Optical flow

Lihi Zelnik-Manor, Computer Vision

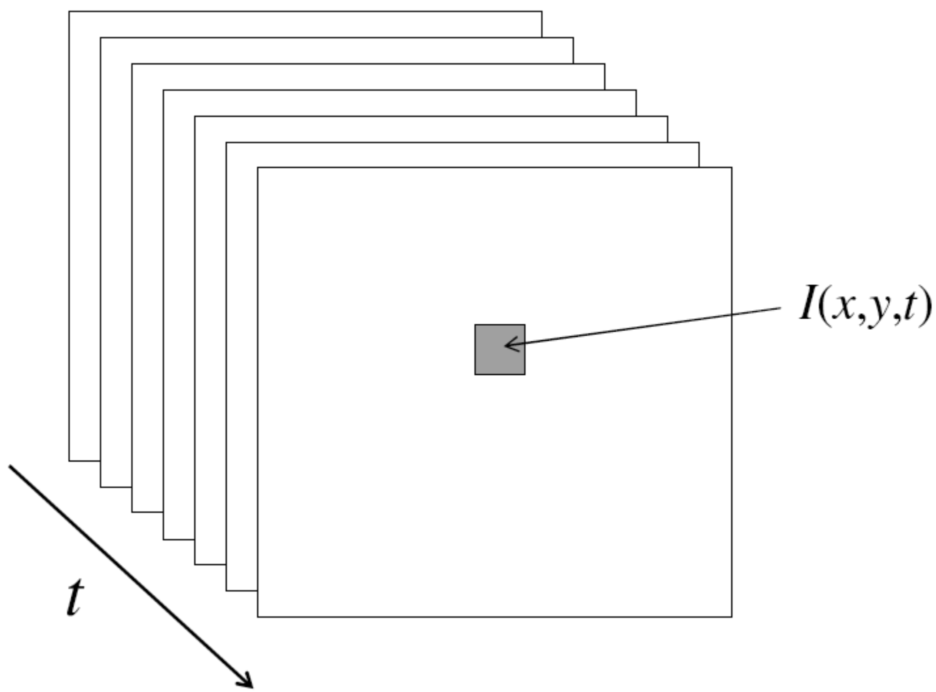
Today

From images to video

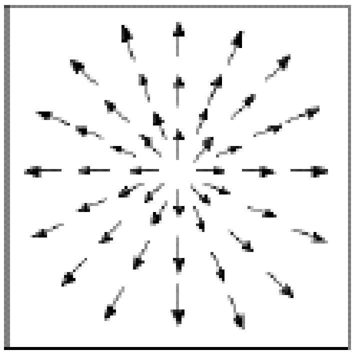
- Feature tracking
- Optical flow
- Motion segmentation
- Applications

From images to video

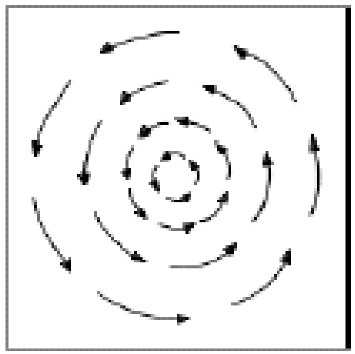
- A video is a sequence of frames captured over time
- Now our image data is a function of space (x,y) and time (t)



