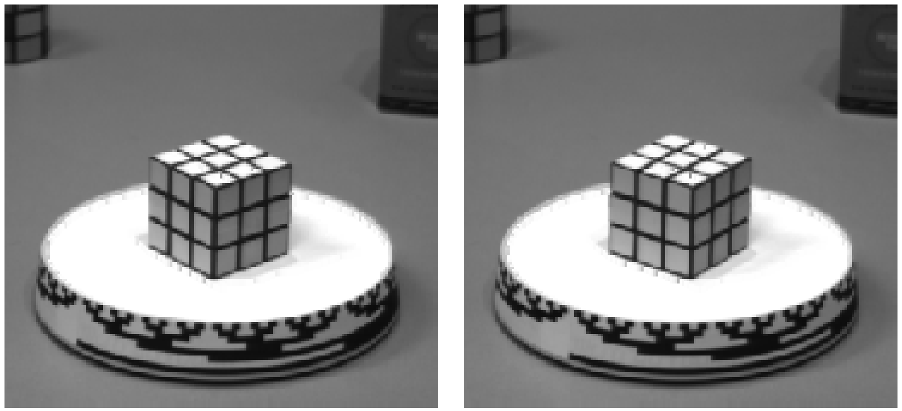
Optical flow

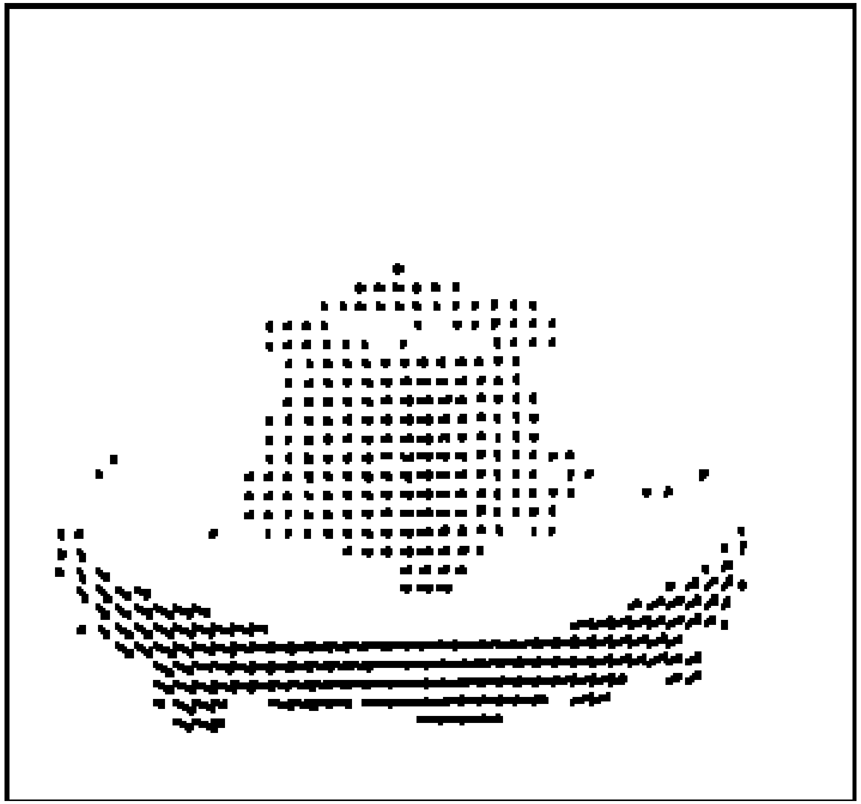
Cordelia Schmid

informatics mathematics

Motion field

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

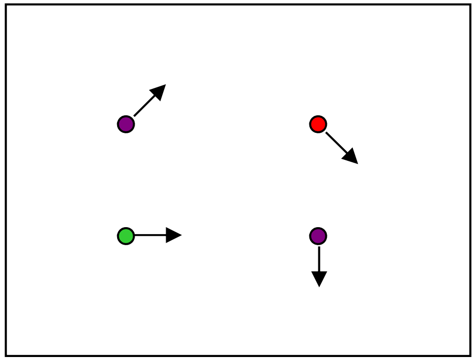




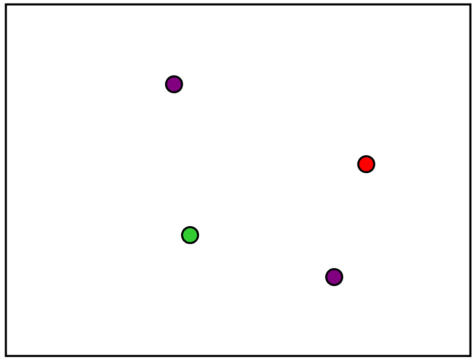
Optical flow

- 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



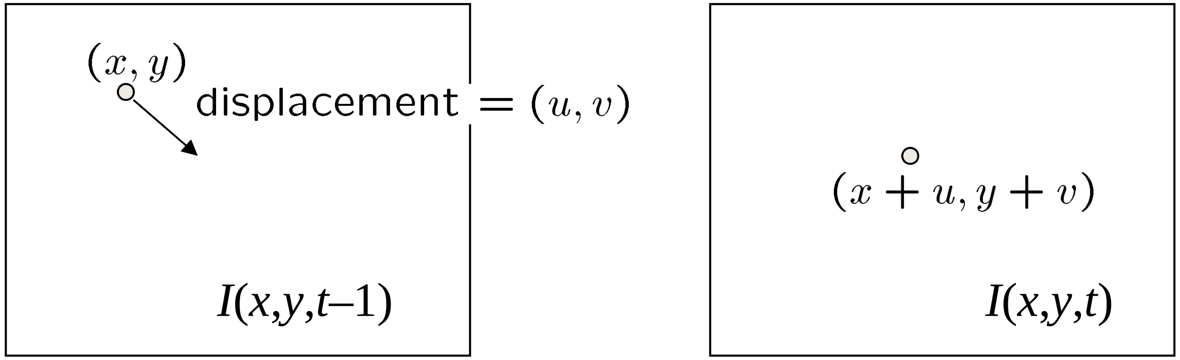
$I(x,y,t-1)$



$I(x,y,t)$

- 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

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

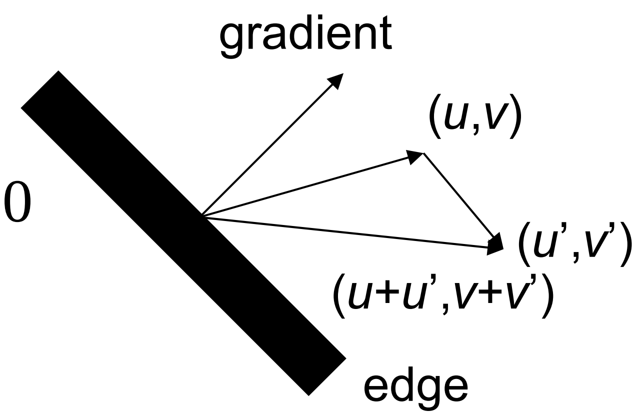
- How many equations and unknowns per pixel?
  - One equation, two unknowns

- What does this constraint mean?

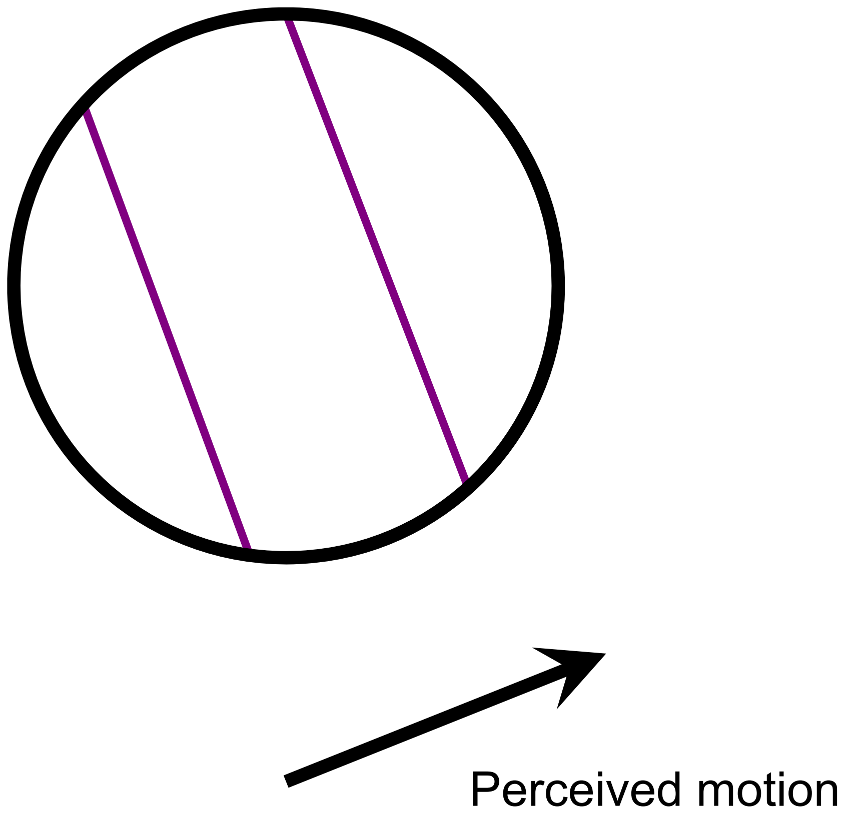
$$\nabla I \cdot (u, v) + I_t = 0$$

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

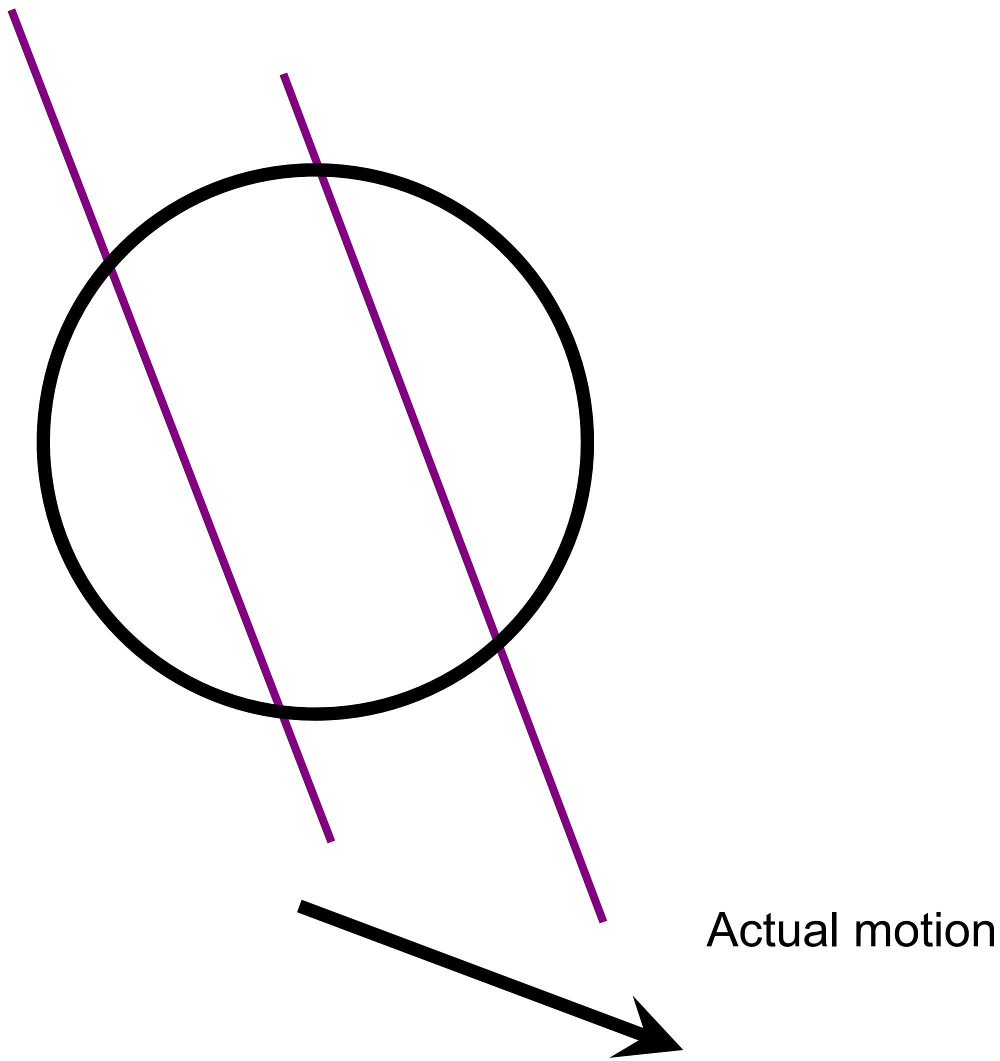
If (*u*, *v*) satisfies the equation,
so does (*u*+*u*', *v*+*v*') if $\nabla I \cdot (u', v') = 0$



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.

