Low-level Vision Tasks

Optical Flow

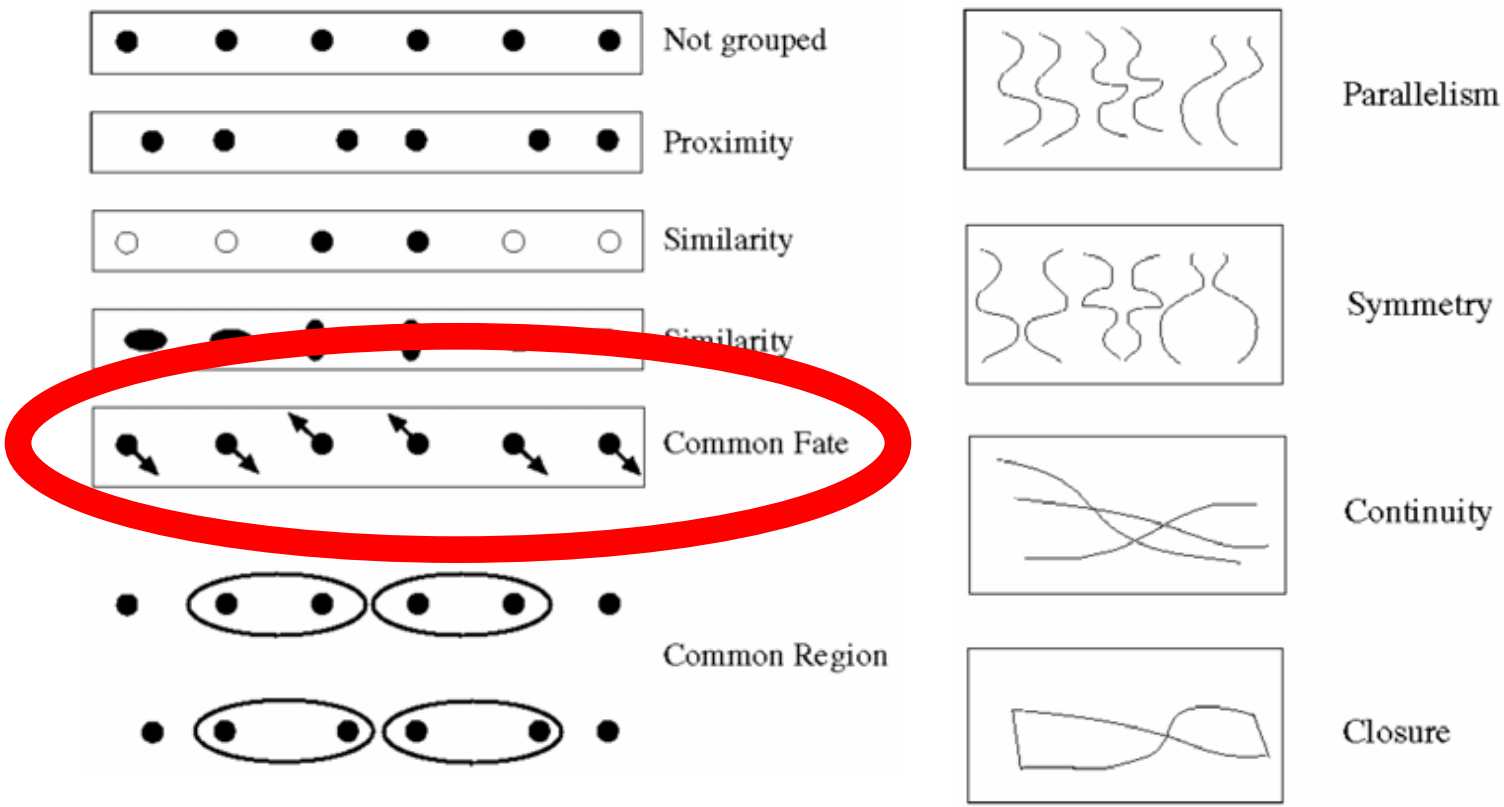
Kuk-Jin Yoon

Visual Intelligence Lab.
Department of Mechanical Engineering

KAIST

Motion and Perceptual Organization

- Sometimes, motion is the only cue



Motion and Perceptual Organization

- Even “impoverished” motion data can evoke a strong percept

G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", Perception and Psychophysics 14, 201-211, 1973.

KAIST

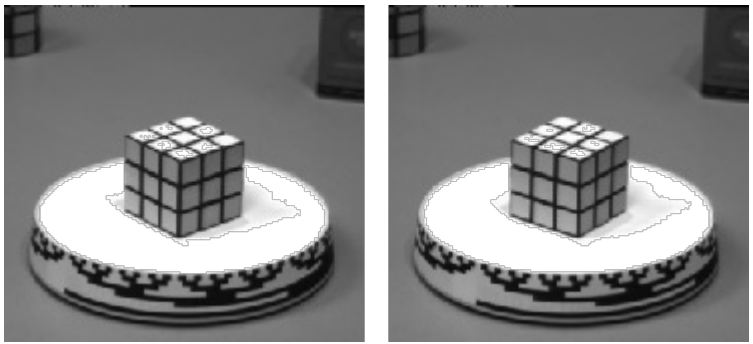
Motion and Perceptual Organization

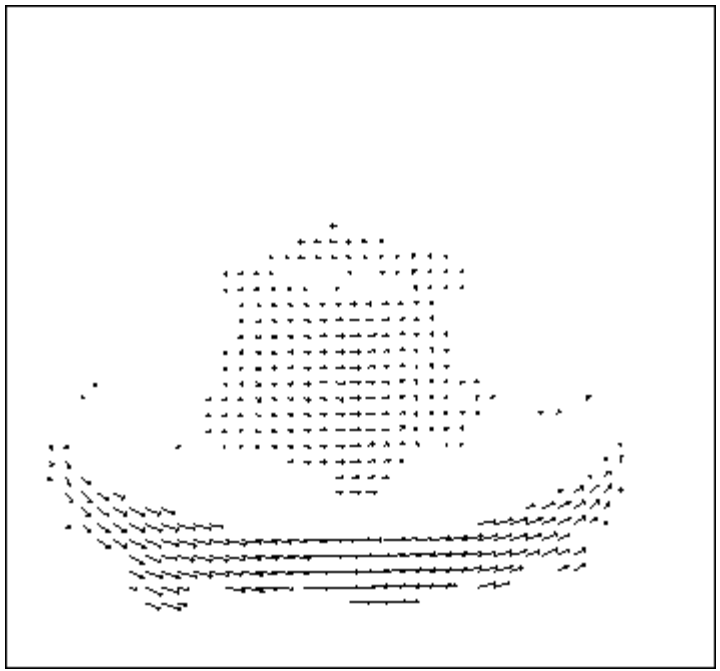
- Even “impoverished” motion data can evoke a strong percept

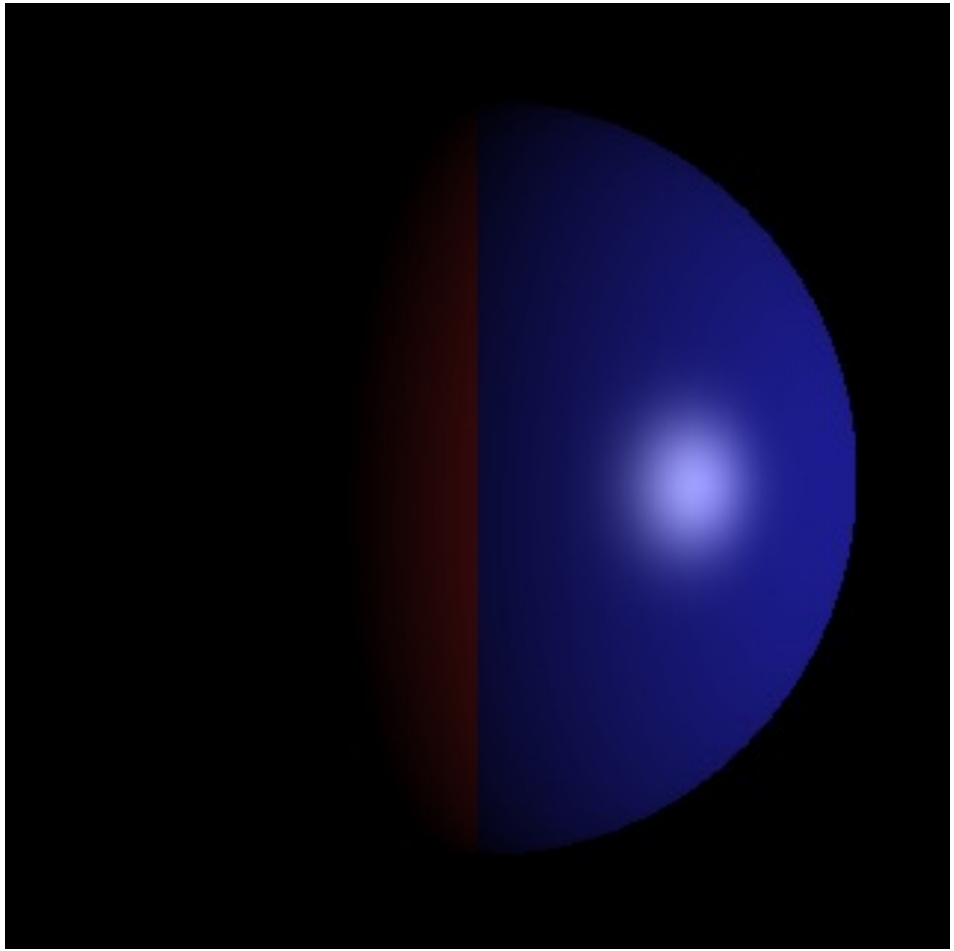
G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", Perception and Psychophysics 14, 201-211, 1973.