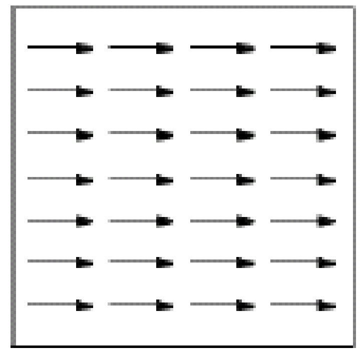
Examples of Motion fields



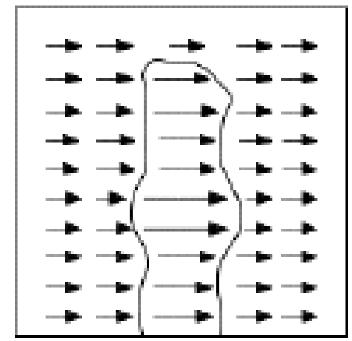
Forward motion



Rotation



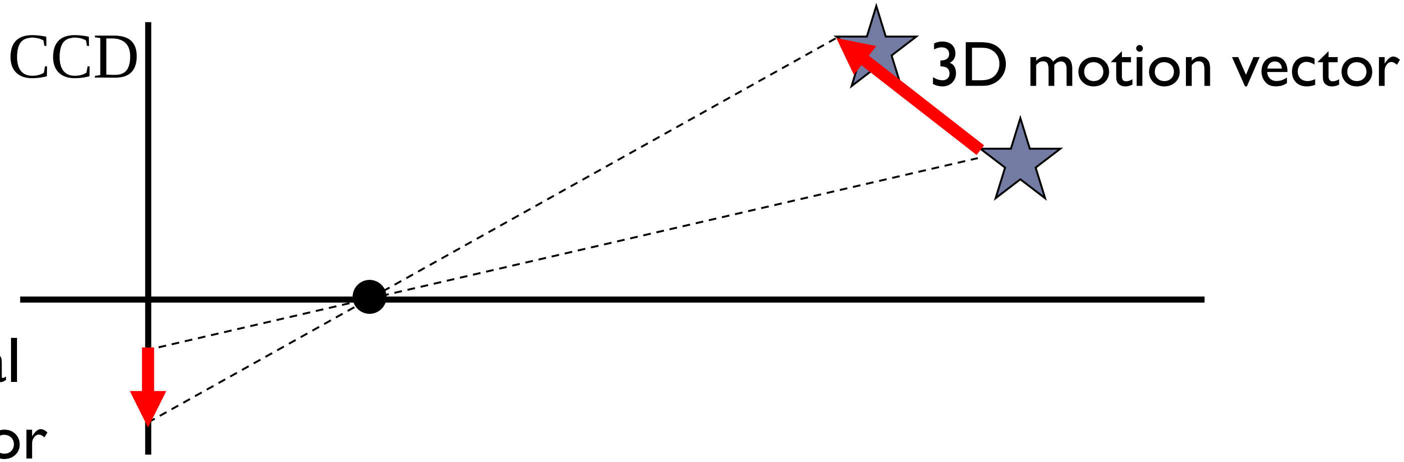
Horizontal translation



Closer objects appear to move faster!!

Motion Field & Optical Flow Field

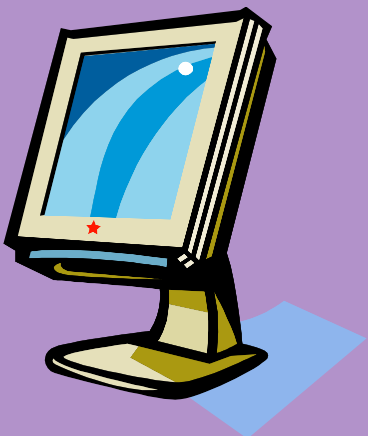
- Underlying assumption:
  The apparent motion field is a projection of the real 3D motion onto the 2d image



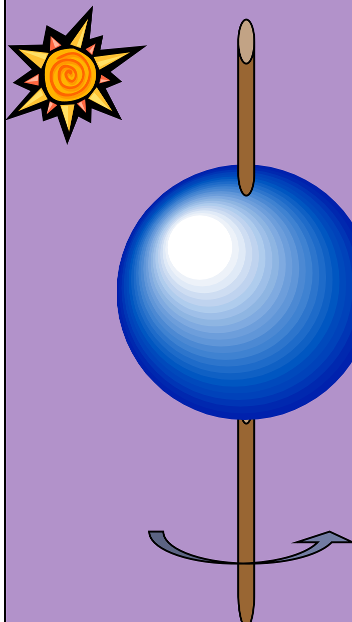
2D optical flow vector

$$\vec{u} = (u, v)$$

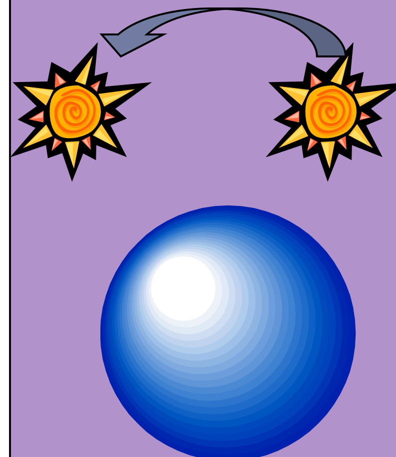
When does it break?



