energy:

$$E(\mathbf{w}) = \underline{E_{\text{color}}(\mathbf{w}) + \gamma E_{\text{gradient}}(\mathbf{w}) + \alpha E_{\text{smooth}}(\mathbf{w})}$$
$$\underline{+\beta E_{\text{match}}(\mathbf{w}, \mathbf{w}_1) + E_{\text{desc}}(\mathbf{w}_1),}$$

- HOG-like / geometric blur
- What's difficult?
  - Descriptor matching results are sparse
  - The solution space is discrete
- What's proposed?
  - Auxiliary descriptor matching variable w_1
  - Only compute matching energy for sparse points

LDOF (Brox & Malik)

- Global Energy:

$$E(\mathbf{w}) = \underline{E_{\text{color}}(\mathbf{w}) + \gamma E_{\text{gradient}}(\mathbf{w}) + \alpha E_{\text{smooth}}(\mathbf{w})}$$
$$\underline{+\beta E_{\text{match}}(\mathbf{w}, \mathbf{w}_1) + E_{\text{desc}}(\mathbf{w}_1),}$$

- HOG-like / geometric blur
- What's difficult?
  - Descriptor matching results are sparse
  - The solution space is discrete
- What's proposed?
  - Auxiliary descriptor matching variable w_1
  - Only compute matching energy for sparse points

LDOF (Brox & Malik)

- Matching Energy:

$$E_{\text{match}}(\mathbf{w}) = \int \underline{\delta(\mathbf{x})}\,\underline{\rho(\mathbf{x})}\Psi\left(\underline{|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2}\right)\,d\mathbf{x}. \quad (5)$$

  - Whether there is a feature matching
  - How much we believe the descriptor matching
  - Flow should be close to feature matching result
- Optimization:
  - Discrete feature matching firstly
  - Continuous flow secondly

LDOF (Brox & Malik)

- Matching Energy:

$$E_{\text{match}}(\mathbf{w}) = \int \underline{\delta(\mathbf{x})}\,\underline{\rho(\mathbf{x})}\,\Psi\left(\underline{|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|^2}\right)\, d\mathbf{x}. \quad (5)$$

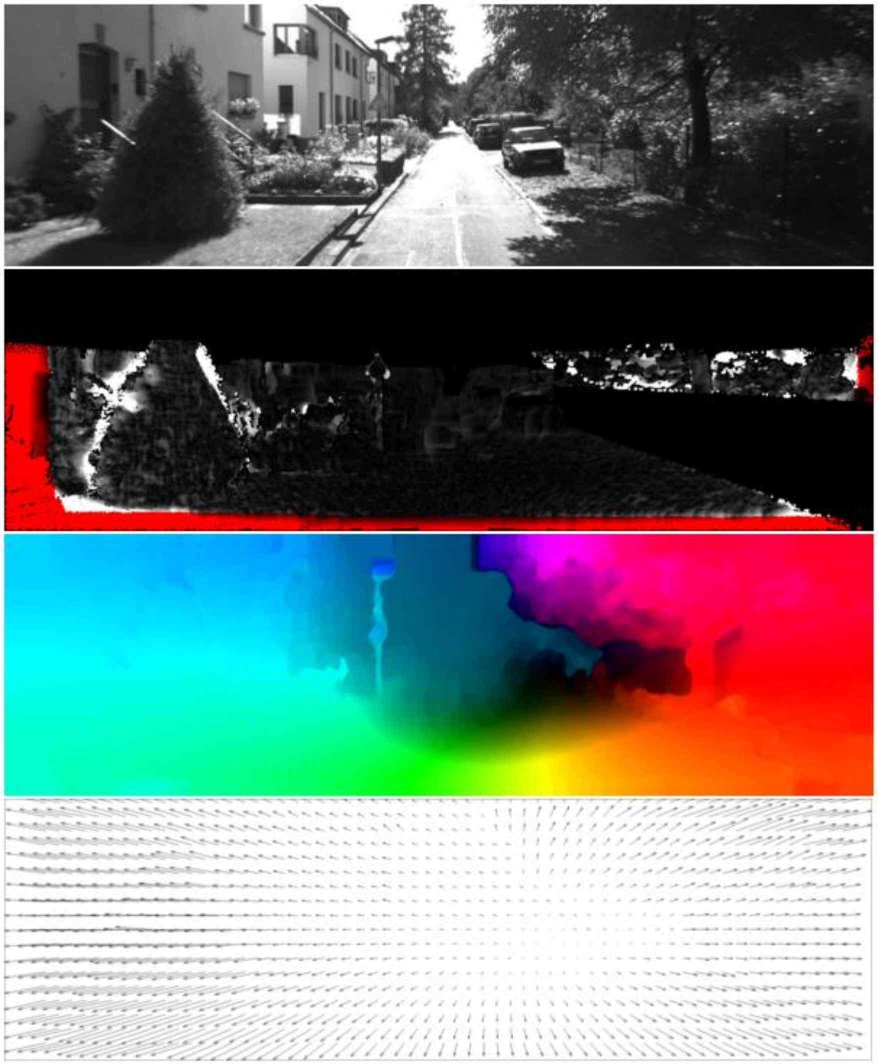
  - Whether there is a feature matching
  - How much we believe the descriptor matching
  - Flow should be close to feature matching result
- Optimization:
  - Discrete feature matching firstly
  - Continuous flow secondly

