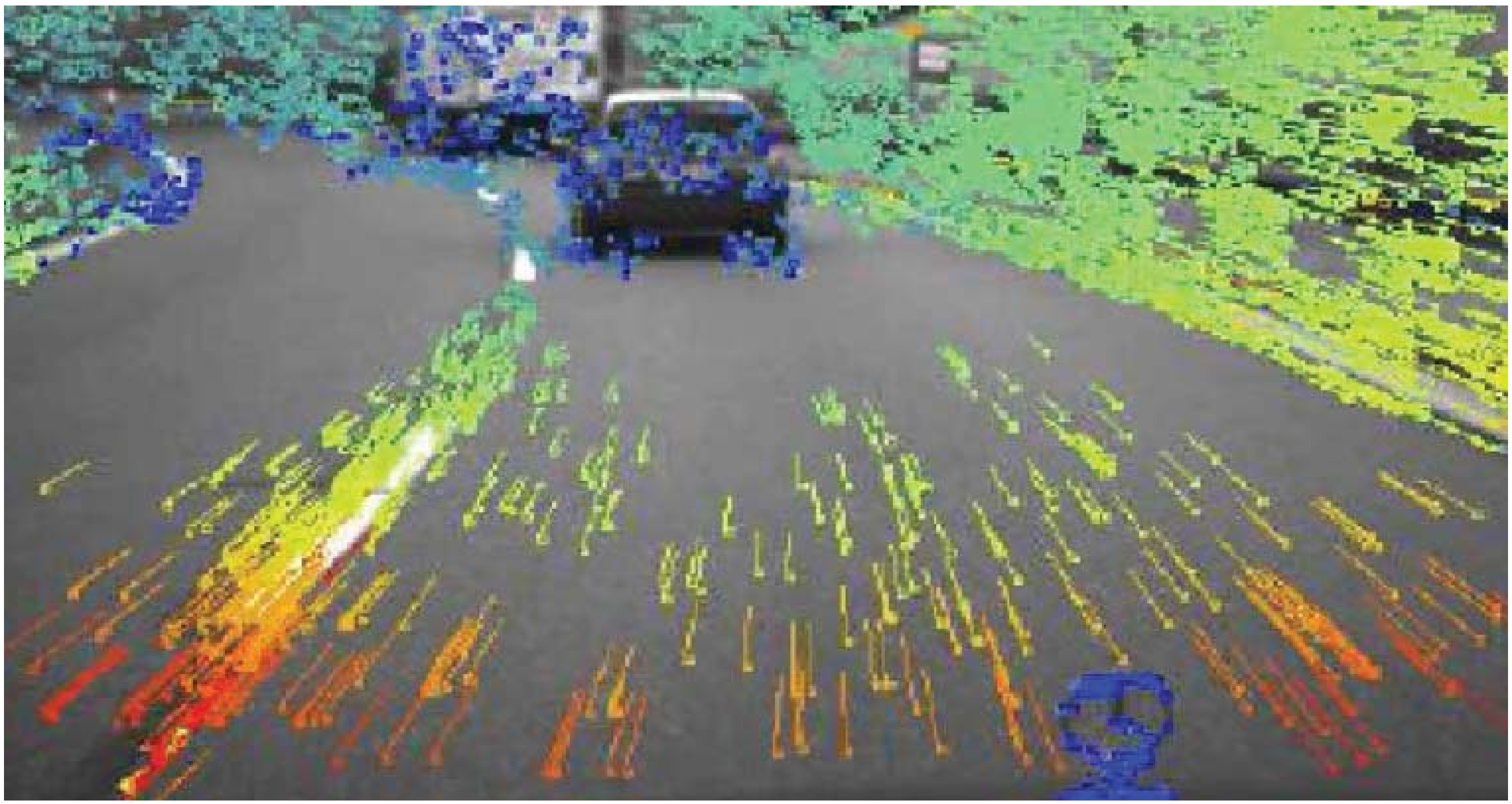
Optical flow



Slide credits: Lihi Zelnik, Tali Tribitz, Kris Kitani, Ioannis Gkioulekas

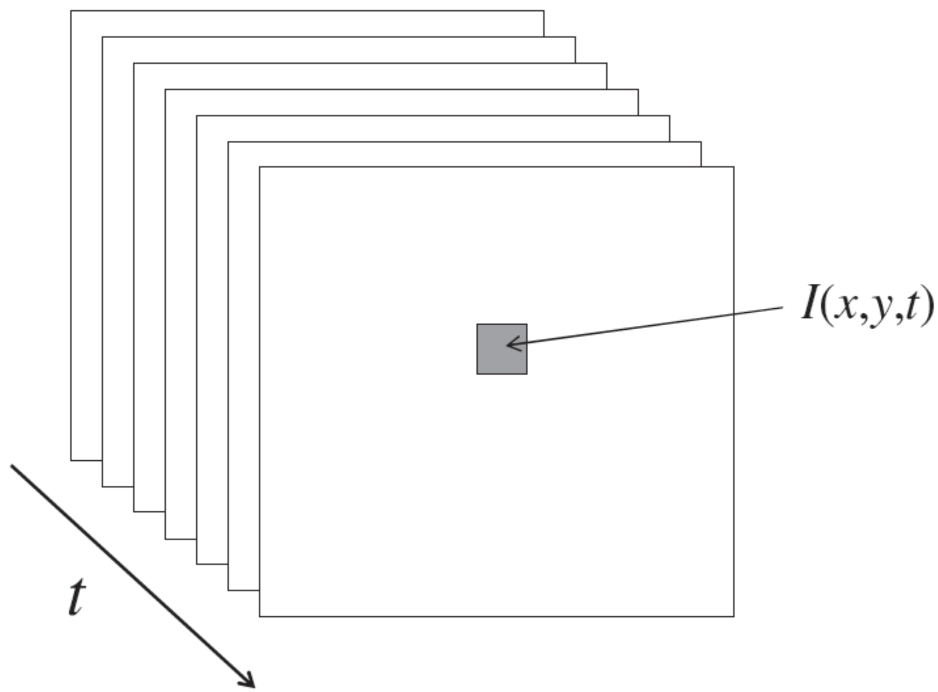
Today

From images to video

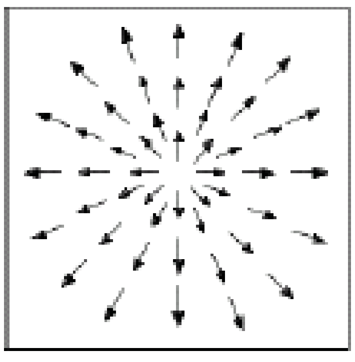
- Feature tracking
- Optical flow
- Motion segmentation
- Applications

From images to video

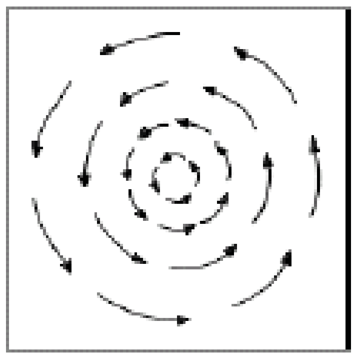
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x,y) and time (t)



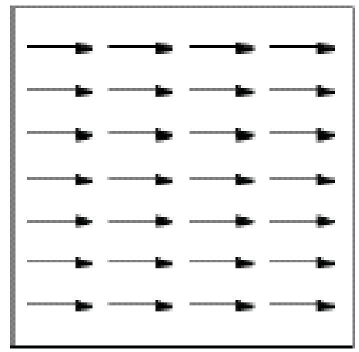
Examples of Motion fields



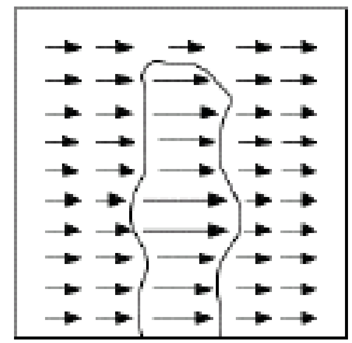
Forward motion



Rotation



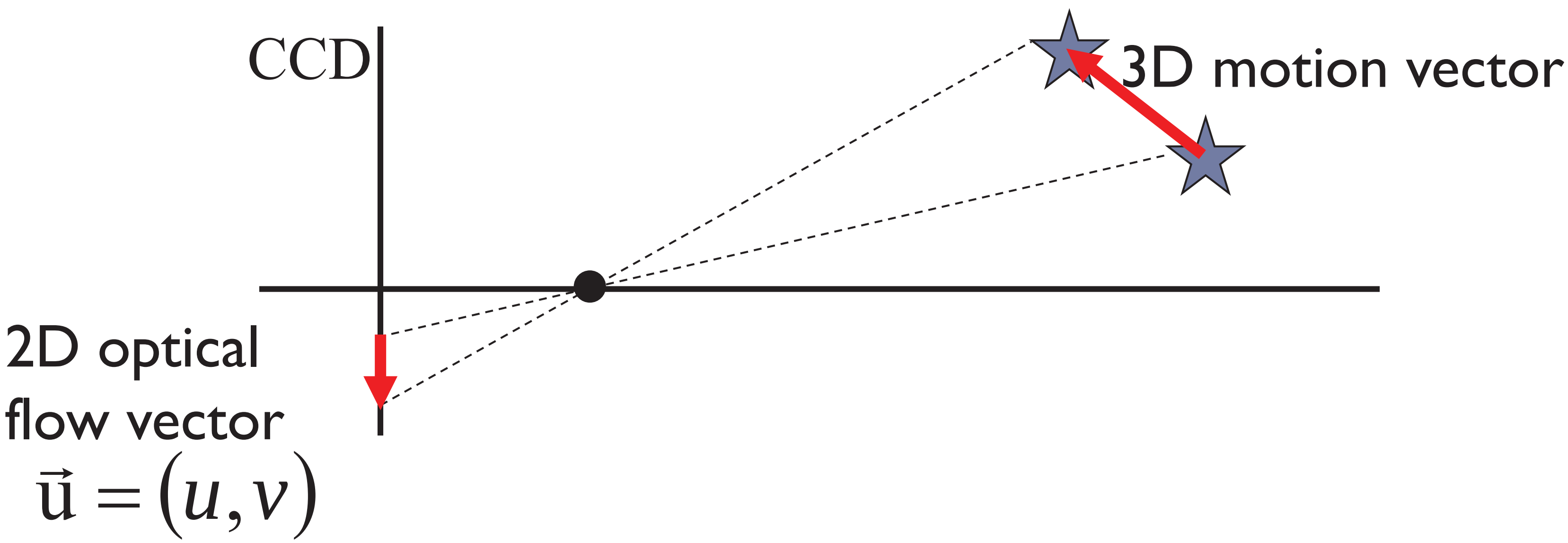
Horizontal translation



Closer objects appear to move faster!!

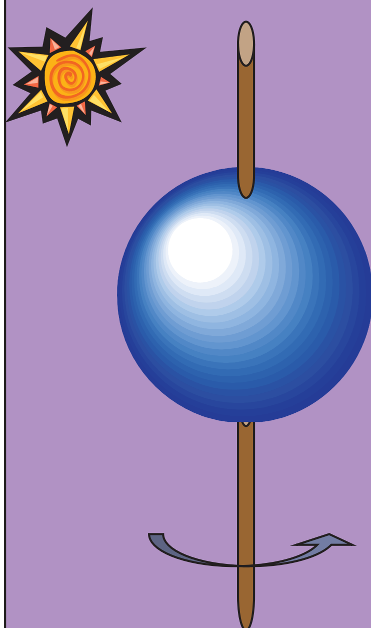
Motion Field & Optical Flow Field

- Underlying assumption:
  The apparent motion field is a projection of the real 3D motion onto the 2d image

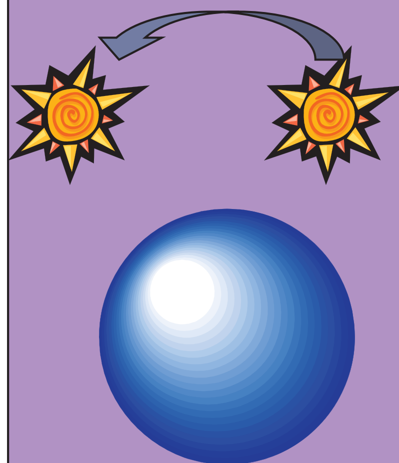


When does it break?

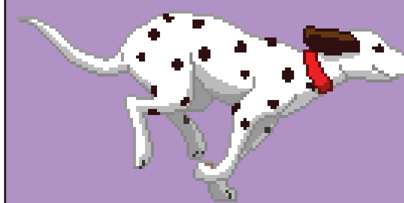
The screen is stationary yet displays motion



Homogeneous objects generate zero optical flow.



Fixed sphere. Changing light source.



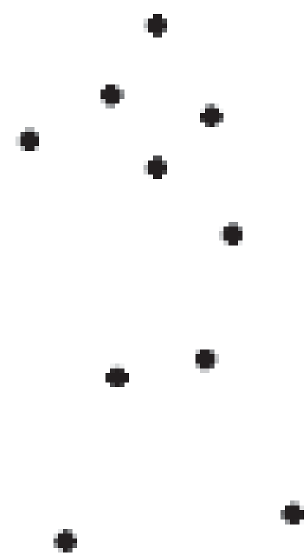
Non-rigid texture motion

Feature tracking vs. optical flow

- Feature tracking
  - Extract visual features and "track" them over multiple frames
- Optical flow
  - Compute image motion at each and every pixel

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.*

Tracking example

Tracking example

GPU4Vision https://www.youtube.com/watch?v=t12lpPCodME&spfreload=1

Optical flow example

- Compute motion for all pixels

http://www.borisfx.com/

http://www.robesafe.com/personal/pablo.alcantarilla/code.html

Today

From images to video

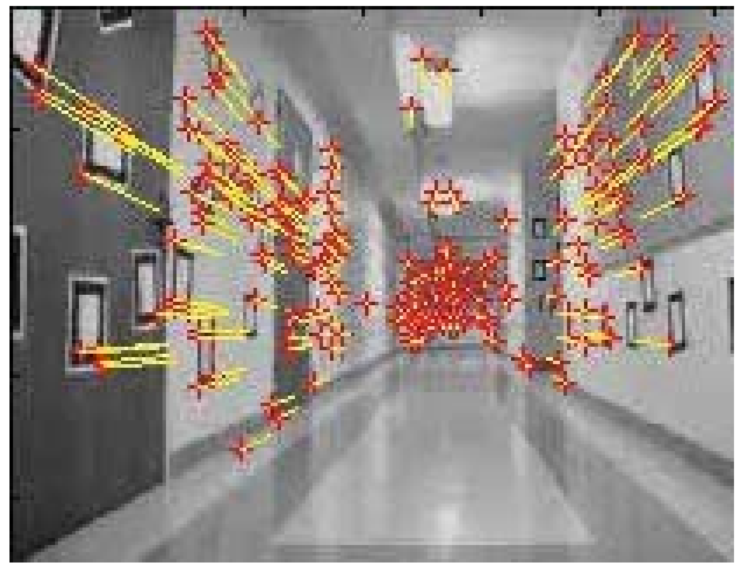
- Feature tracking
- Optical flow
- Motion segmentation
- Applications

Tracking challenges

- Find good features to track
  - Harris, SIFT, etc
- Large motions
- Changes in shape, orientation, color
  - Allow some matching flexibility
- Occlusions, dis-occlusions
  - Need to add/delete features
- Drift (errors accumulate over time)
  - Need to know when to terminate a track

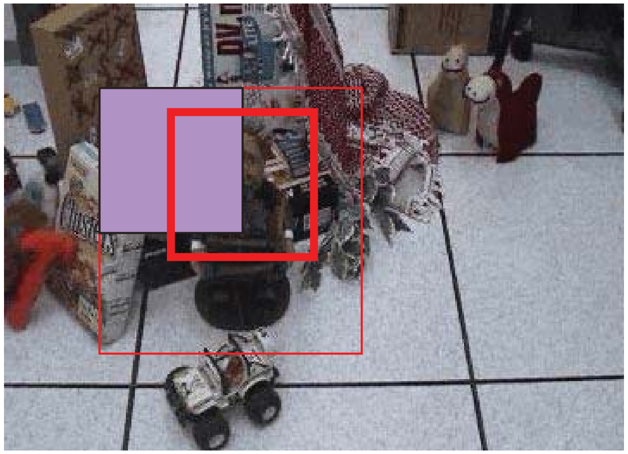
Feature tracking

- Track only "good" features



Tracking by template matching

- The simplest way to track is by template matching
  - Define a small area around a pixel as the template
  - Match the template against each pixel within a search area in next image.
  - Use a match measure such as correlation, normalized correlation, or sum-of-squares difference
  - Choose the maximum (or minimum) as the match



Limitations of template matching

- Slow (need to check more locations)
- Does not give sub-pixel alignment (or becomes much slower)
  - Even pixel alignment may not be good enough to prevent drift
- May be useful as a step in tracking if there are large movements

Today

From images to video

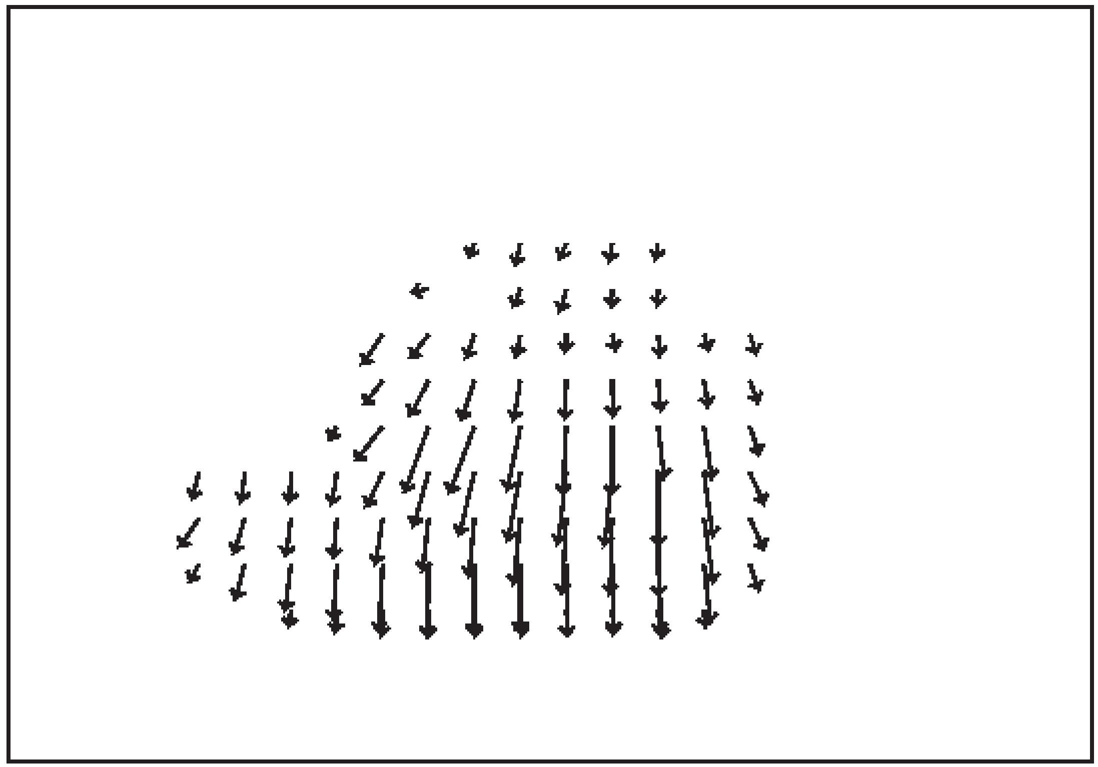
- Feature tracking
- Optical flow
- Motion segmentation
- Applications

Feature tracking vs. optical flow

- Feature tracking
  - Extract visual features and "track" them over multiple frames
- Optical flow
  - Compute image motion at each and every pixel

Optical flow - definition

- The optical flow is the **apparent** motion of brightness patterns in the image



Pierre Kornprobst's Demo

Optical Flow

Problem Definition

Given two consecutive image frames,
estimate the motion of each pixel

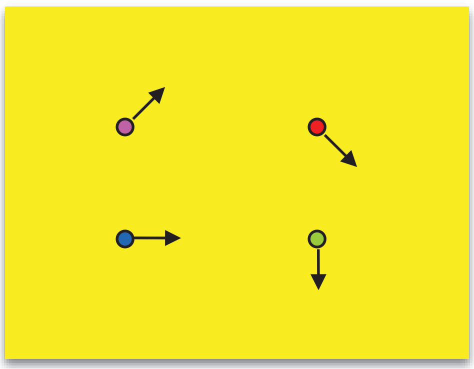
Assumptions

Brightness constancy

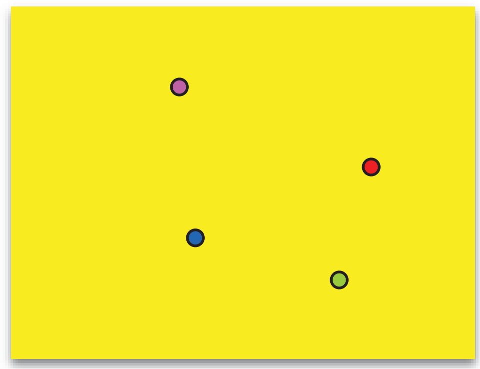
Small motion

Optical Flow

(Problem definition)



$I(x, y, t)$



$I(x, y, t')$

Estimate the motion (flow) between these two consecutive images

How is this different from estimating a 2D transform?