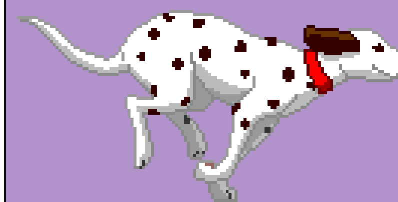
The screen is stationary yet displays motion



Homogeneous objects generate zero optical flow.



Fixed sphere. Changing light source.



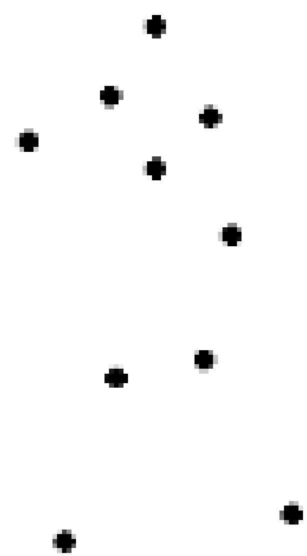
Non-rigid texture motion

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

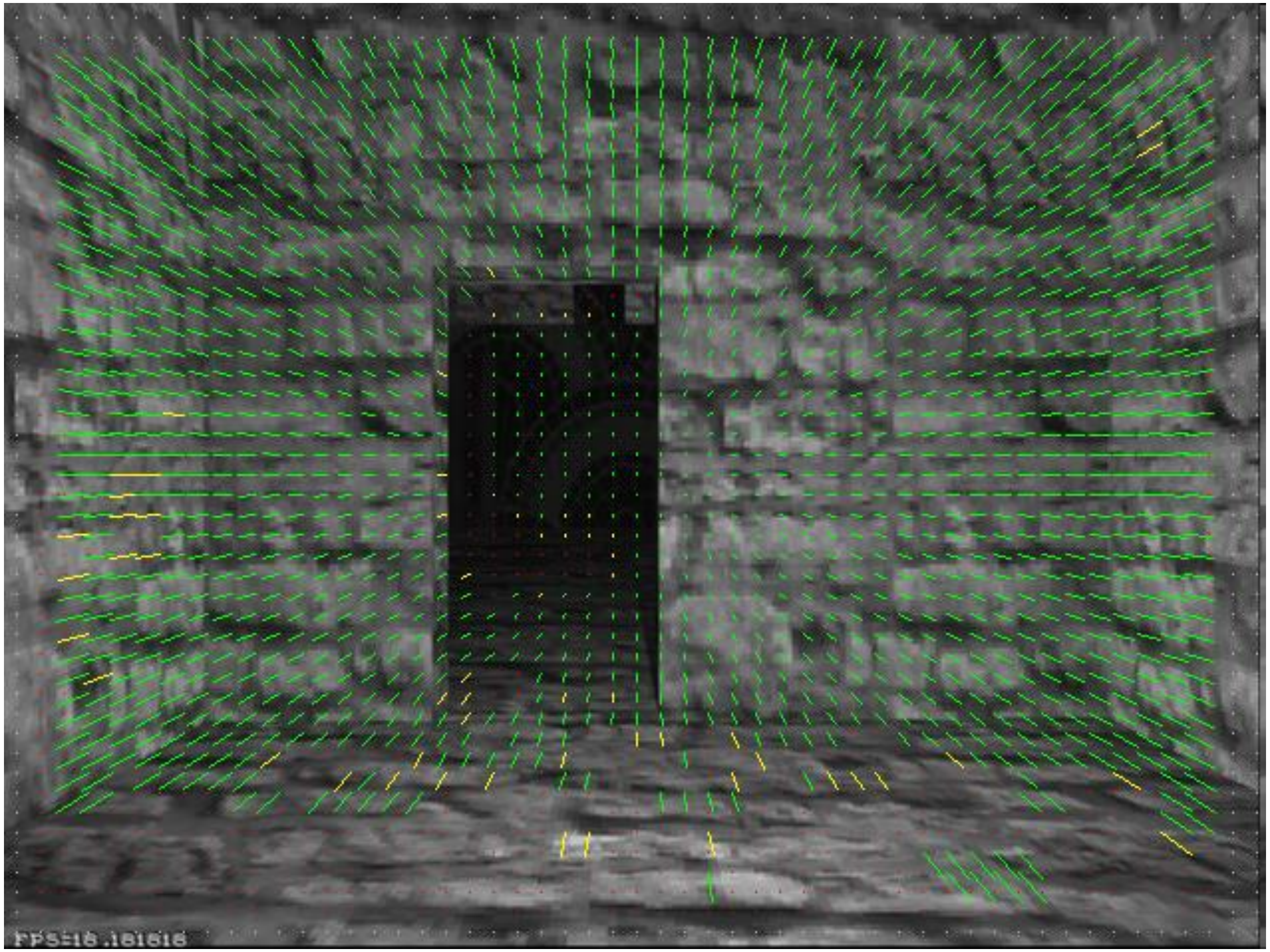


G. Johansson, “Visual Perception of Biological Motion and a Model For Its Analysis", *Perception and Psychophysics 14, 201-211, 1973.*

Tracking example

Optical flow example

- Compute motion for all pixels



Today

From images to video

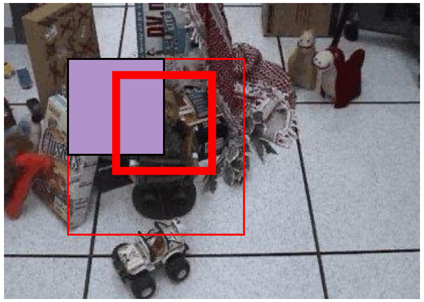
- Feature tracking
- Optical flow
- Motion segmentation
- Applications

Tracking challenges

- Find good features to track
  - Harris, SIFT, etc
- Large motions
  - Discrete search instead of Lucas-Kanade