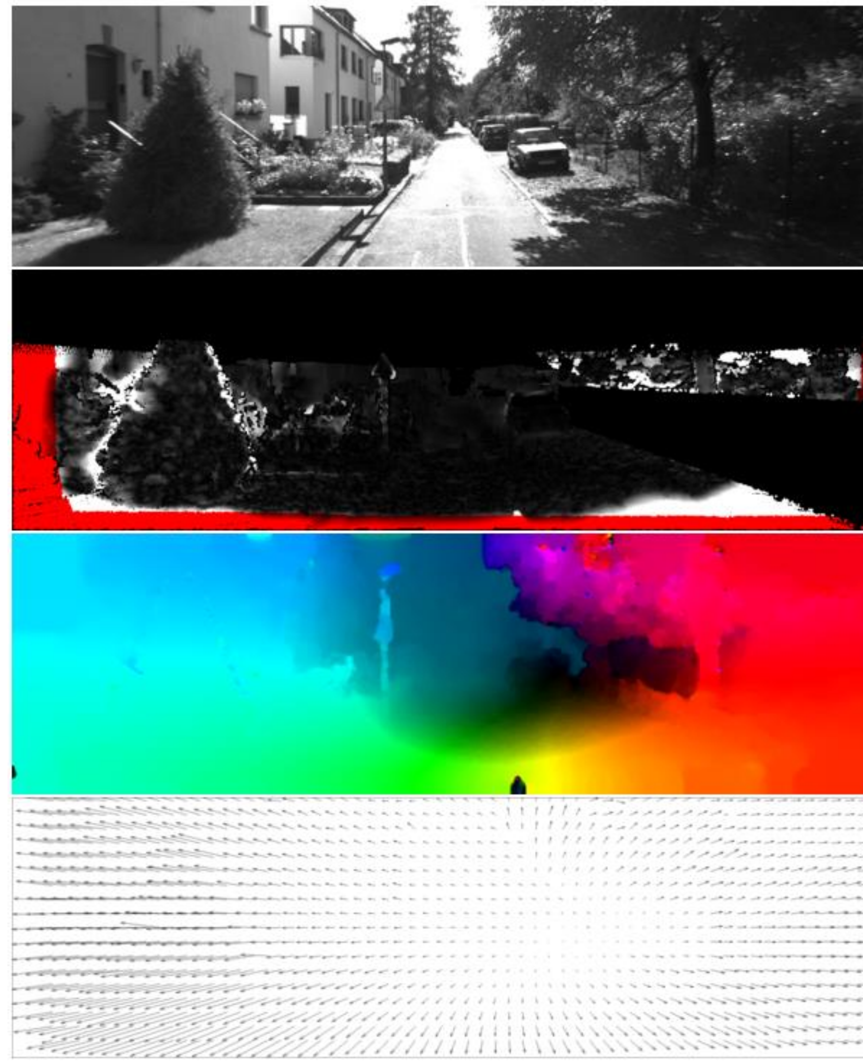
Quantitative Results

- Flow metric:
  - Average angular error

	Warping only ($\beta = 0$)	Regions	HOG	GB
Dimetrodon	1.82	1.74	1.85	1.95
Grove2	2.09	2.25	2.68	2.79
Grove3	5.59	6.55	6.38	6.35
Urban2	2.28	3.05	2.64	3.15
Urban3	3.99	5.76	5.07	5.19
RubberWhale	3.77	3.84	3.94	4.14
Hydrangea	2.32	2.36	2.44	2.54
Venus	5.19	7.37	6.45	6.52
Average	**3.38**	**4.11**	**3.93**	**4.08**