Motion Field

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







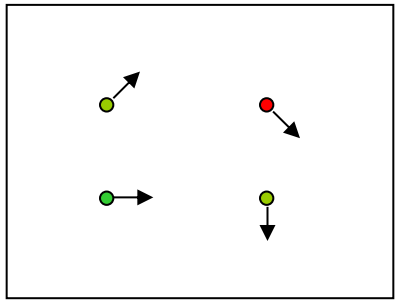
Definition of Image Flow and Optical Flow

- Image flow depicts, at each point in the image surface, a 2D projection of the instantaneous 3D velocity of the corresponding point in the scene.

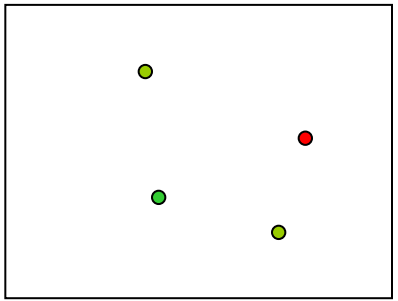
- Optical flow is the 2D distribution of apparent velocities that can be associated with the variation of brightness patterns on the image.
  - No mention of the scene!

Problem Definition: Optical Flow

- How to estimate pixel motion from image H to image I?



$H(x, y)$

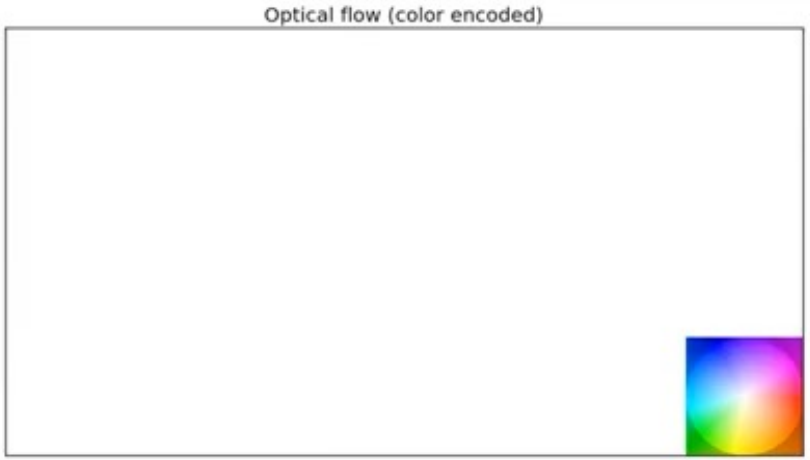


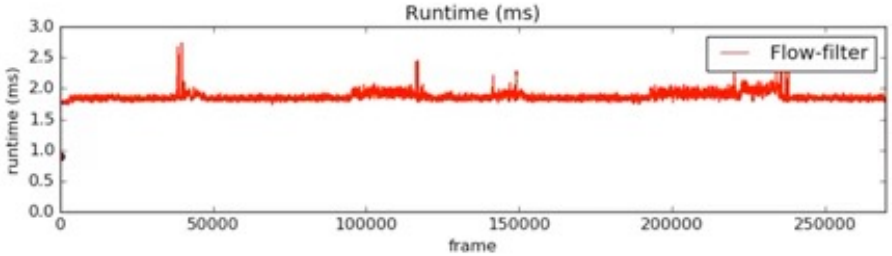
$I(x, y)$

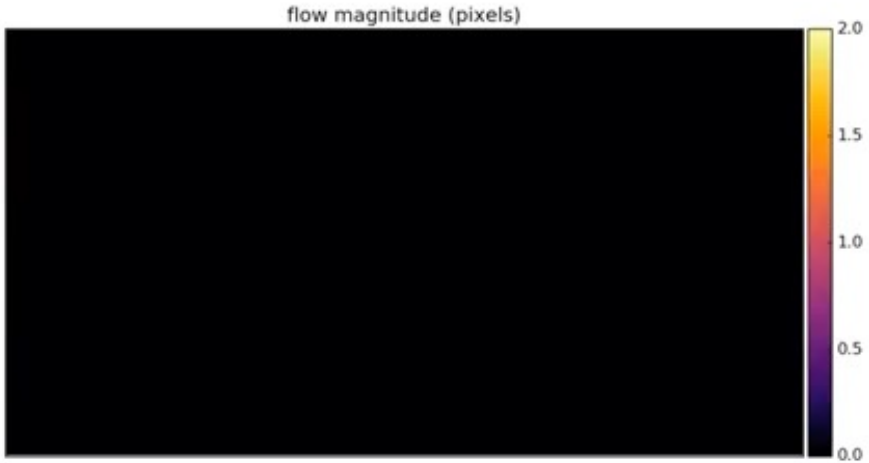
Why is it useful?

- Depth (3D) reconstruction
- Motion detection - tracking moving objects
- Compression
- Mosaics
- And more…

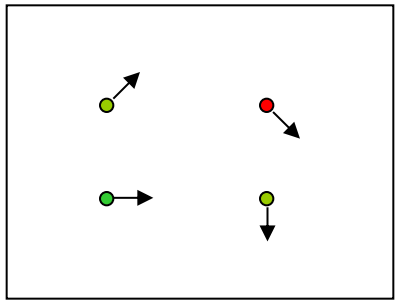
Problem Definition: Optical Flow



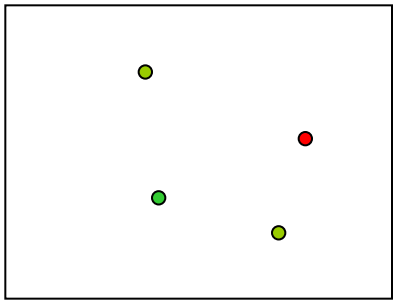




Problem Definition: Optical Flow



$H(x, y)$



$I(x, y)$

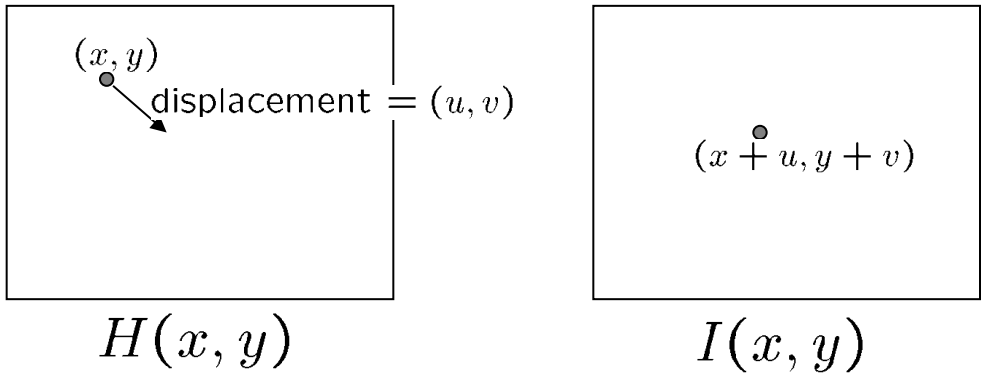
- How to estimate pixel motion from image H to image I?
  - Solve pixel correspondence problem
    - given a pixel in H, look for nearby pixels of the same color in I

  Key assumptions
  - color constancy: a point in H looks the same in I
    - For grayscale images, this is brightness constancy
  - small motion: points do not move very far

  This is called the optical flow problem

Optical Flow Constraints (Grayscale Images)

- Let's take a closer look at these constraints:



- brightness constancy: Q: what's the equation?