- 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

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

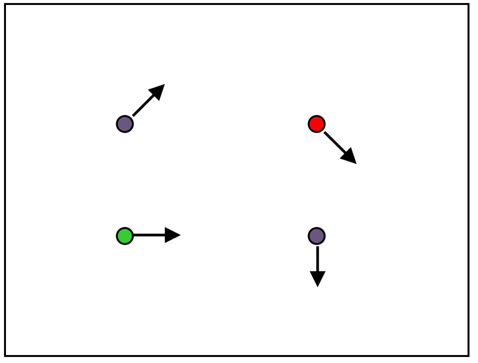


Limitations of template matching

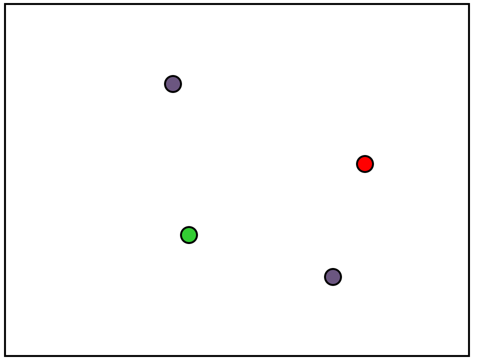
- Slow (need to check more locations)
- Does not give subpixel alignment (or becomes much slower)
  - Even pixel alignment may not be good enough to prevent drift
- May be useful as a step in tracking if there are large movements

The Lucas-Kanade Tracker

Feature tracking



$I(x,y,t)$



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

- Given two subsequent frames, estimate the point translation

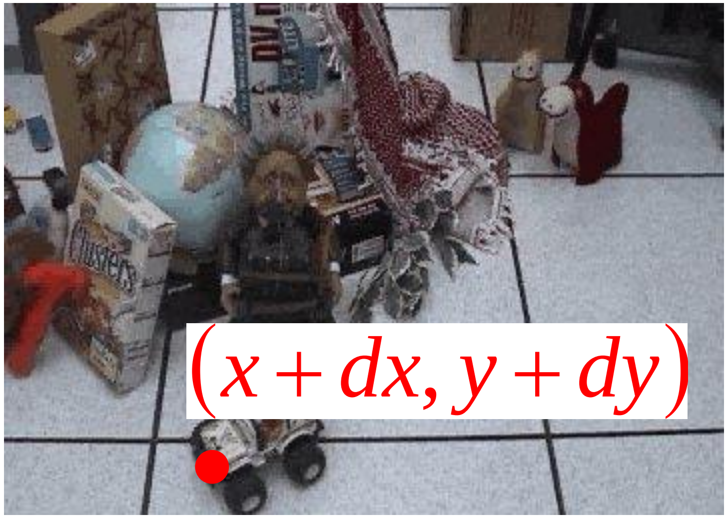
- Key assumptions of Lucas-Kanade Tracker
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far
  - **Spatial coherence:** points move like their neighbors