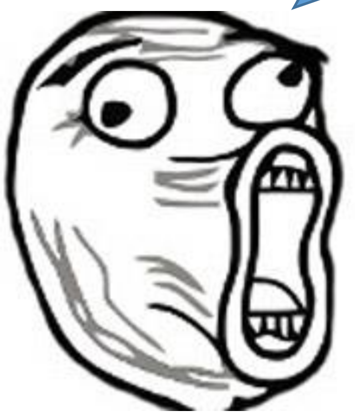
Qualitative Results





End-to-end Learning

Convolutional neural network!

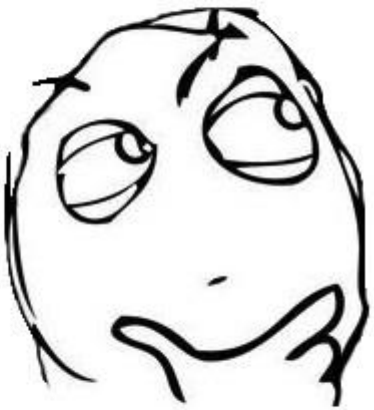


Orz

- Classification
- Detection
- Segmentation
- Boundary
- Stereo
- Action
- Depth
- Enhancing
- …

End-to-end Learning

Hmm...

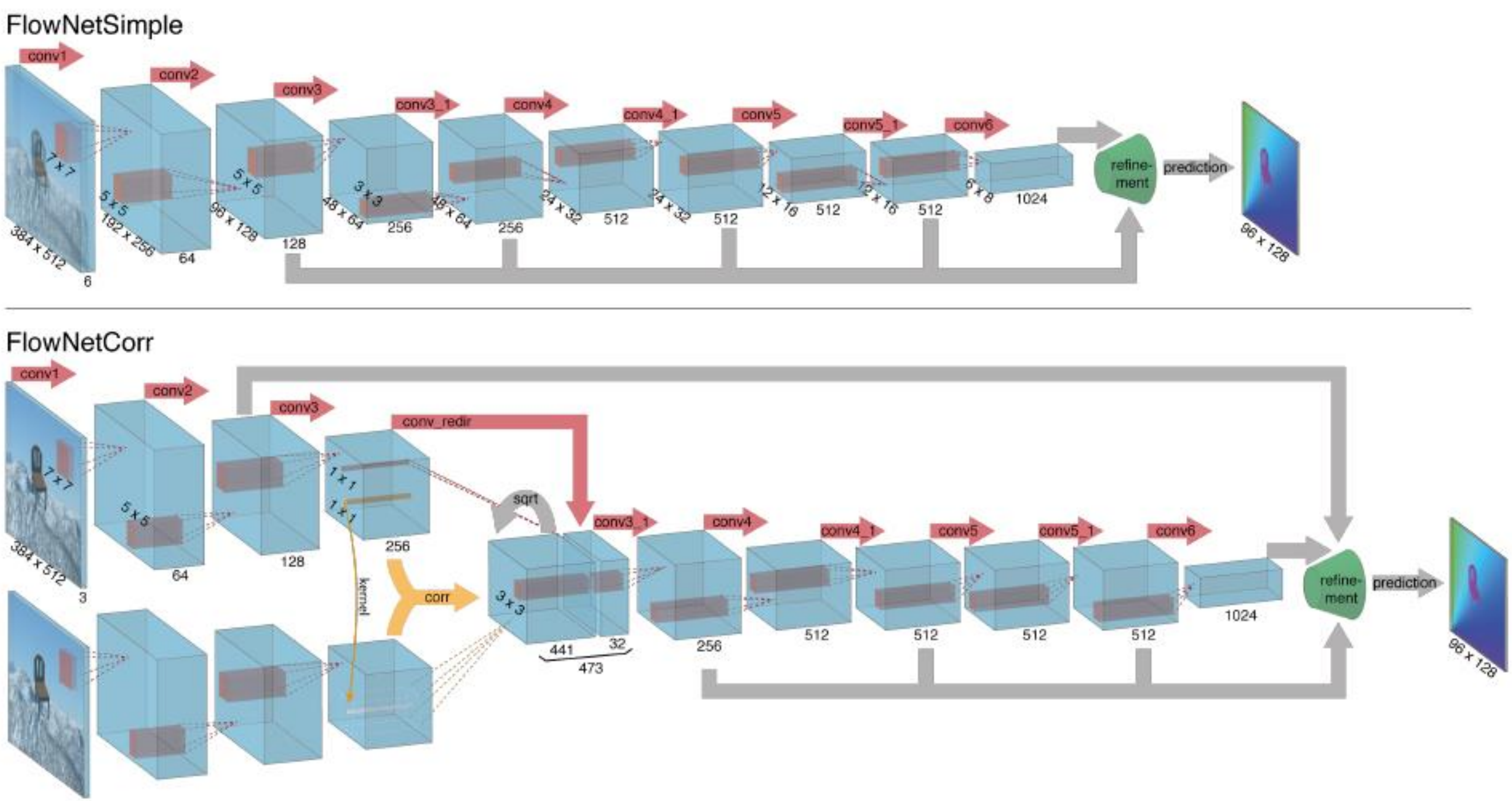


Orz

- Classification
- Detection
- Segmentation
- Boundary
- Stereo
- Action
- Depth
- Enhancing
- …
- Flow? Not yet