- small motion: (u and v are less than 1 pixel)
  - suppose we take the Taylor series expansion of I:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

$$\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Optical Flow Equation

- Combining these two equations

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

$$
\begin{aligned}
0 &= I(x+u, y+v) - H(x,y) \\
&\approx I(x,y) + I_x u + I_y v - H(x,y) \\
&\approx (I(x,y) - H(x,y)) + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot [u \;\; v]
\end{aligned}
$$

In the limit, as u and v go to zero, this approximation is good

$$0 = I_t + \nabla I \cdot [\frac{\partial x}{\partial t} \;\; \frac{\partial y}{\partial t}]$$

Optical Flow Equation

$$0 = I_t + \nabla I \cdot [u \ v]$$

- Q: How many equations and how many unknowns do we have per pixel?

Intuitively, what does this constraint mean?

- The component of the flow in the gradient direction is determined
- The component of the flow parallel to an edge is unknown

This explains the Barber Pole illusion
http://www.sandlotscience.com/Ambiguous/barberpole.htm

Getting More Equations

- How to get more equations for a pixel?
  - Basic idea: impose additional constraints
    - most common is to assume that the flow field is smooth locally
    - one method: assume the pixel's neighbors have the same (u,v)
      - If we use a 5x5 window, this gives us 25 equations per pixel!

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u\ v]$$

$$\underbrace{\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix}}_{A\ 25\times2} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{d\ 2\times1} = - \underbrace{\begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}}_{b\ 25\times1}$$

Lukas-Kanade Flow

- We have over constrained equation set:

$$\underset{25\text{x}2}{A}\ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

Solution: solve least squares problem

- minimum least squares solution given by solution (in d) of:

$$\underset{2\text{x}2}{(A^T A)}\ \underset{2\text{x}1}{d} = \underset{2\text{x}1}{A^T b}$$

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

$A^T A$ $\qquad$ $A^T b$

- The summations are over all pixels in the K x K window
- This technique was first proposed by Lukas & Kanade (1981)

When Can We Solve This ?

- Optimal (u, v) satisfies Lucas-Kanade equation

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

$$A^T A \qquad\qquad A^T b$$

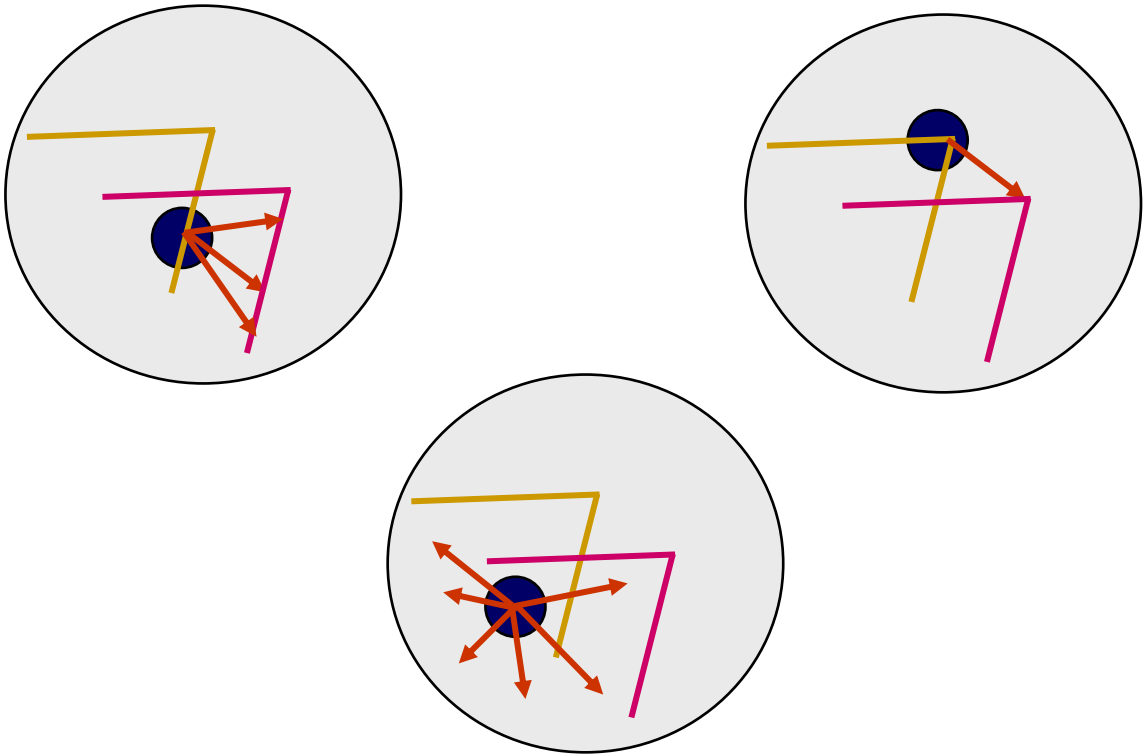
When is this solvable?

- A^TA should be invertible
- The eigenvalues of A^TA should not be too small (noise)
- A^TA should be well-conditioned
  - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

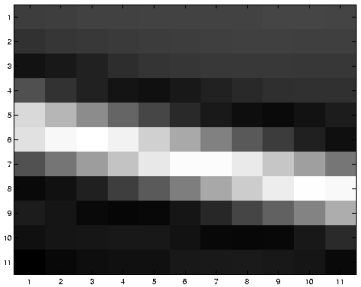
A^TA is invertible when there is no aperture problem

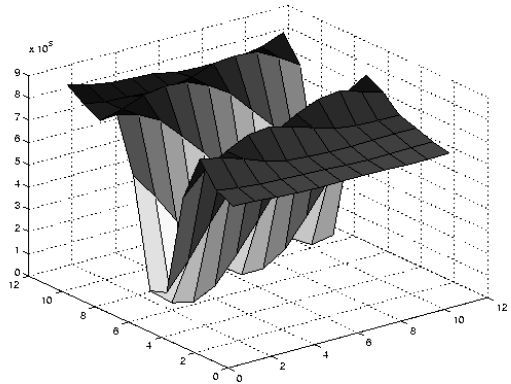
$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Local Patch Analysis



Edge

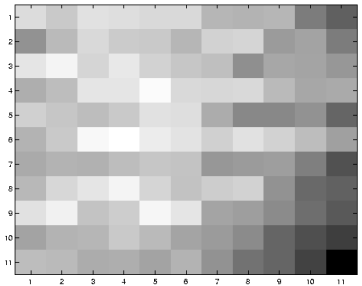


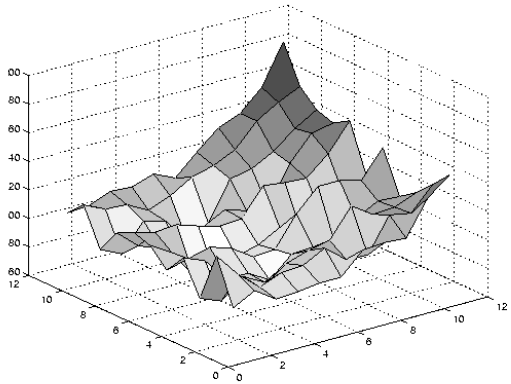


$$\sum \nabla I (\nabla I)^T$$

- large gradients, all the same
- large l_1, small l_2

Low Texture Region



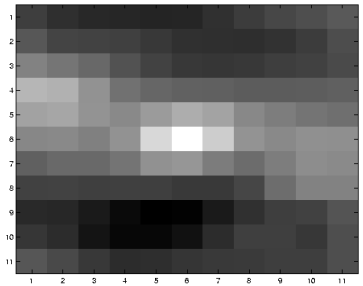


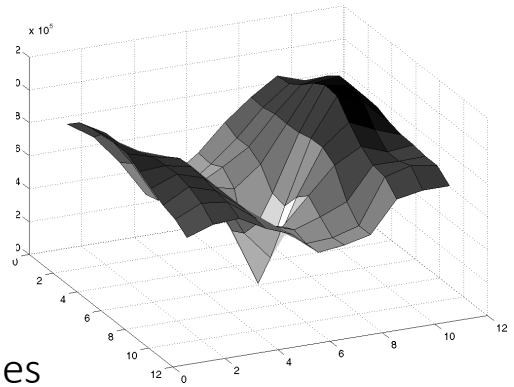
$$\sum \nabla I (\nabla I)^T$$

– gradients have small magnitude

– small l_1, small l_2

High Textured Region





$$\sum \nabla I (\nabla I)^T$$

– gradients are different, large magnitudes

– large l_1, large l_2

Observation

- This is a two image problem BUT
  - Can measure sensitivity/uncertainty by just looking at one of the images!
  - This tells us which pixels are easy to track, which are hard
    - very useful later on when we do feature tracking...