The brightness constancy constraint

- Assumption 1:
  The image intensity I is constant

Time = t

Time = $t+dt$



$$I(x, y, t) = I(x+dx, y+dy, t+dt)$$

Small motion assumption

- The brightness constancy equation

$$I(x,y,t) = I(x+dx, y+dy, t+dt)$$

Small motion assumption

- The brightness constancy equation

$$I(x,y,t) = I(x+dx, y+dy, t+dt)$$

- Assumption 2
  Motion is small
  First order Taylor expansion

$$I(x,y,t) = I(x,y,t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt$$

$$0 = \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt$$

The motion equation

- Simplify notations: $$I_x dx + I_y dy + I_t dt = 0$$
- Divide by *dt* and denote $$u = \frac{dx}{dt} \quad v = \frac{dy}{dt}$$
- Final equation is: $$\boxed{I_x u + I_y v = -I_t}$$

The motion equation

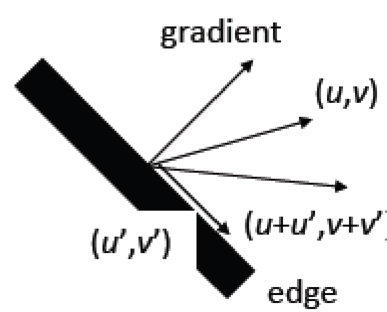
- Can we use this equation to recover image motion at a single pixel (x,y)?

$$I_x u + I_y v = \nabla I \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

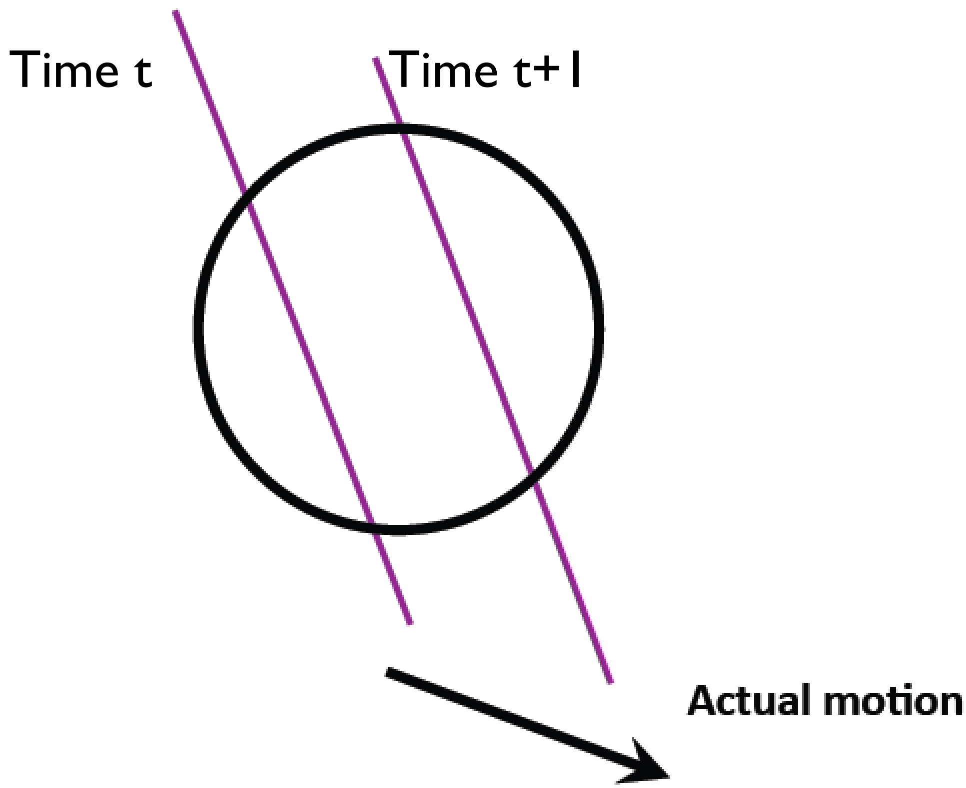
- <u>Problem</u>
  - 1 equation per pixel, 2 unknowns
  - This means we cannot recover the motion component perpendicular to the gradient