End-to-end Learning

- Two choice:
  - Stack them together
  - Adding a correlation layer



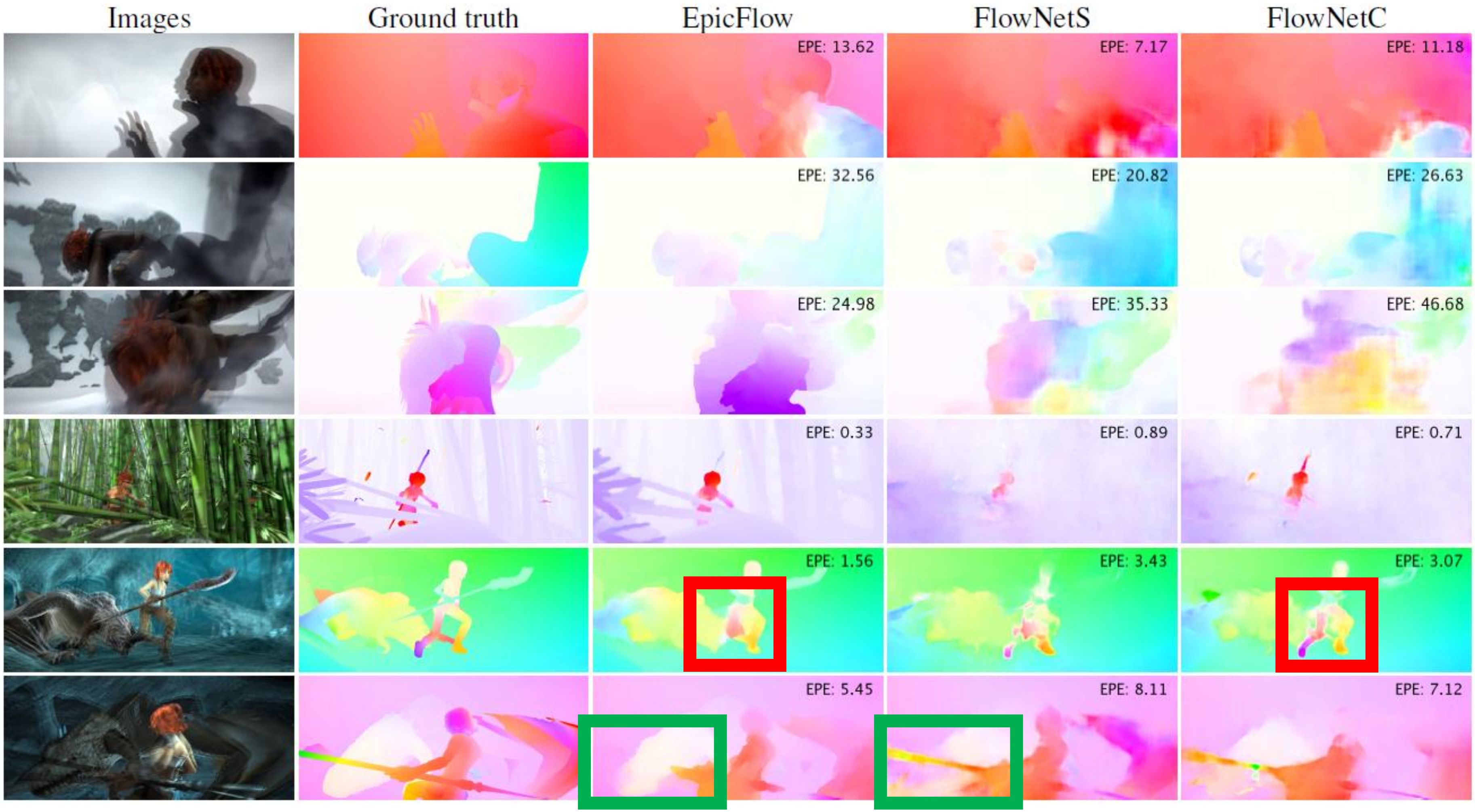
End-to-end Learning

- Difficulties:
  - Output is per-pixel prediction (proved to work)
  - It is about matching between images
  - Lack of labeled data (critical!)
  - Difficult to transfer knowledge from other tasks/dataset
- Solutions:
  - Full convolutional architecture + up-sampling layer
  - Stack channels / Correlation layer
  - Trained on Synthetic data
  - Fine-tuned on Sintel, transfer to real-world datasets

