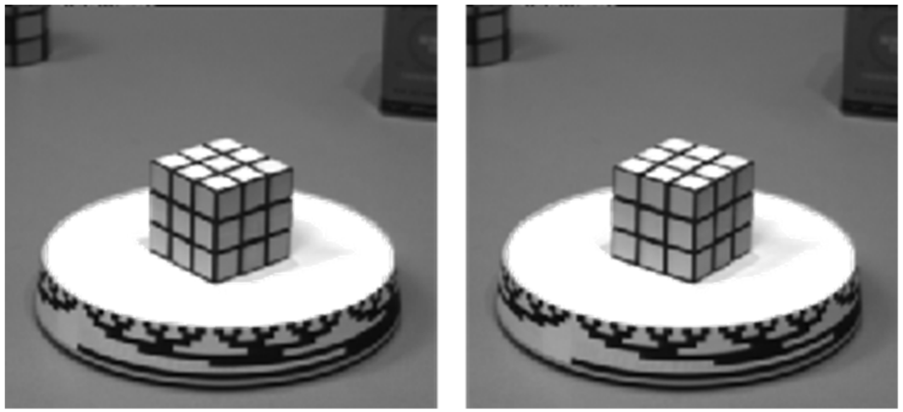
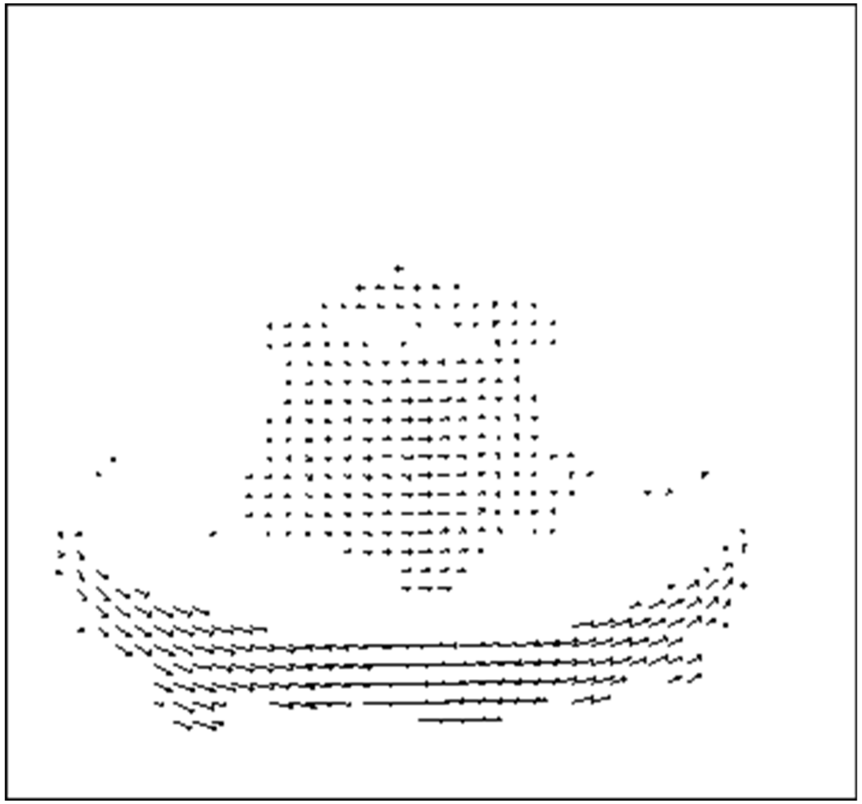
Optical flow

Cordelia Schmid

Motion field

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

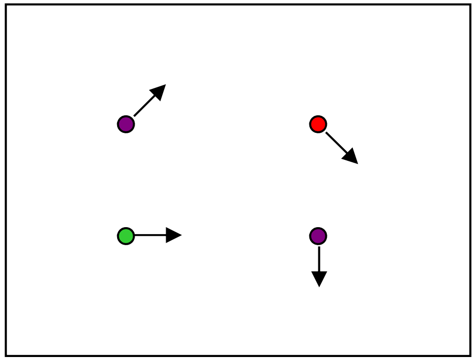




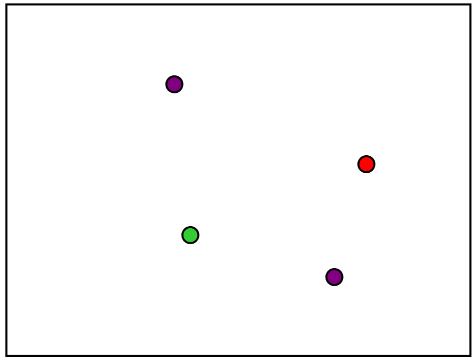
Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Ideally, optical flow would be the same as the motion field
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion
  - Think of a uniform rotating sphere under fixed lighting vs. a stationary sphere under moving illumination

Estimating optical flow



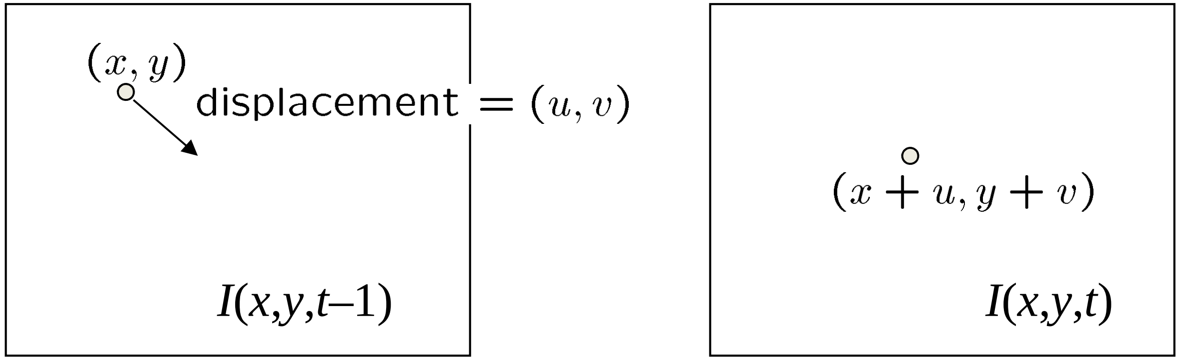
$I(x,y,t-1)$



$I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion field $u(x,y)$ and $v(x,y)$ between them
- Key assumptions
  - Brightness constancy: projection of the same point looks the same in every frame
  - Small motion: points do not move very far
  - Spatial coherence: points move like their neighbors

The brightness constancy constraint



Brightness Constancy Equation:

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

Linearizing the right side using Taylor expansion:

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

Hence, $I_x u + I_y v + I_t \approx 0$

The brightness constancy constraint

$$I_x u + I_y v + I_t = 0$$

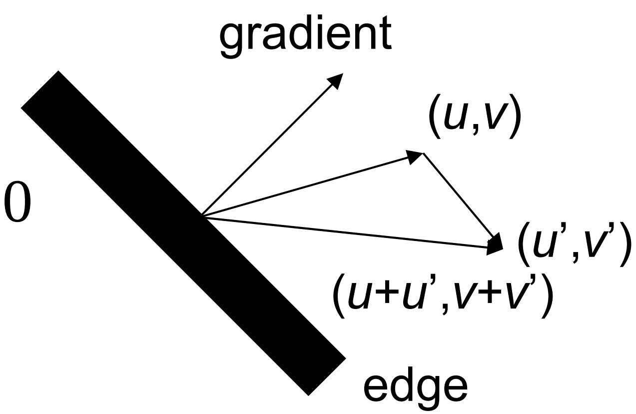
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?

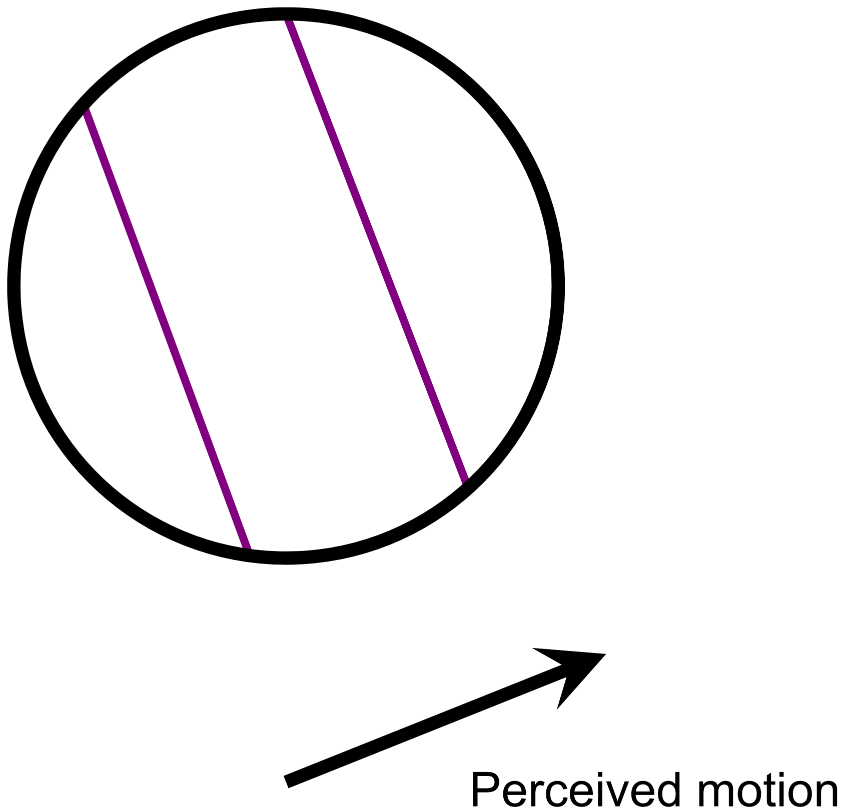
$$\nabla I \cdot (u, v) + I_t = 0$$

- The component of the flow perpendicular to the gradient (i.e., parallel to the edge) is unknown