Key Assumptions

(unique to optical flow)

Color Constancy

(Brightness constancy for intensity images)

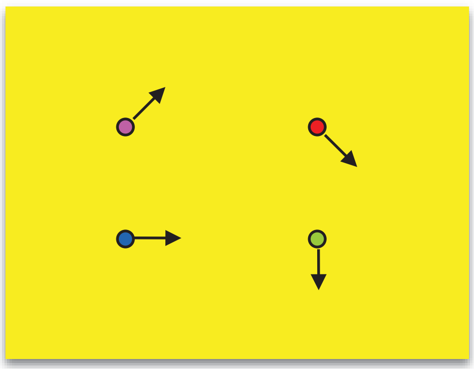
Implication: allows for pixel to pixel comparison
(not image features)

Small Motion

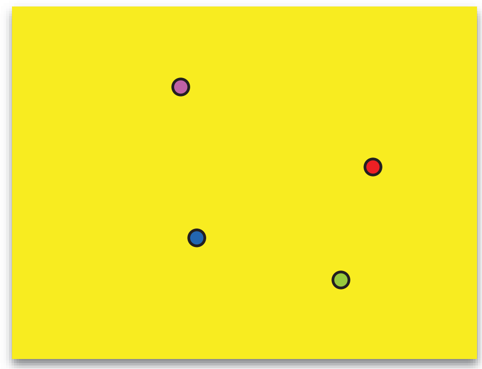
(pixels only move a little bit)

Implication: linearization of the brightness
constancy constraint

Approach



$I(x, y, t)$



$I(x, y, t')$

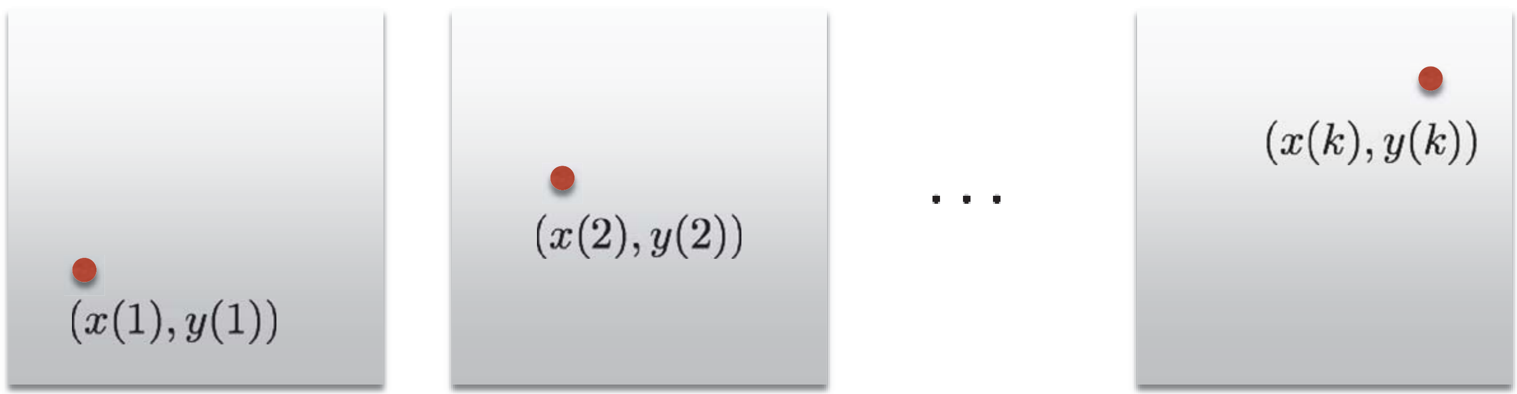
Look for nearby pixels with the same color

(small motion) (color constancy)

Assumption 1

Brightness constancy

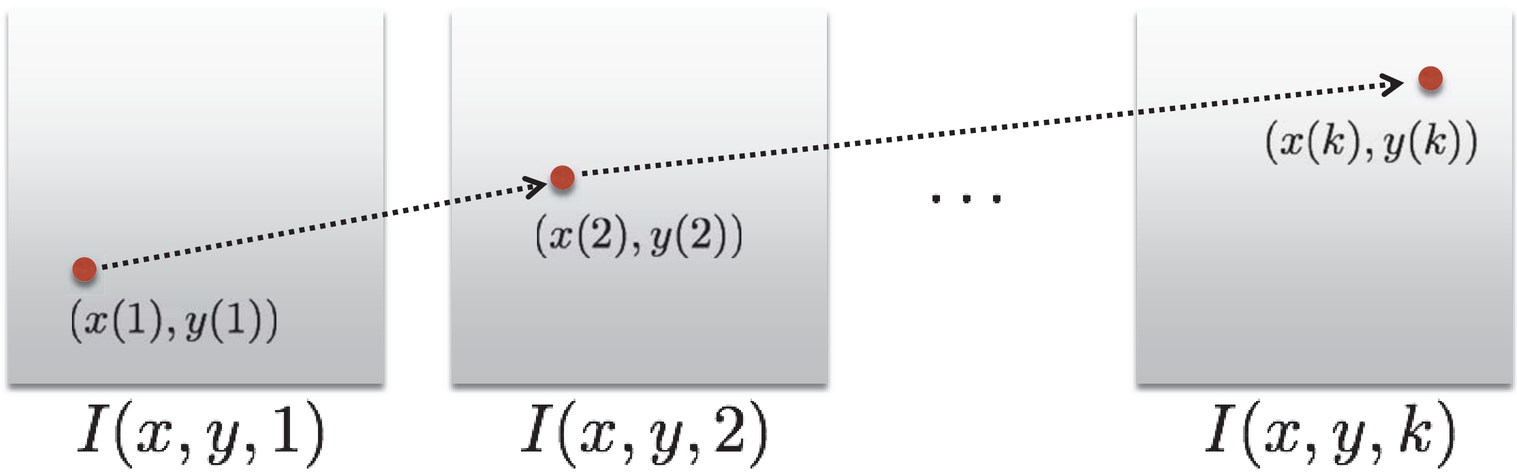
Scene point moving through image sequence



Assumption 1

Brightness constancy

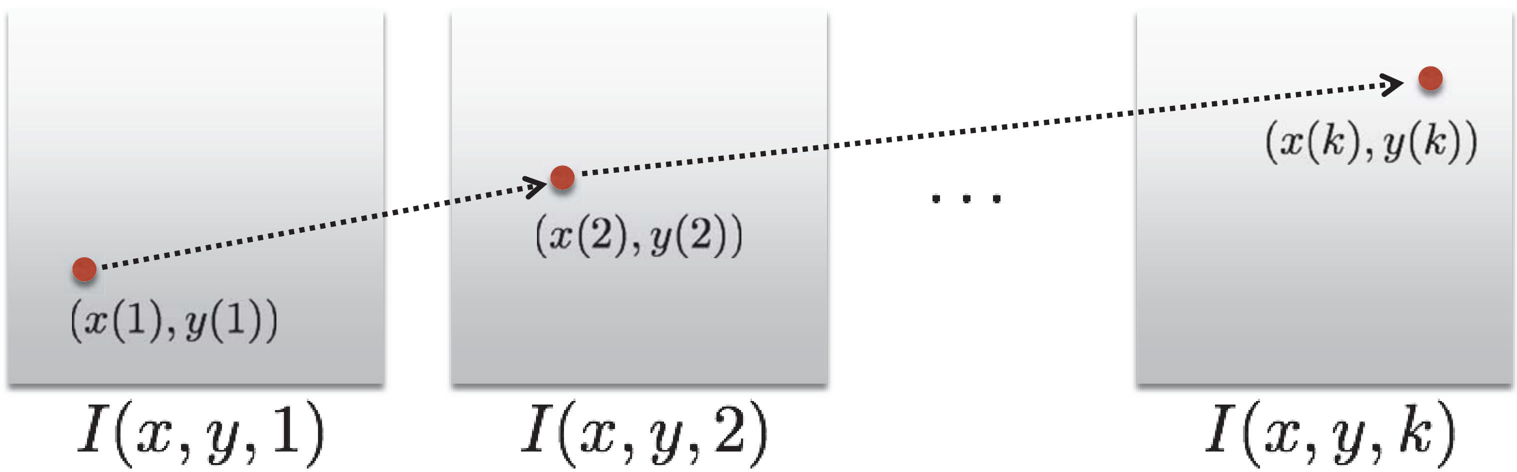
Scene point moving through image sequence



Assumption 1

Brightness constancy

Scene point moving through image sequence

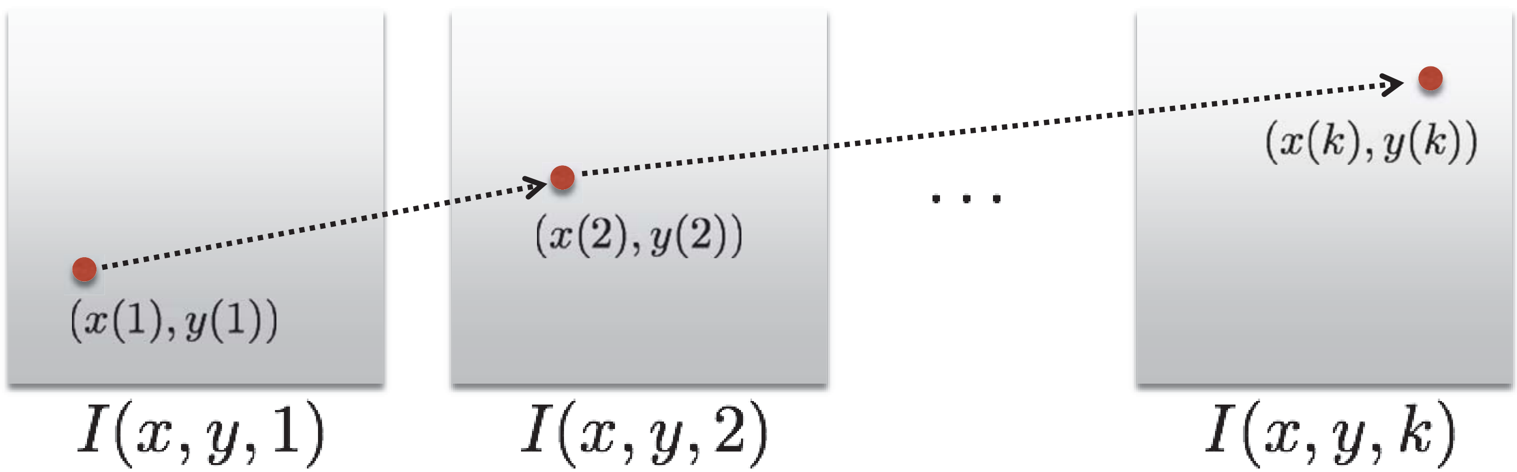


Assumption:Brightness of the point will remain the same

Assumption 1

Brightness constancy

Scene point moving through image sequence



Assumption:Brightness of the point will remain the same

$$I(x(t), y(t), t) = \underbrace{C}_{\text{constant}}$$

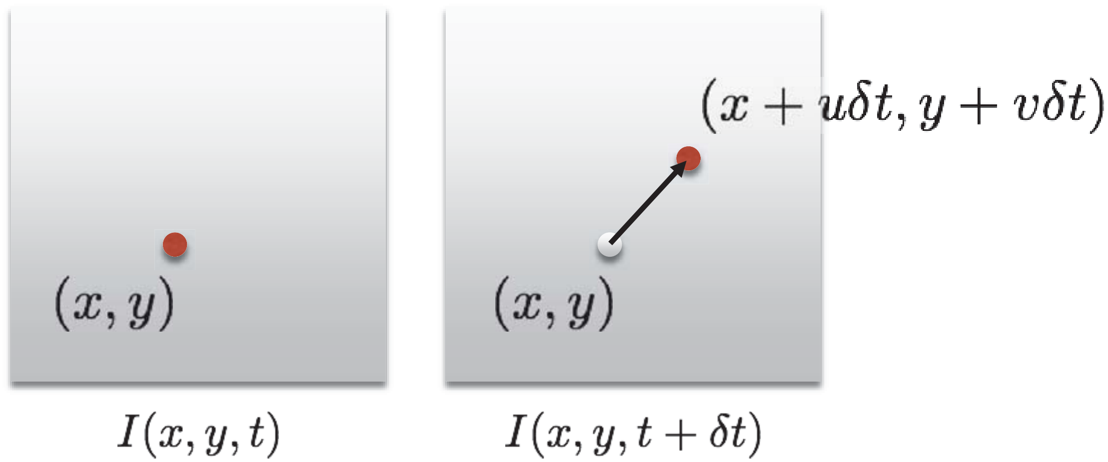
Assumption 2

Small motion

$I(x, y, t)$ $\quad\quad$ $I(x, y, t + \delta t)$

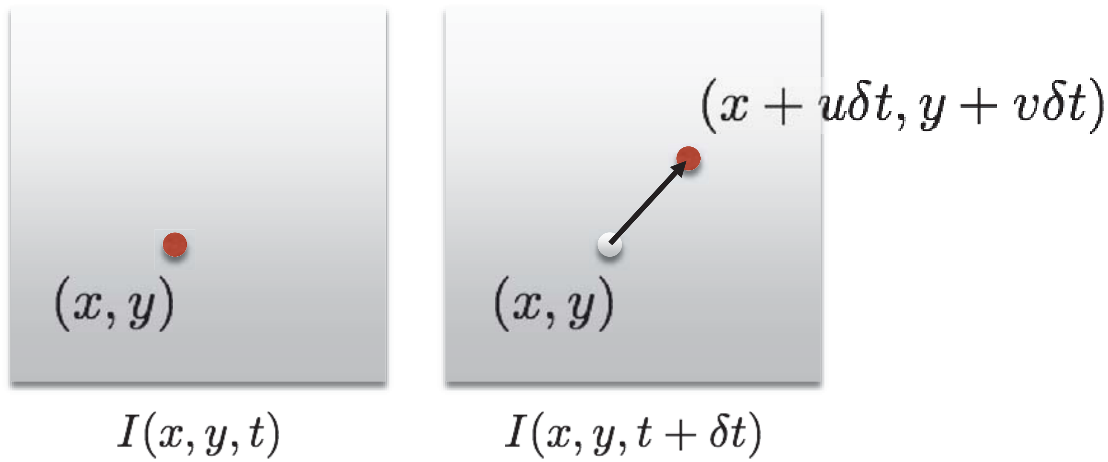
Assumption 2

Small motion



Assumption 2

Small motion

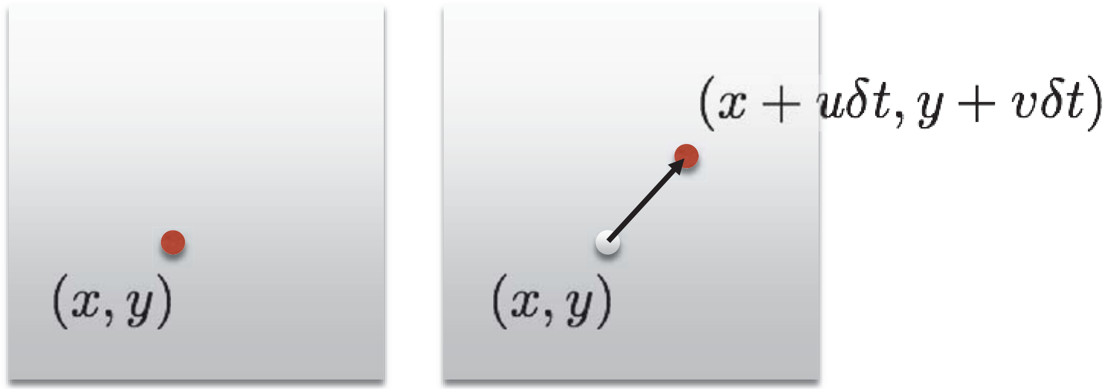


Optical flow (velocities): (u, v)

Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

Assumption 2

Small motion



$I(x, y, t)$ $\qquad$ $I(x, y, t + \delta t)$

Optical flow (velocities): (u, v) $\qquad$ Displacement: $(\delta x, \delta y) = (u\delta t, v\delta t)$

For a ***really small space-time step***…

$$I(x + u\delta t, y + v\delta t, t + \delta t) = I(x, y, t)$$

… the brightness between two consecutive image frames is the same

Small motion assumption

- The brightness constancy equation

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

Small motion assumption

- The brightness constancy equation

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

- Assumption 2
  Motion is small
  First order Taylor expansion

$$I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t$$

$$0 = \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t$$

The motion equation

- Simplify notations: $I_x \delta x + I_y \delta y + I_t \delta t = 0$
- Divide by δt and denote $u = \dfrac{\delta x}{\delta t}$ $\quad v = \dfrac{\delta y}{\delta t}$

(Limit for small δt)

- Final equation is: $\boxed{I_x u + I_y v = -I_t}$

brightness constancy equation

$$I_x u + I_y v + I_t = 0$$ **Brightness Constancy Equation**

(x-flow) (y-flow)

$$\nabla I^\top \boldsymbol{v} + I_t = 0$$ vector form

(1 x 2) (2 x 1)

(putting the math aside for a second…)

What do the term of the
brightness constancy equation represent?

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

(putting the math aside for a second…)

What do the term of the
brightness constancy equation represent?

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

Image gradients
(at a point p)

(putting the math aside for a second…)

What do the term of the
brightness constancy equation represent?

flow velocities

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

Image gradients
(at a point p)

(putting the math aside for a second…)

What do the term of the brightness constancy equation represent?

flow velocities

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

Image gradients
(at a point p)

temporal gradient

How do you compute these terms?

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

How do you compute …

$$\boxed{I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}}$$

spatial derivative

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

How do you compute …

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Scharr filter
…

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

How do you compute …

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

Forward difference
Sobel filter
Scharr filter
…

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

How do you compute …

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

Forward difference
Sobel filter
Scharr filter
…

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

frame differencing

Frame differencing

t

1	1	1	1	1
1	1	1	1	1
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10
1	10	10	10	10

-

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	10	10	10
1	1	10	10	10
1	1	10	10	10

=

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	9	9	9	9
0	9	0	0	0
0	9	0	0	0
0	9	0	0	0

(example of a forward difference)

Example:

3 x 3 patch

t

1	1	1	1	1
1	1	1	1	1
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**
1	**10**	**10**	**10**	**10**

$t+1$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	**10**	**10**	**10**
1	1	**10**	**10**	**10**
1	1	**10**	**10**	**10**

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

-	0	0	0	-
-	0	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-
-	**9**	0	0	-

-1 0 1

$$I_y = \frac{\partial I}{\partial y}$$

-	-	-	-	-
0	0	0	0	0
0	**9**	**9**	**9**	**9**
-	0	0	0	0
0	0	0	0	0
-	-	-	-	-

-1
0
1

$$I_t = \frac{\partial I}{\partial t}$$

0	0	0	0	0
0	0	0	0	0
0	**9**	**9**	**9**	**9**
0	**9**	0	0	0
0	**9**	0	0	0
0	**9**	0	0	0

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

How do you compute …

$I_x = \frac{\partial I}{\partial x}$ $I_y = \frac{\partial I}{\partial y}$ **spatial derivative**	$u = \frac{dx}{dt}$ $v = \frac{dy}{dt}$ **optical flow**	$I_t = \frac{\partial I}{\partial t}$ **temporal derivative**
Forward difference Sobel filter Scharr filter …	How do you compute this?	frame differencing

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

How do you compute …

spatial derivative	optical flow	temporal derivative
$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$	$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$	$I_t = \frac{\partial I}{\partial t}$
Forward difference Sobel filter Scharr filter …	**We need to solve for this!** (this is the unknown in the optical flow problem)	frame differencing

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

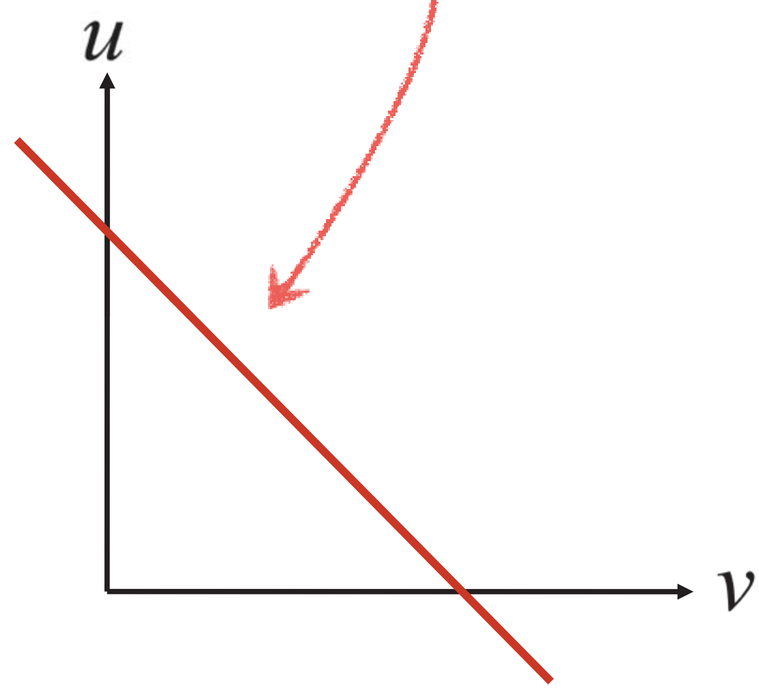
How do you compute …

$I_x = \frac{\partial I}{\partial x}$ $I_y = \frac{\partial I}{\partial y}$ **spatial derivative**	$u = \frac{dx}{dt}$ $v = \frac{dy}{dt}$ **optical flow**	$I_t = \frac{\partial I}{\partial t}$ **temporal derivative**
Forward difference Sobel filter Scharr filter …	(u, v) Solution lies on a line Cannot be found uniquely with a single constraint	frame differencing

Solution lies on a straight line

$$\boxed{I_x u + I_y v + I_t = 0}$$

many combinations of u and v will satisfy the equality



The solution cannot be determined uniquely with a single constraint (a single pixel)

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

$$I_x = \frac{\partial I}{\partial x} \quad I_y = \frac{\partial I}{\partial y}$$

spatial derivative

$$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$

optical flow

$$I_t = \frac{\partial I}{\partial t}$$

temporal derivative

How can we use the brightness constancy equation to estimate the optical flow?

unknown

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

known

We need at least ____ equations to solve for 2 unknowns.

unknown

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

known

Where do we get more equations (constraints)?