Errors in Lukas-Kanade

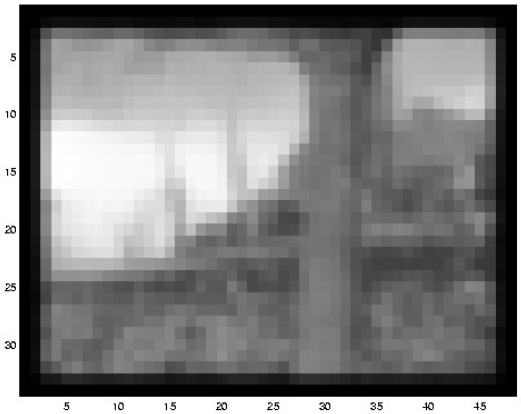
- What are the potential causes of errors in this procedure?
  - Suppose A^TA is easily invertible
  - Suppose there is not much noise in the image

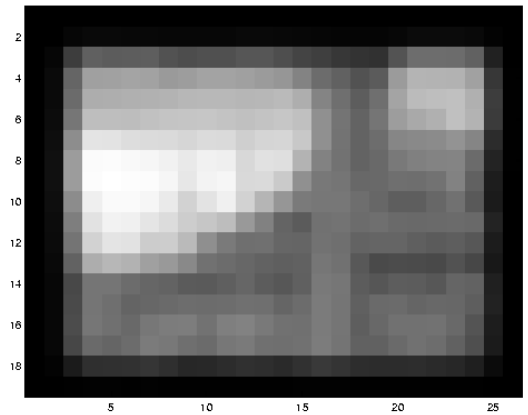
- When our assumptions are violated
  - Brightness constancy is **not** satisfied
  - The motion is **not** small
  - A point does **not** move like its neighbors
    - window size is too large
    - what is the ideal window size?

Revisiting the Small Motion Assumption

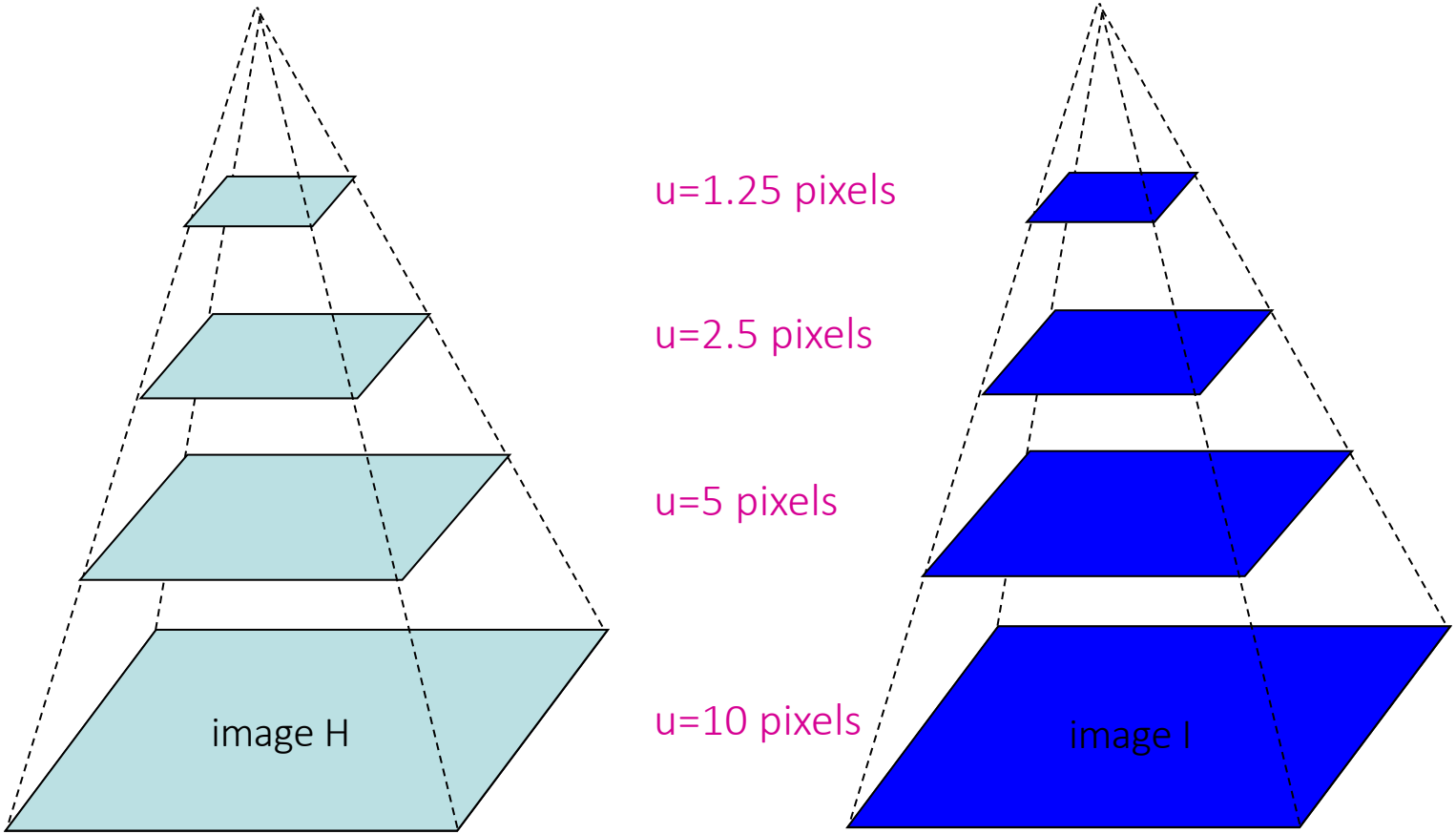
- Is this motion small enough?
  - Probably not—it's much larger than one pixel (2nd order terms dominate)
  - How might we solve this problem?

Reduce the Resolution!