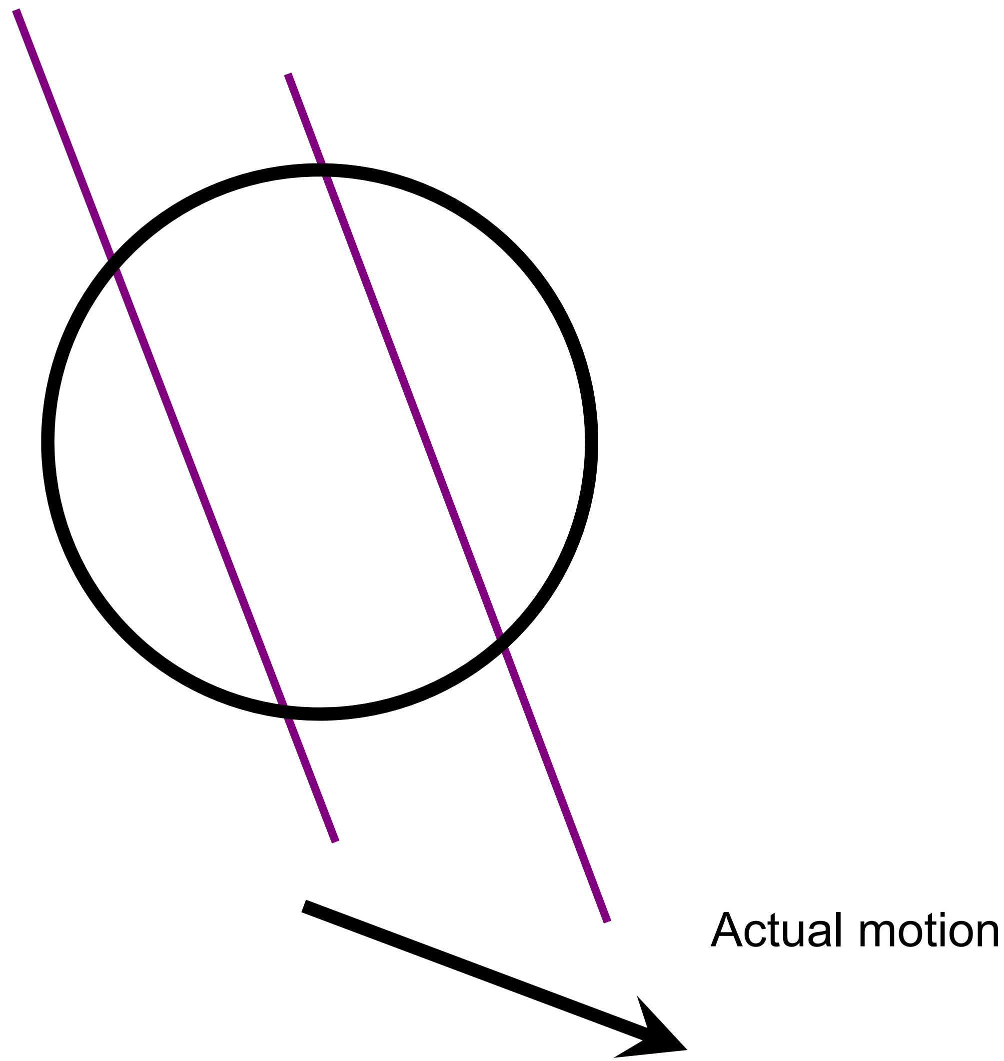
If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if $\nabla I \cdot (u', v') = 0$



The aperture problem



The aperture problem



Solving the aperture problem

- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)
  - E.g., if we use a 5x5 window, that gives us 25 equations per pixel

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*,1981.

informatics mathematics

Lucas-Kanade flow

- Linear least squares problem

$$\begin{bmatrix} I_x(\mathbf{x}_1) & I_y(\mathbf{x}_1) \\ I_x(\mathbf{x}_2) & I_y(\mathbf{x}_2) \\ \vdots & \vdots \\ I_x(\mathbf{x}_n) & I_y(\mathbf{x}_n) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{x}_1) \\ I_t(\mathbf{x}_2) \\ \vdots \\ I_t(\mathbf{x}_n) \end{bmatrix}$$

$$\underset{n\times 2}{\mathbf{A}} \; \underset{2\times 1}{\mathbf{d}} = \underset{n\times 1}{\mathbf{b}}$$

Solution given by $(\mathbf{A}^T\mathbf{A})\mathbf{d} = \mathbf{A}^T\mathbf{b}$

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

The summations are over all pixels in the window

Lucas-Kanade flow

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

- Recall the Harris corner detector: M = $A^T A$ is the *second moment matrix*
- When is the system solvable?
  - By looking at the eigenvalues of the second moment matrix
  - The eigenvectors and eigenvalues of M relate to edge direction and magnitude
  - The eigenvector associated with the larger eigenvalue points in the direction of fastest intensity change, and the other eigenvector is orthogonal to it

Uniform region

– gradients have small magnitude
– small λ_1, small λ_2
– system is ill-conditioned

Edge

– gradients have one dominant direction
– large λ_1, small λ_2
– system is ill-conditioned