Horn-Schunck Optical Flow (1981)	Lucas-Kanade Optical Flow (1981)
brightness constancy	method of differences
small motion	
'smooth' flow (flow can vary from pixel to pixel)	**'constant' flow** (flow is constant for all pixels)
global method (dense)	local method (sparse)

Constant flow

Where do we get more equations (constraints)?

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

Assume that the surrounding patch (say 5x5) has
‘**constant flow**’

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us equations

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$
\begin{gathered}
I_x(\boldsymbol{p}_1)u + I_y(\boldsymbol{p}_1)v = -I_t(\boldsymbol{p}_1) \\
I_x(\boldsymbol{p}_2)u + I_y(\boldsymbol{p}_2)v = -I_t(\boldsymbol{p}_2) \\
\vdots \\
I_x(\boldsymbol{p}_{25})u + I_y(\boldsymbol{p}_{25})v = -I_t(\boldsymbol{p}_{25})
\end{gathered}
$$

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}$$

Matrix form

Assumptions:

Flow is locally smooth

Neighboring pixels have same displacement

Using a 5 x 5 image patch, gives us 25 equations

$$\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}$$

$$\underset{25 \times 2}{A} \qquad \underset{2 \times 1}{x} \qquad \underset{25 \times 1}{b}$$

How many equations? How many unknowns? How do we solve this?

Least squares approximation

$$\hat{x} = \arg\min_{x} ||Ax - b||^2$$ is equivalent to solving $$A^\top A\hat{x} = A^\top b$$

Least squares approximation

$$\hat{x} = \arg\min_x ||Ax - b||^2 \text{ is equivalent to solving } A^\top A\hat{x} = A^\top b$$

To obtain the least squares solution solve:

$$\underset{A^\top A}{\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix}} \underset{\hat{x}}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{A^\top b}{\begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix}}$$

where the summation is over each pixel $\boldsymbol{p}$ in patch $\boldsymbol{P}$

$$x = (A^\top A)^{-1} A^\top b$$

Least squares approximation

$$\hat{x} = \arg\min_x ||Ax - b||^2$$ is equivalent to solving $$A^\top A \hat{x} = A^\top b$$

To obtain the least squares solution solve:

$$\begin{array}{ccc} A^\top A & \hat{x} & A^\top b \\ \begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix} & \begin{bmatrix} u \\ v \end{bmatrix} = - & \begin{bmatrix} \sum_{p \in P} I_x I_t \\ \sum_{p \in P} I_y I_t \end{bmatrix} \end{array}$$

where the summation is over each pixel $\boldsymbol{p}$ in patch $\boldsymbol{P}$

Sometimes called 'Lucas-Kanade Optical Flow'

When is this solvable?

$$A^\top A\hat{x} = A^\top b$$

$A^\mathsf{T}A$ should be invertible

$A^\mathsf{T}A$ should not be too small

λ_1 and λ_2 should not be too small

$A^\mathsf{T}A$ should be well conditioned

λ_1/λ_2 should not be too large (λ_1=larger eigenvalue)

Where have you seen this before?

$$A^\top A = \begin{bmatrix} \sum\limits_{p \in P} I_x I_x & \sum\limits_{p \in P} I_x I_y \\ \sum\limits_{p \in P} I_y I_x & \sum\limits_{p \in P} I_y I_y \end{bmatrix}$$

Where have you seen this before?

$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

Harris Corner Detector!

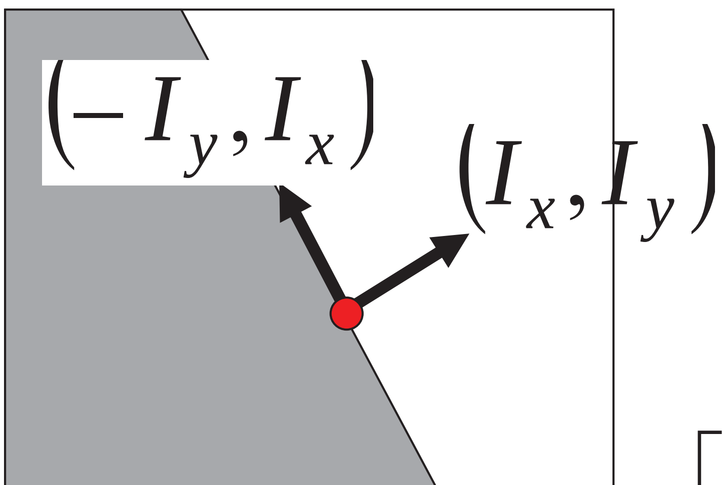
Implications

- Corners are when $\lambda 1$, $\lambda 2$ are big; this is also when Lucas-Kanade optical flow works best
- Corners are regions with two different directions of gradient (at least)
- Corners are good places to compute flow!

What happens when you have no 'corners'?

When is this solvable?

- Edge ➔ $A^T A$ becomes singular



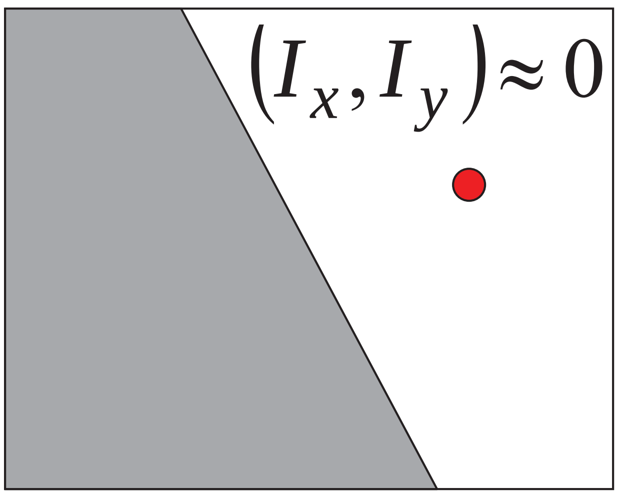
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} -I_y \\ I_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Downarrow$$

$\begin{bmatrix} -I_y \\ I_x \end{bmatrix}$ is eigenvector with eigenvalue 0

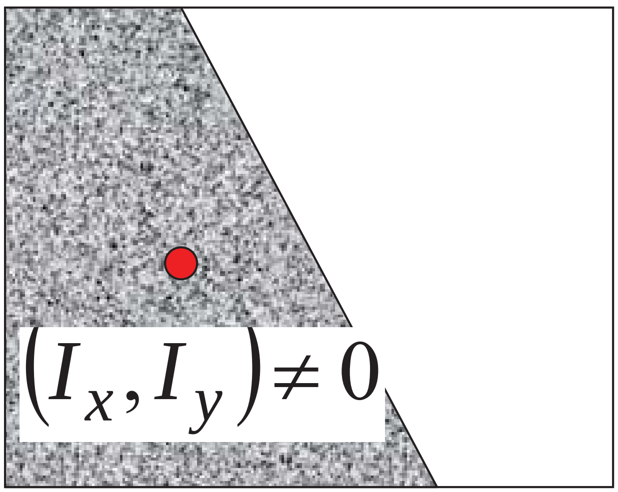
When is this solvable?

- Homogeneous ➔ $A^T A \approx 0$ ➔ eigenvalues are 0



When is this solvable?

- Textured regions ➔ two high eigenvalues



$$\left(I_x, I_y\right) \neq 0$$

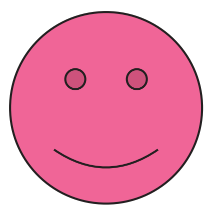
Which features can we track?

- Edge → $A^T A$ becomes singular

Homogeneous regions → low gradients •

$$A^T A \approx 0$$

High texture → •

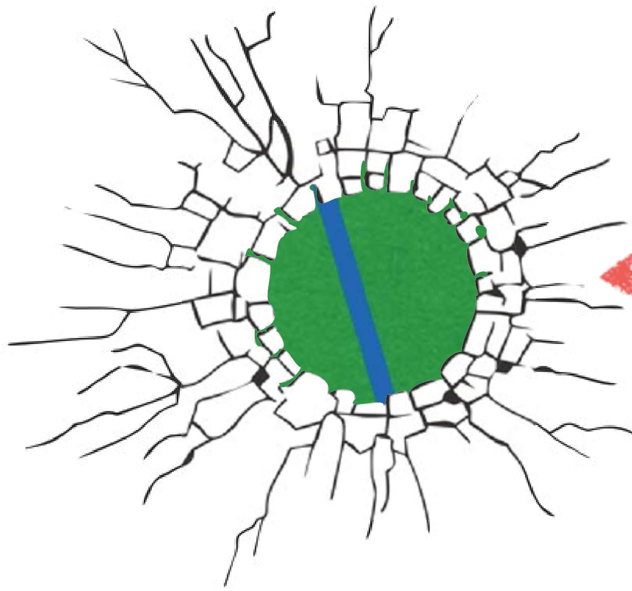


You want to compute optical flow.

What happens if the image patch contains only a line?

Barber's pole illusion

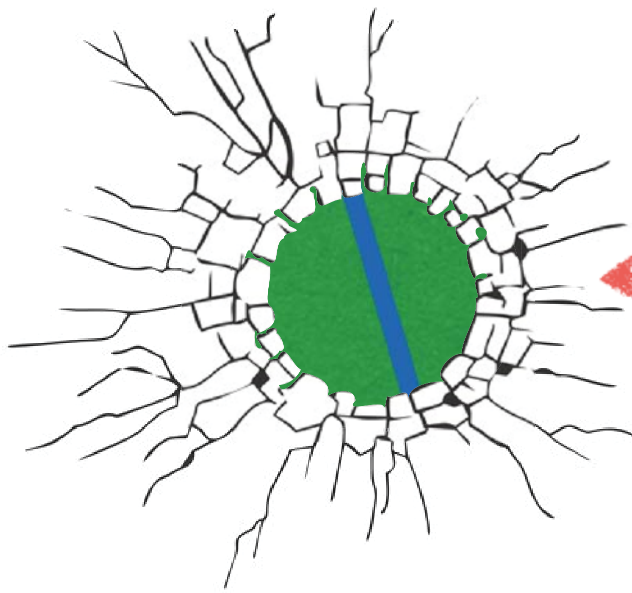
Aperture Problem



small visible image patch

In which direction is the line moving?

Aperture Problem

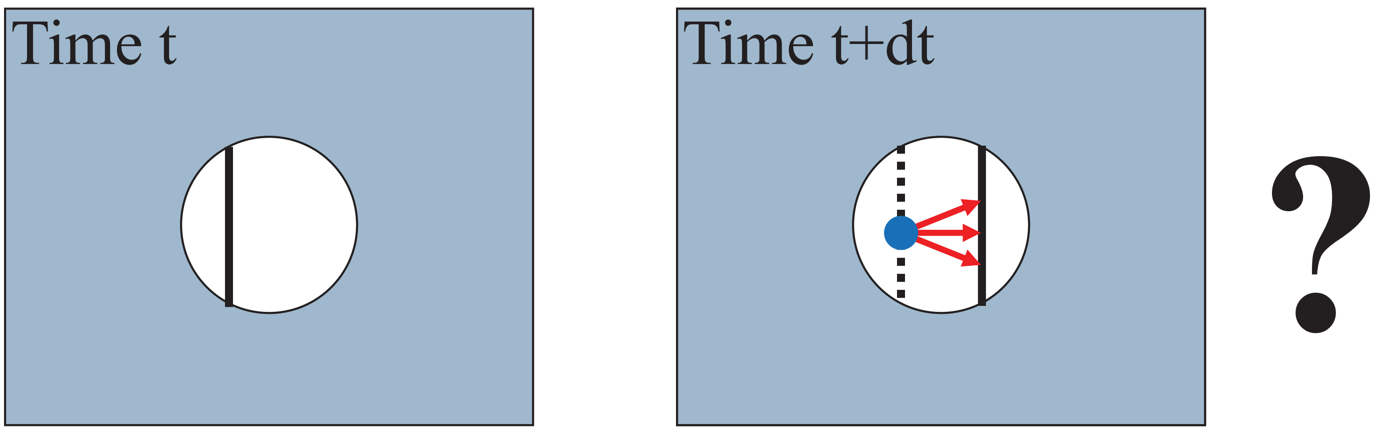


small visible image patch

In which direction is the line moving?

The aperture problem

- For points on a line of fixed intensity we can only recover the normal flow

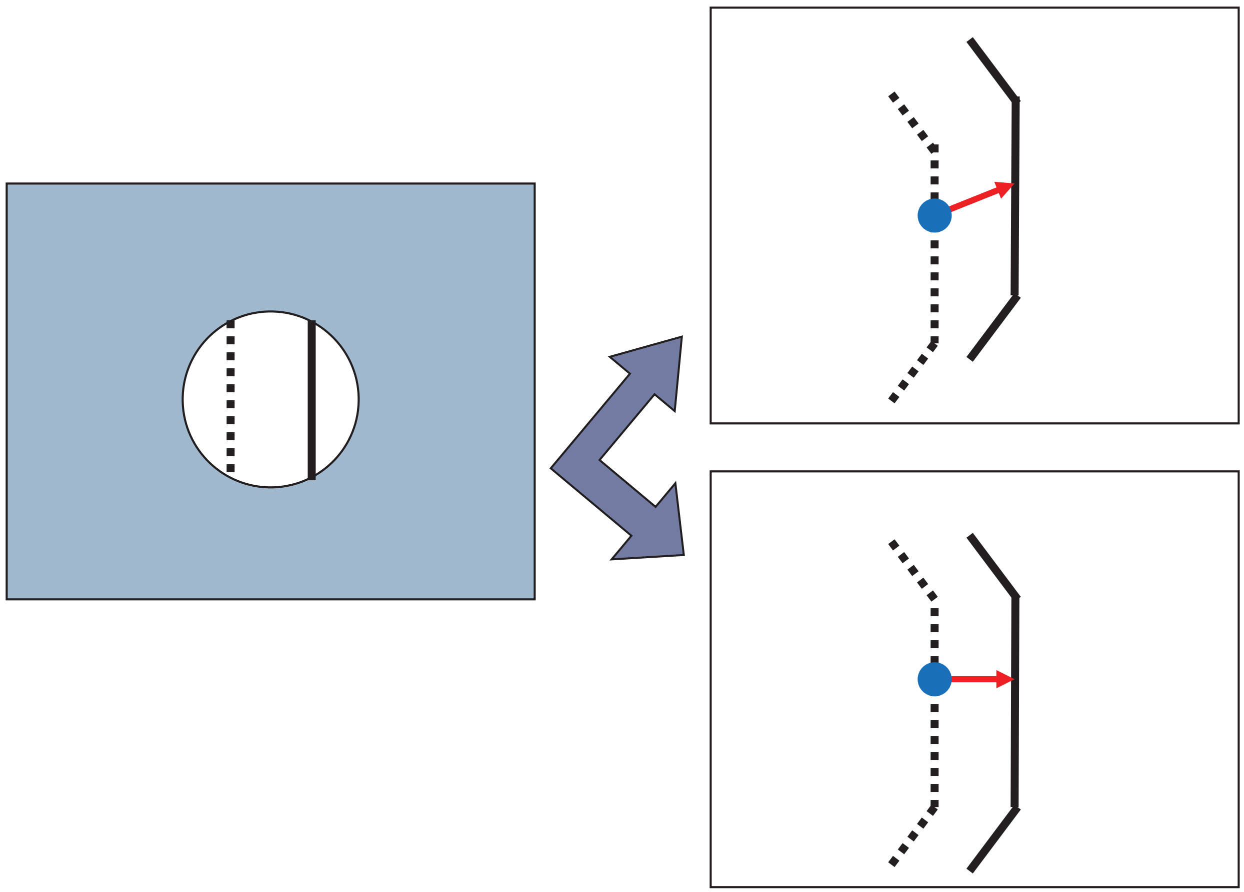


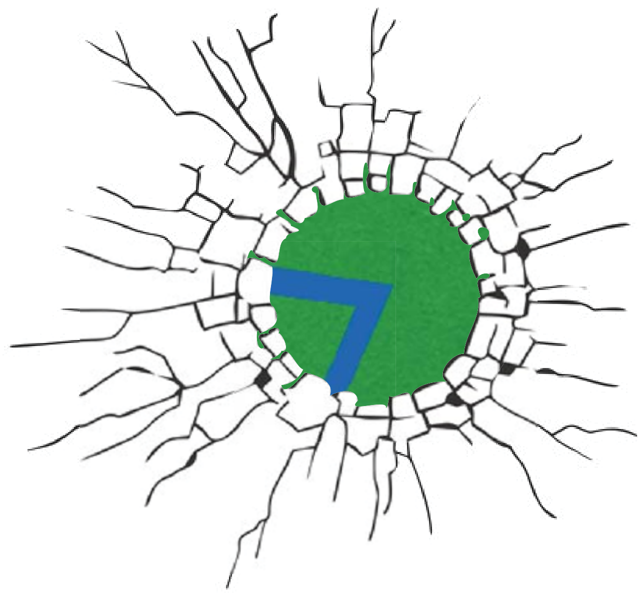
Where did the blue point move to?

We need additional constraints

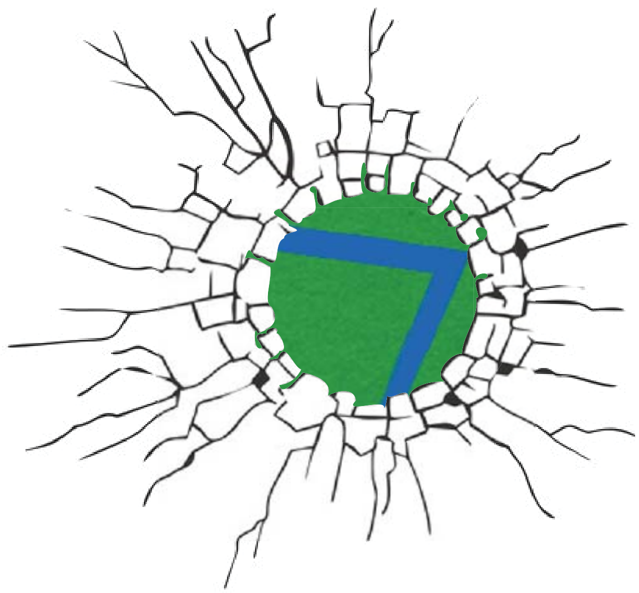
Solving the ambiguity

Sometimes enlarging the aperture can help





Want patches with different gradients to the avoid aperture problem

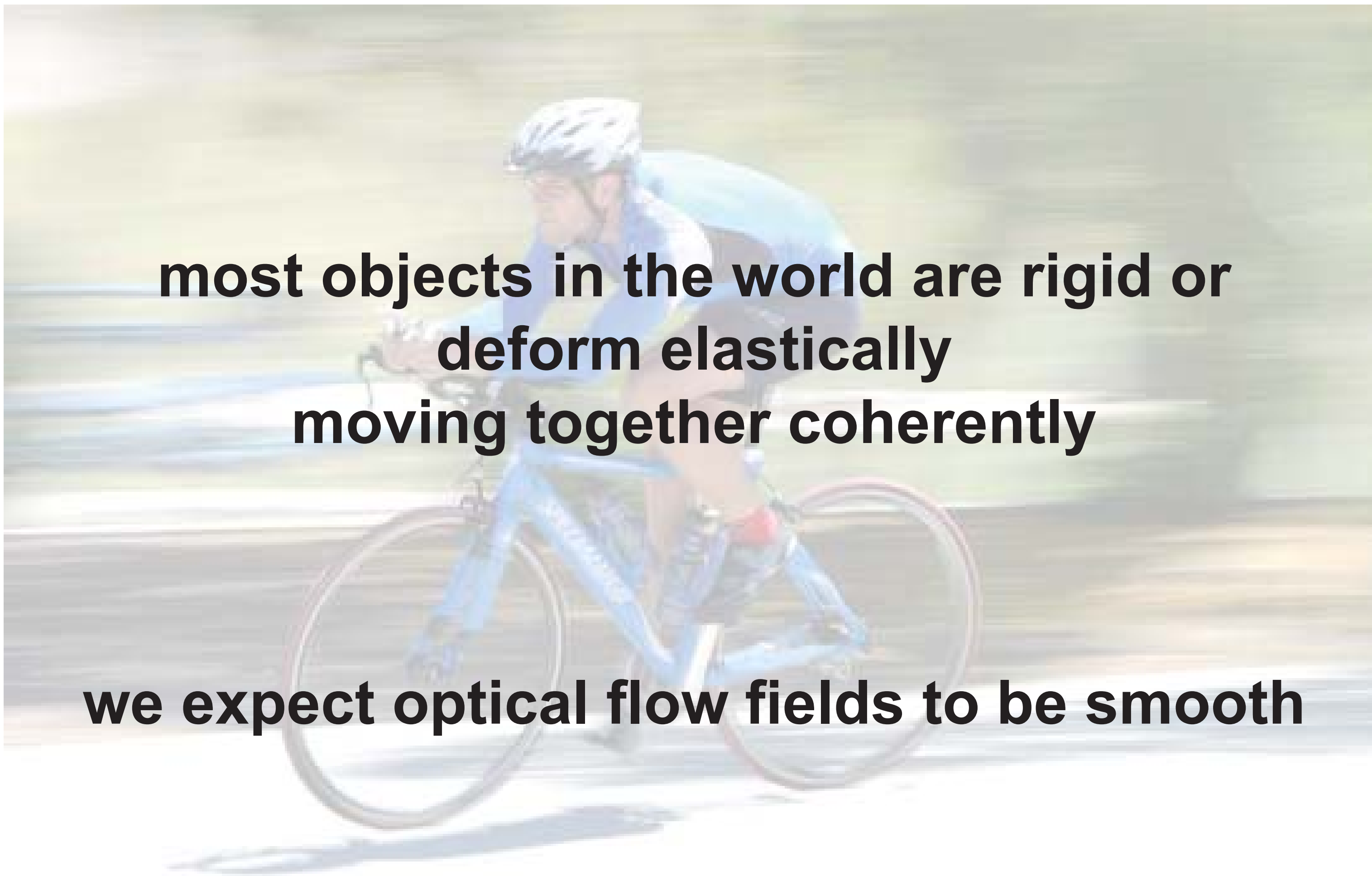


Want patches with different gradients to the avoid aperture problem

Horn-Schunck optical flow

Horn-Schunck Optical Flow (1981)	**Lucas-Kanade Optical Flow (1981)**
brightness constancy small motion	method of differences
'smooth' flow (flow can vary from pixel to pixel)	**'constant' flow** (flow is constant for all pixels)
global method (dense)	local method (sparse)