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

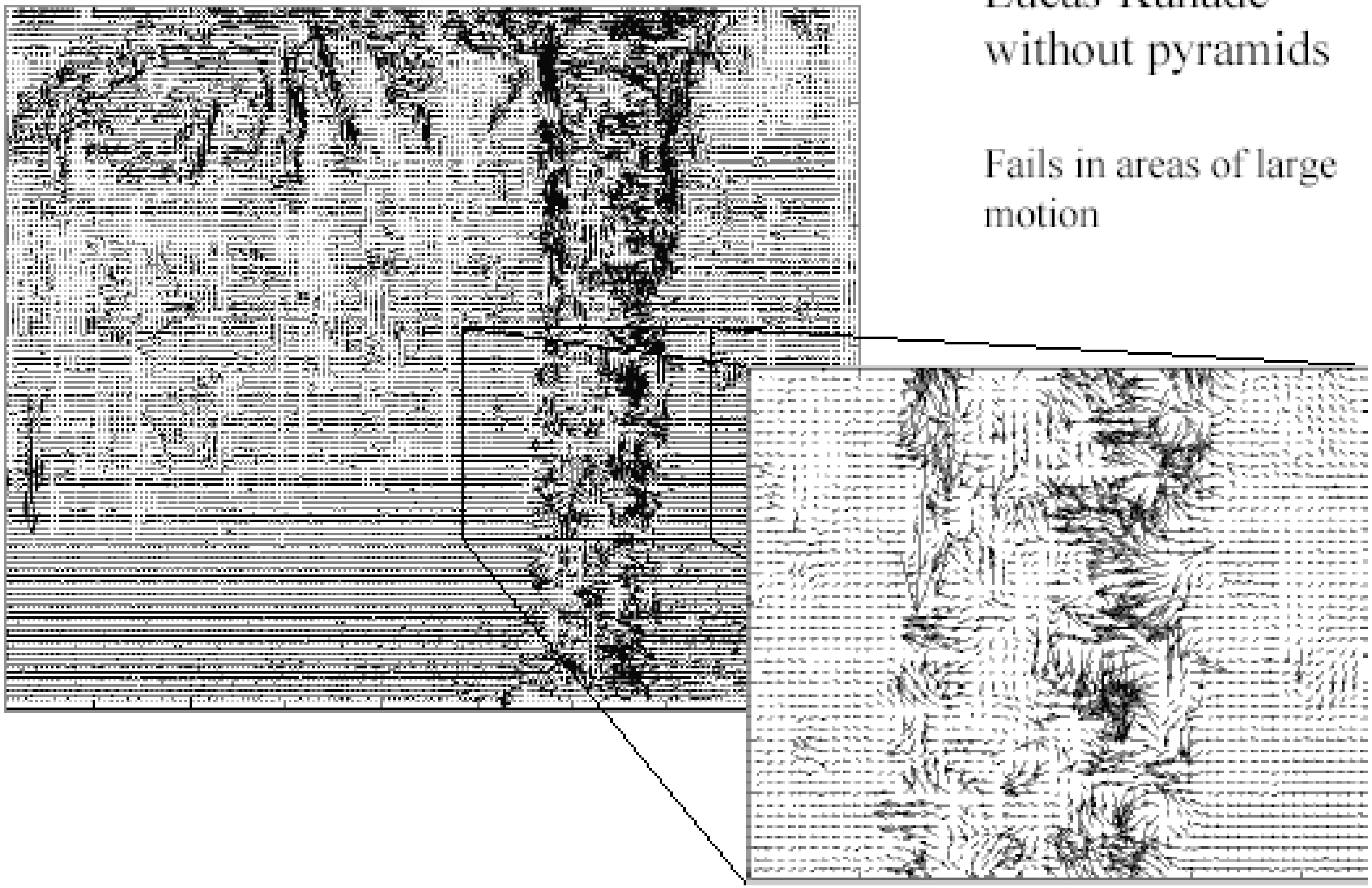
High-texture or corner region

– gradients have different directions, large magnitudes

– large λ_1, large λ_2

– system is well-conditioned

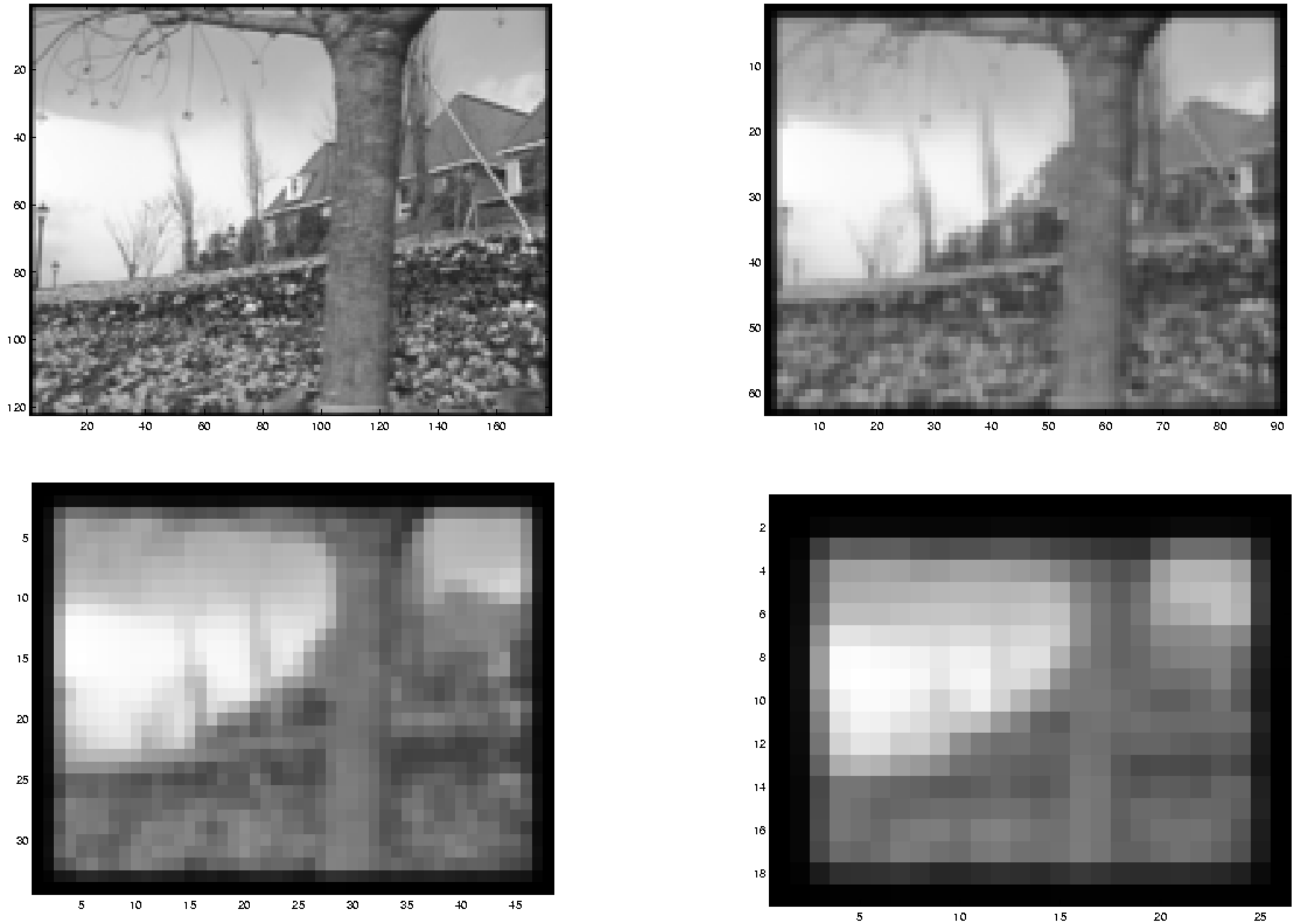
Optical Flow Results



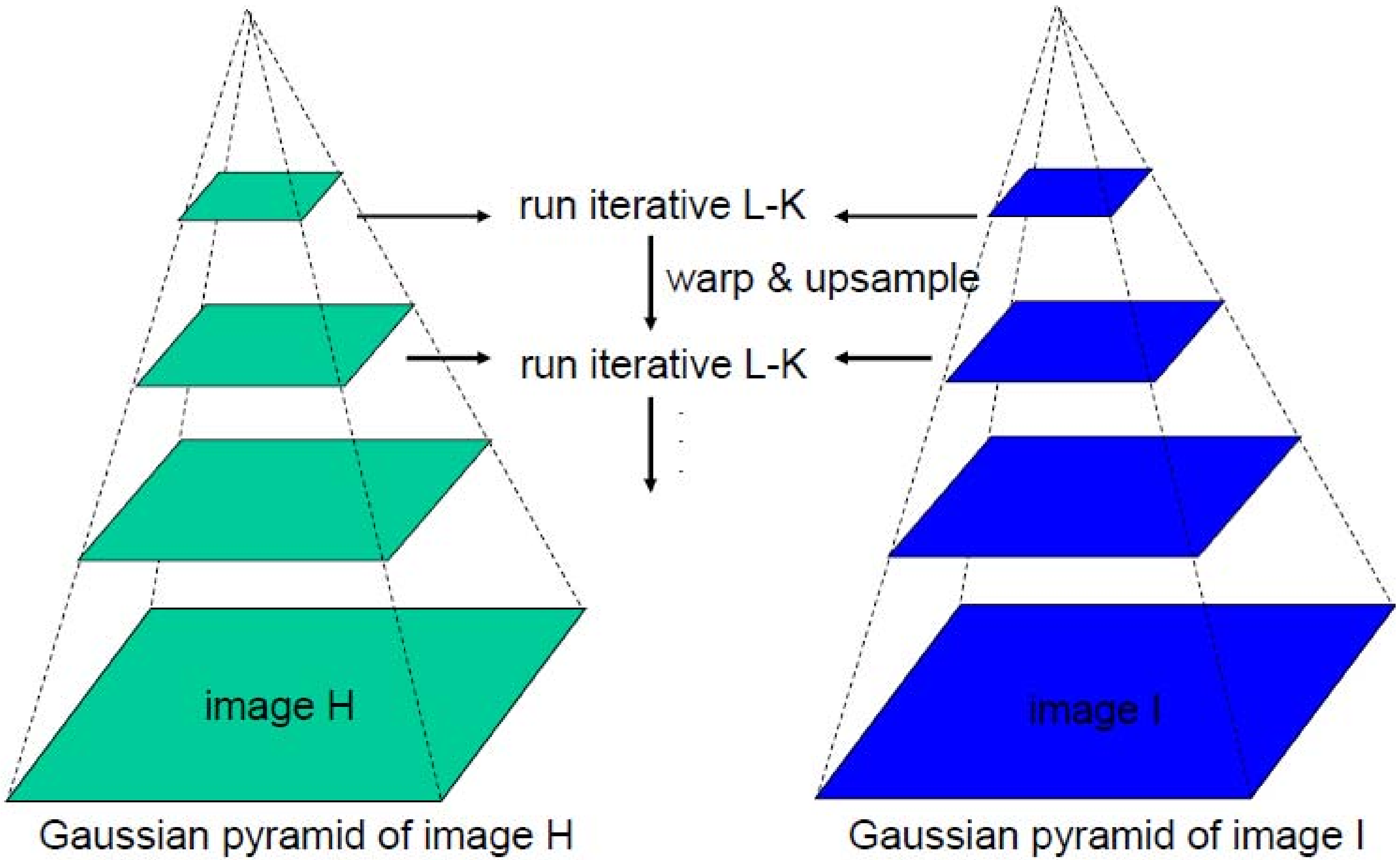
Lucas-Kanade
without pyramids

Fails in areas of large
motion

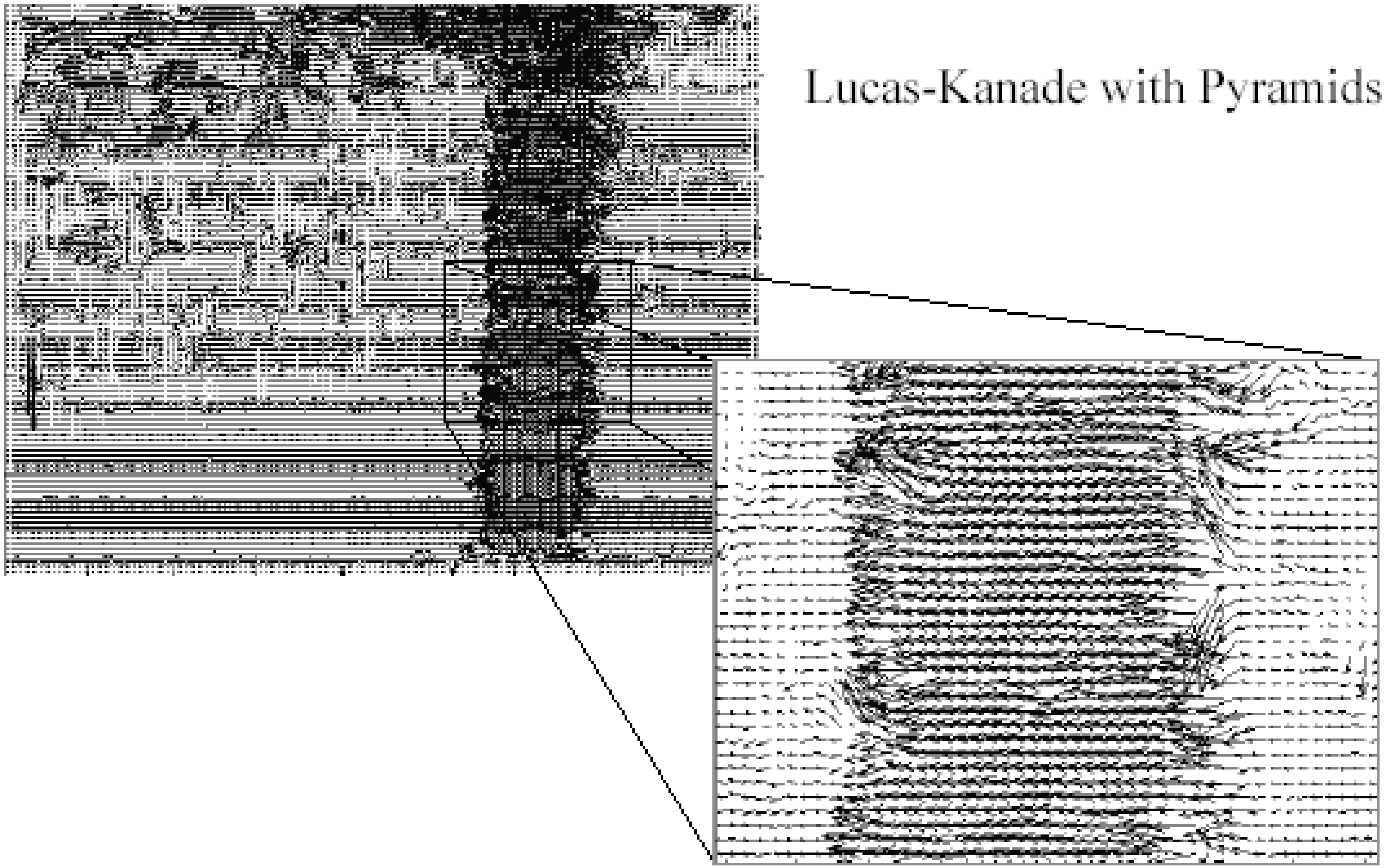
Multi-resolution registration



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

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

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

$$E(u(x,y),v(x,y)) = \iint (I_x u + I_y v + I_t)^2 + \alpha((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy$$