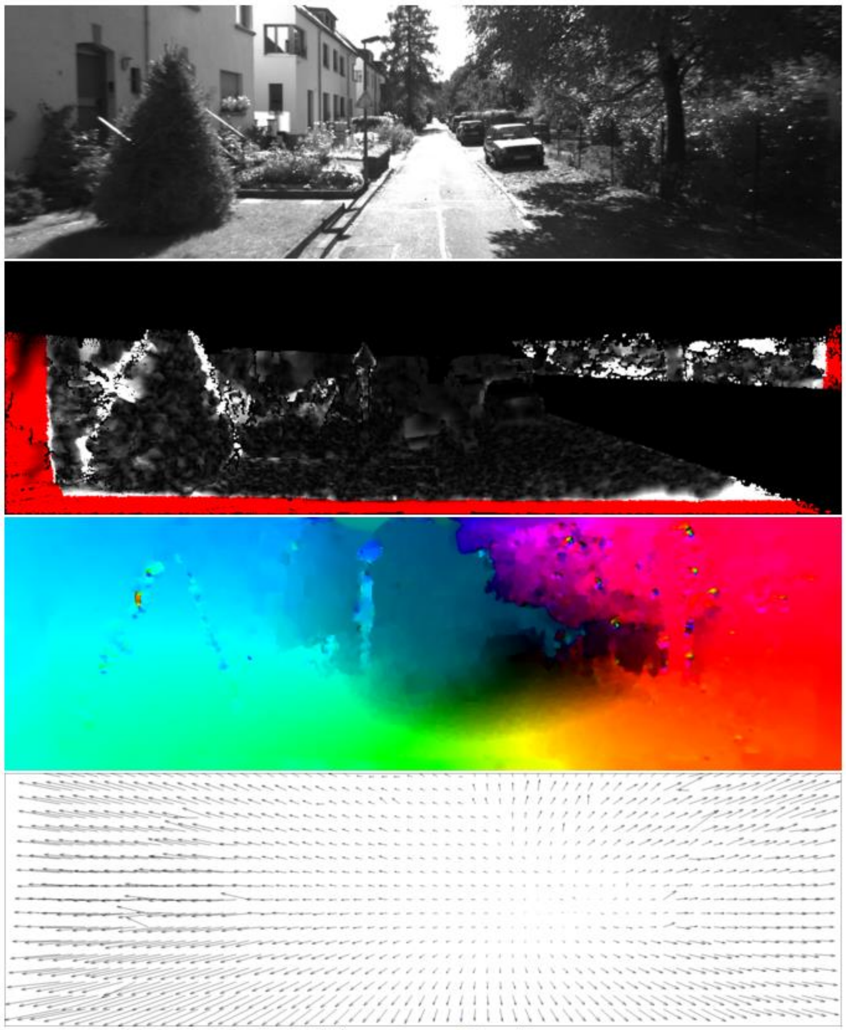
Flying datasets

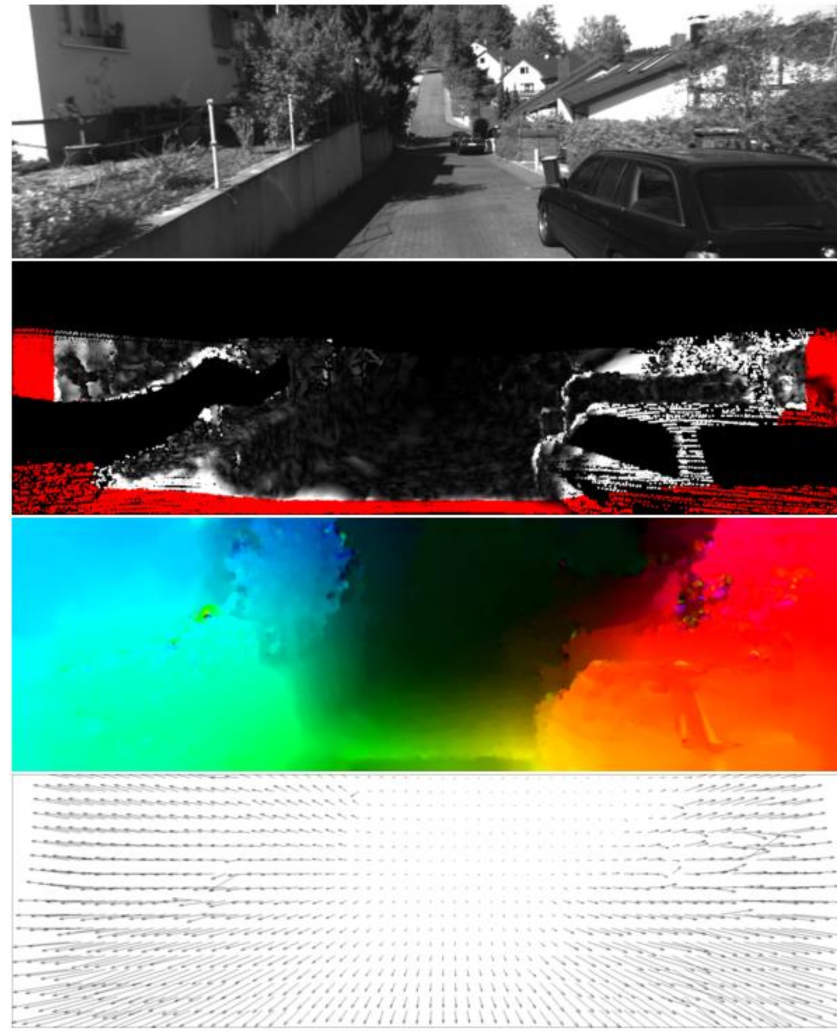
- Dataset configuration
  - Flying chair rendered with 3D shapes
  - Background: static image with affine transform
  - Add some noise
  - Rendering out flow result

Qualitative Results



Qualitative Results





Quantitative Results

Method	Sintel Clean		Sintel Final		KITTI		Middlebury train		Middlebury test		Chairs	Time (sec)	
	train	test	train	test	train	test	AEE	AAE	AEE	AAE	test	CPU	GPU
EpicFlow [30]	2.27	4.12	3.57	6.29	3.47	3.8	0.31	3.24	0.39	3.55	2.94	16	-
DeepFlow [35]	3.19	5.38	4.40	7.21	4.58	5.8	0.21	3.04	0.42	4.22	3.53	17	-