Smoothness



most objects in the world are rigid or deform elastically moving together coherently

we expect optical flow fields to be smooth

Key idea

(of Horn-Schunck optical flow)

Enforce

brightness constancy

Enforce

smooth flow field

to compute optical flow

Key idea

(of Horn-Schunck optical flow)

Enforce
brightness constancy

Enforce
smooth flow field

to compute optical flow

Enforce
brightness constancy

$$\boxed{I_x u + I_y v + I_t = 0}$$

For every pixel,

$$\min_{u,v} \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

Enforce

brightness constancy

$$\boxed{I_x u + I_y v + I_t = 0}$$

For every pixel,

$$\min_{u,v} \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

lazy notation for $I_x(i,j)$

Key idea

(of Horn-Schunck optical flow)

Enforce

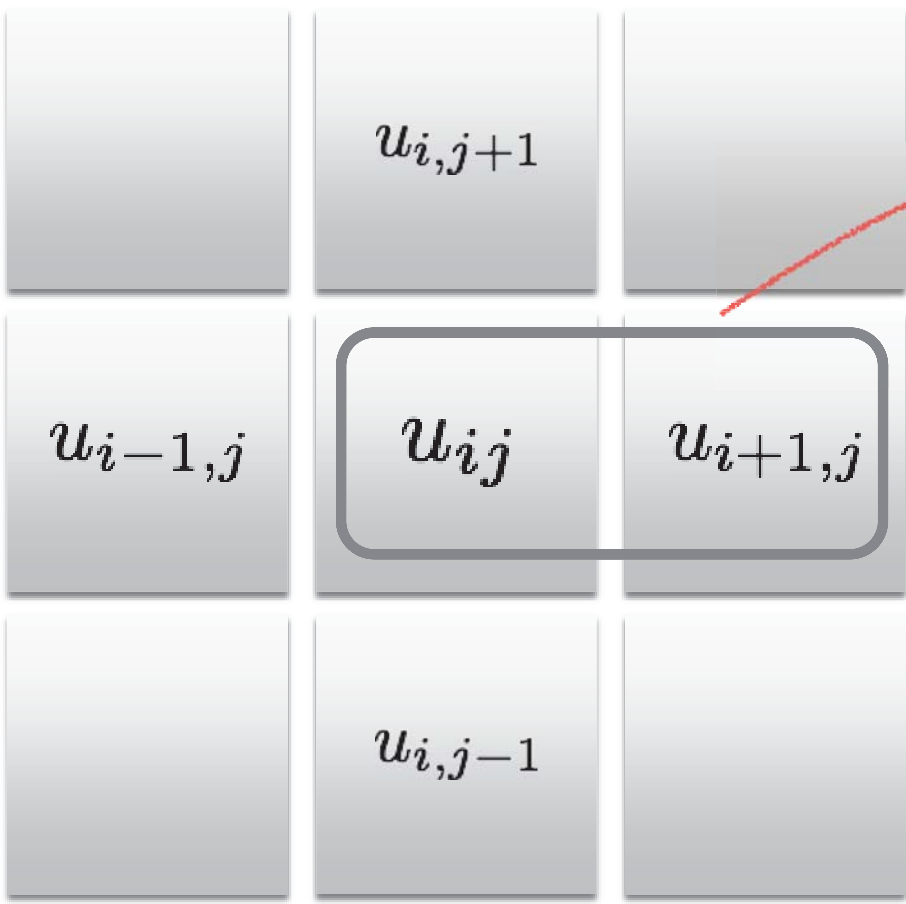
brightness constancy

Enforce

smooth flow field

to compute optical flow

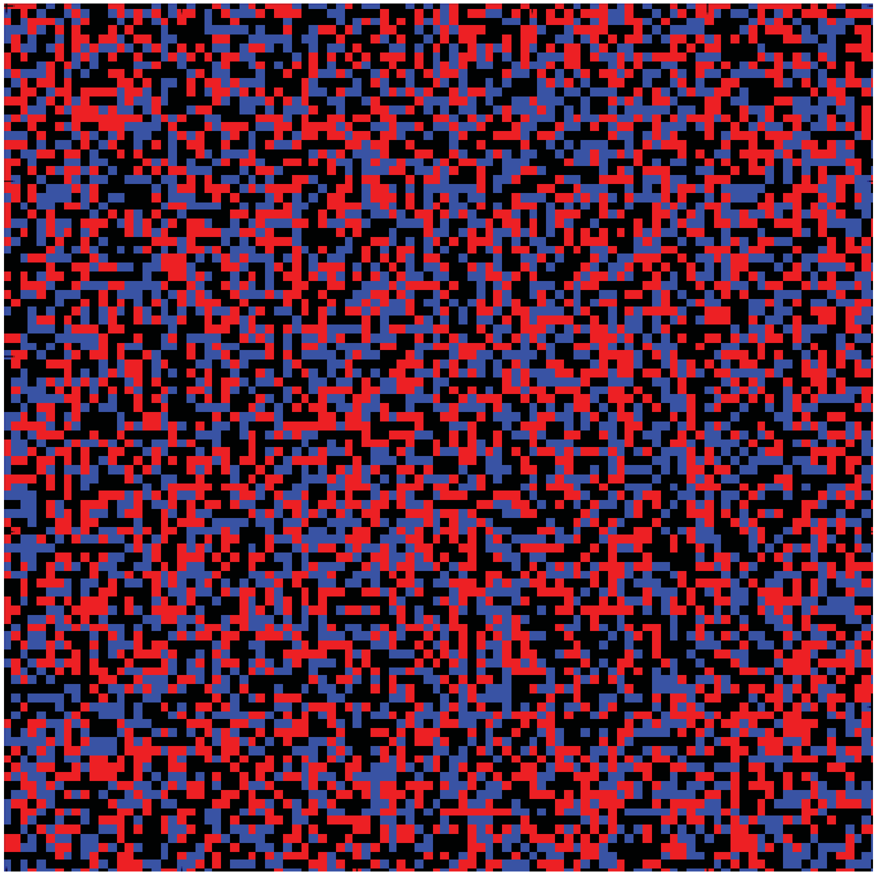
Enforce **smooth flow field**



u-component of flow

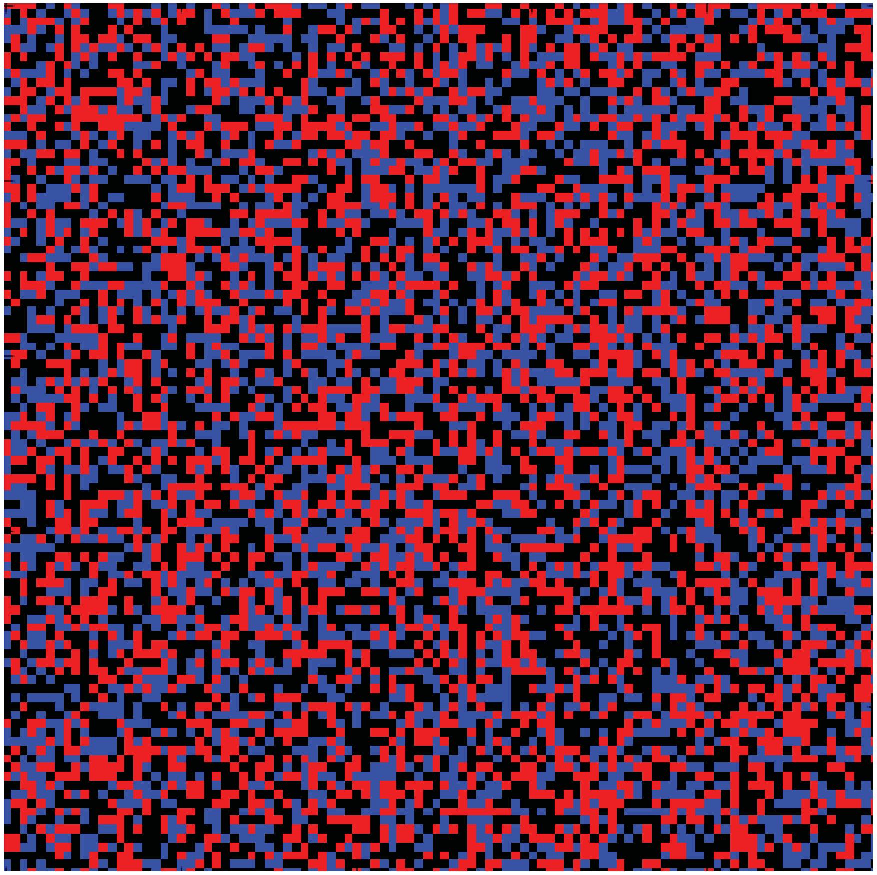
$$\min_{\boldsymbol{u}}(u_{i,j} - u_{i+1,j})^2$$

Which flow field optimizes the objective? $\min_{\boldsymbol{u}}(u_{i,j} - u_{i+1,j})^2$



$$\sum_{ij}(u_{ij} - u_{i+1,j})^2 \quad ? \quad \sum_{ij}(u_{ij} - u_{i+1,j})^2$$

Which flow field optimizes the objective? $\min_{\boldsymbol{u}} (u_{i,j} - u_{i+1,j})^2$



big

small

Key idea

(of Horn-Schunck optical flow)

Enforce

brightness constancy

Enforce

smooth flow field

to compute optical flow

bringing it all together…

Horn-Schunck optical flow

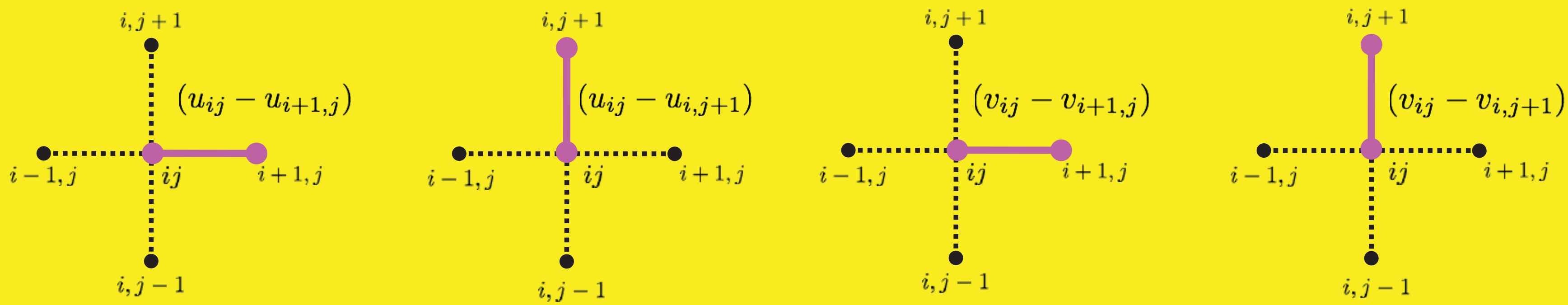
$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ \overset{\text{smoothness}}{E_s(i,j)} + \overset{\text{brightness constancy}}{\underset{\text{weight}}{\lambda} E_d(i,j)} \right\}$$

HS optical flow objective function

Brightness constancy $$E_d(i,j) = \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2$$

Smoothness

$$E_s(i,j) = \frac{1}{4}\left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right]$$



why not all four neighbors?

How do we solve this minimization problem?

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$$

How do we solve this minimization problem?

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$$

Compute partial derivative, derive update equations

(gradient decent!)

Compute the partial derivatives of this huge sum!

$$\sum_{ij} \left\{ \underbrace{\frac{1}{4}\left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2\right]}_{\text{smoothness term}} + \lambda \underbrace{\left[I_x u_{ij} + I_y v_{ij} + I_t\right]^2}_{\text{brightness constancy}} \right\}$$

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

it's not so bad…

$$\frac{\partial E}{\partial u_{kl}} =$$

how many u terms depend on k and l?

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

it's not so bad…

$$\frac{\partial E}{\partial u_{kl}}=$$

how many u terms depend on k and l?

FOUR from smoothness

ONE from brightness constancy

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

it's not so bad…

$$\frac{\partial E}{\partial u_{kl}}=2(u_{kl}-\bar{u}_{kl})+2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_x$$

how many u terms depend on k and l?

FOUR from smoothness

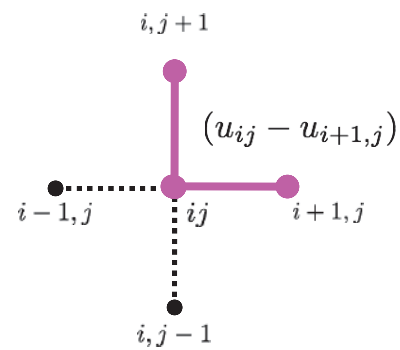
ONE from brightness constancy

Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

$$(u_{ij}^2-2u_{ij}u_{i+1,j}+u_{i+1,j}^2)\qquad (u_{ij}^2-2u_{ij}u_{i,j+1}+u_{i,j+1}^2)$$

(variable will appear four times in sum)

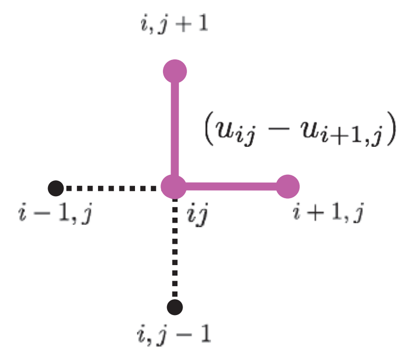


Compute the partial derivatives of this huge sum!

$$\sum_{ij}\left\{\frac{1}{4}\Big[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\Big]+\lambda\Big[I_x u_{ij}+I_y v_{ij}+I_t\Big]^2\right\}$$

$$(u_{ij}^2-2u_{ij}u_{i+1,j}+u_{i+1,j}^2) \qquad (u_{ij}^2-2u_{ij}u_{i,j+1}+u_{i,j+1}^2)$$

(variable will appear four times in sum)



$$\frac{\partial E}{\partial u_{kl}}=2(u_{kl}-\bar{u}_{kl})+2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}}=2(v_{kl}-\bar{v}_{kl})+2\lambda(I_x u_{kl}+I_y v_{kl}+I_t)I_y$$

short hand for
local average

$$\bar{u}_{ij}=\frac{1}{4}\left\{u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}\right\}$$

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

this is a linear system $\quad \mathbf{A}\boldsymbol{x} = \boldsymbol{b} \quad$ *how do you solve this?*

ok, take a step back, why are we doing all this math?

We are solving for the optical flow (u,v) given two constraints

$$\sum_{ij} \left\{ \frac{1}{4} \left[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2 \right] + \lambda \left[I_x u_{ij} + I_y v_{ij} + I_t \right]^2 \right\}$$

smoothness

brightness constancy

We need the math to minimize this
(back to the math)

Partial derivatives of Horn-Schunck objective function E:

$$\frac{\partial E}{\partial u_{kl}} = 2(u_{kl} - \bar{u}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_x$$

$$\frac{\partial E}{\partial v_{kl}} = 2(v_{kl} - \bar{v}_{kl}) + 2\lambda(I_x u_{kl} + I_y v_{kl} + I_t)I_y$$

Where are the extrema of E?

(set derivatives to zero and solve for unknowns u and v)

$$(1 + \lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1 + \lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

$$\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$$ *how do you solve this?*

$$(1+\lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1+\lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

Recall $\boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b} = \frac{\operatorname{adj}\mathbf{A}}{\det \mathbf{A}}\boldsymbol{b}$

$$(1+\lambda I_x^2)u_{kl} + \lambda I_x I_y v_{kl} = \bar{u}_{kl} - \lambda I_x I_t$$

$$\lambda I_x I_y u_{kl} + (1+\lambda I_y^2)v_{kl} = \bar{v}_{kl} - \lambda I_y I_t$$

Recall $\boldsymbol{x} = \mathbf{A}^{-1}\boldsymbol{b} = \dfrac{\operatorname{adj}\mathbf{A}}{\det \mathbf{A}}\boldsymbol{b}$

Same as the linear system:

$$\underbrace{\{1+\lambda(I_x^2+I_y^2)\}}_{\text{(det A)}}u_{kl} = (1+\lambda I_x^2)\bar{u}_{kl} - \lambda I_x I_y \bar{v}_{kl} - \lambda I_x I_t$$

$$\underbrace{\{1+\lambda(I_x^2+I_y^2)\}}_{\text{(det A)}}v_{kl} = (1+\lambda I_y^2)\bar{v}_{kl} - \lambda I_x I_y \bar{u}_{kl} - \lambda I_y I_t$$

$$\{1+\lambda(I_x^2+I_y^2)\}u_{kl} = (1+\lambda I_x^2)\bar{u}_{kl} - \lambda I_x I_y \bar{v}_{kl} - \lambda I_x I_t$$

$$\{1+\lambda(I_x^2+I_y^2)\}v_{kl} = (1+\lambda I_y^2)\bar{v}_{kl} - \lambda I_x I_y \bar{u}_{kl} - \lambda I_y I_t$$

Rearrange to get update equations:

$$\boxed{\begin{aligned} \underbrace{\hat{u}_{kl}}_{\text{new value}} &= \underbrace{\bar{u}_{kl}}_{\text{old average}} - \frac{I_x\bar{u}_{kl} + I_y\bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x \\ \hat{v}_{kl} &= \bar{v}_{kl} - \frac{I_x\bar{u}_{kl} + I_y\bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y \end{aligned}}$$

Recall: $\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$

When lambda is small (lambda inverse is big)…

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x$$

new value — old average

$$\hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y$$

Recall: $$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$$

When lambda is small (lambda inverse is big)…

$$\underbrace{\hat{u}_{kl}}_{\text{new value}} = \underbrace{\bar{u}_{kl}}_{\text{old average}} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\cancel{\lambda^{-1}} + I_x^2 + I_y^2} I_x$$

goes to zero

$$\hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\cancel{\lambda^{-1}} + I_x^2 + I_y^2} I_y$$

goes to zero

Recall: $\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{i,j} \left\{ E_s(i,j) + \lambda E_d(i,j) \right\}$

When lambda is small (lambda inverse is big)…

$$\hat{u}_{kl} = \bar{u}_{kl} - \cancel{\frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2}} I_x$$

goes to zero

new value

old average

$$\hat{v}_{kl} = \bar{v}_{kl} - \cancel{\frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2}} I_y$$

goes to zero

…we only care about smoothness.

ok, take a step back, why did we do all this math?

We are solving for the optical flow (u,v) given two constraints

$$\sum_{ij}\left\{\frac{1}{4}\left[(u_{ij}-u_{i+1,j})^2+(u_{ij}-u_{i,j+1})^2+(v_{ij}-v_{i+1,j})^2+(v_{ij}-v_{i,j+1})^2\right]+\lambda\left[I_x u_{ij}+I_y v_{ij}+I_t\right]^2\right\}$$

smoothness

brightness constancy

We needed the math to minimize this
(now to the algorithm)

Horn-Schunck
Optical Flow Algorithm

1. Precompute image gradients $I_y \quad I_x$

2. Precompute temporal gradients I_t

3. Initialize flow field $\quad \boldsymbol{u} = \mathbf{0}$

$\boldsymbol{v} = \mathbf{0}$

4. While not converged

Compute flow field updates for each pixel:

$$\hat{u}_{kl} = \bar{u}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_x \qquad \hat{v}_{kl} = \bar{v}_{kl} - \frac{I_x \bar{u}_{kl} + I_y \bar{v}_{kl} + I_t}{\lambda^{-1} + I_x^2 + I_y^2} I_y$$

Just 8 lines of code!

When assumptions break

- Brightness constancy is **not** satisfied

  Correlation based methods

- A point does **not** move like its neighbors
  - what is the ideal window size?