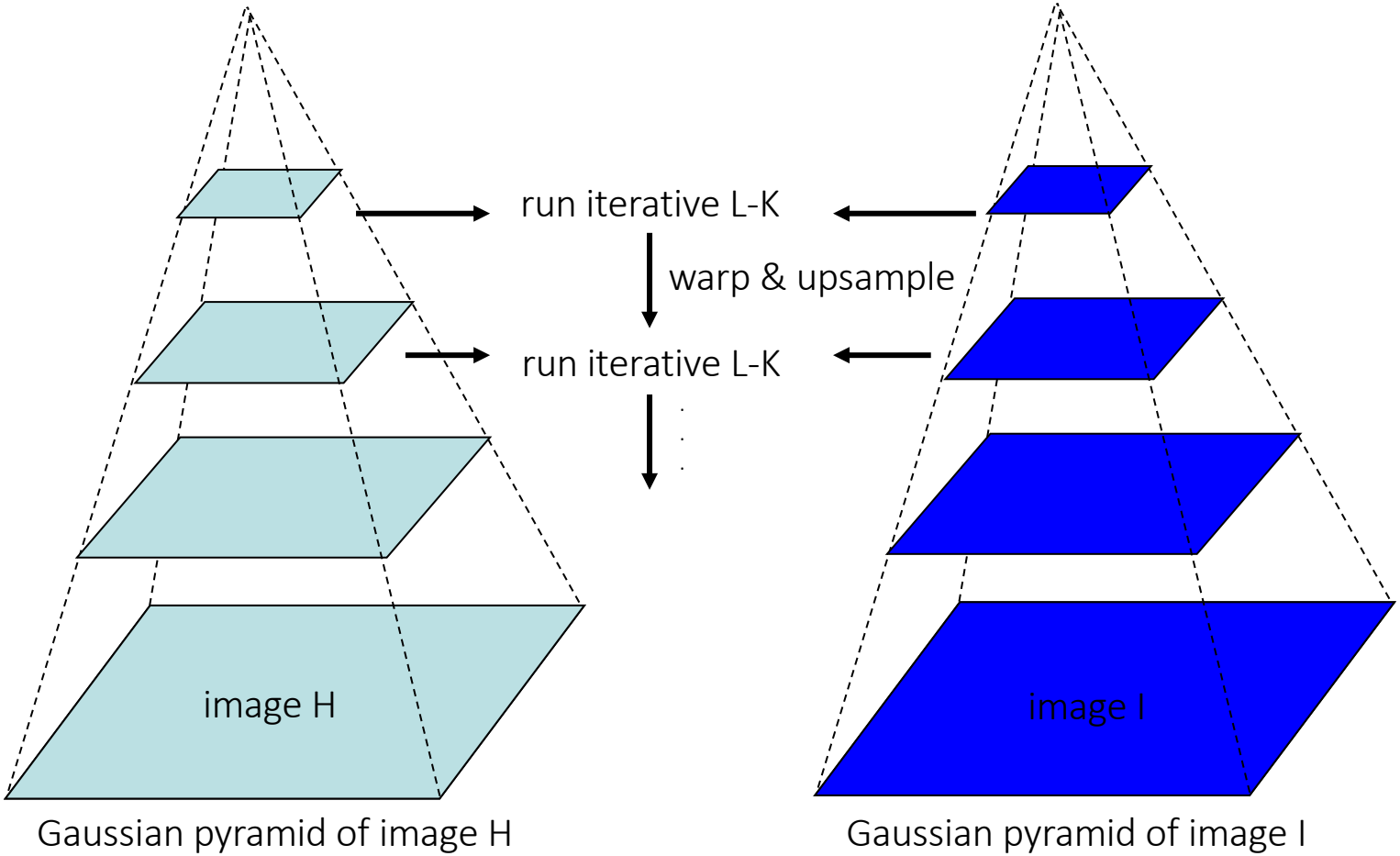




Coarse-to-fine Optical Flow Estimation



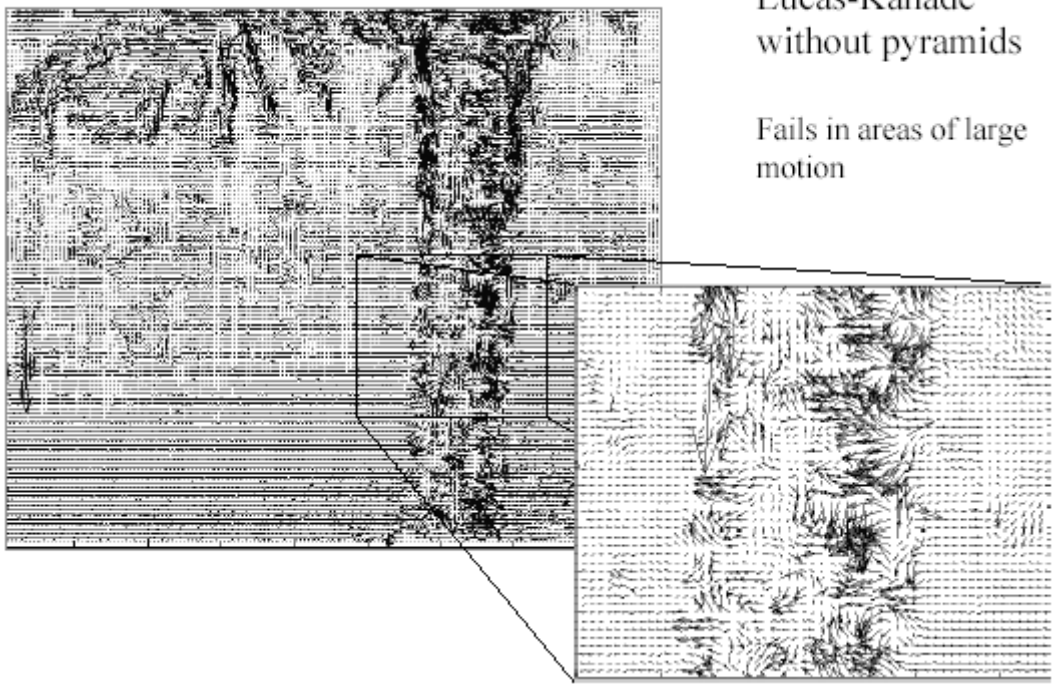
Coarse-to-fine Optical Flow Estimation



Multi-resolution Lucas Kanade Algorithm

- Compute Iterative LK at highest level
- For Each Level i
  - Take flow u(i-1), v(i-1) from level i-1
  - Upsample the flow to create u*(i), v*(i) matrices of twice resolution for level i.
  - Multiply u*(i), v*(i) by 2
  - Compute It from a block displaced by u*(i), v*(i)
  - Apply LK to get u'(i), v'(i) (the correction in flow)
  - Add corrections u'(i), v'(i) to obtain the flow u(i), v(i) at the ith level, i.e., u(i)=u*(i)+u'(i), v(i)=v*(i)+v'(i)

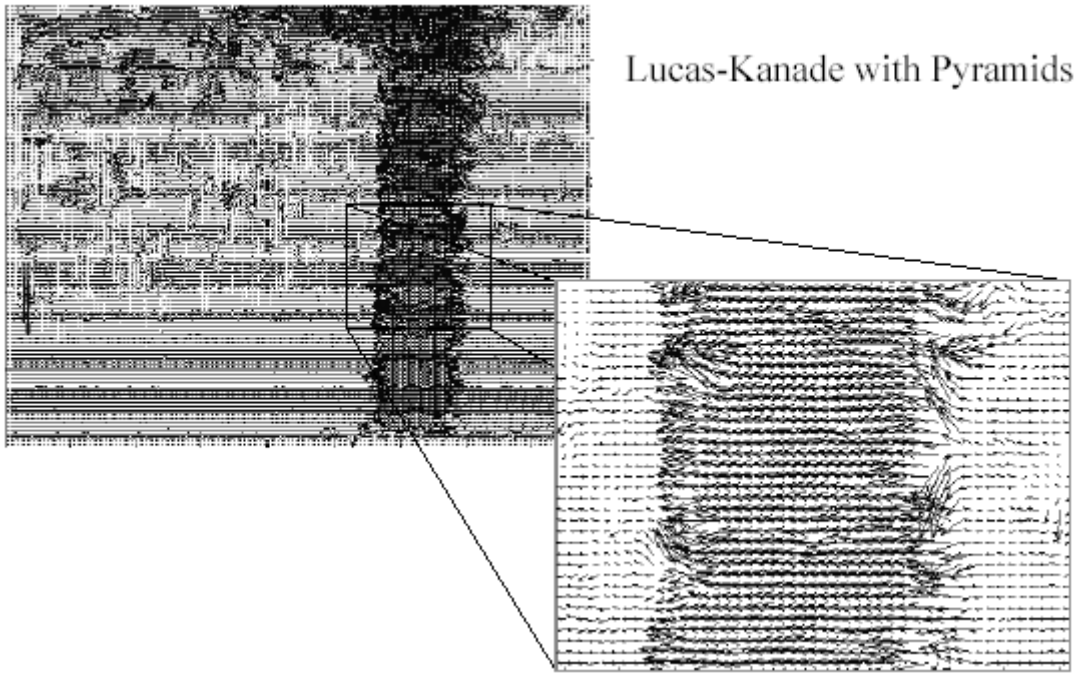
Optical Flow Results



Lucas-Kanade
without pyramids

Fails in areas of large
motion

Optical Flow Results



Horn & Schunck Algorithm

- The Horn–Schunck method of estimating optical flow is a global method which introduces a global constraint of smoothness to solve the aperture problem.

besides OF constraint equation term

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

Additional smoothness constraint : minimize $e_s + \lambda e_c$

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

KAIST

Horn & Schunck Algorithm

- The Horn–Schunck method of estimating optical flow is a global method which introduces a global constraint of smoothness to solve the aperture problem.

- The Horn-Schunck algorithm assumes smoothness in the flow over the whole image. Thus, it tries to minimize distortions in flow and prefers solutions which show more smoothness.

- The flow is formulated as a global energy functional which is then sought to be minimized.

Horn & Schunck Algorithm

- A global energy functional is given for two-dimensional image streams as:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x\mathrm{d}y$$

  - where I_x, I_y and I_t are the derivatives of the image intensity values along the x, y and time dimensions respectively, $\vec{V} = [u(x,y), v(x,y)]^\top$ is the optical flow vector, and the parameter α is a regularization constant. Larger values of α lead to a smoother flow.