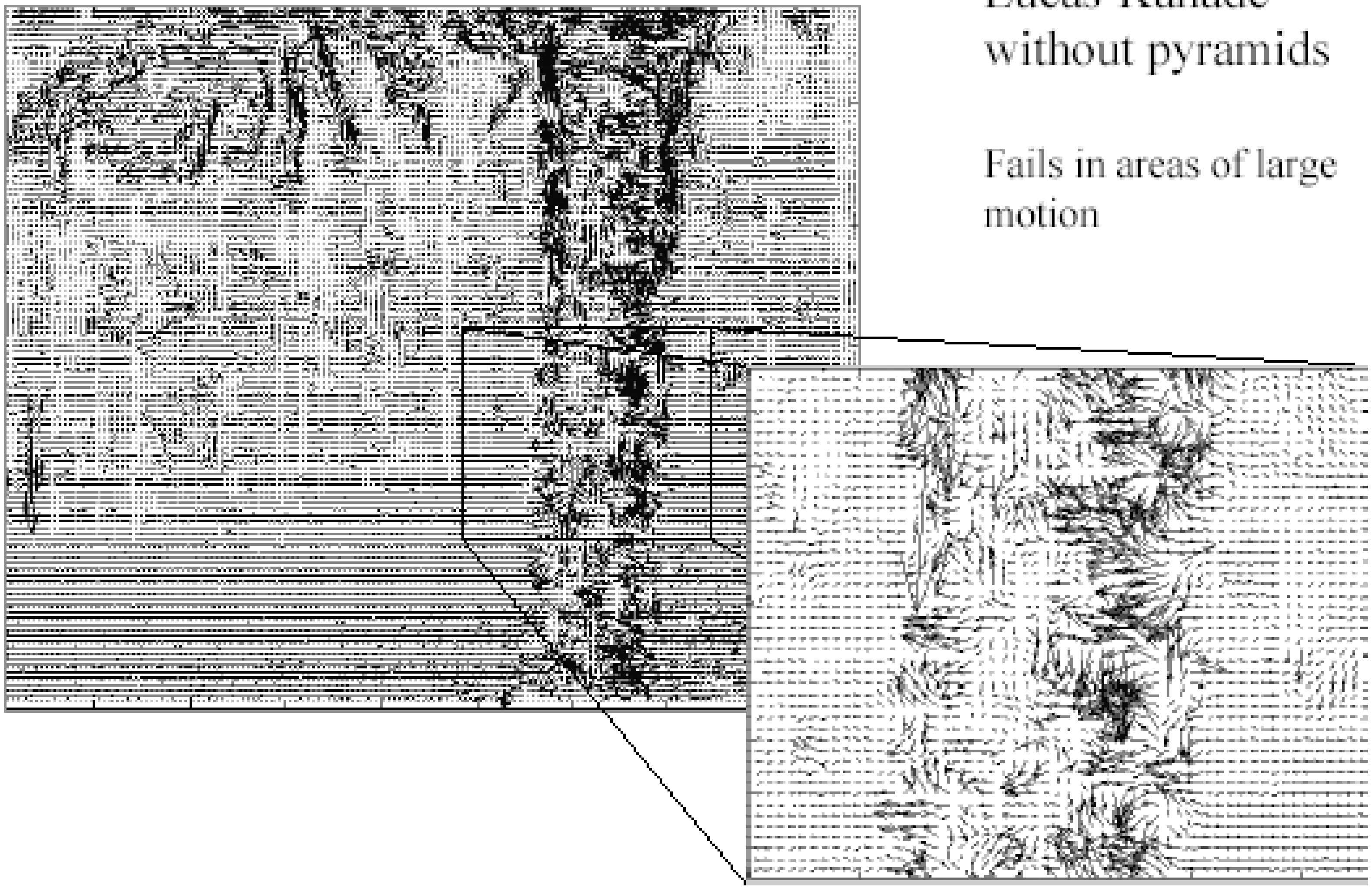
High-texture or corner region

– gradients have different directions, large magnitudes

– large λ_1, large λ_2

– system is well-conditioned

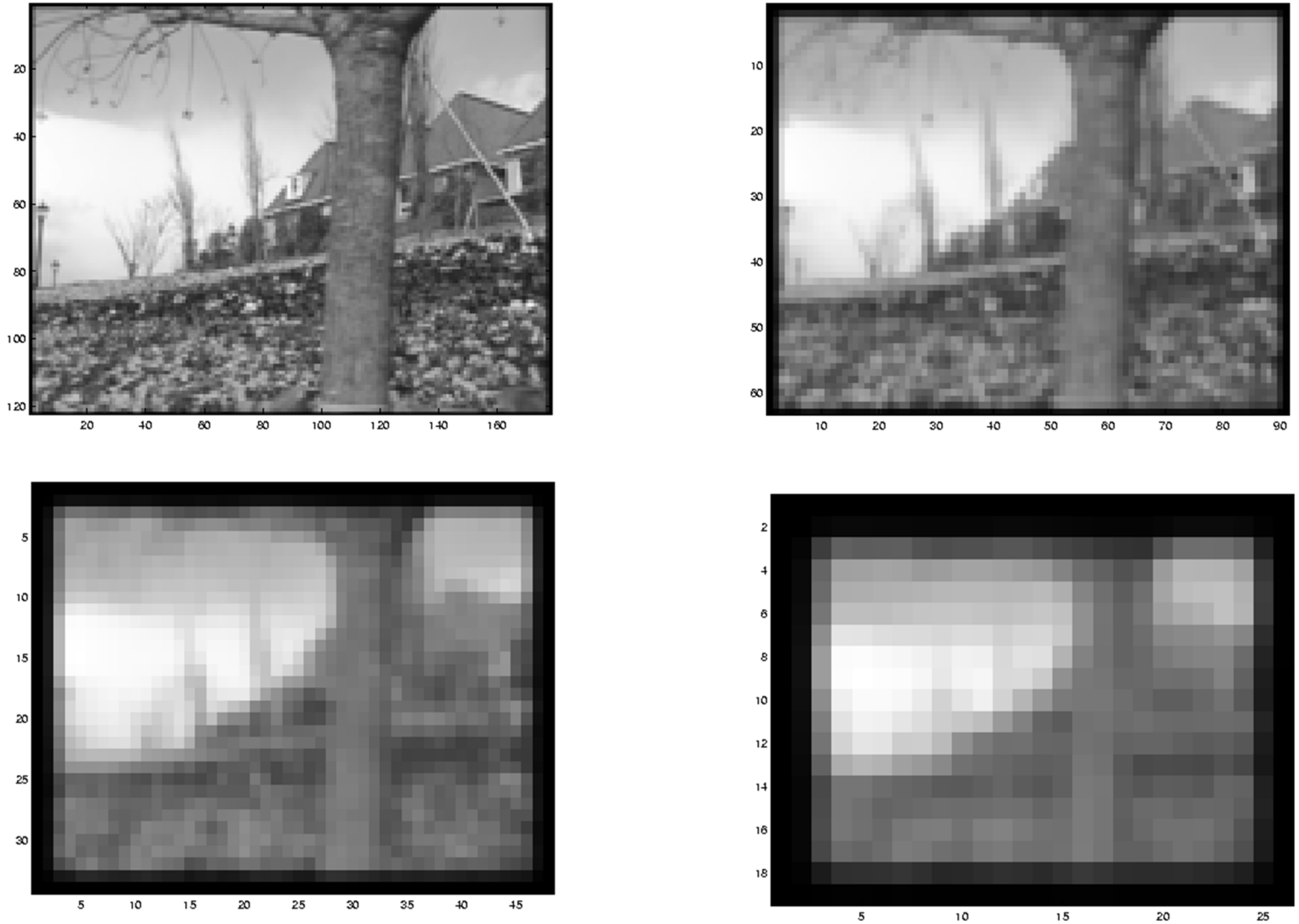
Optical Flow Results



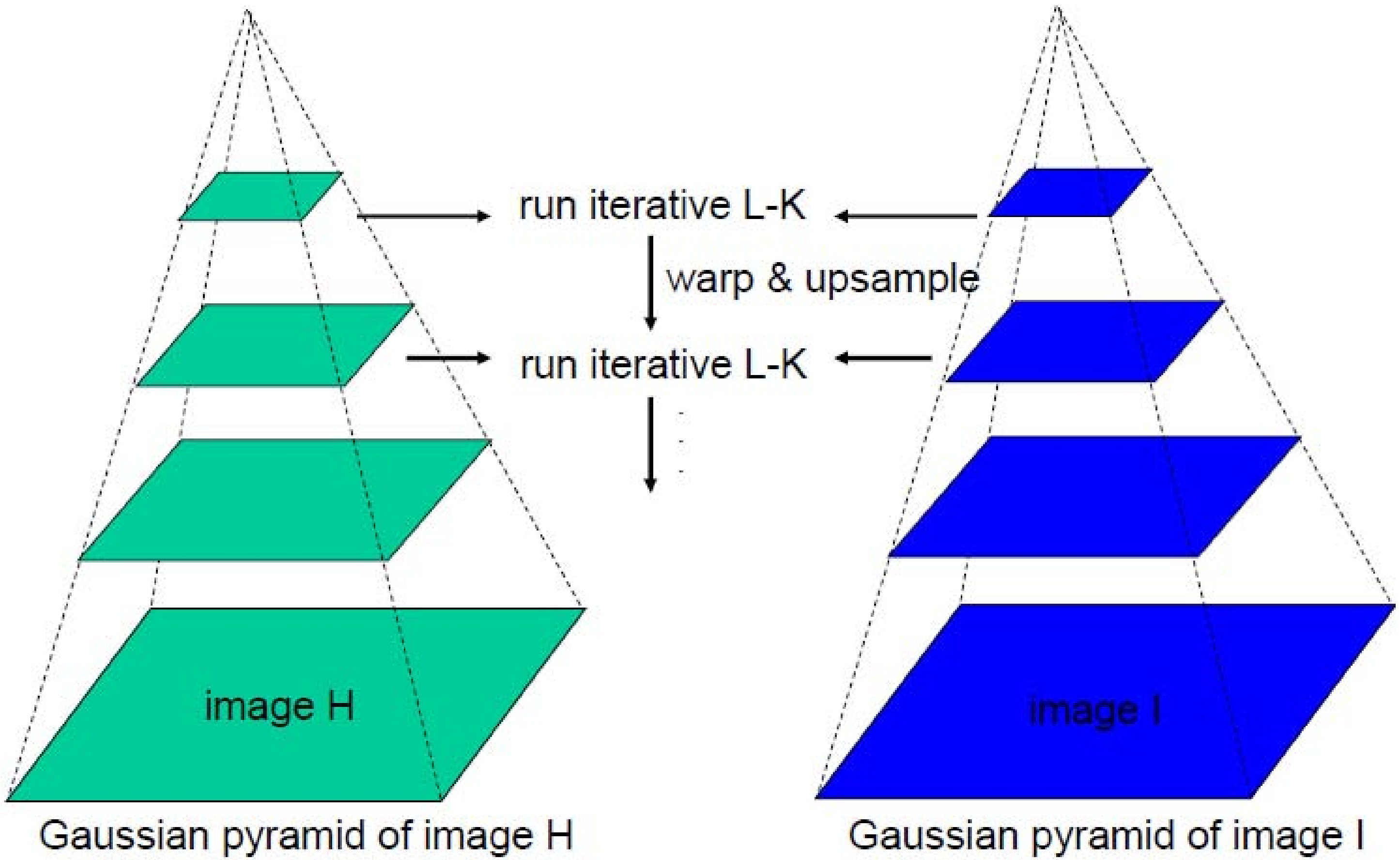
Lucas-Kanade
without pyramids

Fails in areas of large
motion

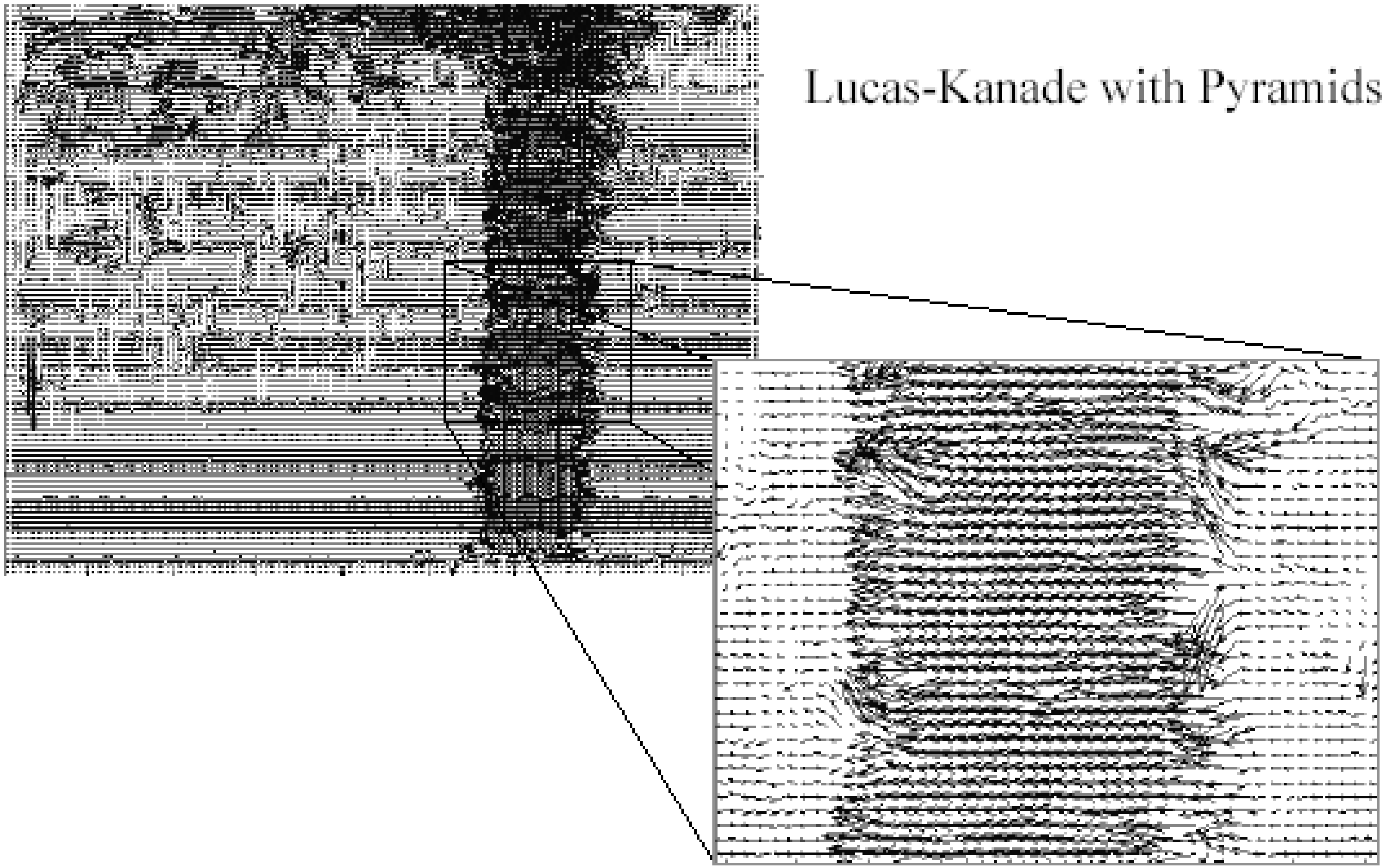
Multi-resolution registration



Coarse to fine optical flow estimation



Optical Flow Results



Lucas-Kanade with Pyramids

Horn & Schunck algorithm

Additional smoothness constraint :

- nearby point have similar optical flow
- Addition constraint $||\nabla u||^2, \ ||\nabla v||^2$

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy,$$

B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*,1981

Horn & Schunck algorithm

Additional smoothness constraint :

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy,$$

besides OF constraint equation term

$$e_c = \iint (I_x u + I_y v + I_t)^2 dxdy,$$

$\boxed{\text{minimize } e_s + \lambda e_c}$ λ regularization parameter