  Regularization based methods

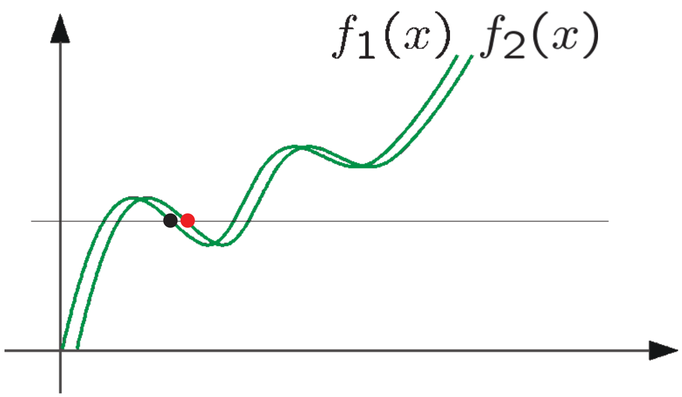
- The motion is **not** small (Taylor expansion doesn't hold)
- Aliasing

  Use multi-scale estimation

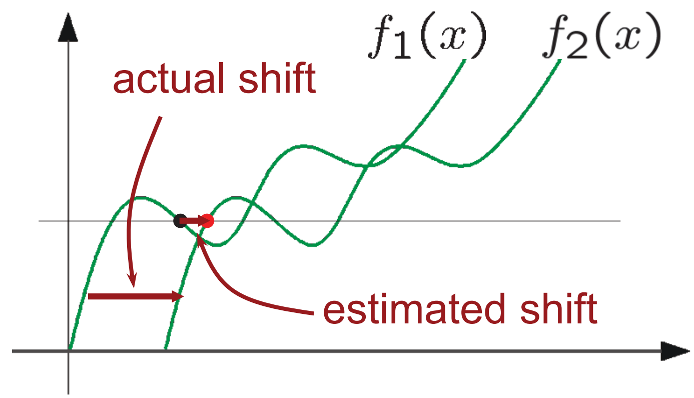
Optical Flow: Aliasing

Temporal aliasing causes ambiguities in optical flow because images can have many pixels with the same intensity.

I.e., how do we know which 'correspondence' is correct?



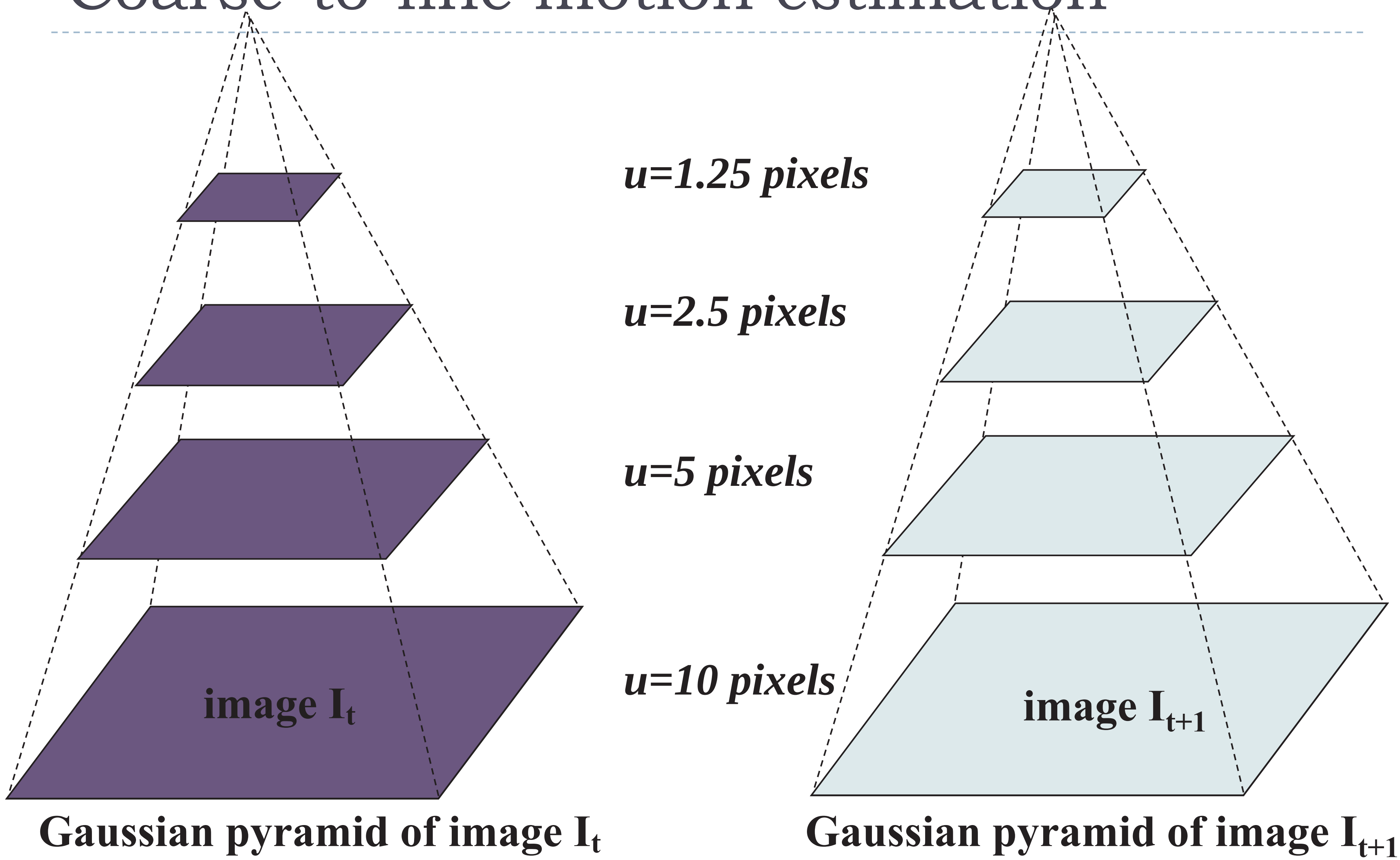
nearest match is correct (no aliasing)



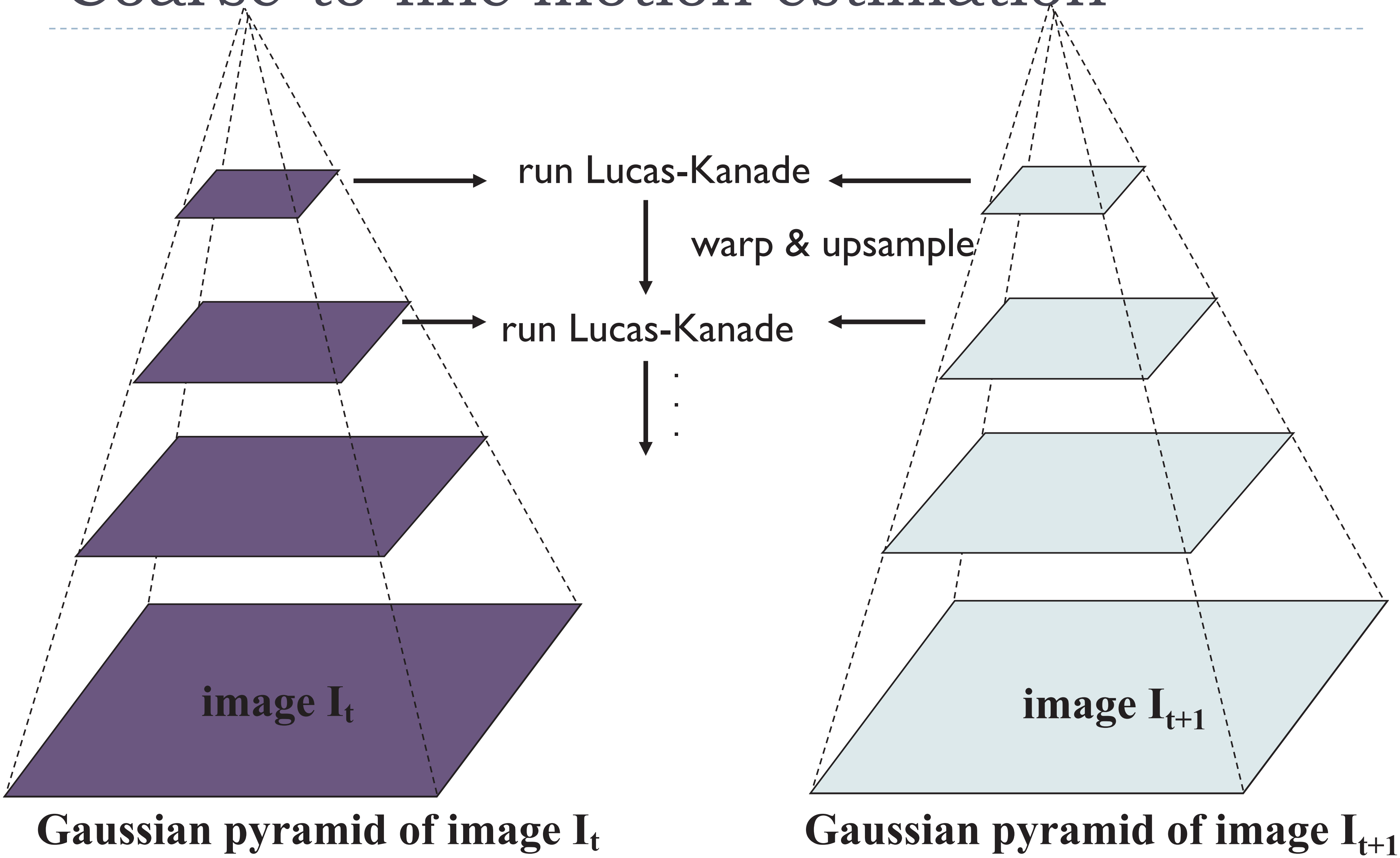
nearest match is incorrect (aliasing)

To overcome aliasing: coarse-to-fine estimation.

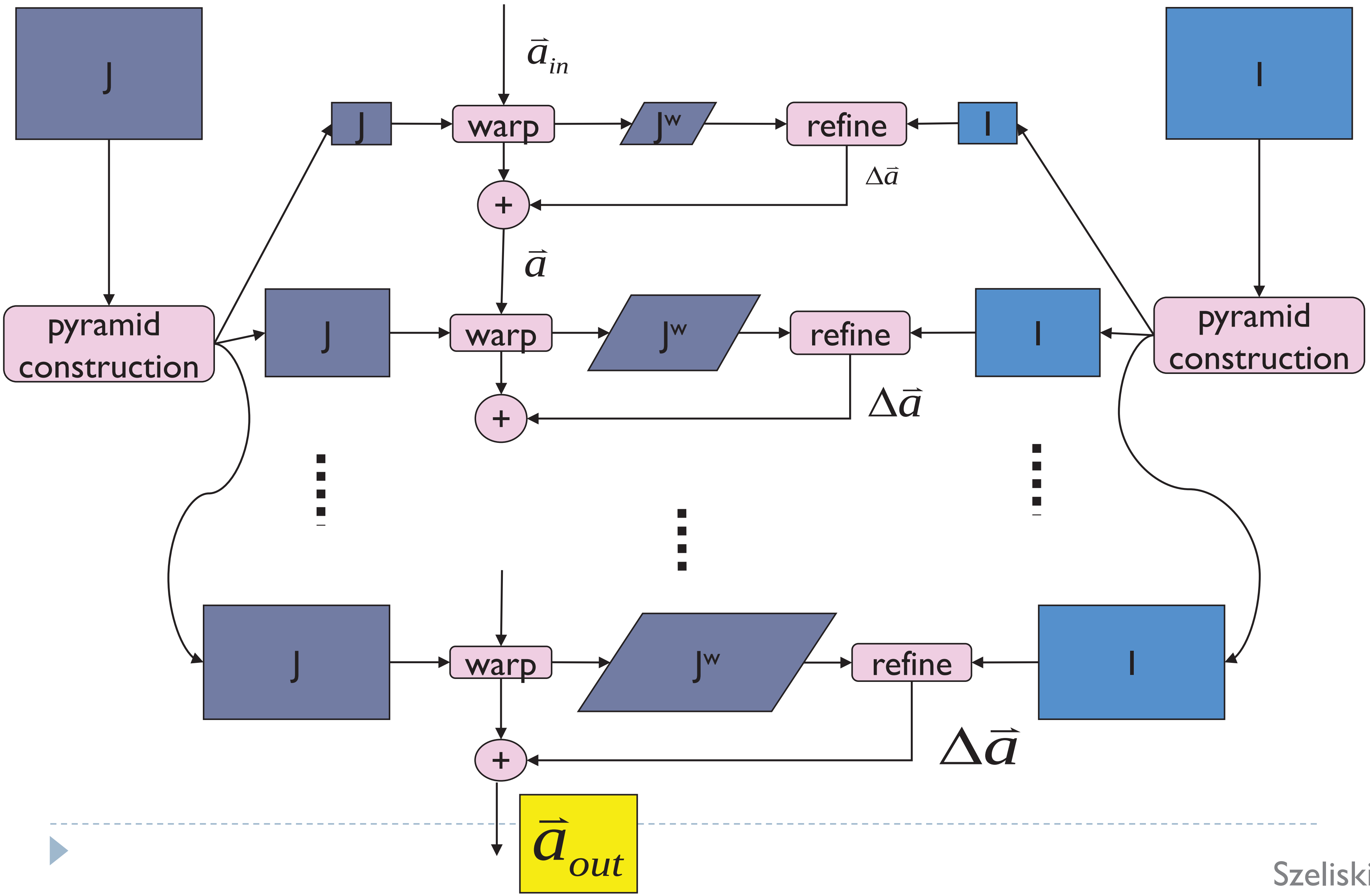
Coarse-to-fine motion estimation



Coarse-to-fine motion estimation



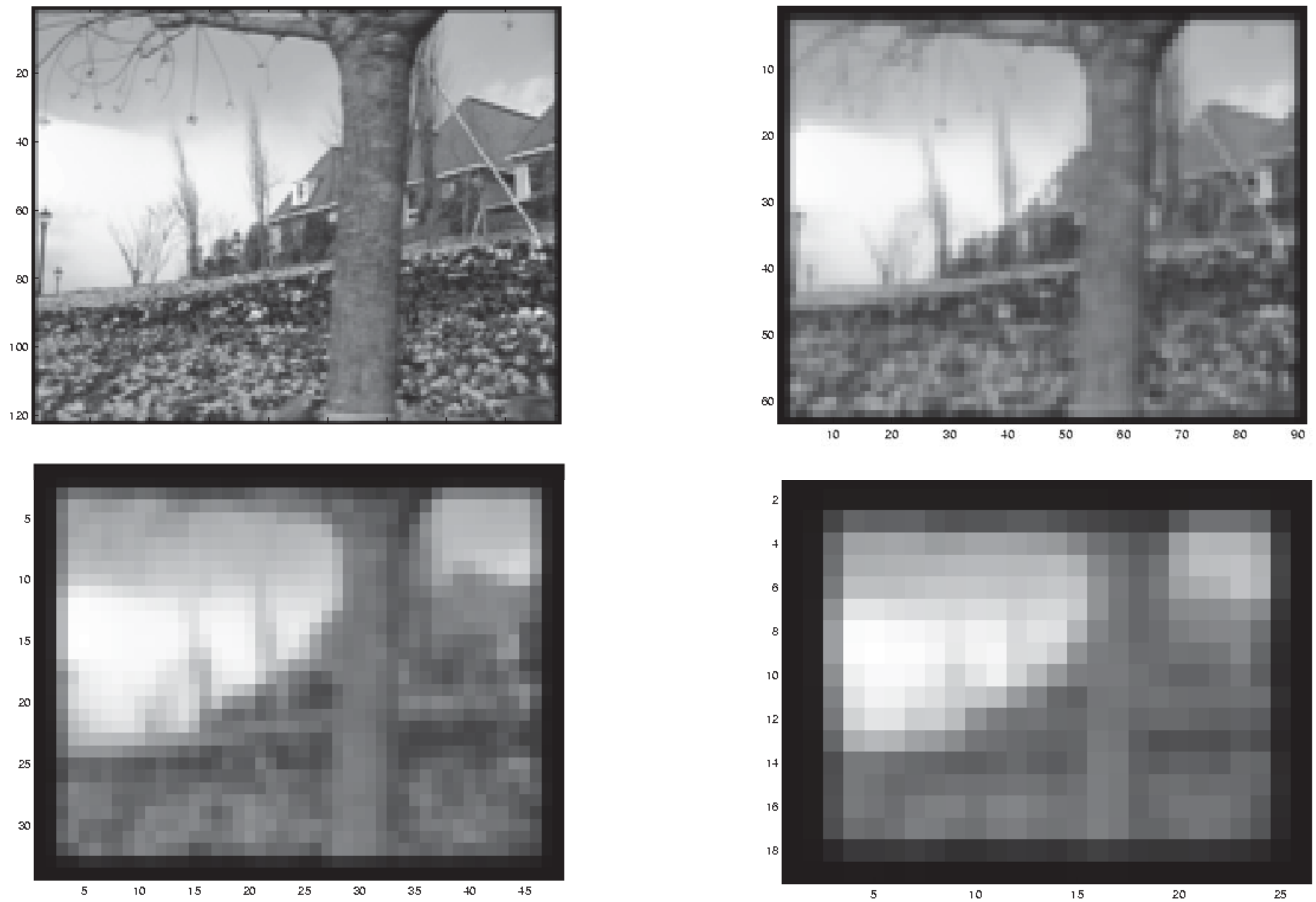
Coarse-to-Fine Estimation



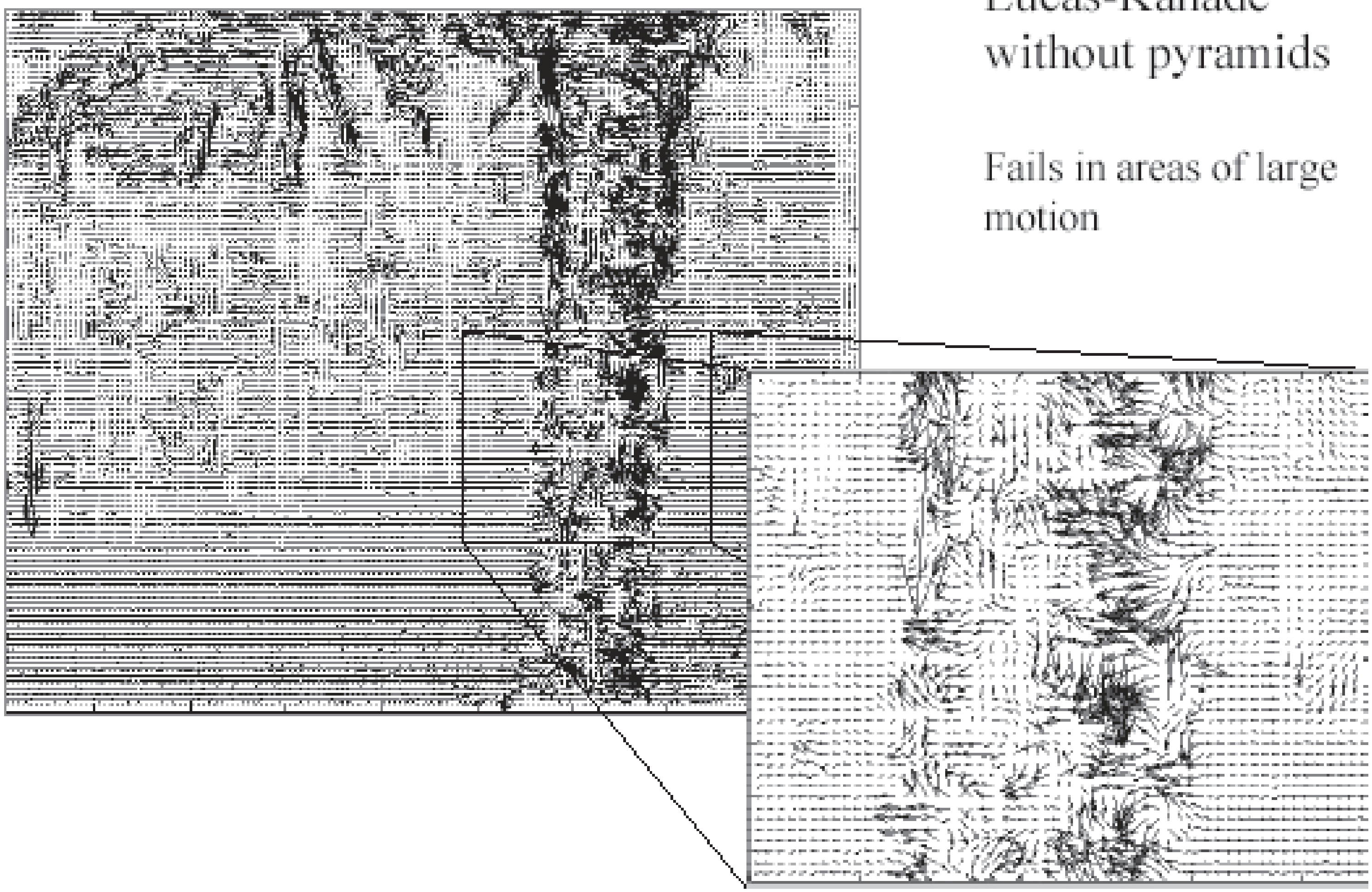
Szeliski

Example

Multi-resolution registration



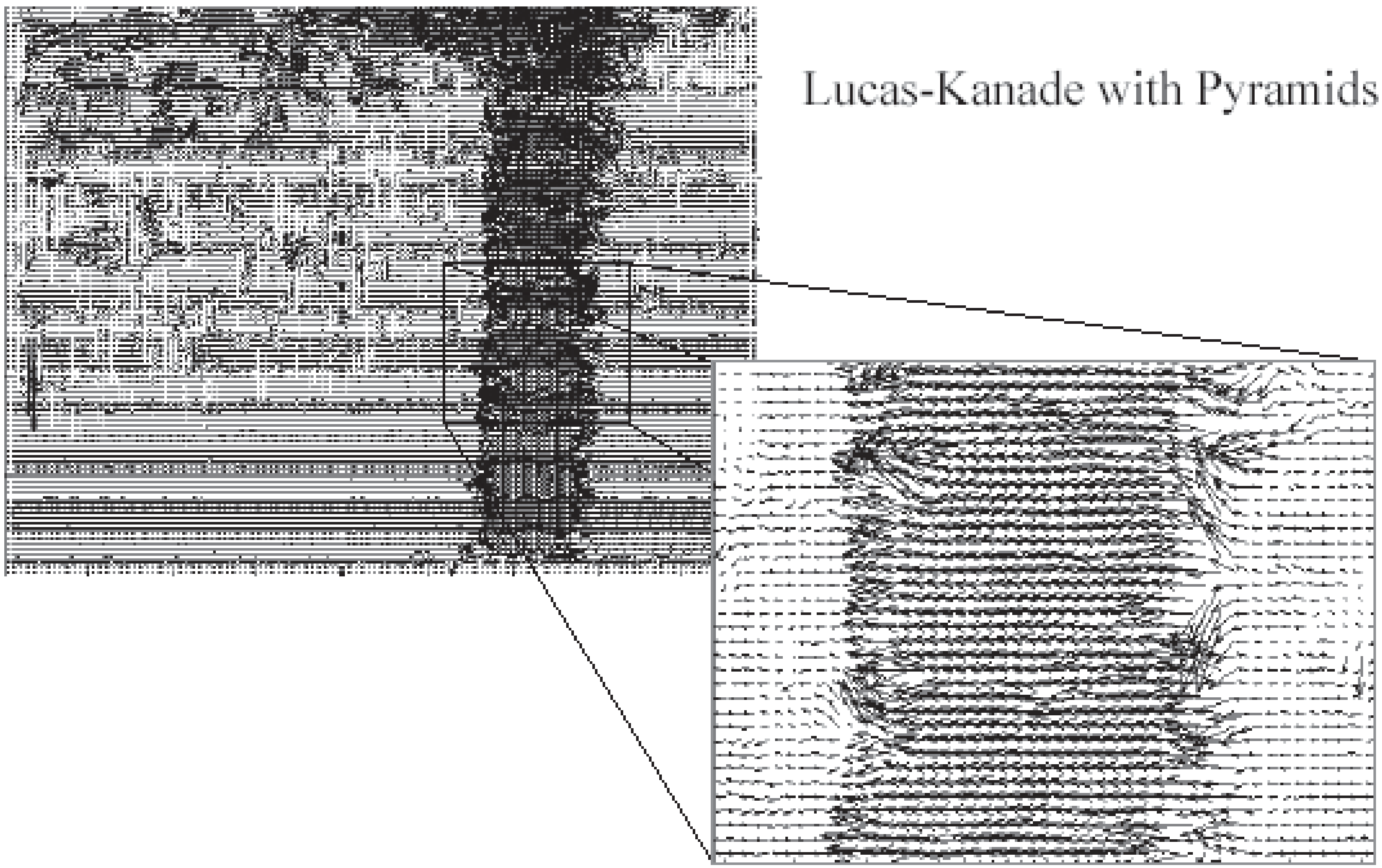
Optical flow results



Lucas-Kanade without pyramids

Fails in areas of large motion

Optical Flow Results



Lucas-Kanade with Pyramids

When assumptions break

- Brightness constancy is **not** satisfied

  → Correlation based methods

- A point does **not** move like its neighbors
  - what is the ideal window size?