Data term brightness constancy

Smoothness term

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

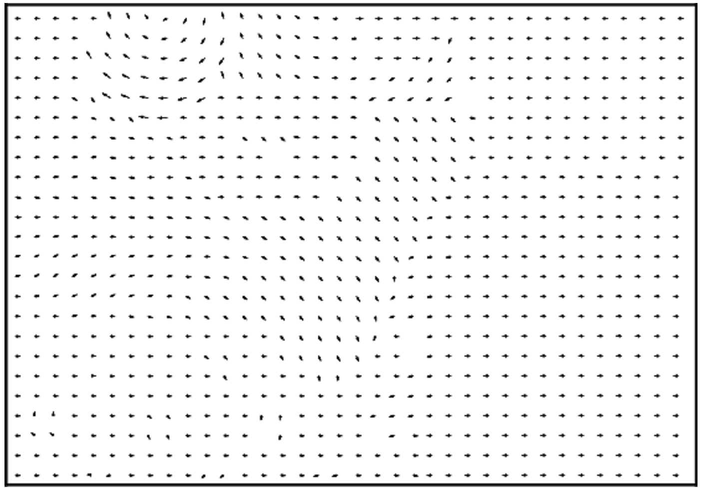
Euler-Lagrange equations

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

Coupled PDEs solved using iterative methods and finite differences

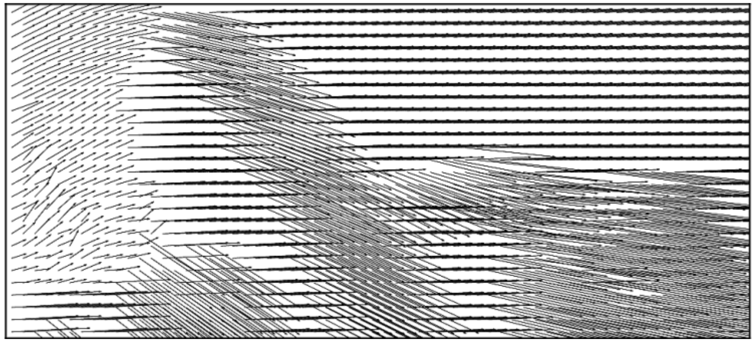
Horn & Schunck

- 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

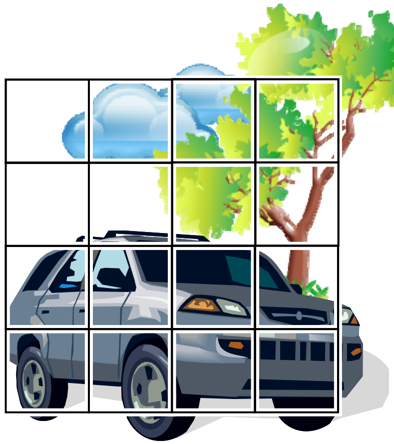
- Classical optical flow [Horn and Schunck 1981]
  - energy: $$E(\mathbf{w}) = \iint E_{data} + \alpha E_{smooth} \mathbf{dx}$$

    color/gradient constancy — smoothness constraint

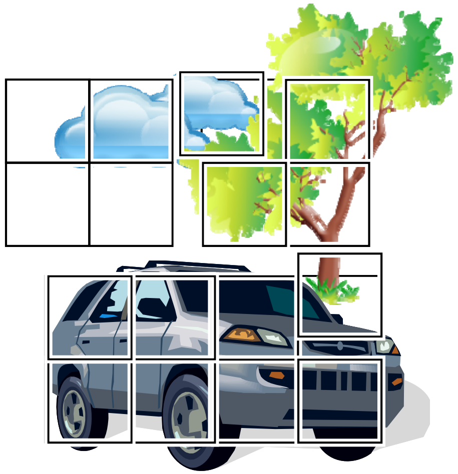
  - minimization using a coarse-to-fine scheme