B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*,1981

Horn & Schunck algorithm

$$E(u(x,y),v(x,y)) = \iint \underbrace{(I_x u + I_y v + I_t)^2}_{\text{Data term brightness constancy}} + \alpha\underbrace{((u_x^2 + u_y^2) + (v_x^2 + v_y^2))}_{\text{Smoothness term}} dxdy$$

$$E(u,v) = \int_{\Omega} F(x,y,u,v,u_x,u_y,v_x,v_y)dxdy$$

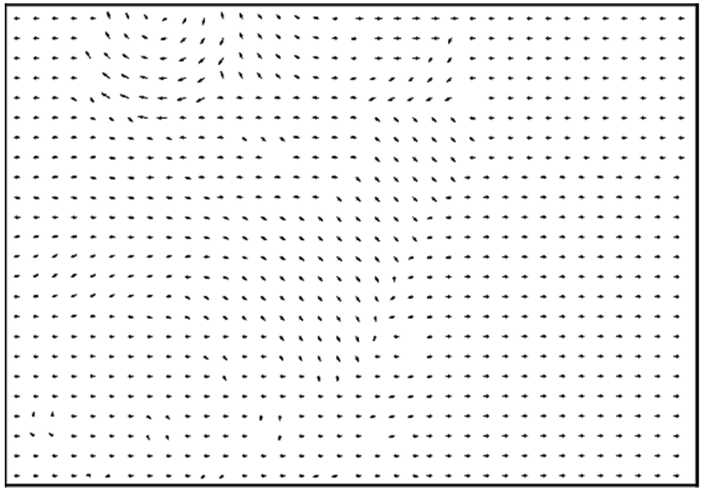
Euler-Lagrange equations

$$F_u - \frac{\partial}{\partial x}F_{u_x} - \frac{\partial}{\partial y}F_{u_y} = 0 \qquad F_v - \frac{\partial}{\partial x}F_{v_x} - \frac{\partial}{\partial y}F_{v_y} = 0$$

Coupled PDEs solved using iterative methods and finite differences

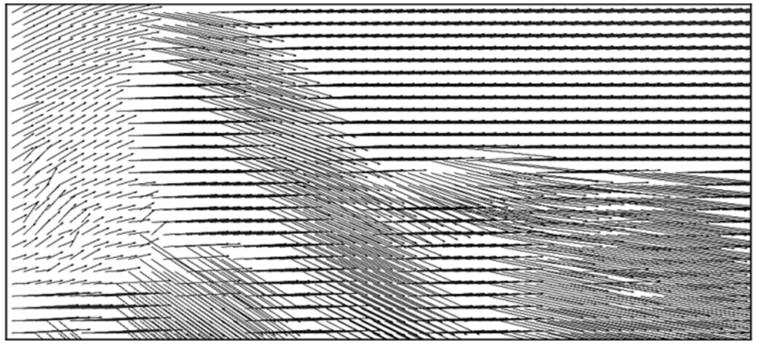
Horn & Schunck

- Works well for small displacements
  - For example Middlebury sequence



Large displacement estimation in optical flow

- Large displacement is still an open problem in optical flow estimation



MPI Sintel dataset

Large displacement optical flow

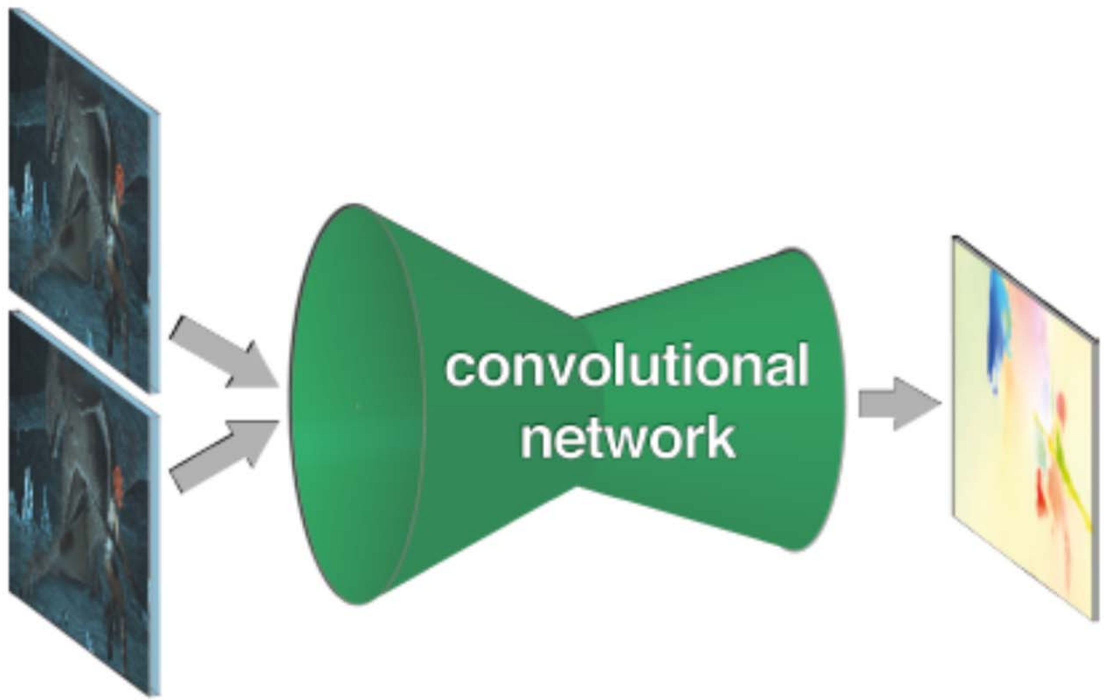
- Classical optical flow [Horn and Schunck 1981]
  - energy: $$E(\mathbf{w}) = \iint E_{data} + \alpha E_{smooth} \mathbf{dx}$$