OF constraint equation term

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

Additional smoothness constraint :

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

minimize $e_c + \alpha^2 e_s$

Horn & Schunck Algorithm

- A global energy functional is given for two-dimensional image streams as:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x\mathrm{d}y$$

  - This functional can be minimized by solving the associated multi-dimensional Euler-Lagrange equations.

The Euler-Lagrange equations :

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0 \quad \frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0$$

where L is the integrand of the energy expression, giving

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0 \quad I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

where subscripts again denote partial differentiation and

$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplace operator

Horn & Schunck Algorithm

- A global energy functional is given for two-dimensional image streams as:

$$E = \iint \left[(I_x u + I_y v + I_t)^2 + \alpha^2 (\|\nabla u\|^2 + \|\nabla v\|^2) \right] \mathrm{d}x\mathrm{d}y$$

  - This functional can be minimized by solving the associated multi-dimensional Euler-Lagrange equations.

The Euler-Lagrange equations :

$$\frac{\partial L}{\partial u} - \frac{\partial}{\partial x}\frac{\partial L}{\partial u_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial u_y} = 0 \quad \frac{\partial L}{\partial v} - \frac{\partial}{\partial x}\frac{\partial L}{\partial v_x} - \frac{\partial}{\partial y}\frac{\partial L}{\partial v_y} = 0$$

where L is the integrand of the energy expression, giving

$$I_x(I_x u + I_y v + I_t) - \alpha^2 \Delta u = 0 \quad I_y(I_x u + I_y v + I_t) - \alpha^2 \Delta v = 0$$

where subscripts again denote partial differentiation and

$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ denotes the Laplace operator

Horn & Schunck Algorithm

In practice the Laplacian is approximated numerically using finite differences, and may be written $\Delta u(x,y) = \overline{u}(x,y) - u(x,y)$ where $\overline{u}(x,y)$ is a weighted average of u calculated in a neighborhood around the pixel at location (x,y). Using this notation the above equation system may be written

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

which is linear in u and v and may be solved for each pixel in the image. However, since the solution depends on the neighboring values of the flow field, it must be repeated once the neighbors have been updated. The following iterative scheme is derived:

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

where the superscript *k+1* denotes the next iteration, which is to be calculated and *k* is the last calculated result. This is in essence the Jacobi method applied to the large, sparse system arising when solving for all pixels simultaneously.

Horn & Schunck Algorithm

- Remarks :
  - More than two frames allow a better estimation of I_t
  - Information spreads from corner-type patterns
  - Errors at boundaries (due to the smoothness constraint)
  - Example of regularization
    - Selection principle for the solution of ill-posed problems

DL-based Optical Flow Estimation

- FlowNet: Learning Optical Flow with Convolutional Networks, ICCV 2015

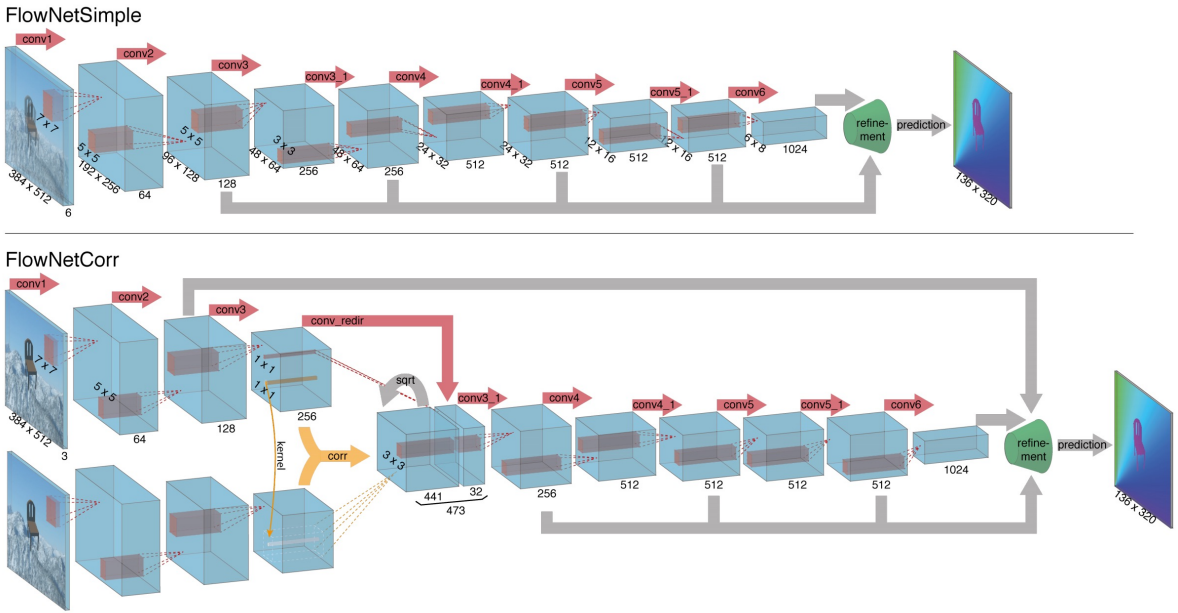


Figure 2. The two network architectures: FlowNetSimple (top) and FlowNetCorr (bottom).

Figure 7. Examples of optical flow prediction on the Sintel dataset. In each row left to right: overlaid image pair, ground truth flow and 3 predictions: EpicFlow, FlowNetS and FlowNetC. Endpoint error is shown for every frame. Note that even though the EPE of FlowNets is usually worse than that of EpicFlow, the networks often better preserve fine details.

DL-based Optical Flow Estimation

- FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, CVPR 2017

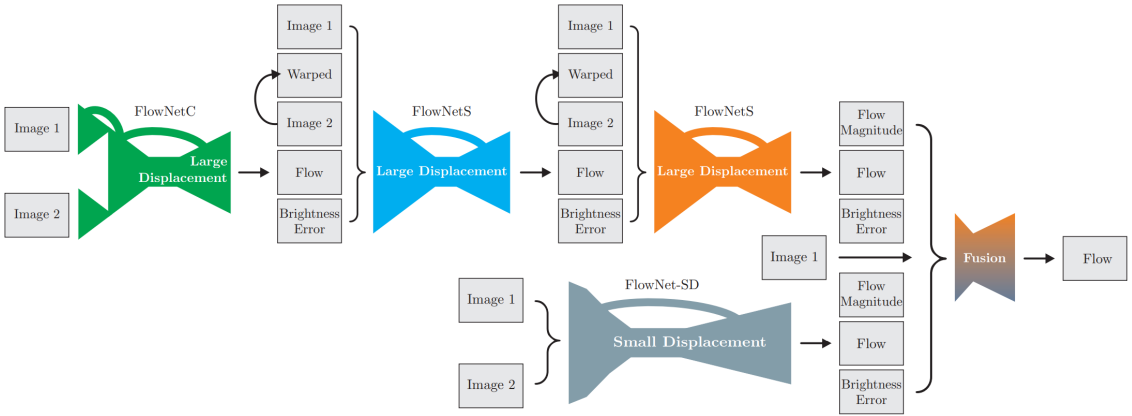


Figure 2. Schematic view of complete architecture: To compute large displacement optical flow we combine multiple FlowNets. Braces indicate concatenation of inputs. *Brightness Error* is the difference between the first image and the second image warped with the previously estimated flow. To optimally deal with small displacements, we introduce smaller strides in the beginning and convolutions between upconvolutions into the FlowNetS architecture. Finally we apply a small fusion network to provide the final estimate.

Figure 1. We present an extension of FlowNet. FlowNet 2.0 yields smooth flow fields, preserves fine motion details and runs at 8 to 140fps. The accuracy on this example is four times higher than with the original FlowNet.