- 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

Deep Matching: main idea



First image



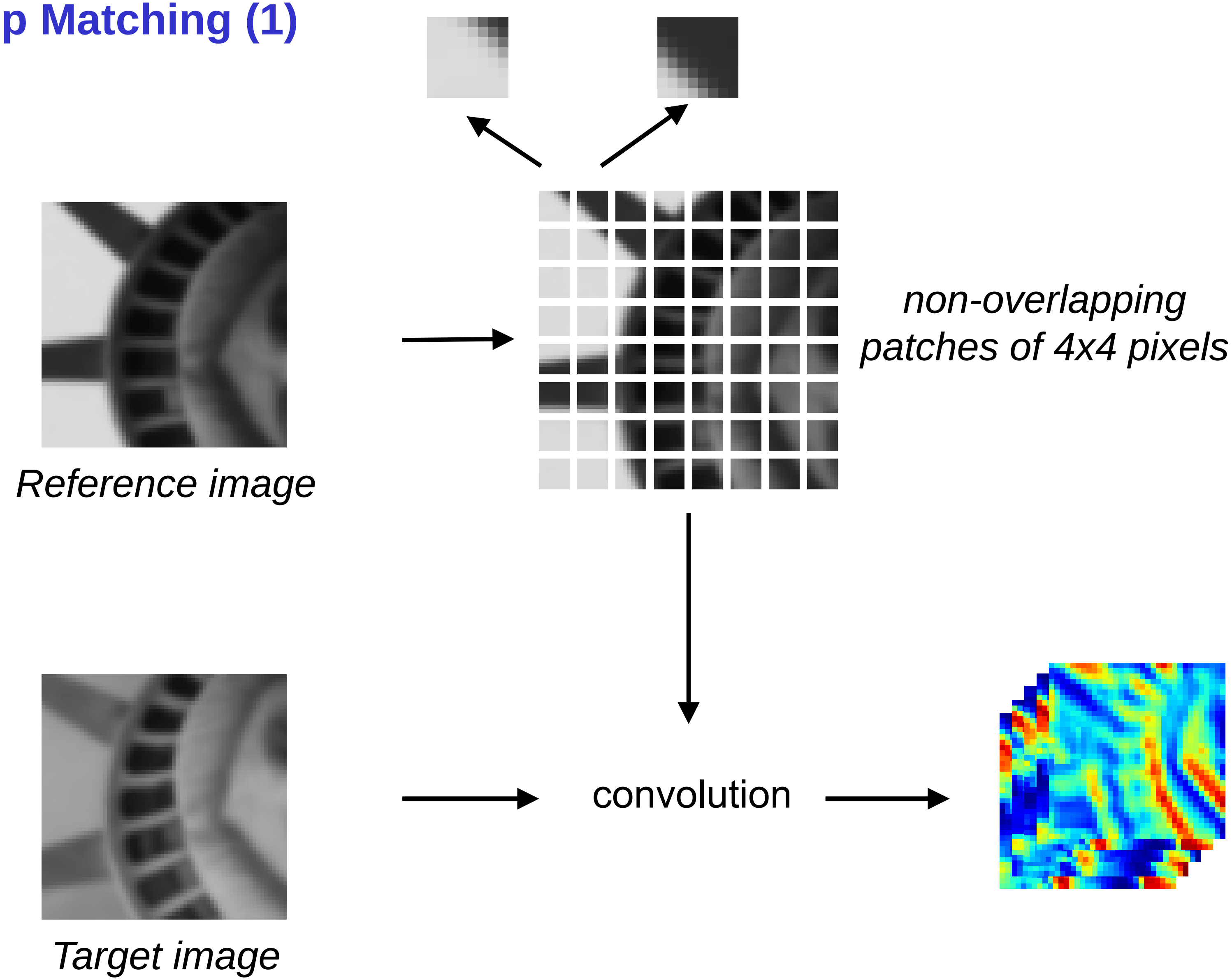
Second image

- Each subpatch is allowed to move:
  - independently
  - in a limited range depending on its size

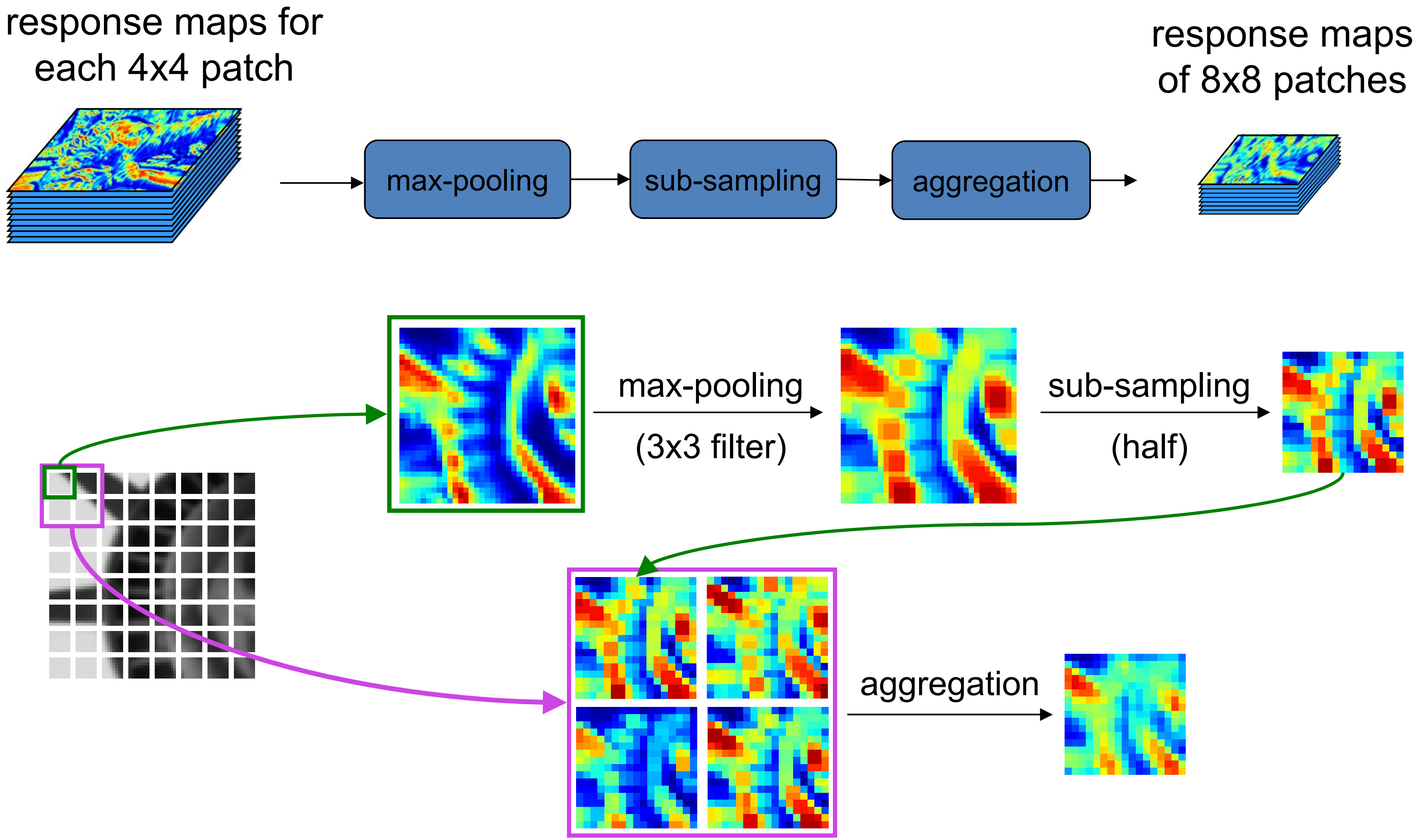
- The approach is fast to compute using convolution and max-pooling

- The idea is applied recursively

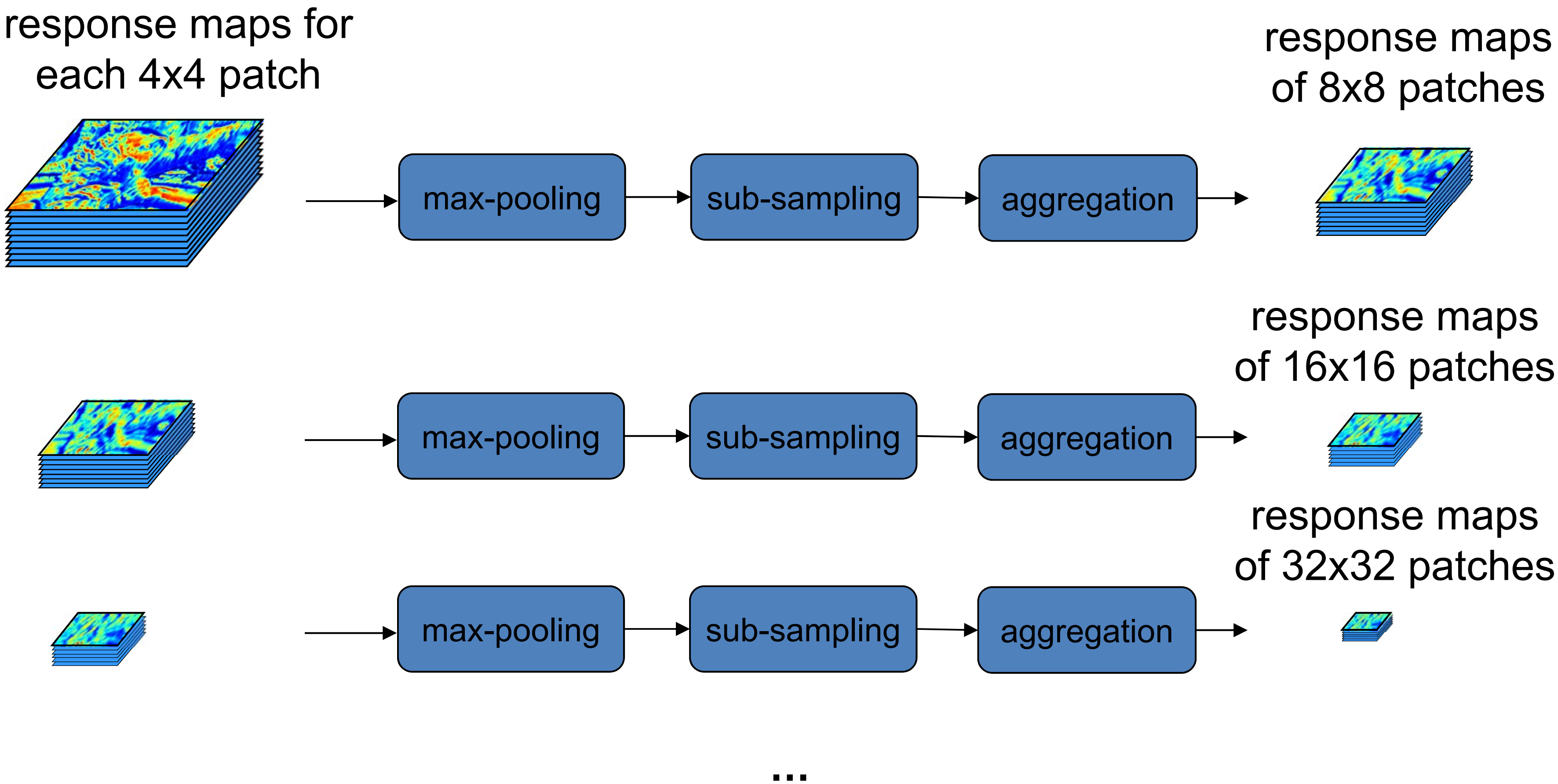
Deep Matching (1)



Deep Matching (2)

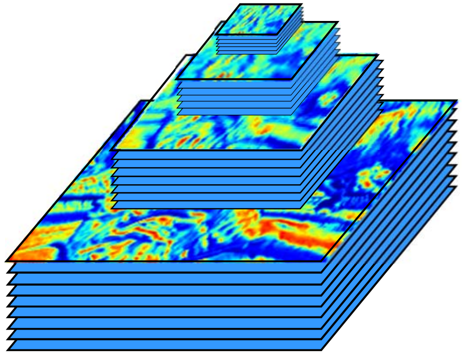


Deep Matching (2)



Pipeline similar in spirit to **deep** convolutional nets [Lecun *et al.* 1998]

Deep Matching (3)