    color/gradient constancy   smoothness constraint

  - minimization using a coarse-to-fine scheme

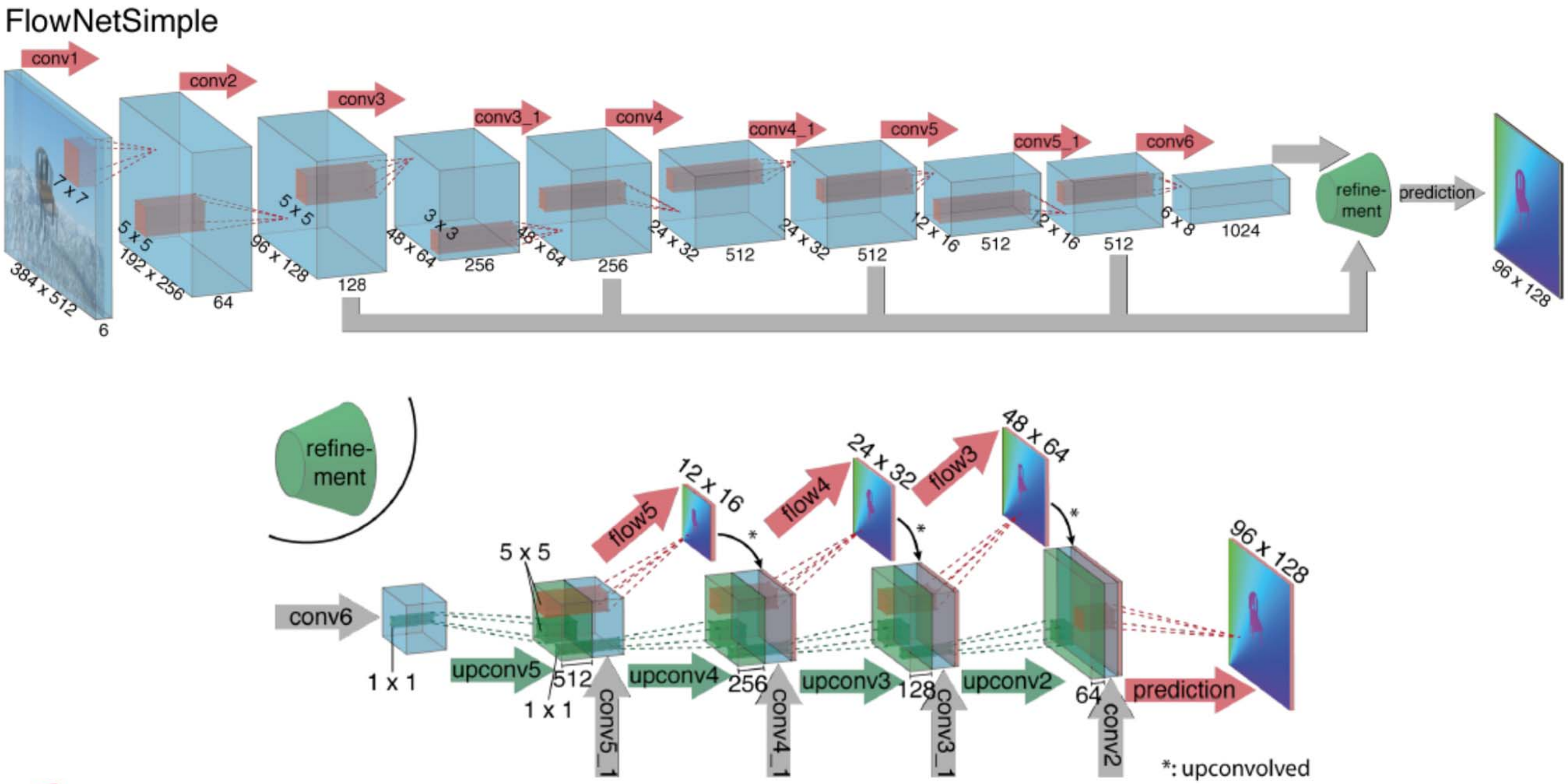
- Large displacement approaches:
  - LDOF [Brox and Malik 2011]
    a matching term, penalizing the difference between flow and HOG matches
    $$E(\mathbf{w}) = \iint E_{data} + \alpha E_{smooth} + \beta E_{match} \mathbf{dx}$$
  - MDP-Flow2 [Xu *et al.* 2012]
    expensive fusion of matches (SIFT + PatchMatch) and estimated flow at each level
  - DeepFlow [Weinzaepfel *et al.* 2013]
    deep matching + flow refinement with variational approach

CNN to estimate optical flow: FlowNet

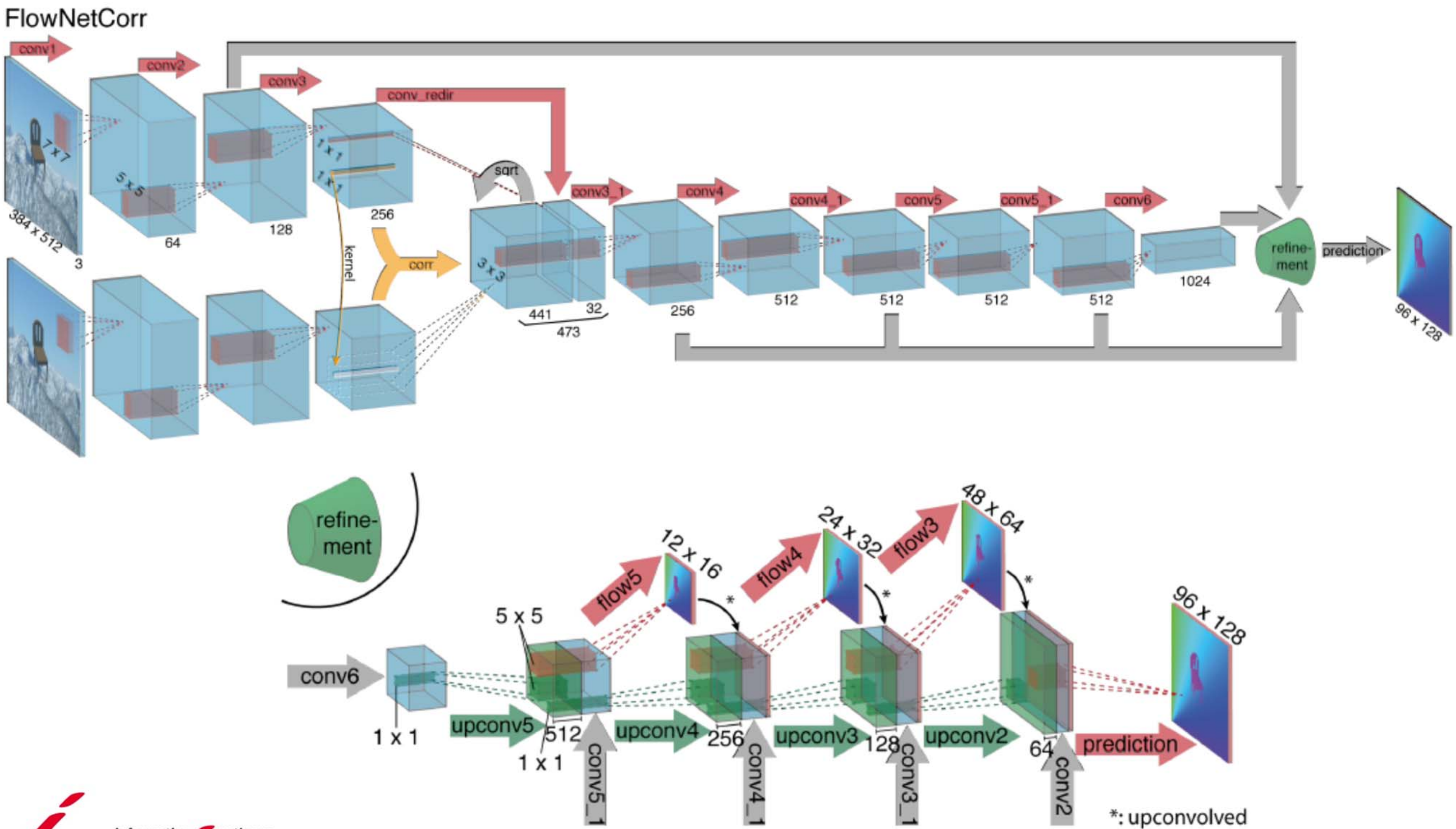


[A. Dosovitskiy et al. ICCV'15]