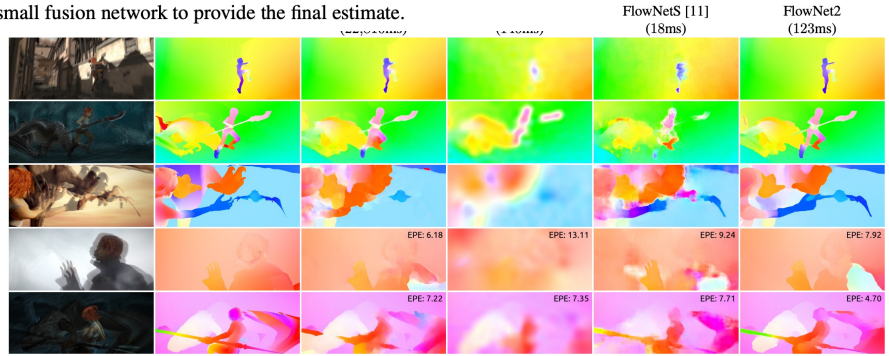
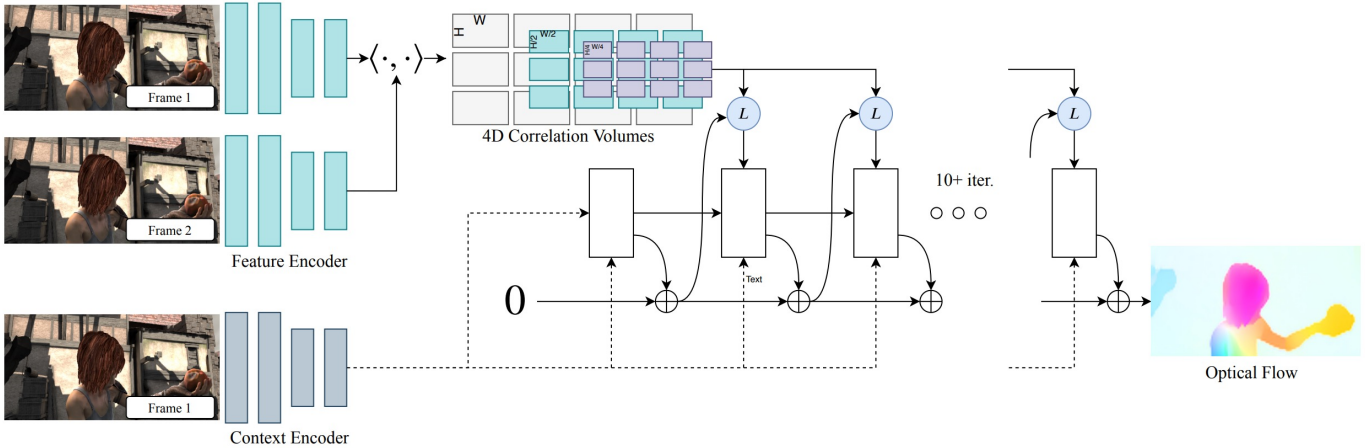
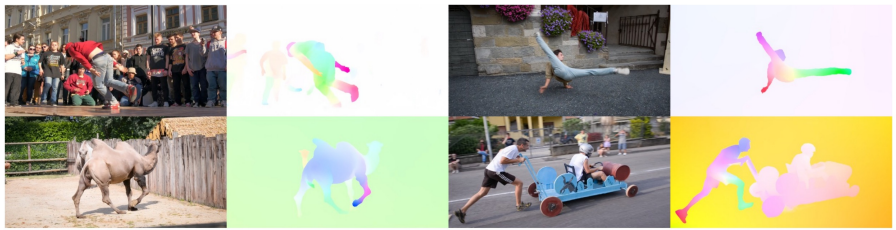


Figure 6. Examples of flow fields from different methods estimated on Sintel. FlowNet2 performs similar to FlowFields and is able to extract fine details, while methods running at comparable speeds perform much worse (PCA-Flow and FlowNetS).

DL-based Optical Flow Estimation

- RAFT: Recurrent All Pairs Field Transforms for Optical Flow, ECCV 2020

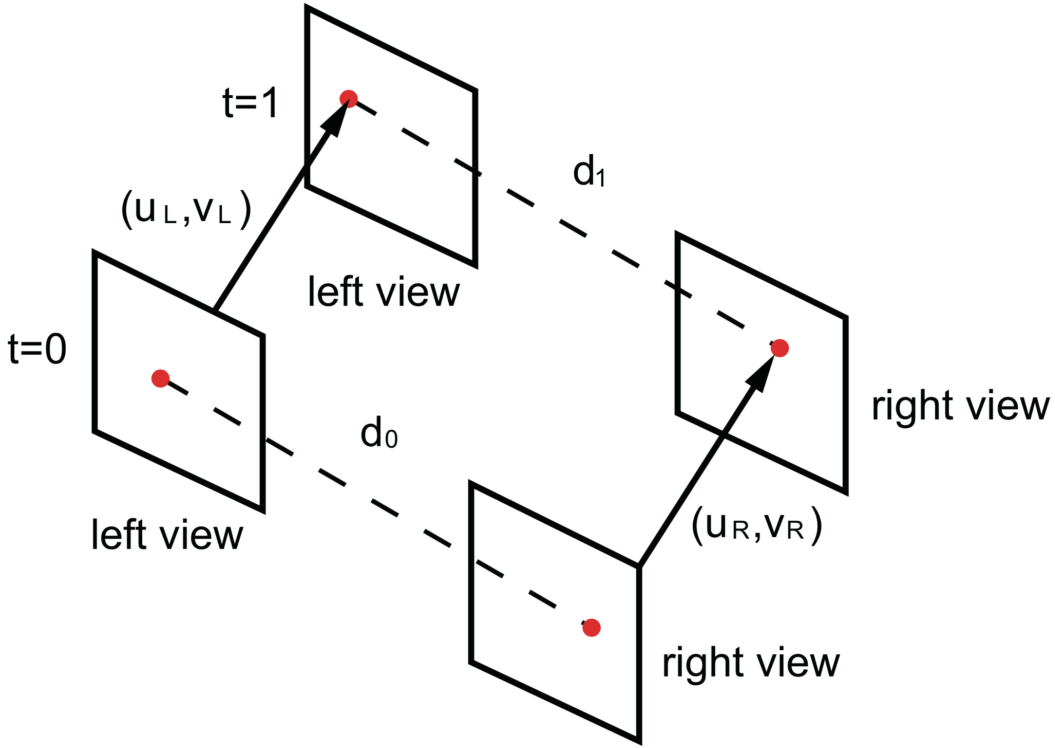




RAFT consists of 3 main components: (1) A feature encoder that extracts per-pixel features from both input images, along with a context encoder that extracts features from only I1. (2) A correlation layer which constructs a 4D W × H × W × H correlation volume by taking the inner product of all pairs of feature vectors. The last 2-dimensions of the 4D volume are pooled at multiple scales to construct a set of multi-scale volumes. (3) An update operator which recurrently updates optical flow by using the current estimate to look up values from the set of correlation volumes.

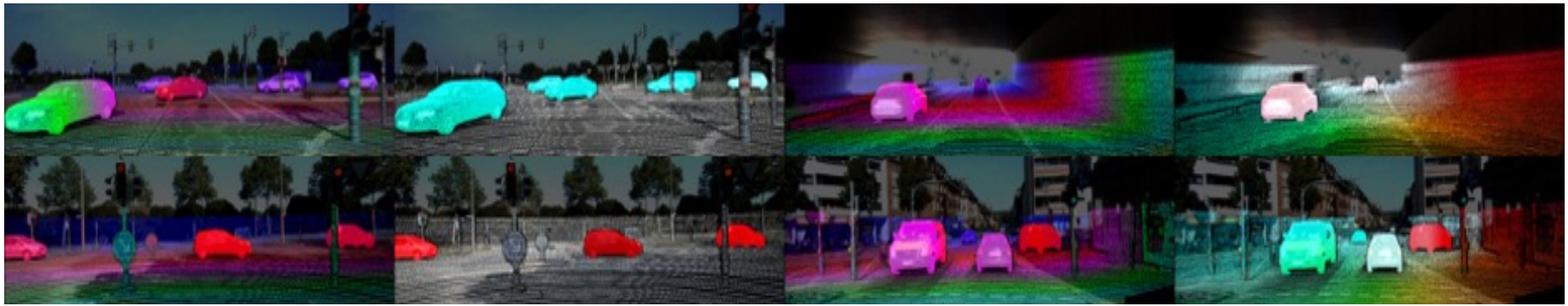
Scene Flow

- (Optical flow + stereo matching) defines the scene flow, motion in 3D space.



Scene Flow

- (Opical flow + stereo matching) defines the scene flow, motion in 3D space.