Multi-scale response pyramid

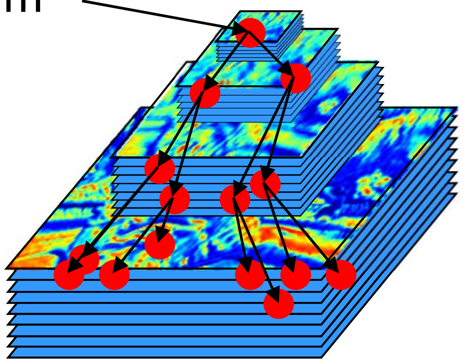
Extract scale-space local maxima

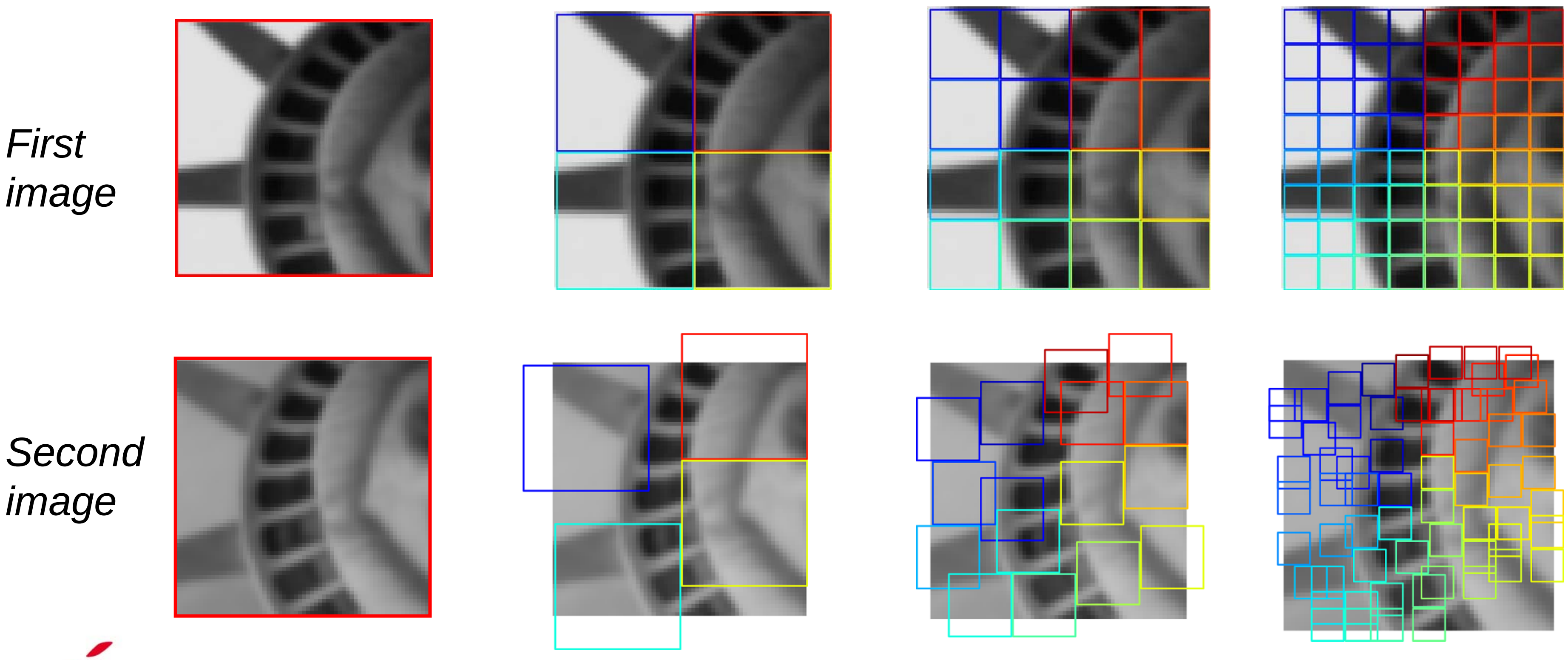
Backtrack quasi-dense correspondences

Bottom-up

Top-down

Deep Matching (3)



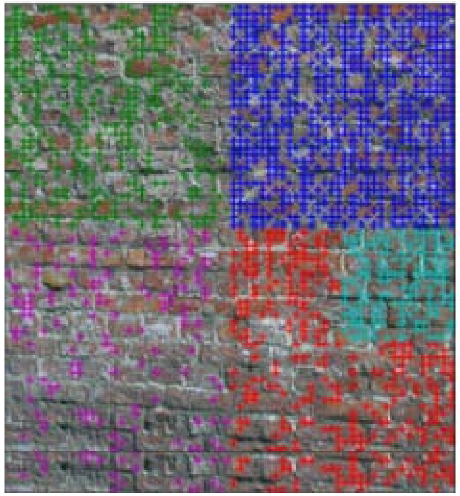


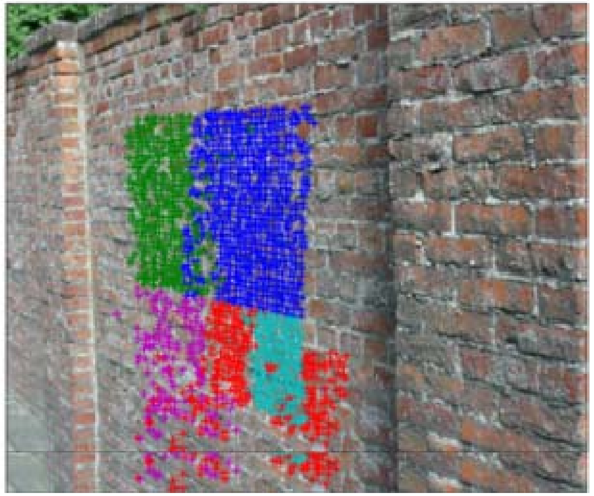
Deep Matching: example results

- Repetitive textures

First image

Second image

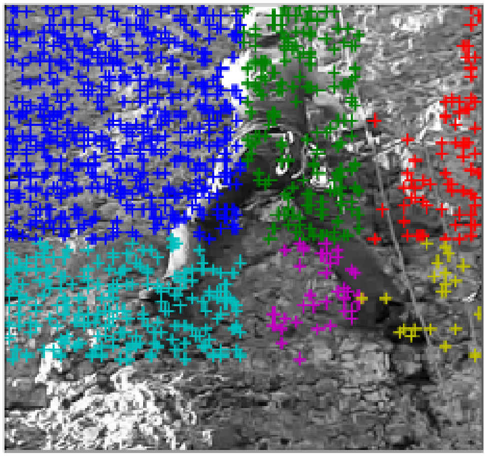


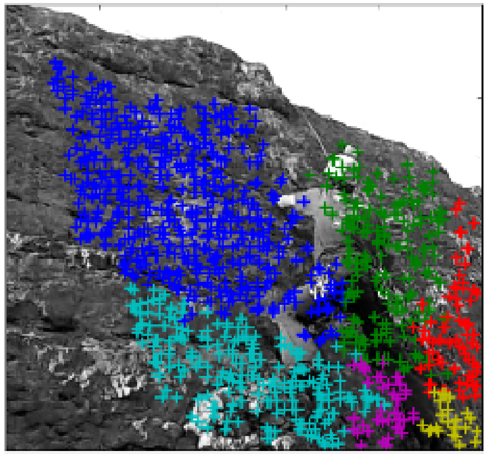


Deep Matching: example results

- Non-rigid deformation

First image *Second image*





DeepFlow

- Classical optical flow [Horn and Schunck 1981]
  - energy $E(\mathbf{w}) = \iint E_{data} + \alpha E_{smooth} \mathbf{dx}$
- Integration of Deep Matching
  - energy $E(\mathbf{w}) = \iint E_{data} + \alpha E_{smooth} + \beta E_{match} \mathbf{dx}$
  - matches guide the flow
  - similar to [Brox and Malik 2011]
- Minimization using:
  - coarse-to-fine strategy
  - fixed point iterations
  - Successive Over Relaxation (SOR)

Experimental results: datasets

- MPI-Sintel [Butler *et al.* 2012]
  - ► sequences from a realistic animated movie
  - ► large displacements (>20px for 17.5% of pixels)
  - ► atmospheric effects and motion blur

informatics mathematics

Experimental results: datasets

- KITTI [Geiger *et al.* 2013]
  - sequences captured from a driving platform
  - large displacements (>20px for 16% of pixels)
  - real-world: lightings, surfaces, materials

Experimental results: sample results

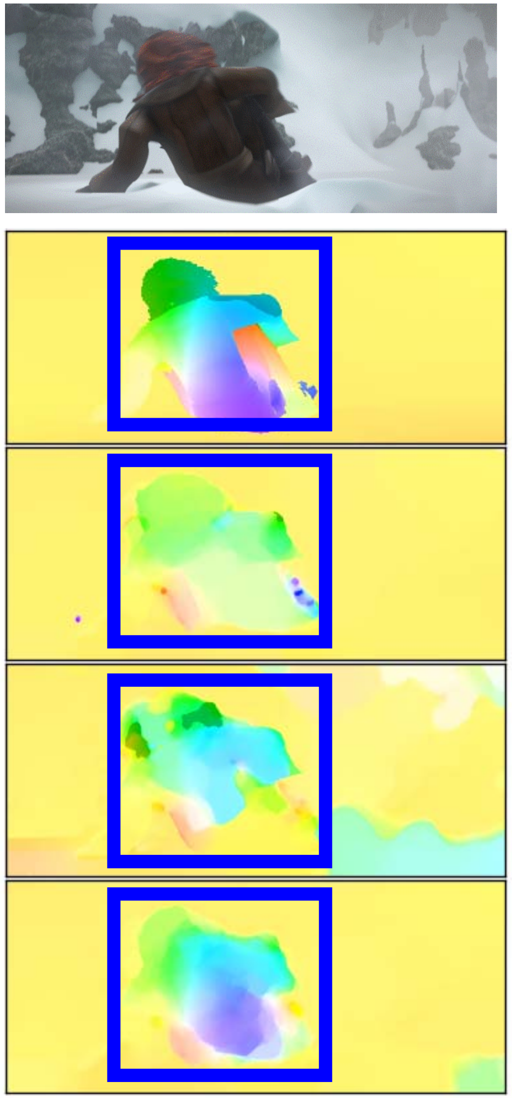
Ground-truth

LDOF [Brox & Malik 2011]

MDP-Flow2 [Xu *et al.* 2012]

DeepFlow

Experimental results: sample results



Ground-truth

LDOF [Brox & Malik 2011]

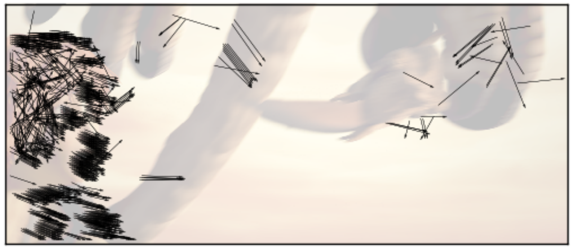
MDP-Flow2 [Xu *et al.* 2012]

DeepFlow

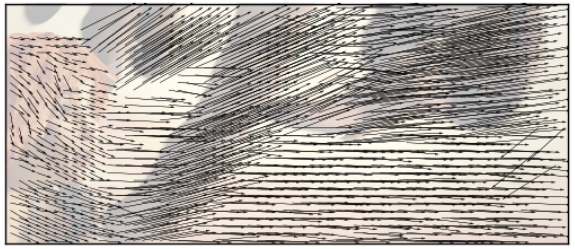
Experimental results: improvements due to Deep Matching

- Comparison on MPI-Sintel training set
  - AEE: average endpoint error
  - s40+: only on large displacements

Matching	Flow evaluation	
	AEE	s40+
No match	5.54	39.86
KLT [OpenCV]	5.51	39.20
SIFT-NN	5.44	38.28
HOG-NN	5.27	37.86
Deep Matching	**4.42**	**29.23**