Architecture FlowNetSimple



Architecture FlowNetCorrelation



Synthetic dataset for training: Flying chairs

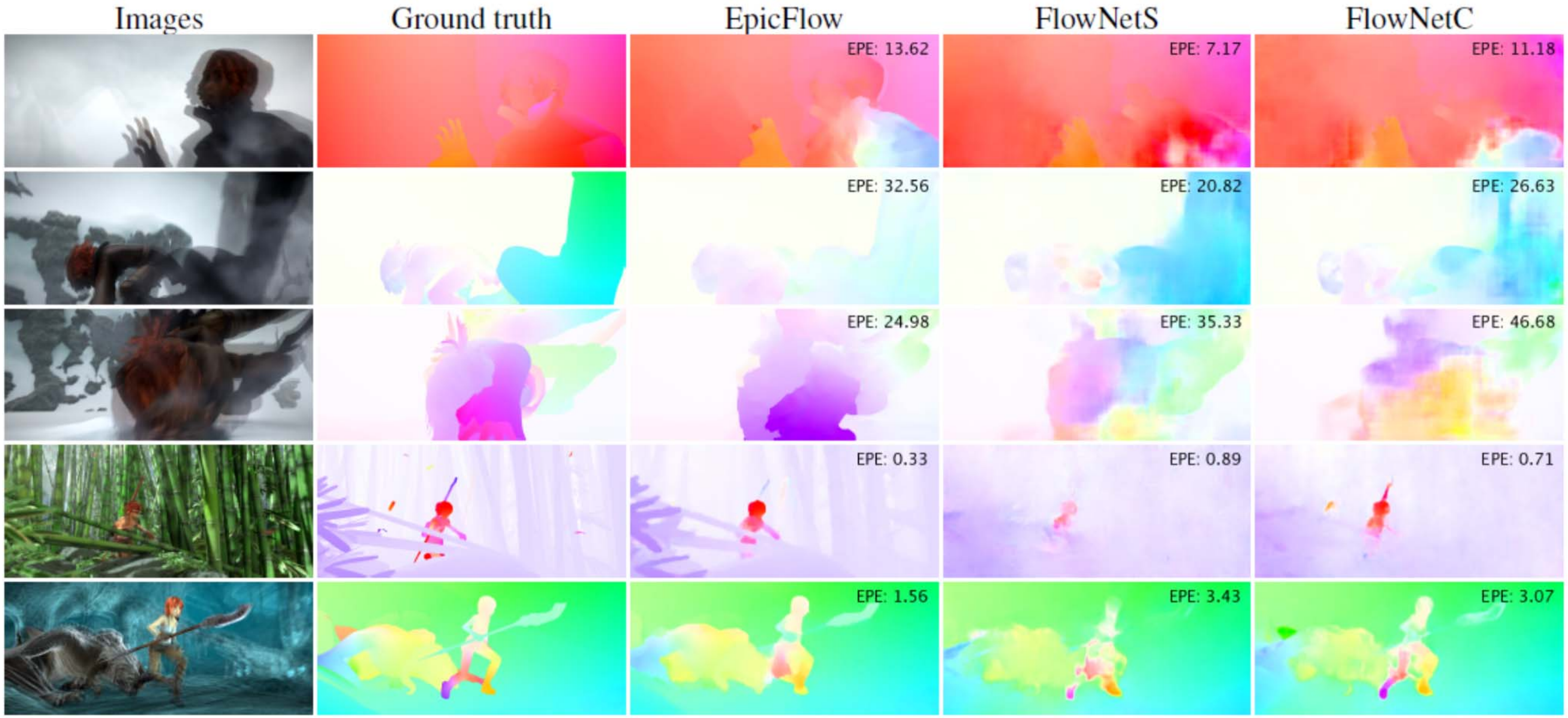
A dataset of approx. 23k image pairs

Experimental results

Method	Sintel Clean		Sintel Final	
	train	test	train	test
EpicFlow [30]	2.27	4.12	3.57	6.29
DeepFlow [35]	3.19	5.38	4.40	7.21
EPPM [3]	-	6.49	-	8.38
LDOF [6]	4.19	7.56	6.28	9.12
FlowNetS	4.50	7.42	5.45	8.43
FlowNetS+v	3.66	6.45	4.76	7.67
FlowNetS+ft	(3.66)	6.96	(4.44)	7.76
FlowNetS+ft+v	(2.97)	6.16	(4.07)	7.22
FlowNetC	4.31	7.28	5.87	8.81
FlowNetC+v	3.57	6.27	5.25	8.01
FlowNetC+ft	(3.78)	6.85	(5.28)	8.51
FlowNetC+ft+v	(3.20)	6.08	(4.83)	7.88

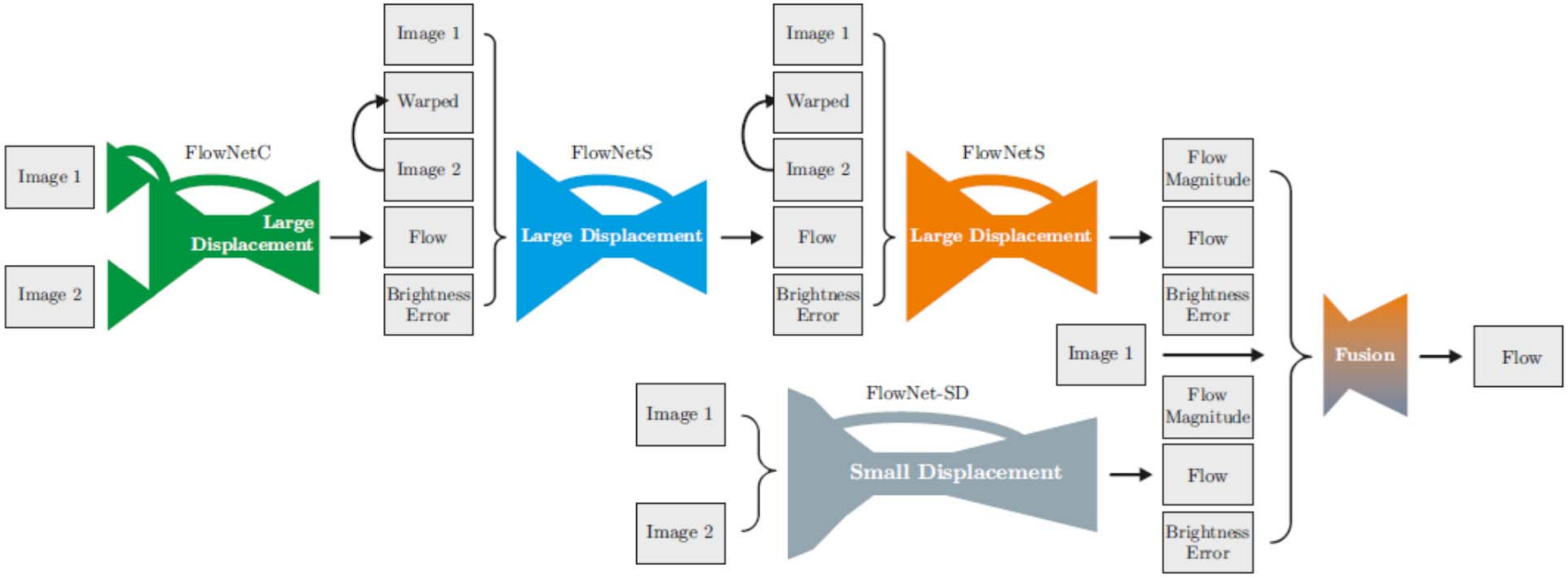
S: simple, C: correlation, v: variational refinement, ft:fine-tuning