EPPM [3]	-	6.49	-	8.38	-	9.2	-	-	0.33	3.36	-	-	0.2
LDOF [6]	4.19	7.56	6.28	9.12	13.73	12.4	0.45	4.97	0.56	4.55	3.47	65	2.5
FlowNetS	4.50	7.42	5.45	8.43	8.26	-	1.09	13.28	-	-	2.71	-	0.08
FlowNetS+v	3.66	6.45	4.76	7.67	6.50	-	0.33	3.87	-	-	2.86	-	1.05
FlowNetS+ft	(3.66)	6.96	(4.44)	7.76	7.52	9.1	0.98	15.20	-	-	3.04	-	0.08
FlowNetS+ft+v	(2.97)	6.16	(4.07)	7.22	6.07	7.6	0.32	3.84	0.47	4.58	3.03	-	1.05
FlowNetC	4.31	7.28	5.87	8.81	9.35	-	1.15	15.64	-	-	2.19	-	0.15
FlowNetC+v	3.57	6.27	5.25	8.01	7.45	-	0.34	3.92	-	-	2.61	-	1.12
FlowNetC+ft	(3.78)	6.85	(5.28)	8.51	8.79	-	0.93	12.33	-	-	2.27	-	0.15
FlowNetC+ft+v	(3.20)	6.08	(4.83)	7.88	7.31	-	0.33	3.81	0.50	4.52	2.67	-	1.12