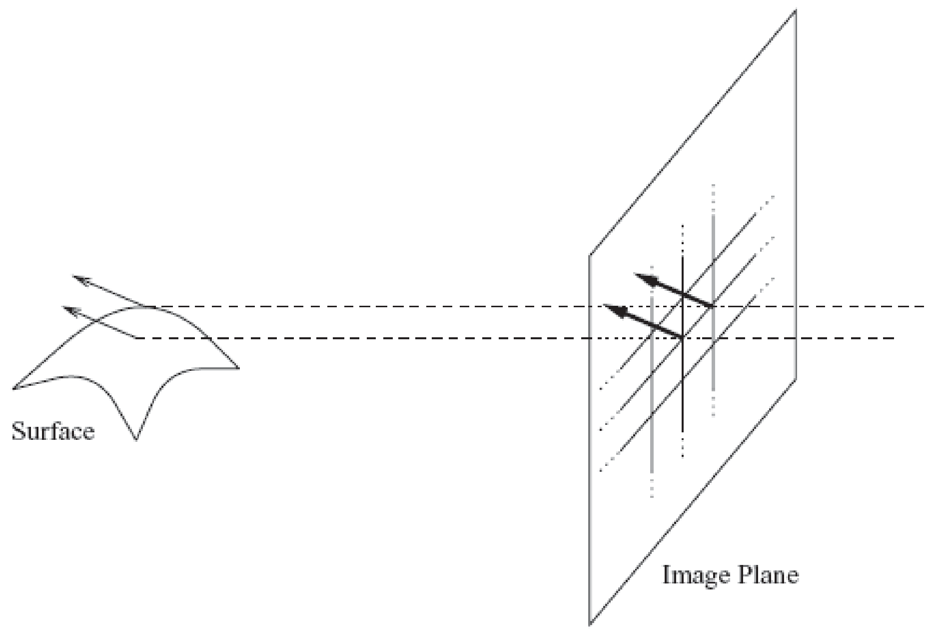
  → Regularization based methods

- The motion is **not** small (Taylor expansion doesn't hold)
- Aliasing

  → Use multi-scale estimation

Spatial coherence

- Neighboring points in the scene typically belong to the same surface and hence typically have similar motions.
- Since they also project to nearby points in the image, we expect spatial coherence in image flow.



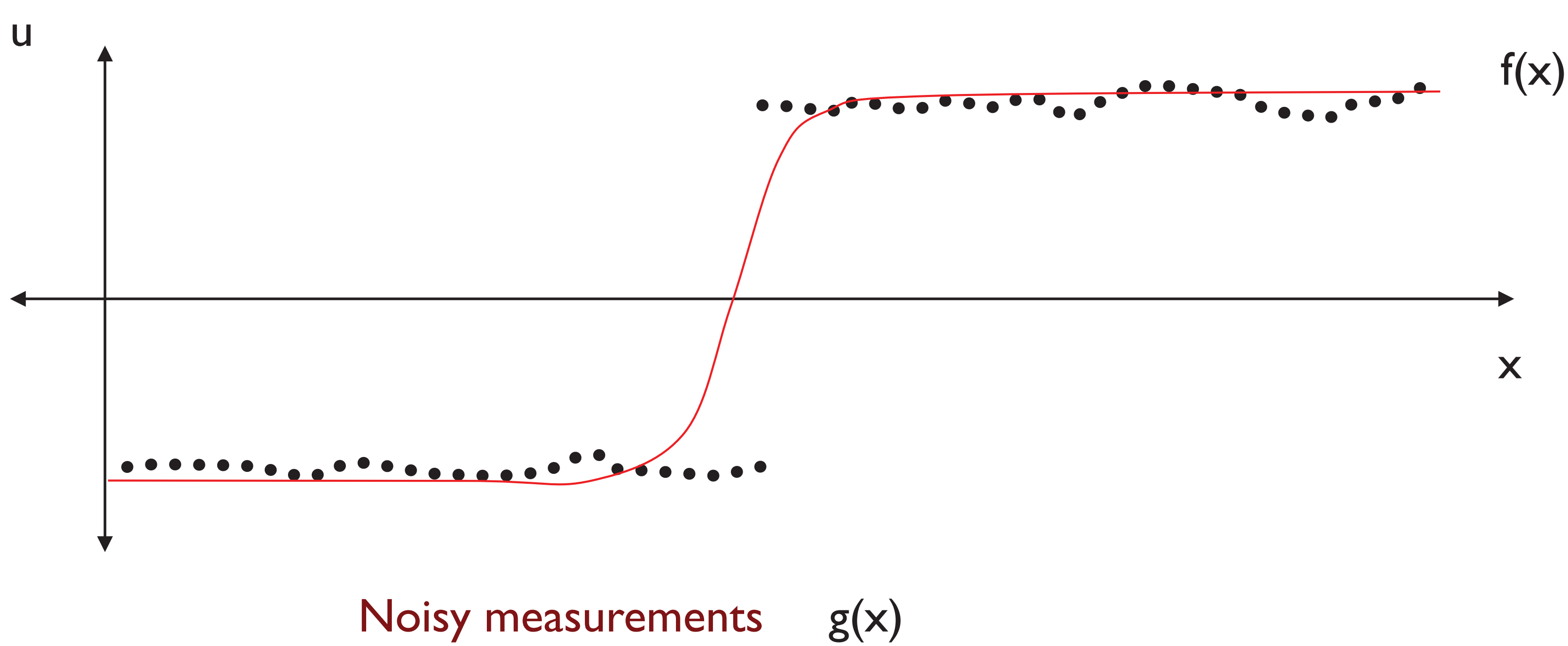
Formalize this Idea

Noisy 1D signal:

Noisy measurements   g(x)

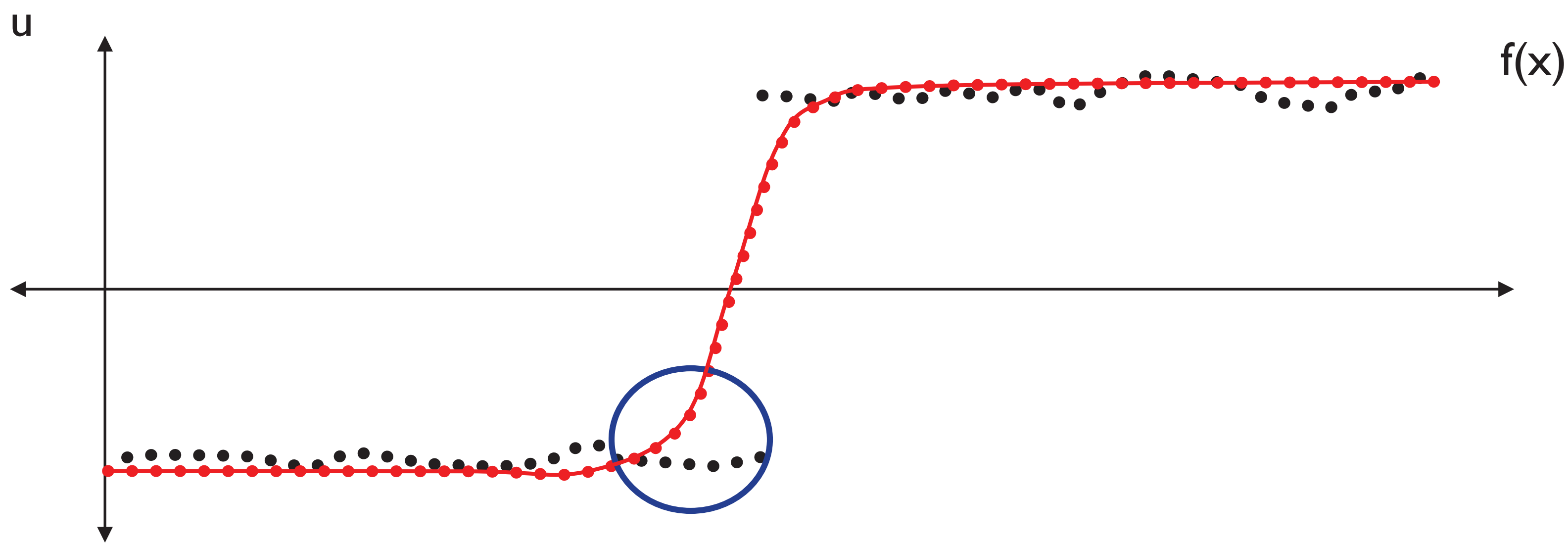
Spatial Regularization

Find the "best fitting" smoothed function f(x)

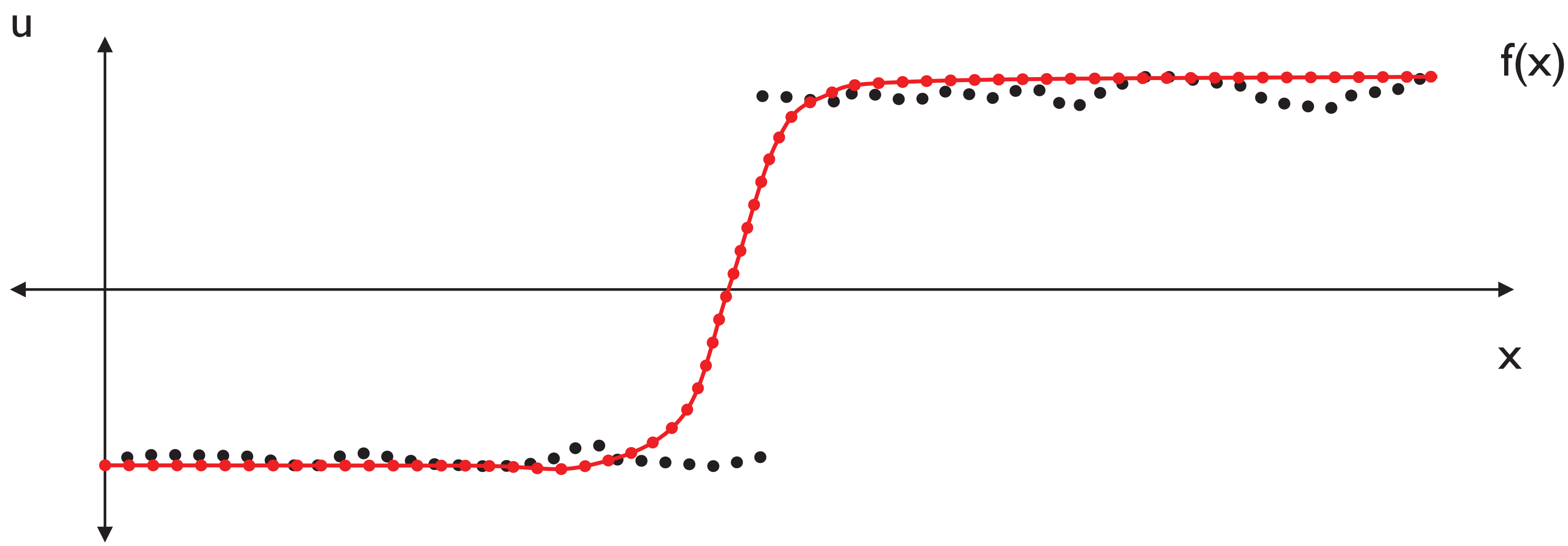


Spatial Regularization

Find the "best fitting" smoothed function f(x)



Regularization



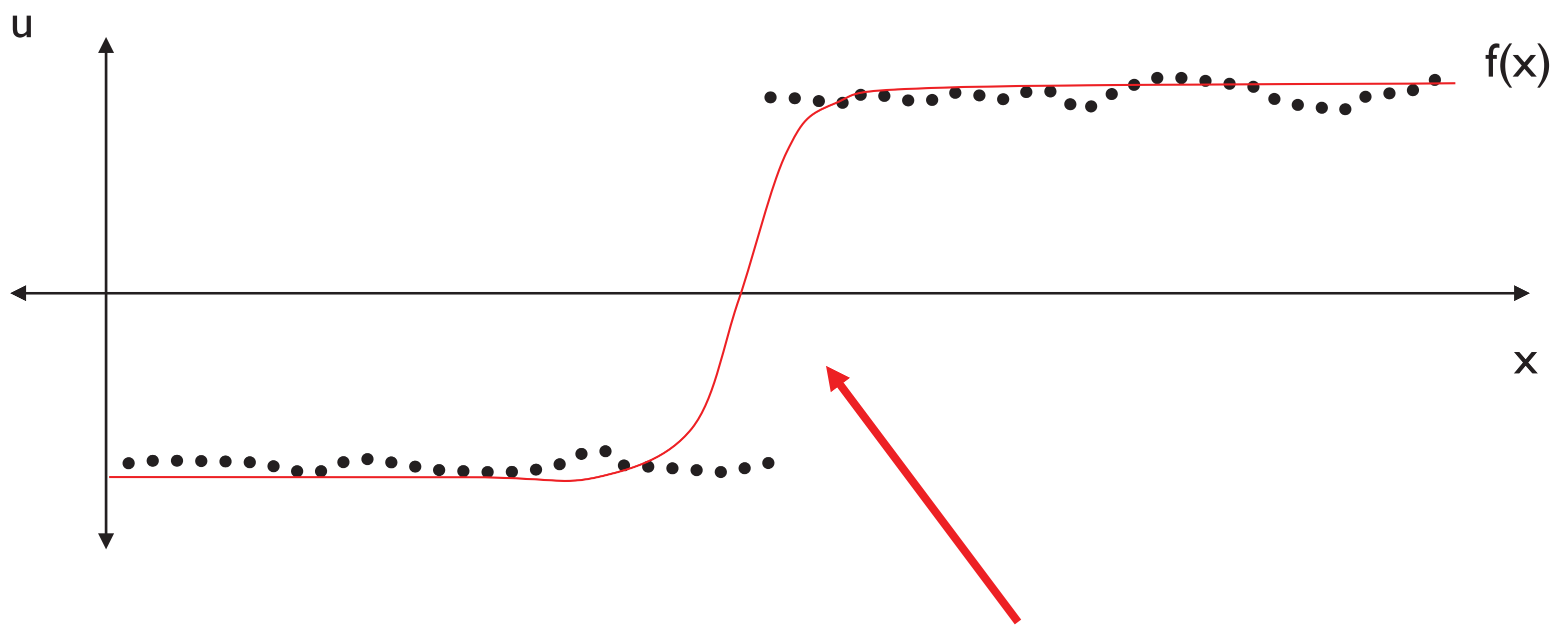
Minimize:

Faithful to the data

Spatial smoothness assumption

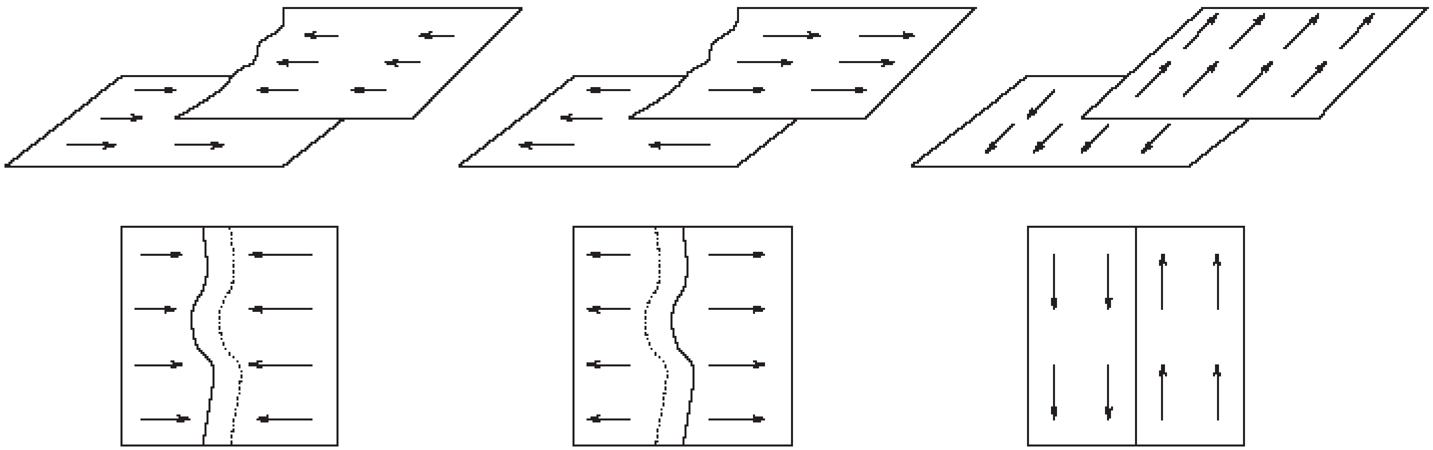
$$E(f)=\sum_{x=1}^{N}\left[f(x)-g(x)\right]^2+\lambda\sum_{x=1}^{N-1}\left[f(x+1)-f(x)\right]^2$$

Discontinuities



What about this discontinuity?
What is happening here?
What can we do?

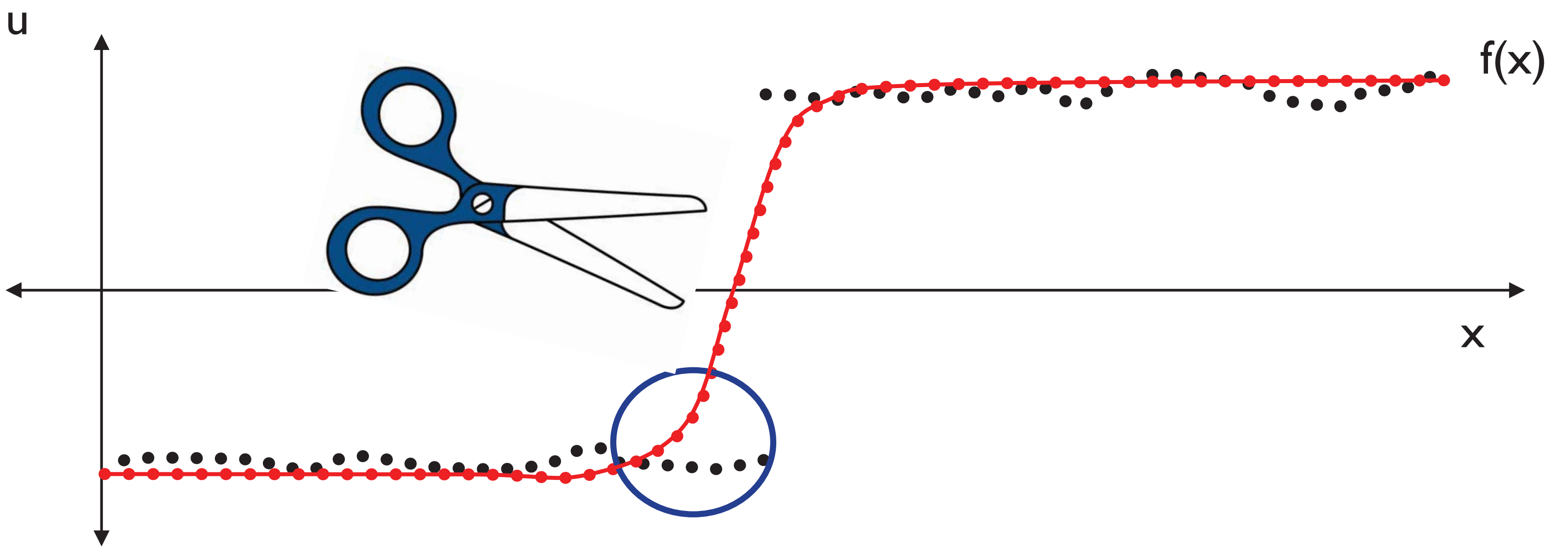
Occlusion



occlusion disocclusion shear

Multiple motions within a finite region.