Experimental results

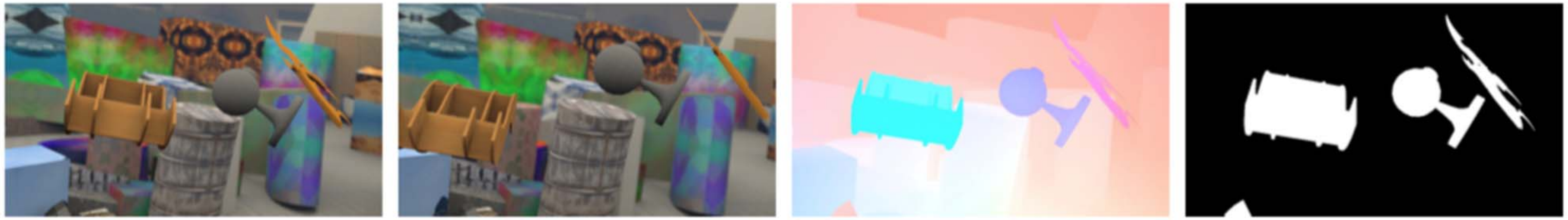


informatics mathematics

FlowNet2.0 [Ilg et al. CVPR'17]



FlyingThings3D [Mayer et al., CVPR'16]



Comparison training data

Architecture	Datasets	S_{short}	S_{long}	S_{fine}
FlowNetS	Chairs	4.45	-	-
	Chairs	-	4.24	4.21
	Things3D	-	5.07	4.50
	mixed	-	4.52	4.10
	Chairs→Things3D	-	4.24	**3.79**
FlowNetC	Chairs	3.77	-	-
	Chairs→Things3D	-	3.58	**3.04**

Best: pretraining on a simpler dataset, then fine tuning on a more complex set
FlowNetC better than FlowNetS

Stacking of networks

Stack architecture	Training enabled		Warping included	Warping gradient enabled	Loss after		EPE on Chairs test	EPE on Sintel train *clean*
	Net1	Net2			Net1	Net2		
Net1	✓	–	–	–	✓	–	3.01	3.79
Net1 + Net2	✗	✓	✗	–	–	✓	2.60	4.29
Net1 + Net2	✓	✓	✗	–	✗	✓	2.55	4.29
Net1 + Net2	✓	✓	✗	–	✓	✓	2.38	3.94
Net1 + W + Net2	✗	✓	✓	–	–	✓	1.94	**2.93**
Net1 + W + Net2	✓	✓	✓	✓	✗	✓	1.96	3.49
Net1 + W + Net2	✓	✓	✓	✓	✓	✓	**1.78**	3.33

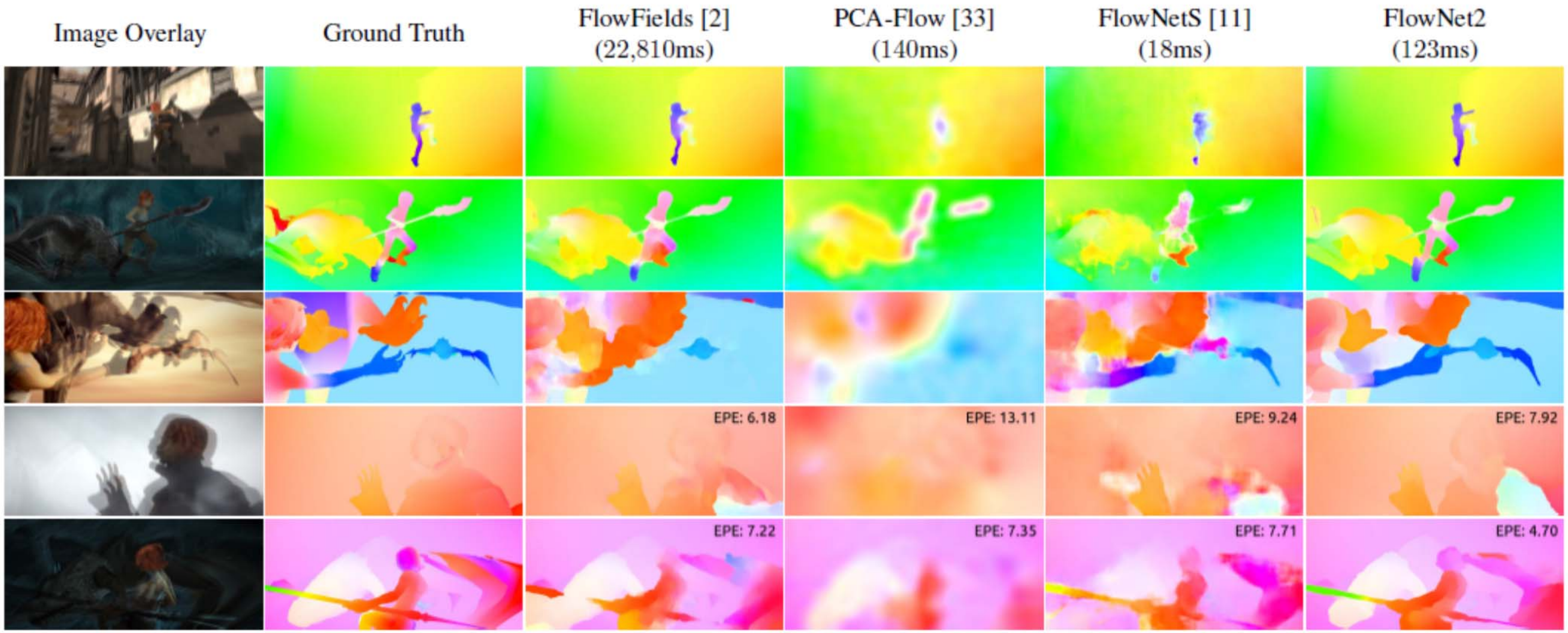
Importance of warping

Comparison to the state of the art

	Method	Sintel *clean* AEE		Sintel *final* AEE		KITTI 2012 AEE		KITTI 2015 AEE	KITTI 2015 Fl-all	KITTI 2015 Fl-all	Middlebury AEE		Runtime ms per frame	
		train	*test*	*train*	*test*	*train*	*test*	*train*	*train*	*test*	*train*	*test*	CPU	GPU
Accurate	EpicFlow† [22]	2.27	4.12	3.56	6.29	**3.09**	**3.8**	9.27	27.18%	**27.10%**	0.31	0.39	42,600	–
	DeepFlow† [32]	2.66	5.38	3.57	7.21	4.48	5.8	10.63	26.52%	29.18%	**0.25**	0.42	51,940	–
	FlowFields [2]	**1.86**	**3.75**	**3.06**	**5.81**	3.33	3.5	**8.33**	**24.43%**	–	0.27	**0.33**	22,810	–
	LDOF (CPU) [7]	4.64	7.56	5.96	9.12	10.94	12.4	18.19	38.11%	–	0.44	0.56	64,900	–
	LDOF (GPU) [27]	4.76	–	6.32	–	10.43	–	18.20	38.05%	–	0.36	–	–	6,270
	PCA-Layers [33]	3.22	5.73	4.52	7.89	5.99	5.2	12.74	27.26%	–	0.66	–	**3,300**	–
Fast	EPPM [4]	–	**6.49**	–	**8.38**	–	9.2	–	–	–	–	**0.33**	–	200
	PCA-Flow [33]	**4.04**	6.83	**5.18**	8.65	**5.48**	**6.2**	**14.01**	**39.59%**	–	**0.70**	–	140	–
	DIS-Fast [16]	5.61	9.35	6.31	10.13	11.01	14.4	21.20	53.73%	–	0.92	–	70	–
	FlowNetS [11]	4.50	6.96‡	5.45	7.52‡	8.26	–	–	–	–	1.09	–	–	**18**
	FlowNetC [11]	4.31	6.85‡	5.87	8.51‡	9.35	–	–	–	–	1.15	–	–	32
FlowNet 2.0	FlowNet2-s	4.55	–	5.21	–	8.89	–	16.42	56.81%	–	1.27	–	–	**7**
	FlowNet2-ss	3.22	–	3.85	–	5.45	–	12.84	41.03%	–	0.68	–	–	14
	FlowNet2-css	2.51	–	3.54	–	4.49	–	11.01	35.19%	–	0.54	–	–	31
	FlowNet2-css-ft-sd	2.50	–	3.50	–	4.71	–	11.18	34.10%	–	0.43	–	–	31
	FlowNet2-CSS	2.10	–	3.23	–	**3.55**	–	**8.94**	29.77%	–	0.44	–	–	69
	FlowNet2-CSS-ft-sd	2.08	–	3.17	–	4.05	–	10.07	30.73%	–	0.38	–	–	69
	FlowNet2	**2.02**	**3.96**	**3.14**	6.02	4.09	–	10.06	30.37%	–	**0.35**	**0.52**	–	123
	FlowNet2-ft-sintel	(1.45)	4.16	(2.01)	**5.74**	3.61	–	9.84	**28.20%**	–	0.35	–	–	123
	FlowNet2-ft-kitti	3.43	–	4.66	–	(1.28)	**1.8**	(2.30)	(8.61%)	**11.48%**	0.56	–	–	123