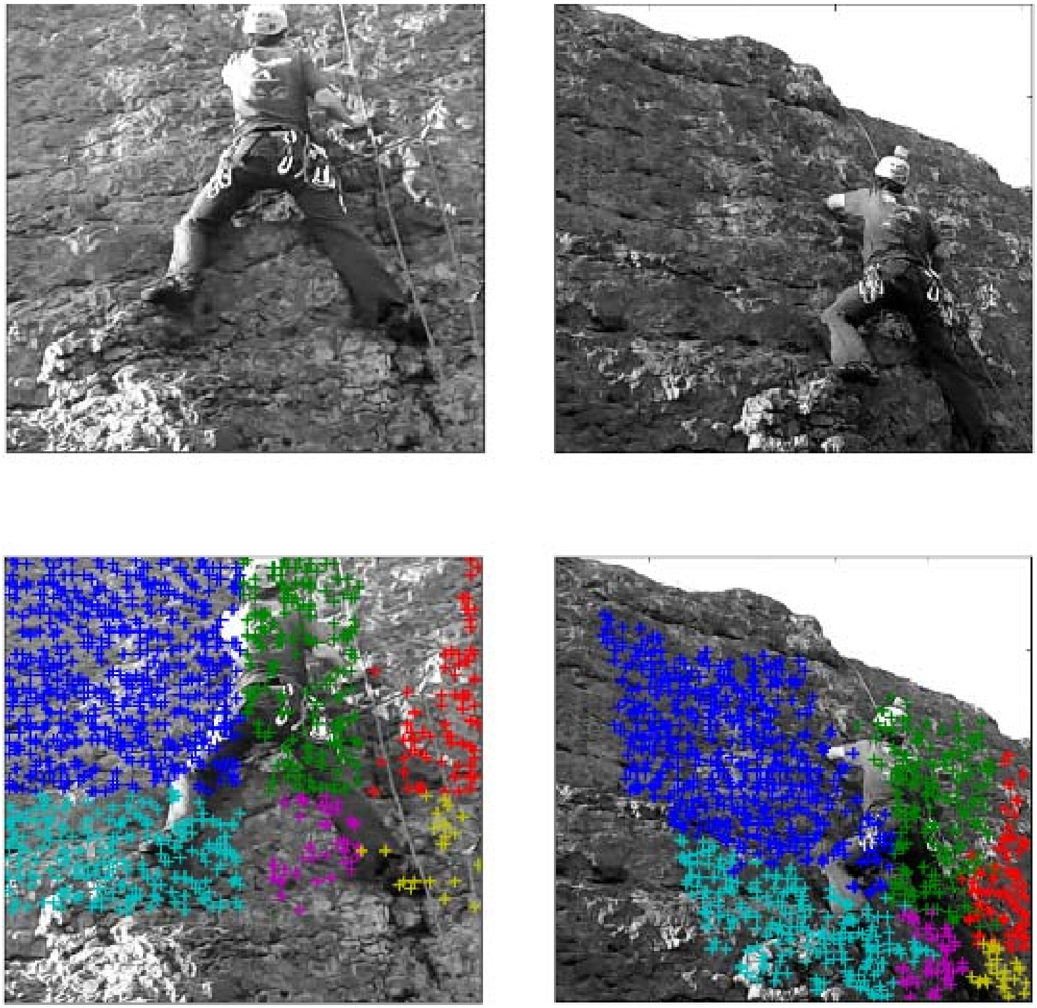
HOG matching



Deep Matching

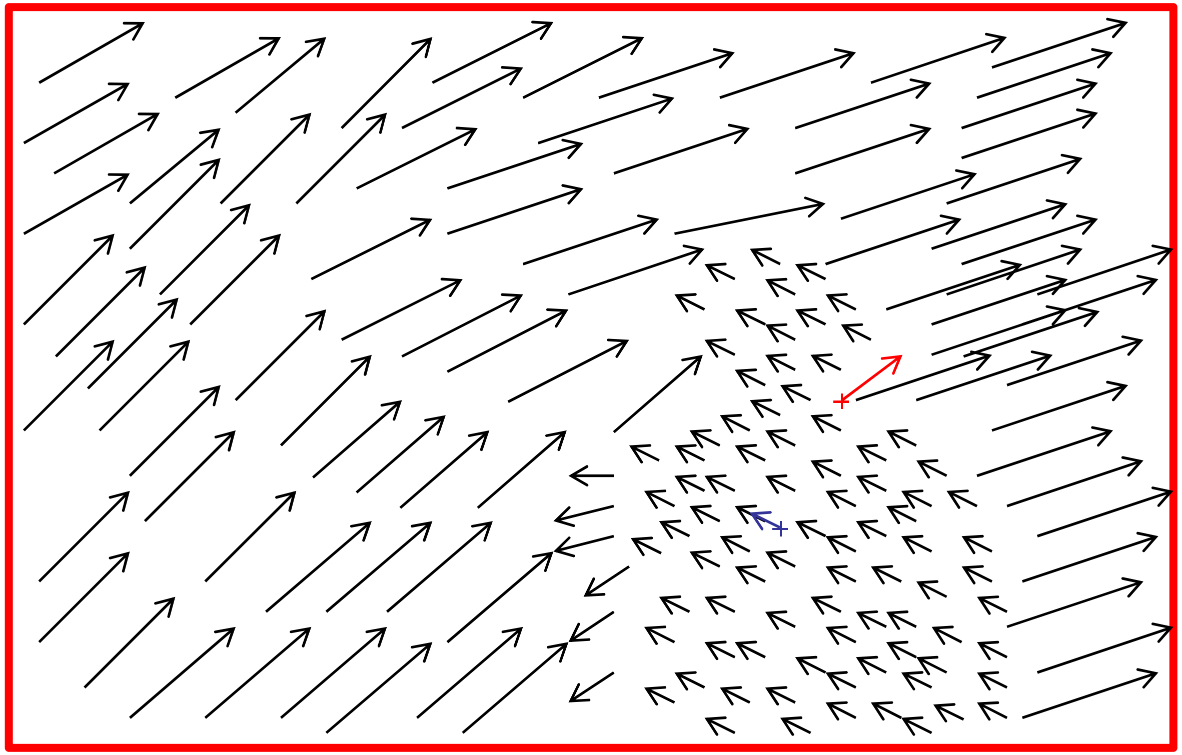
EpicFlow: Sparse-to-dense interpolation based on Deep Matching

- accurate quasi dense matches with DeepMatching



[Revaud et al., CVPR'15]

Approach: Sparse-to-dense interpolation based on Deep Matching



Does not respect motion boundaries

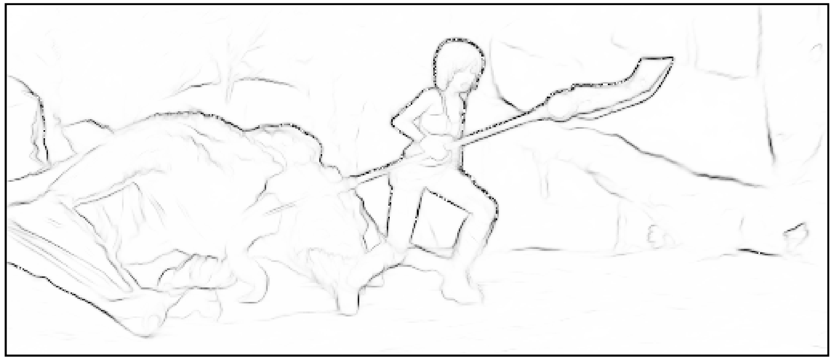
Interpolation

Ground-Truth

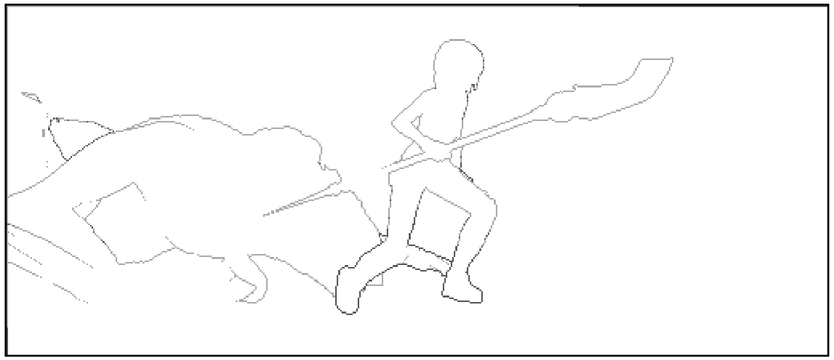
Approach: Sparse-dense interpolation with motion boundaries

- image edges often coincide with motion boundaries (recall 95%)
- state-of-the-art SED detector [structured forest for edge detection, Dollar'13]

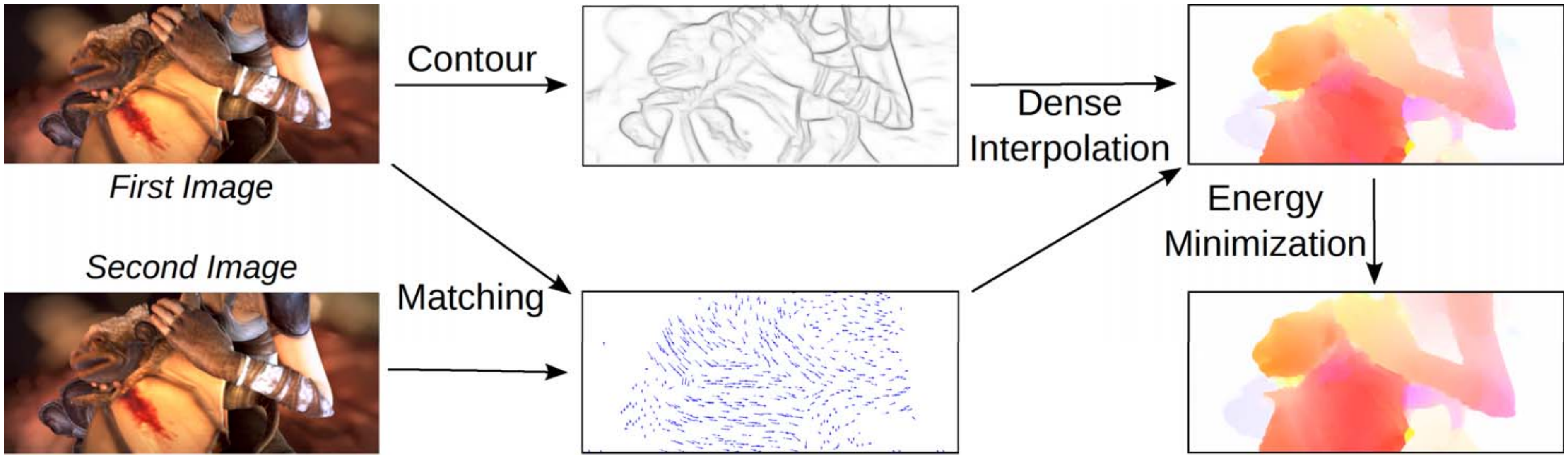

image


SED edges


ground-truth flow


ground-truth motion boundaries

Approach: Sparse-dense interpolation with motion boundaries



EpicFlow:

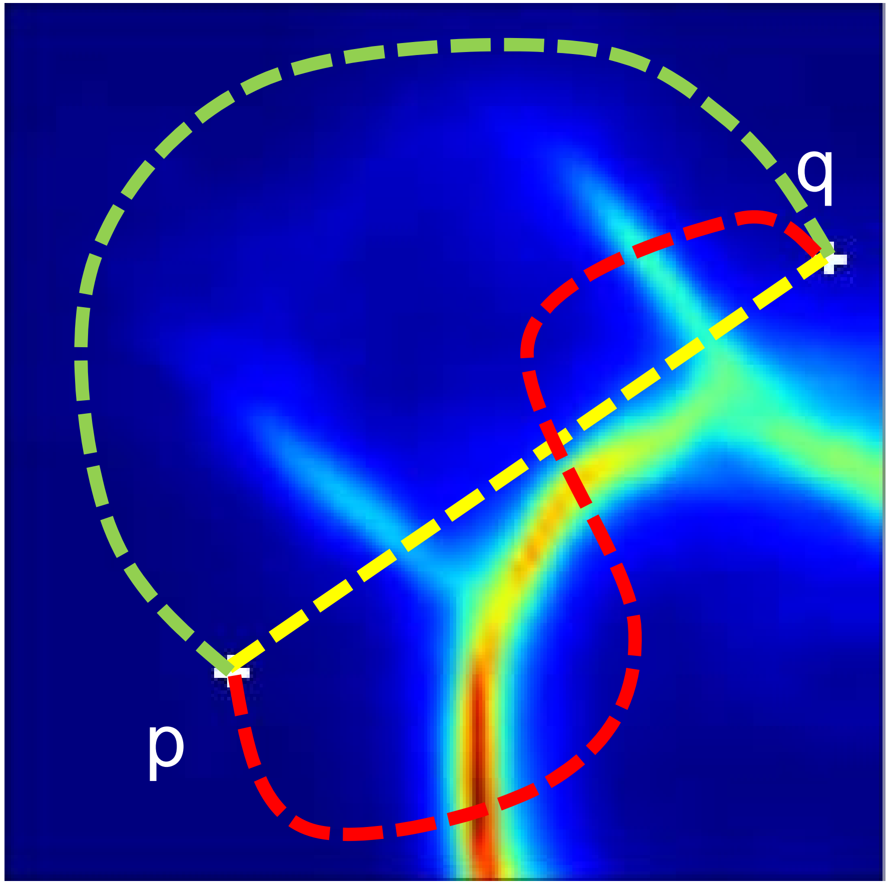
- Matching [Deep Matching]
- Sparse-dense interpolation preserving motion boundaries
  → Geodesic distance based on edges [SED]
- Refinement: One-level energy minimization with variational approach

Distance $D(\boldsymbol{p}_m, \boldsymbol{p})$: edge-aware geodesic distance

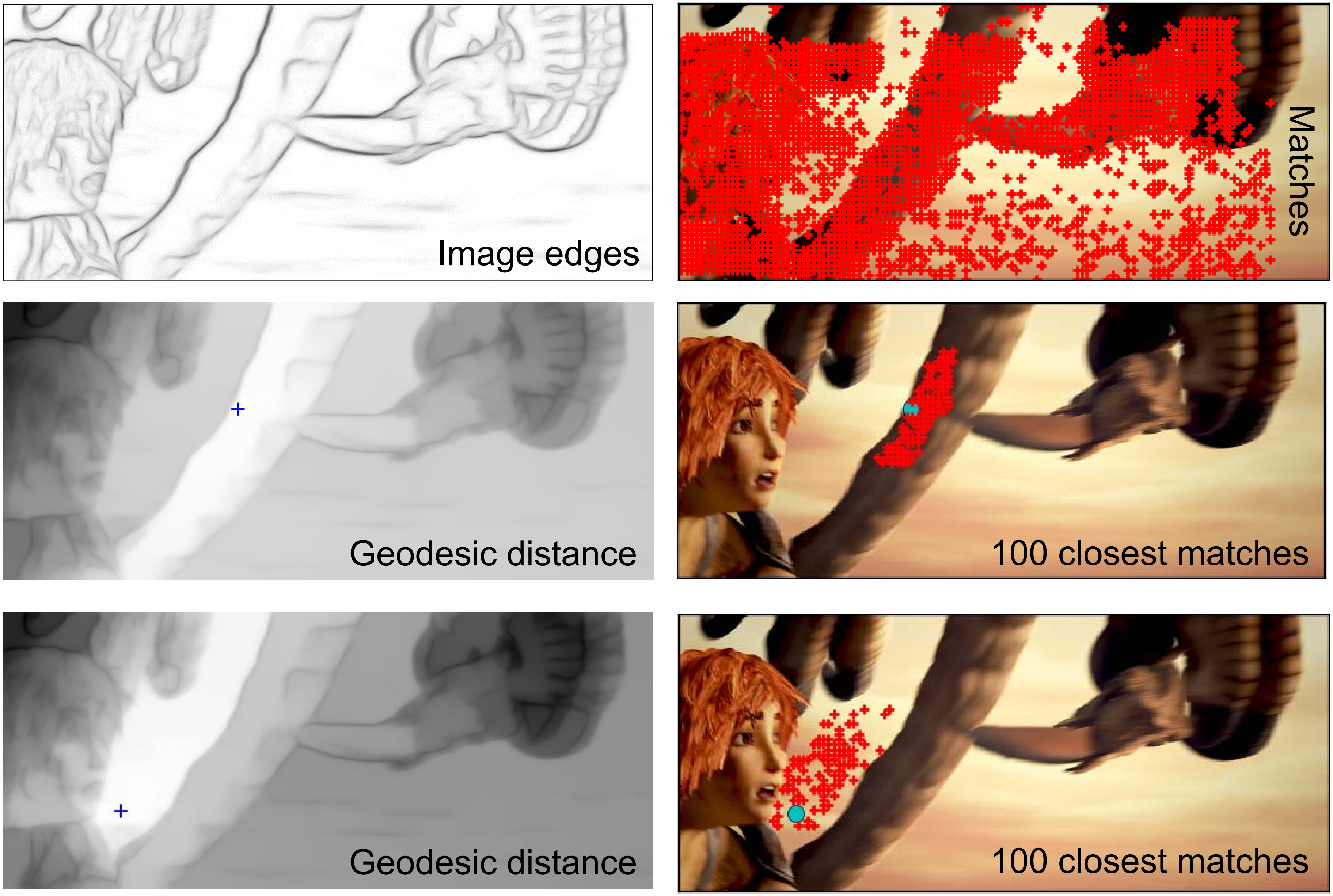
- geodesic distance:
  - shortest distance
  - knowing a cost map C

$$D_G(\boldsymbol{p}, \boldsymbol{q}) = \inf_{\Gamma \in \mathcal{P}_{\boldsymbol{p},\boldsymbol{q}}} \int_\Gamma C(\boldsymbol{p}_s) d\boldsymbol{p}_s$$

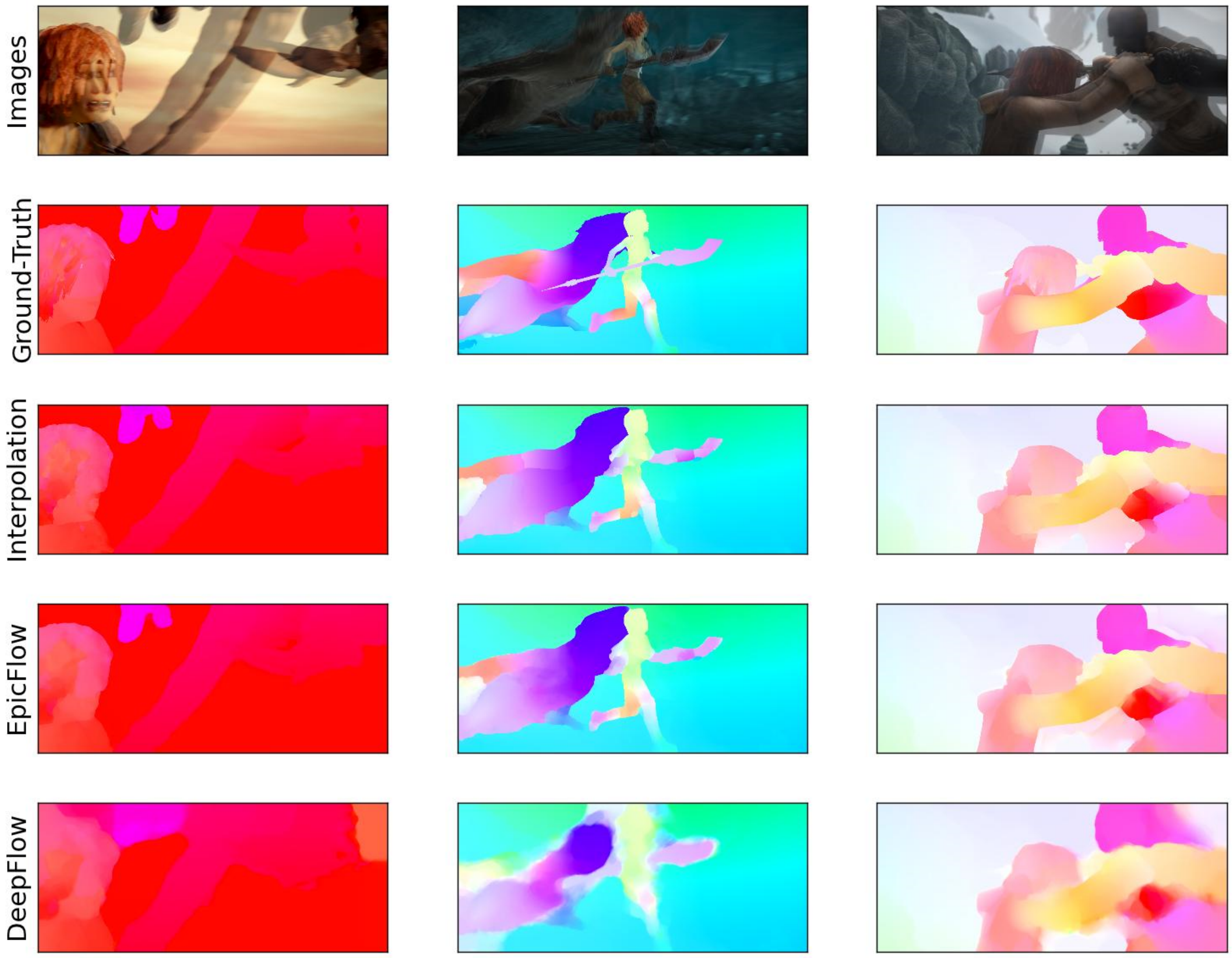
- Cost map C:
  - image edges
  - here computed with SED