Weak membrane model



$$E(f,l)=\sum_{x=1}^{N}\left[f(x)-g(x)\right]^2+\lambda\sum_{x=1}^{N-1}\left\{l(x)\left[f(x+1)-f(x)\right]^2+\beta\left[1-l(x)\right]\right\}$$

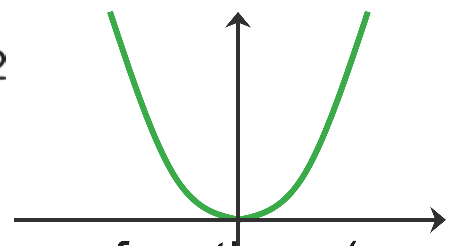
$$l(x)\in\{0,1\}$$ (binary)

Robust estimation

Problem: Least-squares estimators penalize deviations between data & model with quadratic error f^2 (extremely sensitive to outliers)

error penalty function

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



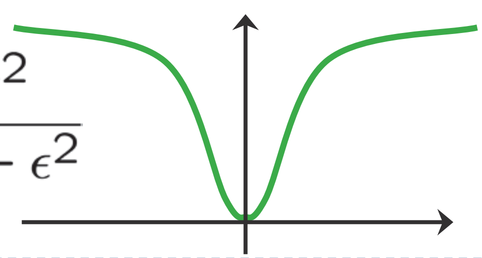
influence function

$$\psi(\epsilon) = \frac{\partial \rho(\epsilon)}{\partial \epsilon} = 2\,\epsilon$$

Redescending error functions (e.g., Geman-McClure) help to reduce the influence of outlying measurements.

error penalty function

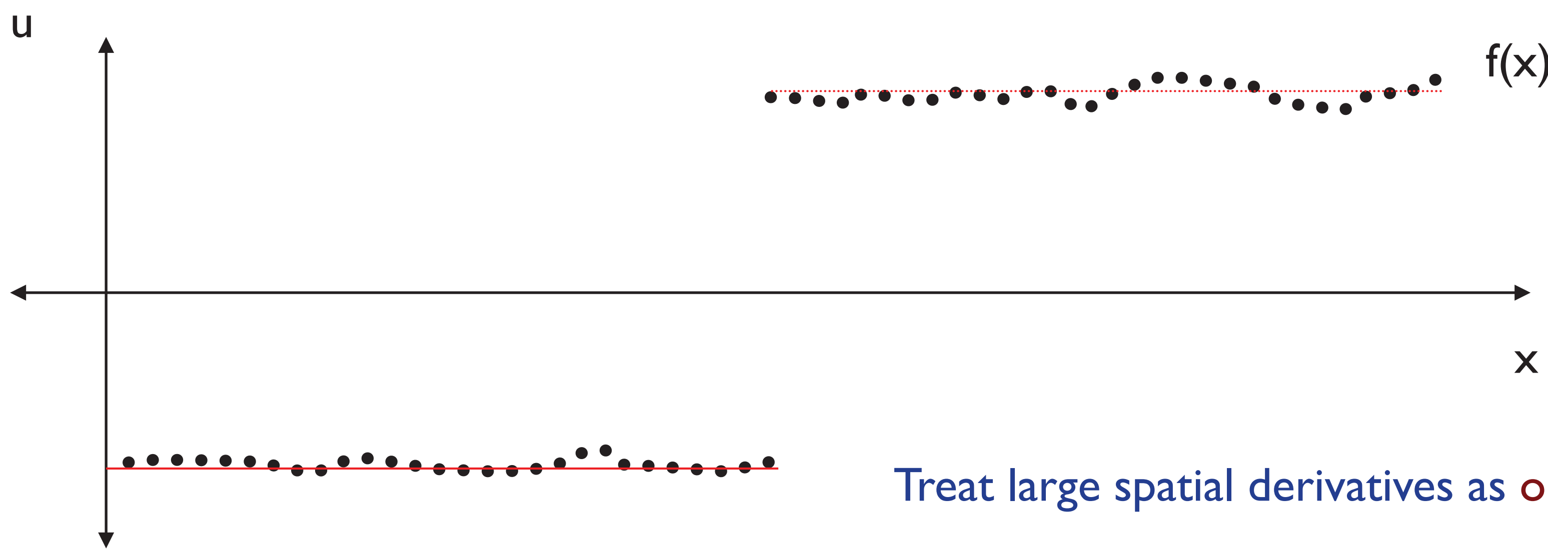
$$\rho(\epsilon;\, s) = \frac{\epsilon^2}{s + \epsilon^2}$$



influence function

$$\psi(\epsilon;\, s) = \frac{2\,\epsilon\, s}{(s + \epsilon^2)^2}$$

Robust regularization



Treat large spatial derivatives as outliers.

Minimize:

$$E(f) = \sum_{x=1}^{N} \rho(f(x) - g(x), \sigma_1) + \lambda \sum_{x=1}^{N-1} \rho(f(x+1) - f(x), \sigma_2)$$

SIFT Flow

Displacement between SIFT features, not between intensity images

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \left\| s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}) \right\|_1 + \frac{1}{\sigma^2} \sum_{\mathbf{p}} \left(u^2(\mathbf{p}) + v^2(\mathbf{p}) \right) +$$
$$\sum_{(\mathbf{p},\mathbf{q}) \in \varepsilon} \min\left(\alpha |u(\mathbf{p}) - u(\mathbf{q})|, d \right) + \min\left(\alpha |v(\mathbf{p}) - v(\mathbf{q})|, d \right)$$

http://people.csail.mit.edu/celiu/ECCV2008/

EpicFlow:
Edge-Preserving Interpolation of Correspondences for Optical Flow

Jerome Revaud

Joint work with:

Philippe Weinzaepfel
Zaid Harchaoui
Cordelia Schmid

Inria

https://thoth.inrialpes.fr/src/epicflow/
CVPR 2015

Optical flow estimation is challenging!

- Main remaining problems:
  - large displacements
  - occlusions
  - motion discontinuities

Optical flow estimation is challenging!

- Main remaining problems:
  - large displacements
  - occlusions
  - motion discontinuities

- Our approach: « **EpicFlow** »

  Epic: **E**dge-**P**reserving **I**nterpolation of **C**orrespondences

  - leverages state-of-the-art matching algorithm
    - invariant to large displacements
  - incorporate an edge-aware distance:
    - handles occlusions and motion discontinuities
  - state-of-the-art results

Related work:

- Variational optical flow [Horn and Schunck 1981]
  - energy:

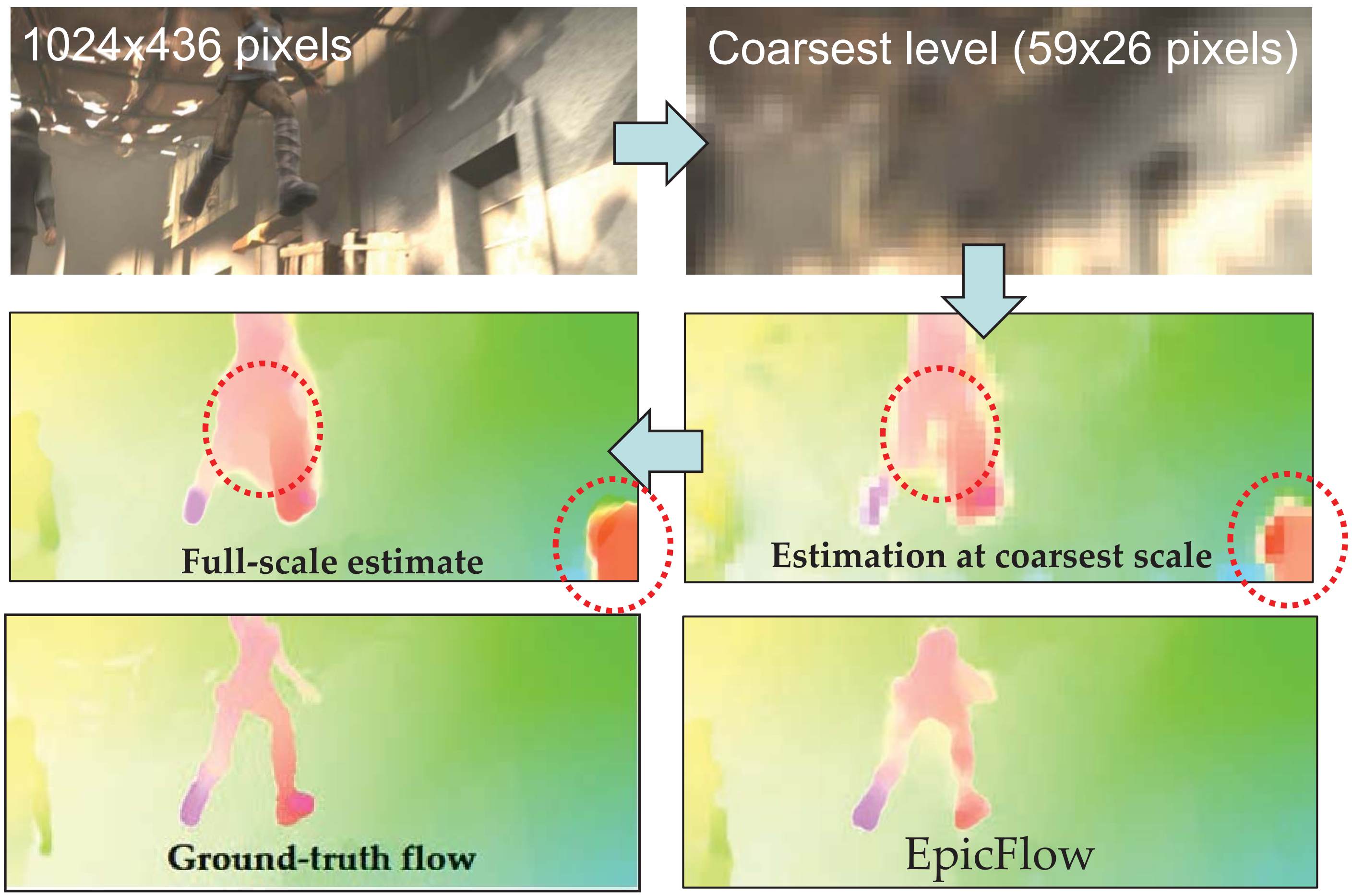
$$E = \underbrace{E_{data}}_{\text{color/gradient constancy}} + \alpha \underbrace{E_{smooth}}_{\text{smoothness constraint}}$$

- Solutions for handling large displacements
  - Minimization using a coarse-to-fine scheme [Horn'81, Brox'04, Sun'13, ..]
    - iterative energy minimization at several scales
    - displacements are small at coarse scales
  - Addition of a matching term [Brox'11, Braux-Zin'13, Weinzaepfel'13, ..]
    - penalizing the difference between flow and HOG matches

$$E = E_{data} + \alpha E_{smooth} + \beta E_{match}$$

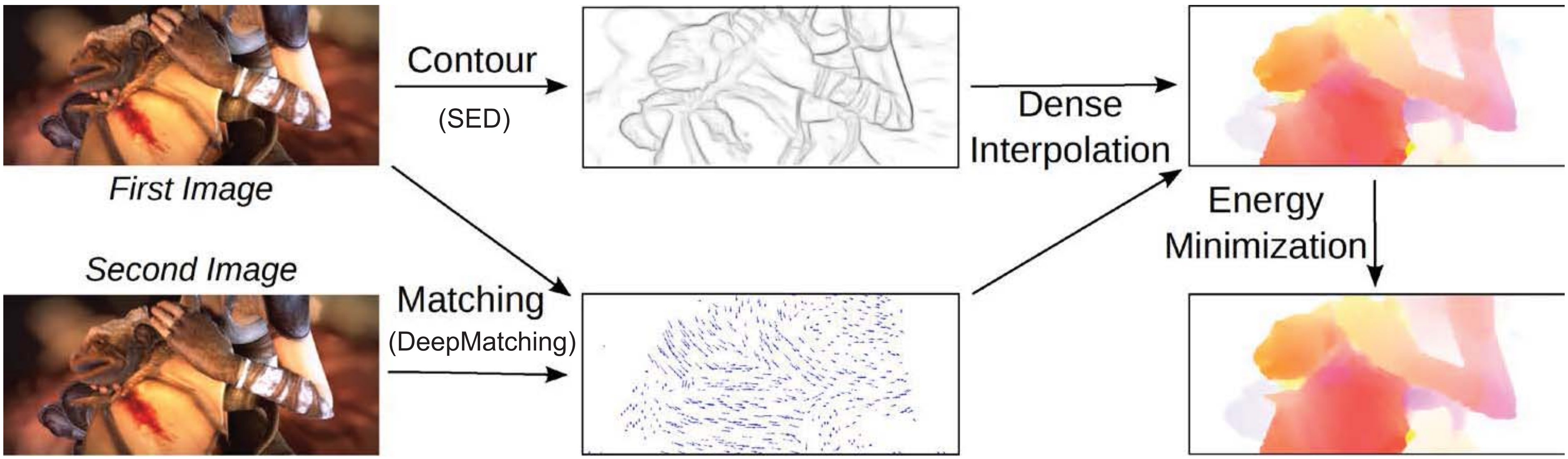
Related work: coarse-to-fine minimization

- Problems with coarse-to-fine:
  - flow discontinuities overlap at coarsest scales
  - errors are propagated across scales
  - no theoretical guarantees or proof of convergence!



Overview of our approach

based on sparse matches



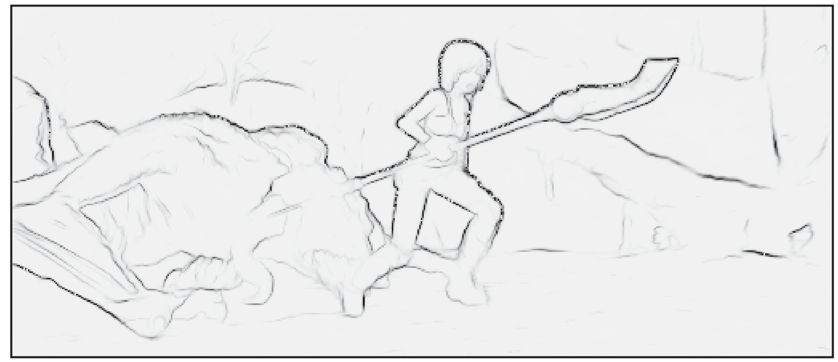
- Avoid coarse-to-fine scheme

- Other matching-based approaches [Chen'13, Lu'13, Leordeanu'13]
  - less robust matching algorithms (eg. PatchMatch)
  - complex schemes to handle occlusions & motion discontinuities
  - no geodesic distance

Dense Interpolation

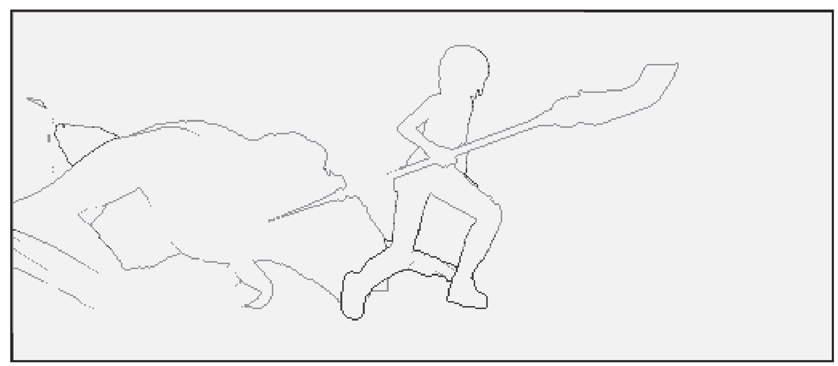
- Assumption:
  - motion boundaries are mostly included in image edges

image



contours (SED, [Dollar and Zitnick, 2013])

ground-truth flow



motion boundaries

Today

From images to video

- Optical flow
- Feature tracking
- Motion segmentation
  - Layered representation
- Applications

- How do we evaluate success?
  [Baker et al., A Database and Evaluation Methodology for Optical Flow, ICCV'07]

Synthetic video sequence

- Synthetic sequences can be used for quantitative evaluation

Yosemite

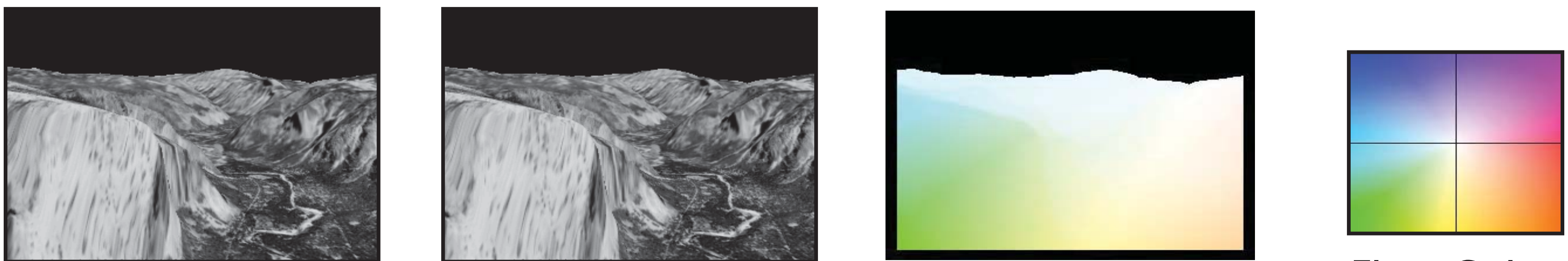


Image 7 Image 8 Ground-Truth Flow Flow Color Coding

- Limitation
  - Hard to make these a true representative of real video and its noise and blur

Real video with ground-truth

- Paint scene with textured fluorescent paint
- Take 2 images: One in visible light, one in UV light
- Move scene in very small steps using robot
- Generate ground-truth by tracking the UV images

Visible

UV

Setup Lights Image Cropped

Middlebury dataset (Baker et al. 2007)

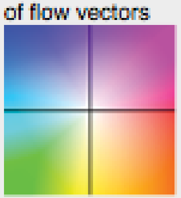
http://vision.middlebury.edu/flow/

Optical flow evaluation results Choose error measures: Average SD R1.0 R3.0 R5.0 A50 A75 A95

Average angle error	avg. rank	Dimetrodon (Hidden texture) GT im0 im1 all	disc	untext	Seashell (Hidden texture) GT im0 im1 all	disc	untext	Rock (Synthetic) GT im0 im1 all	disc	untext
Bruhn et al.	1.6	10.99_{3}	9.41_{1}	14.22_{3}	11.09_{2}	19.48_{2}	16.21_{2}	6.14_{1}	17.41_{1}	12.86_{2}
Black and Anandan	2.1	9.26_{1}	10.11_{3}	12.08_{1}	11.20_{3}	19.83_{3}	17.01_{3}	7.67_{3}	18.44_{3}	16.80_{4}
Pyramid LK	2.8	10.27_{2}	9.71_{2}	13.63_{2}	9.46_{1}	18.62_{1}	12.07_{1}	6.53_{2}	18.43_{2}	10.95_{1}
MediaPlayer™	4.1	15.82_{4}	26.42_{4}	16.96_{4}	23.18_{4}	27.71_{5}	21.78_{4}	9.44_{4}	22.25_{4}	15.03_{3}
Zitnick et al.	4.2	30.10_{5}	34.27_{5}	31.58_{5}	29.07_{5}	27.55_{4}	21.78_{4}	12.38_{5}	23.93_{5}	17.59_{5}

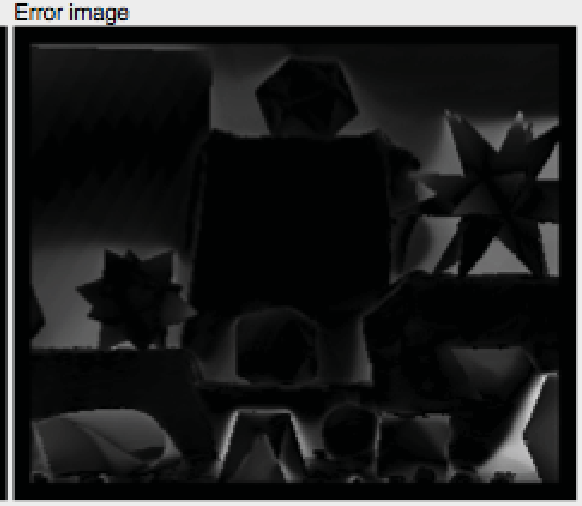
Average angle error	Grove (Synthetic) GT im0 im1 all	disc	untext	Yosemite (Synthetic) GT im0 im1 all	disc	untext	Venus (Stereo) GT im0 im1 all	disc	untext	Moebius (Stereo) GT im0 im1 all	disc	untext
Bruhn et al.	6.32_{1}	12.41_{1}	10.98_{1}	1.69_{1}	2.86_{1}	1.05_{1}	8.73_{2}	31.46_{2}	8.15_{2}	5.85_{1}	10.12_{2}	8.80_{2}
Black and Anandan	7.89_{2}	13.55_{2}	13.96_{4}	2.65_{2}	4.18_{2}	1.88_{2}	7.64_{1}	30.13_{1}	7.31_{1}	7.05_{2}	10.02_{1}	8.41_{1}
Pyramid LK	8.14_{3}	15.08_{3}	12.78_{2}	5.22_{3}	6.64_{3}	4.29_{3}	14.61_{4}	36.18_{4}	24.67_{5}	12.98_{5}	13.85_{4}	20.61_{5}
MediaPlayer™	10.99_{4}	18.15_{5}	13.64_{3}	11.09_{4}	17.16_{4}	10.66_{5}	15.48_{5}	43.56_{5}	15.09_{4}	9.98_{4}	15.04_{5}	9.47_{3}
Zitnick et al.	12.55_{5}	15.56_{4}	17.35_{5}	18.50_{5}	28.00_{5}	9.41_{4}	11.42_{3}	31.46_{2}	11.12_{3}	9.88_{3}	12.83_{3}	11.28_{4}

Color encoding of flow vectors



Flow image

Error image



KITTI dataset (Geiger et al. 2012)

http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow

The **stereo 2015** / **flow 2015** / **scene flow 2015** benchmark consists of 200 training scenes and 200 test scenes (4 color images per scene, saved in loss less png format). Compared to the stereo 2012 and flow 2012 benchmarks, it comprises dynamic scenes for which the ground truth has been established in a semi-automatic process. Our evaluation server computes the percentage of bad pixels averaged over all ground truth pixels of all 200 test images. For this benchmark, we consider a pixel to be correctly estimated if the disparity or flow end-point error is **<3px or <5%** (for scene flow this criterion needs to be fulfilled for both disparity maps and the flow map). We require that all methods use the same parameter set for all test pairs. Our development kit provides details about the data format as well as MATLAB / C++ utility functions for reading and writing disparity maps and flow fields. More details can be found in Object Scene Flow for Autonomous Vehicles (CVPR 2015).