informatics mathematics

Optical flow results on Sintel



informatics mathematics
Inria

Video object segmentation

- Segment the moving object in all the frames of a video

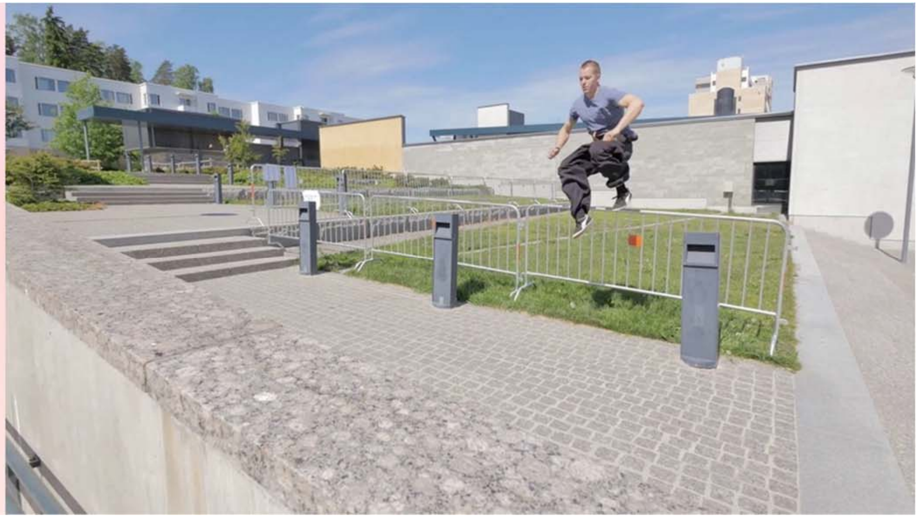
DAVIS (ground-truth)

[Tokmakov et al., CVPR 2017]

Challenges

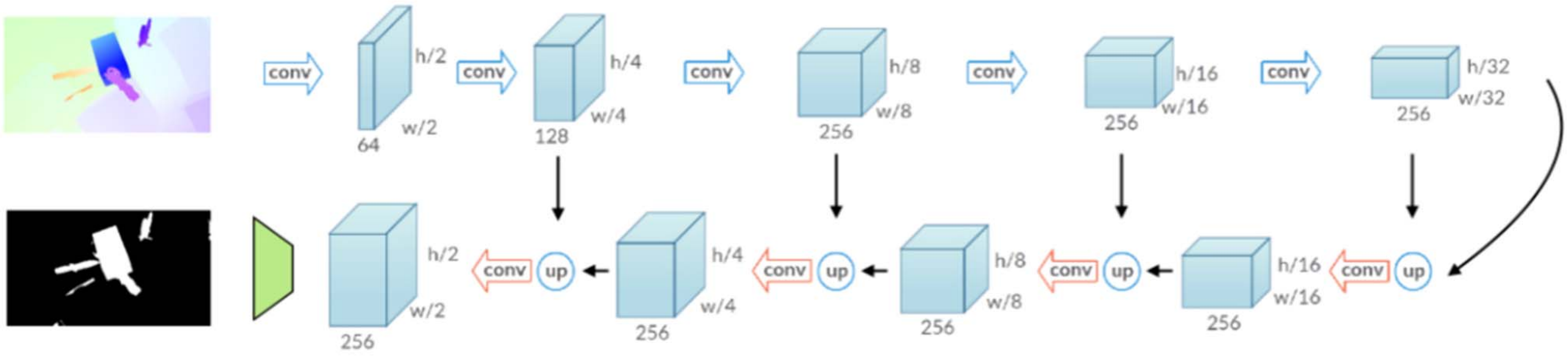
- Strong camera or background motion

LDOF flow



DAVIS

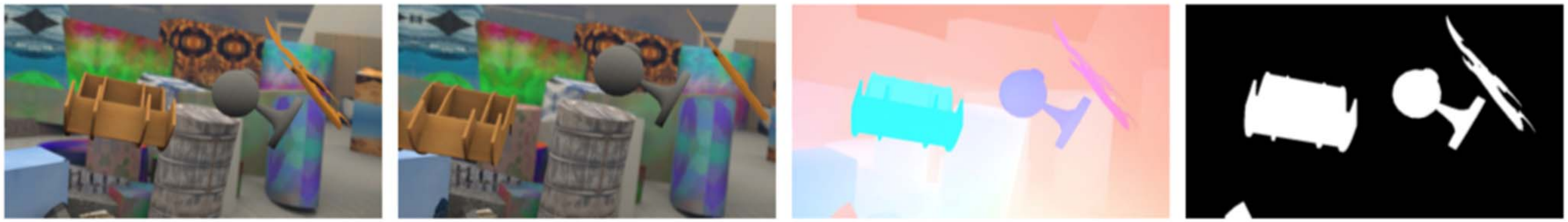
Network architecture – MP-Net



Convolutional/deconvolutional network, similar to U-Net

Training data

- FlyingThings3D dataset [Mayer et al., CVPR'16]
- 2700 synthetic, 10-frame stereo videos of random object flying in random trajectories (2250/450 training/test split)
- Ground-truth optical flow and camera data available
- Labels for moving object can be obtained from the data



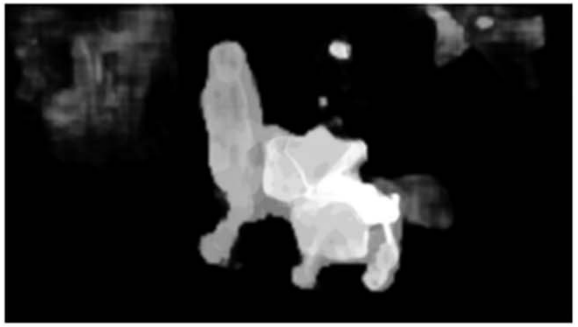
Results on FlyingThings3D test set

Motion estimation in real videos

- Flow estimation inaccuracies

DAVIS

LDOF

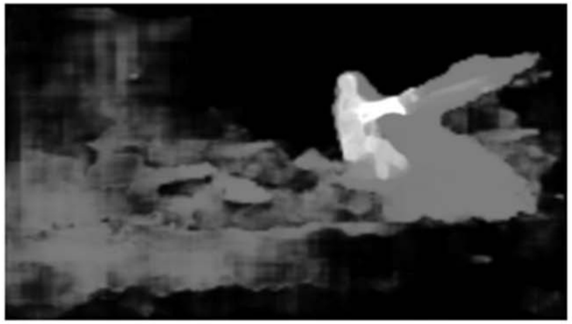


MP-Net

- Background motion

DAVIS

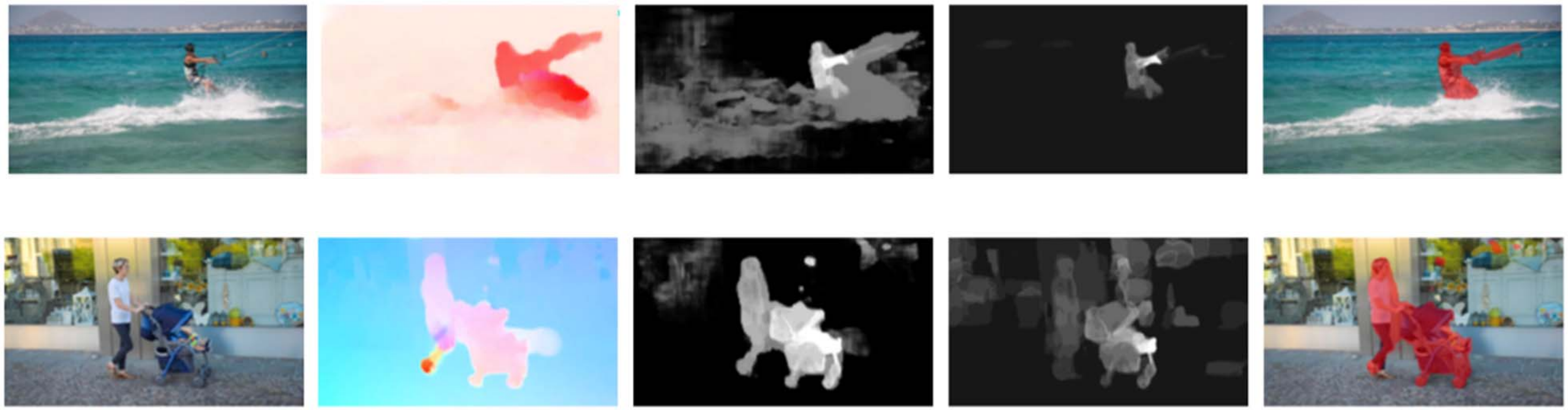
LDOF



MP-Net

Addition of an objectness measure

- Extract 100 object proposals per frame with SharpMask [Pinheiro et al., ECCV'16]
- Aggregate to obtain pixel-level objectness scores o_i
- Combine with the motion predictions m_i



DAVIS LDOF MP-Net Objectness Result

FlowNet 2.0 Evaluation

Setting	LDOF flow	FLowNet 2.0 flow
MP-Net	52.4	62.6
MP-Net + Obj	63.3	69.0
MP-Net + Obj + CRF	69.7	72.5

Mean IoU on DAVIS trainval set