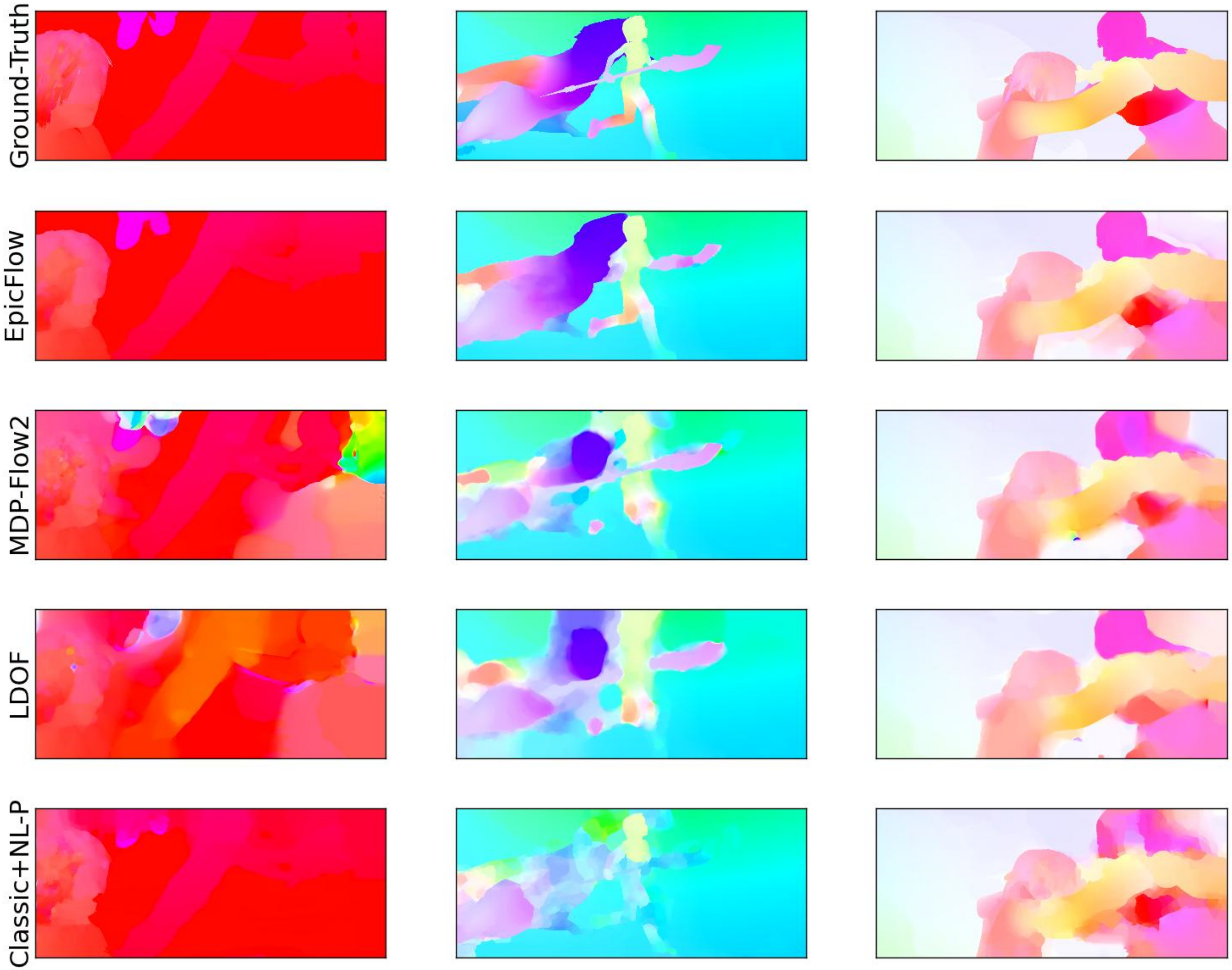
Sparse-to-dense Interpolation: edge-aware geodesic distance



Comparing Interpolation/EpicFlow/DeepFlow



Comparison to the state of the art



Comparison to the state of the art (AEE)

Method	Error on MPI-Sintel	Error on Kitti	Timings
EpicFlow	**6.28**	**3.8**	**16.4s**
TF+OFM	6.73	5.0	~500s
DeepFlow	7.21	5.8	19s
NLTGV-SC	8.75	**3.8**	16s (GPU)

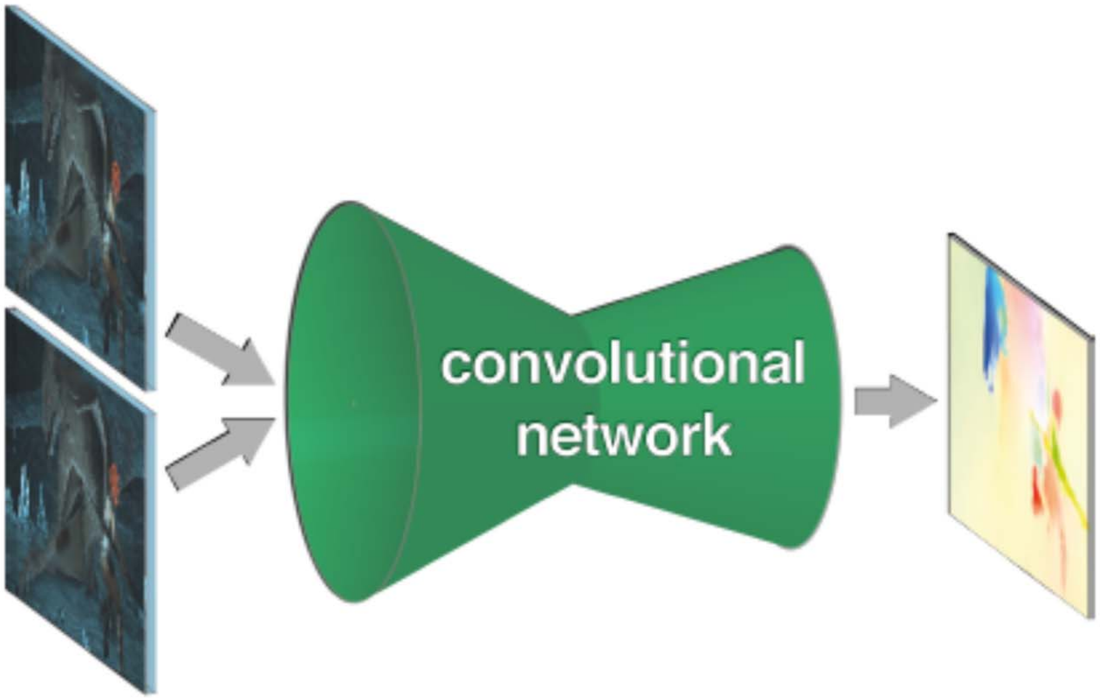
TF + OFM : Kennedy'15. Optical flow with geometric occlusion estimation and fusion of multiple frames
NLTGV-SC: Ranftl'14. Non-local total generalized variation for optical flow estimation.

Failure cases

Missing matches
(spear and horns of dragon)

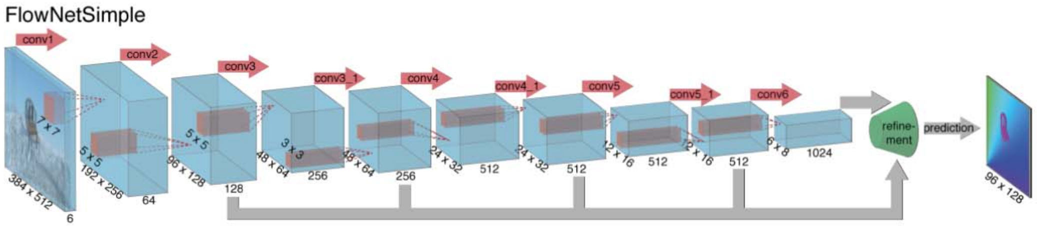
Missing contours
(arm)

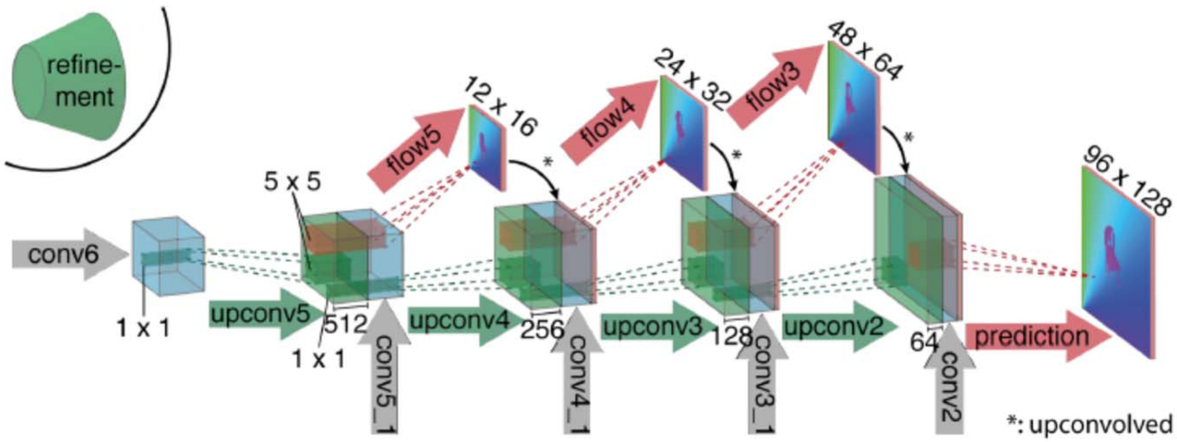
CNN to estimate optical flow: FlowNet



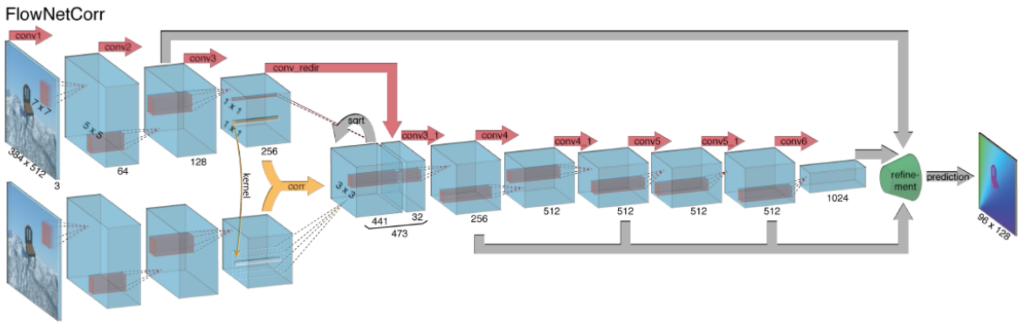
[A. Dosovitskiy et al. ICCV'15]

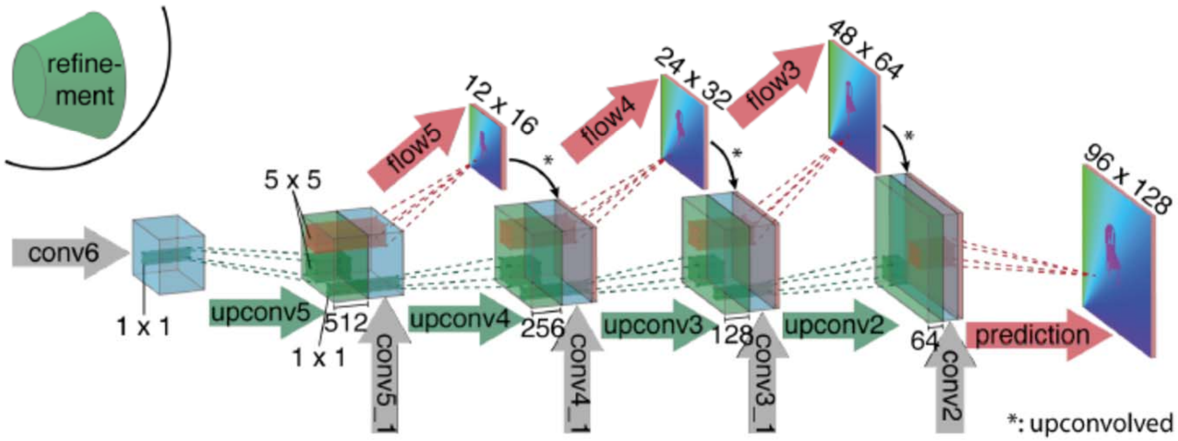
Architecture FlowNetSimple





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