DL-based Scene Flow

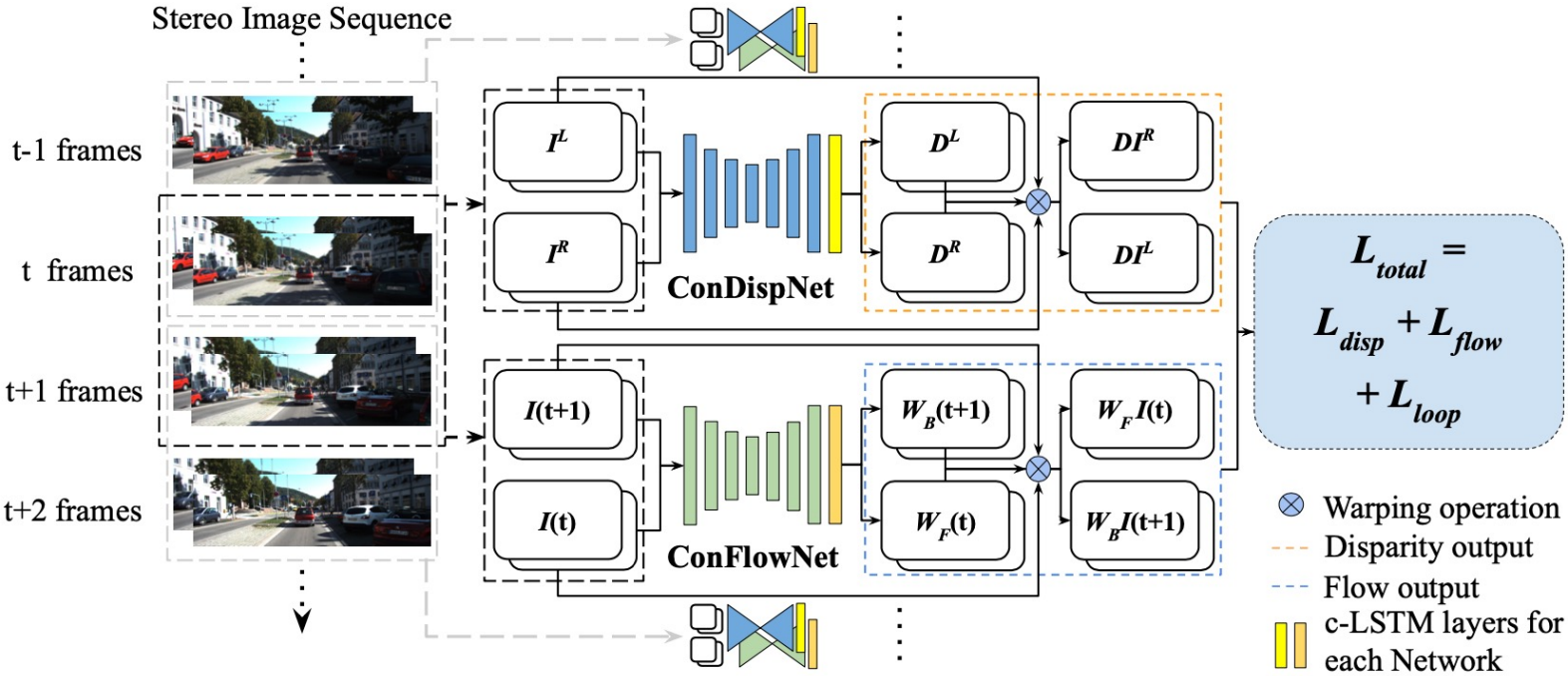


Figure 2: Overview of our framework. Our network consists of two subnetworks, ConDispNet and ConFlowNet, that estimate the disparity and the optical flow, respectively. Using the estimated dense correspondence map from each subnetwork, the reference image is warped to the target image. DI^s and $WI(t)$ denote the warped image corresponding to the target image I^s and $I(t)$, respectively.

Loop-Net: Joint Unsupervised Disparity and Optical Flow Estimation of Stereo Videos with Spatiotemporal Loop Consistency (IROS 2020)

DL-based Scene Flow

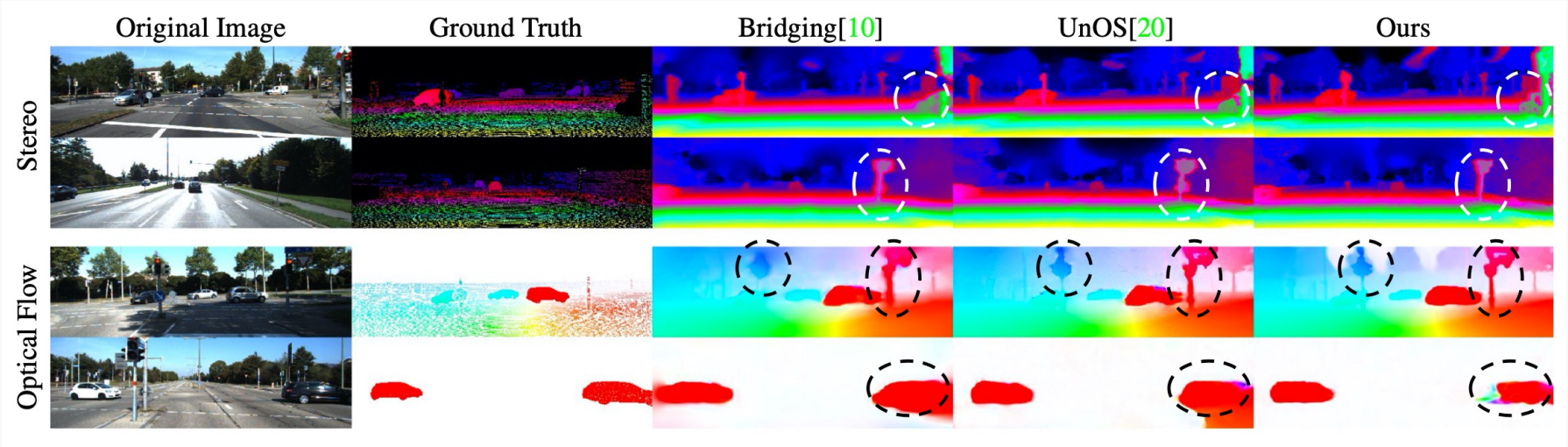


Table 4: Quantitative evaluation of the optical flow task on KITTI 2015 training/test dataset. "noc", "occ" denote non-occlusion and occlusion regions, respectively. The boldface and underscore denote the best and second best performances, respectively.

Method	Joint learning	Super vised	Train EPE	Train Fl	Train EPE noc	Test Fl
Flownet2 [6]		✓	10.06	30.37%	-	-
SpyNet [13]		✓	20.56	44.78%	-	-
UnFlow-C [12]			8.80	28.94%	-	29.46%
UnFlow-CSS [12]			8.10	23.27%	-	-
GeoNet [25]	✓		10.81	-	8.05	-
Wang *et al.* [21]			8.88	-	-	-
Janai *et al.* [7]			6.59	-	3.22	22.94%
DFnet [34]	✓		8.98	26.01%	-	25.70%
CC [14]	✓		6.21	26.41%	-	-
CC-uft [14]	✓		5.66	20.93%	-	25.27%
Bridging-R [10]	✓		7.02	27.34%	4.26	-
Bridging-P [10]	✓		6.66	21.50%	-	-
UnOS [20]	✓		5.58	-	-	**18.00%**
Ours						
(Flow only - SSIM)			6.15	19.91%	2.94	-
(Flow only - ZNCC)			6.08	18.63%	2.78	-
(Full - w/o c-LSTM)			5.77	18.09%	2.88	-
(Full-1)	✓		5.53	18.00%	2.63	-
(Full-2)	✓		**5.27**	**17.57%**	**2.53**	18.42%

Table 5: Quantitative evaluation of spatiotemporal consistency on KITTI 2015 dataset. ξ is a set of LC(loop consistency). μ, med denote mean and median, respectively. Note that high accuracy does not guarantee consistent results.

Method	Higher the better			Lower the better	
	$\xi < 2$	$\xi < 3$	$\xi < 4$	$\mu(\xi)$	$med(\xi)$
Bridging-R [10]	0.5114	0.5608	0.5964	4.9295	1.3659
Bridging-P [10]	0.5868	0.6213	0.6475	4.4213	0.9663
UnOS [20]	0.3412	0.3822	0.4184	16.6032	6.2568
Ours(Full-2)	**0.7581**	**0.7822**	**0.8003**	**3.4095**	**0.7855**

Loop-Net: Joint Unsupervised Disparity and Optical Flow Estimation of Stereo Videos with Spatiotemporal Loop Consistency (IROS 2020)