KITTI dataset (Geiger et al. 2012)

http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow

Evaluation ground truth: All pixels   Evaluation area: All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	UberATG-DRISF			**3.59 %**	10.40 %	**4.73 %**	100.00 %	0.75 s	CPU+GPU @ 2.5 Ghz (Python)	
W. Ma, S. Wang, R. Hu, Y. Xiong and R. Urtasun: Deep Rigid Instance Scene Flow. CVPR 2019.										
2	DH-SF			4.12 %	12.07 %	5.45 %	100.00 %	350 s	1 core @ 2.5 Ghz (Matlab + C/C++)	
3	ISF			5.40 %	10.29 %	6.22 %	100.00 %	10 min	1 core @ 3 Ghz (C/C++)	
A. Behl, O. Jafari, S. Mustikovela, H. Alhaija, C. Rother and A. Geiger: Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?. International Conference on Computer Vision (ICCV) 2017.										
4	HD^3-Flow		code	6.05 %	9.02 %	6.55 %	100.00 %	0.10 s	NVIDIA Pascal Titan XP	
Z. Yin, T. Darrell and F. Yu: Hierarchical Discrete Distribution Decomposition for Match Density Estimation. CVPR 2019.										
5	PRSM		code	5.33 %	13.40 %	6.68 %	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.										
6	OSF+TC			5.76 %	13.31 %	7.02 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)	
M. Neoral and J. Šochman: Object Scene Flow with Temporal Consistency. 22nd Computer Vision Winter Workshop (CVWW) 2017.										
7	SSF			5.63 %	14.71 %	7.14 %	100.00 %	5 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
Z. Ren, D. Sun, J. Kautz and E. Sudderth: Cascaded Scene Flow Prediction using Semantic Segmentation. International Conference on 3D Vision (3DV) 2017.										
8	SPOSF			5.41 %	15.96 %	7.16 %	100.00 %	10 min	1 core @ 3.5 Ghz (Matlab + C/C++)	
9	MFF			7.15 %	7.25 %	7.17 %	100.00 %	0.05 s	NVIDIA Pascal Titan X (Python)	
Z. Ren, O. Gallo, D. Sun, M. Yang, E. Sudderth and J. Kautz: A Fusion Approach for Multi-Frame Optical Flow Estimation. IEEE Winter Conference on Applications of Computer Vision 2019.										
10	OSF 2018		code	5.38 %	17.61 %	7.41 %	100.00 %	390 s	1 core @ 2.5 Ghz (Matlab + C/C++)	
M. Menze, C. Heipke and A. Geiger: Object Scene Flow. ISPRS Journal of Photogrammetry and Remote Sensing (JPRS) 2018.										

Today

From images to video

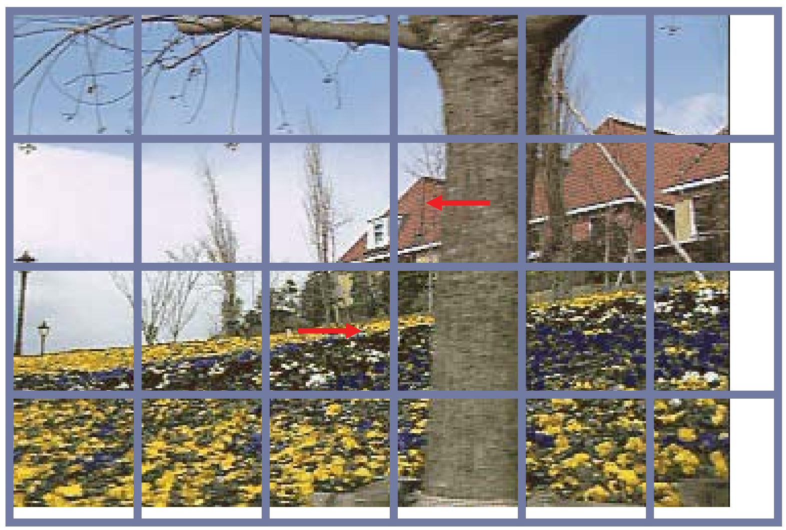
- Optical flow
- Feature tracking
- Motion segmentation
  - Layered representation
- Applications

Motion representations

- How can we describe the motion in the scene?

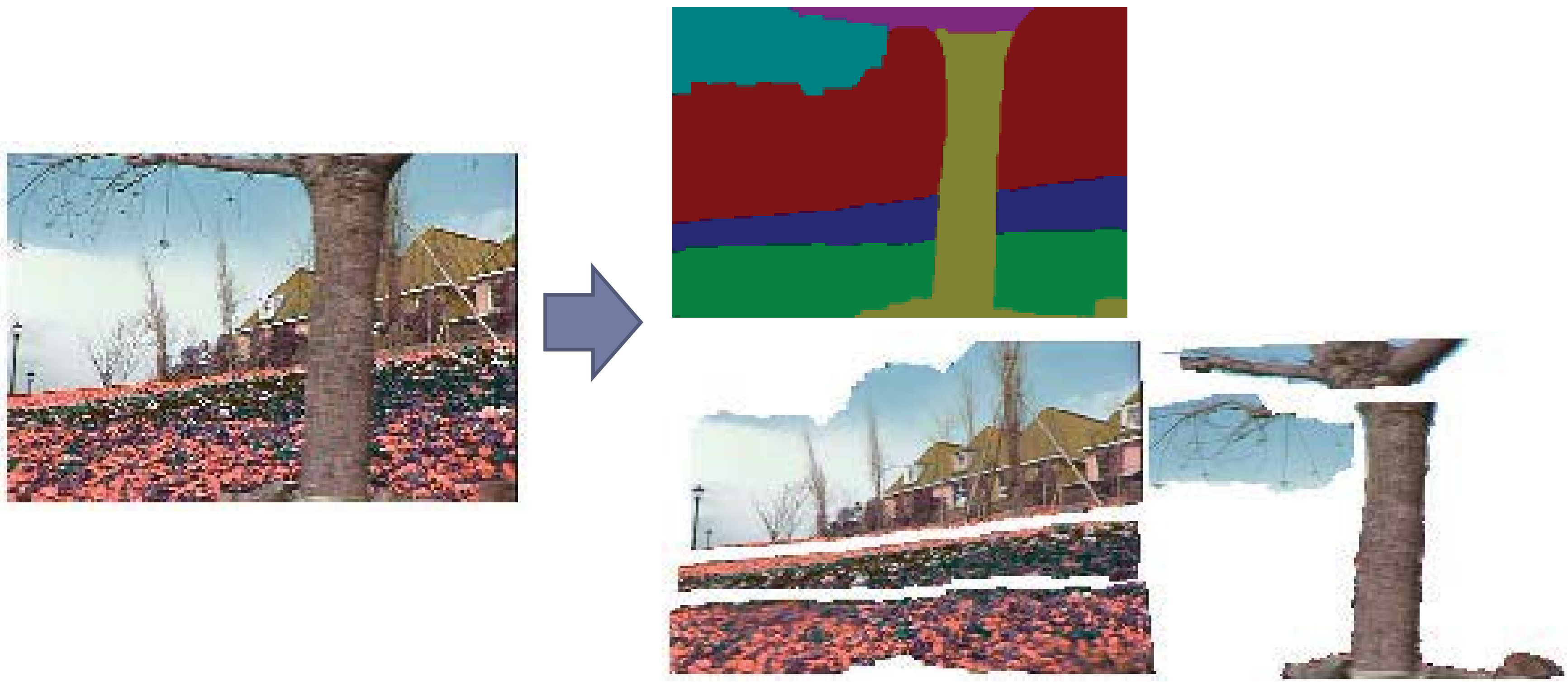
Block-based motion prediction

- Break image up into square blocks
- Estimate translation for each block
- Use this to predict next frame, code difference (MPEG-2)



Layered motion representation

- Break image sequence up into "layers" of coherent motion
- Each layer's motion is represented by a parametric model



Affine motion (dense)

- Recall the brightness constancy equation

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

- Assume affine motion

$$u = a_1 + a_2 x + a_3 y$$

$$v = a_4 + a_5 x + a_6 y$$

- Combine the equations

$$I_x\left(a_1 + a_2 x + a_3 y\right) + I_y\left(a_4 + a_5 x + a_6 y\right) + I_t = 0$$

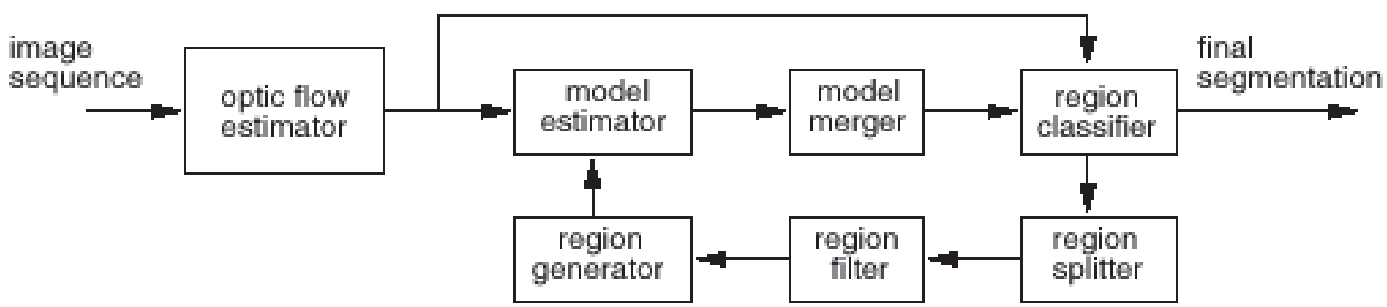
  - Each pixel provides one equation
  - Solve with Least-squares

Layered motion

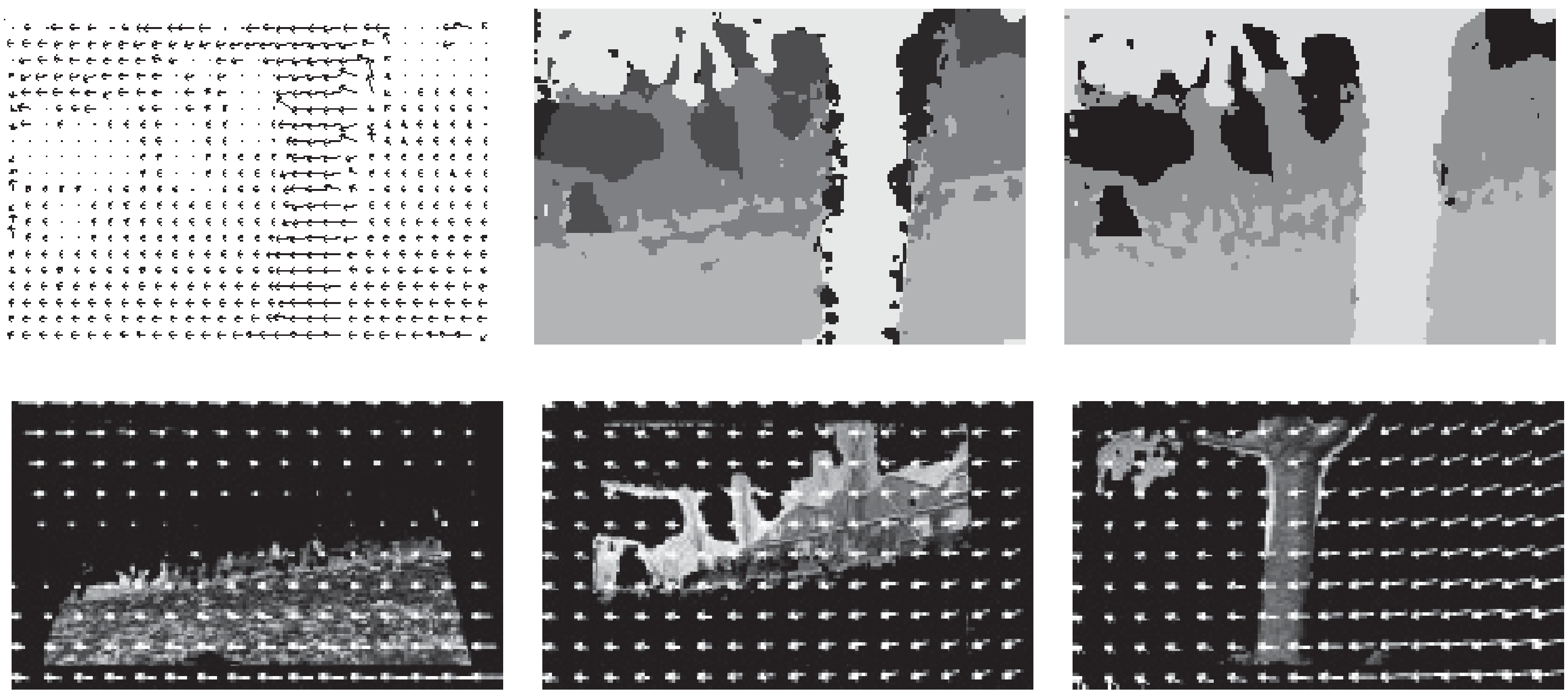
- Advantages
  - can represent occlusions / disocclusions
  - each layer's motion can be smooth
  - video segmentation for semantic processing
- Difficulties:
  - how do we determine the correct number?
  - how do we assign pixels?
  - how do we model the motion?

How do we estimate the layers?

1. compute coarse-to-fine flow
2. estimate affine motion for each block
3. cluster with *k-means*
4. assign pixels to best fitting affine region
5. re-estimate affine motions in each region…



Layered motion result



[Wang & Adelson, CVPR'93]

Layered motion representation (option2)

For scenes with multiple parametric motions

Estimate dominant motion parameters

Reject pixels which do not fit

Convergence

Restart on remaining pixels

Segmentation of Affine Motion

Input

Segmentation result

[Zelnik-Manor & Irani, PAMI 2000]

Today

From images to video

- Optical flow
- Feature tracking
- Motion segmentation
  - Layered representation
- Applications

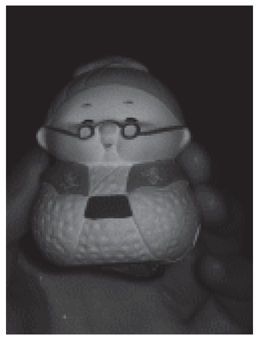
Panoramas

Input

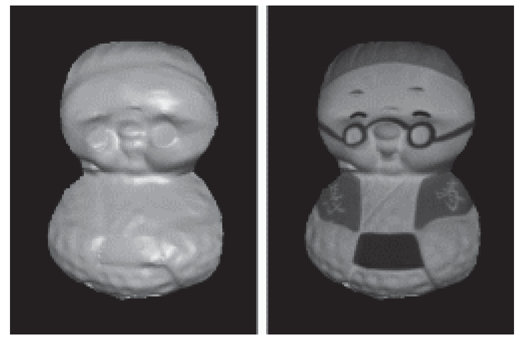
Camera ego-motion

Result by MobilEye (www.mobileye.com)

Structure from Motion



Input



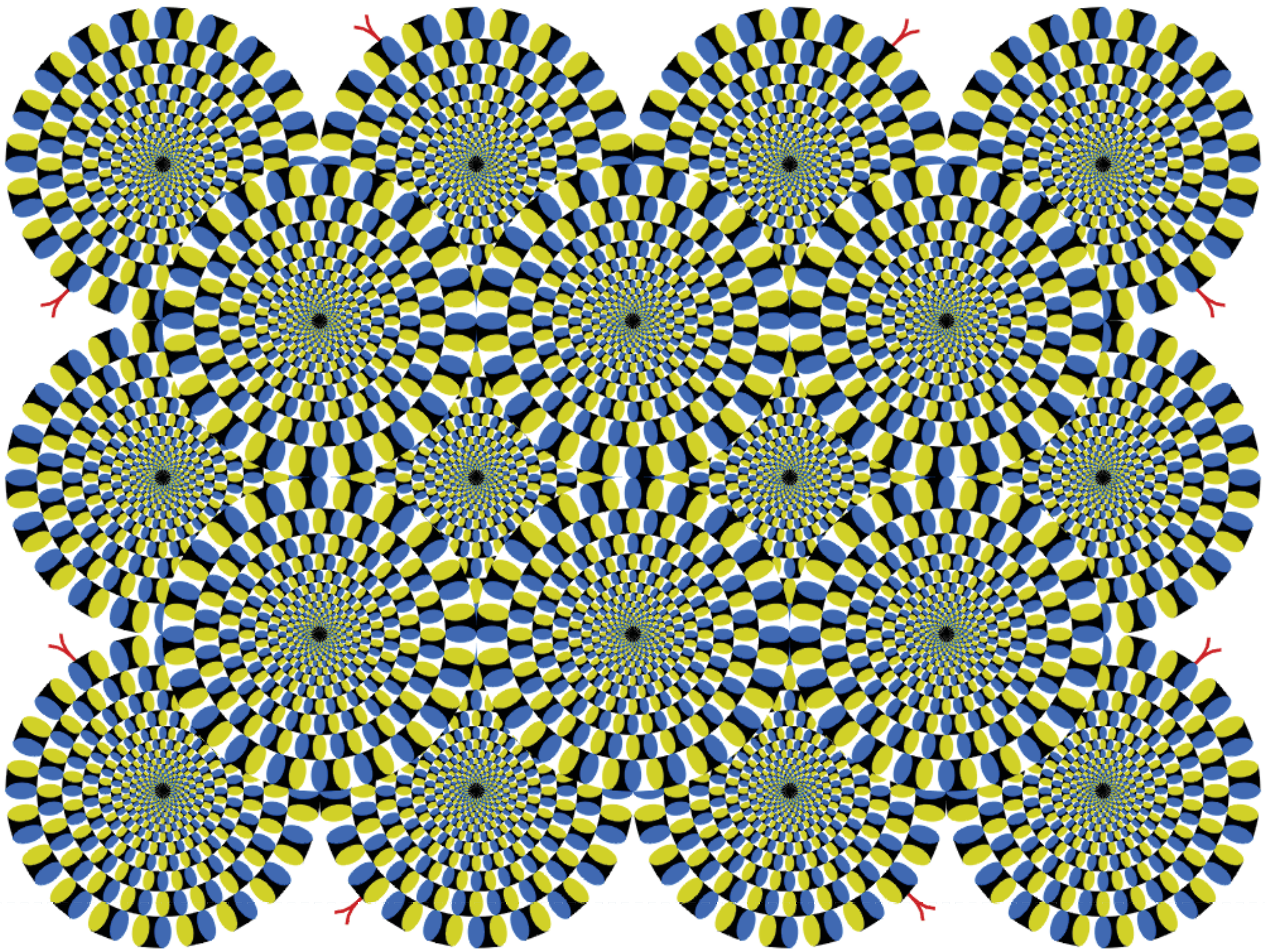
Reconstructed shape

[Zhang, et al. ICCV'03]

Stabilization

[Zelnik-Manor & Irani, PAMI 2000]

Optical flow without motion



References on Optical Flow

Lucas-Kanade method:

- B.D. Lucas and T. Kanade "***An Iterative Image Registration Technique with an Application to Stereo Vision***" IJCAI '81 pp. 674-679
- S. Baker and I. Matthews "***Lucas-Kanade 20 Years On: A Unifying Framework***" IJCV, Vol. 56, No. 3, March, 2004, pp. 221 - 255. http://www.ri.cmu.edu/projects/project_515.html (papers + code)

Regularization based methods:

- B. K. P. Horn and B. Schunck, "***Determining Optical Flow***," Artificial Intelligence, 17 (1981), pp. 185-203
- Black, M. J. and Anandan, P., "***A framework for the robust estimation of optical flow***", ICCV' 93, May, 1993, pp. 231-236 (papers + code)

Comparison of various optical flow techniques:
Barron, J.L., Fleet, D.J., and Beauchemin, S. "***Performance of optical flow techniques***". IJCV, 1994, 12(1):43-77

Layered representation (affine):
James R. Bergen P. Anandan Keith J. Hanna Rajesh Hingorani "***Hierarchical Model-Based Motion Estimation***" ECCV' 92, pp. 237-- 252