If (u, v) satisfies the equation,
so does ($u+u'$, $v+v'$) if

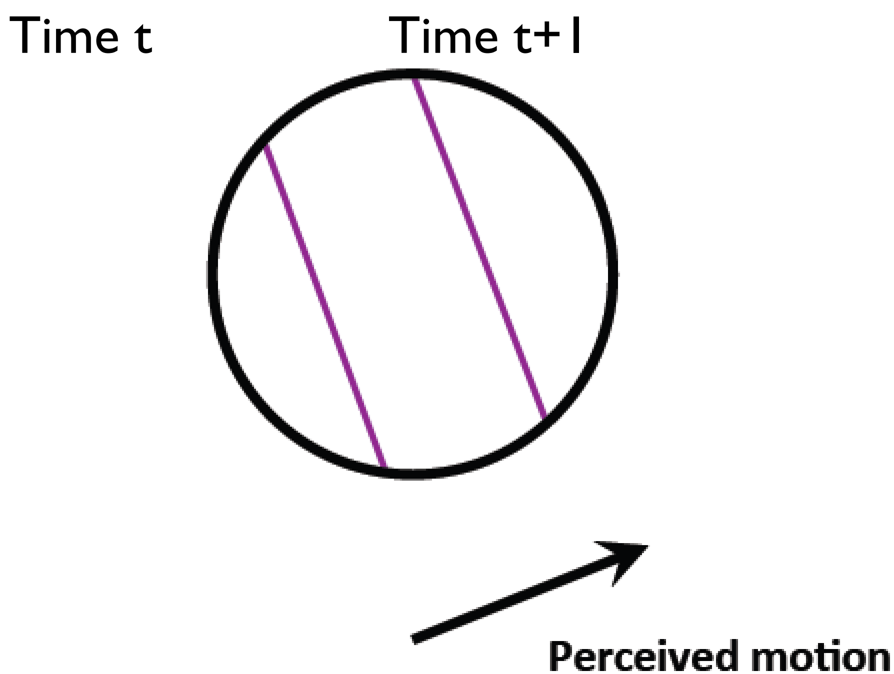
$$\nabla \mathrm{I} \cdot [\mathrm{u}' \ \ \mathrm{v}']^{\mathrm{T}} = 0$$



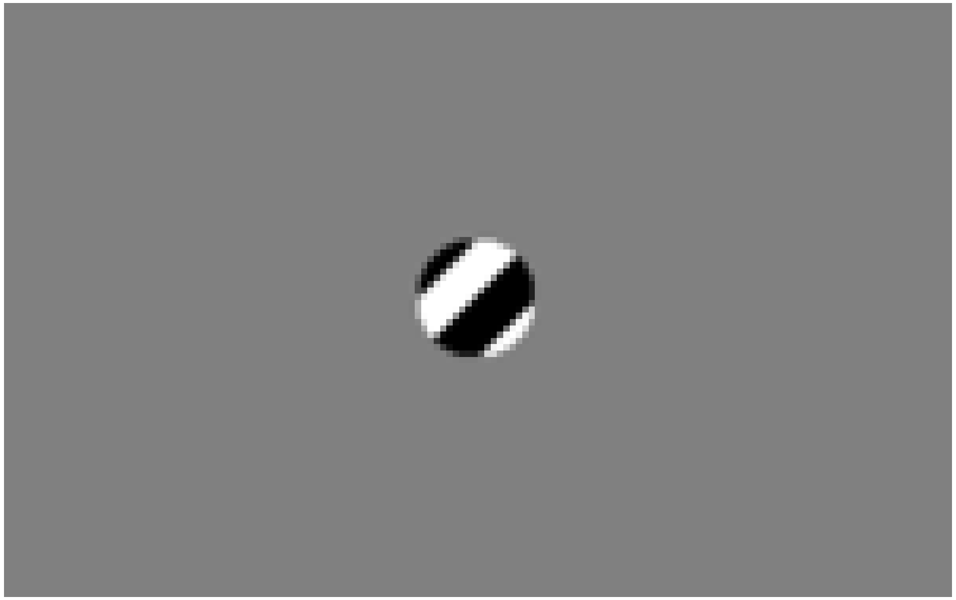
The aperture problem



The aperture problem

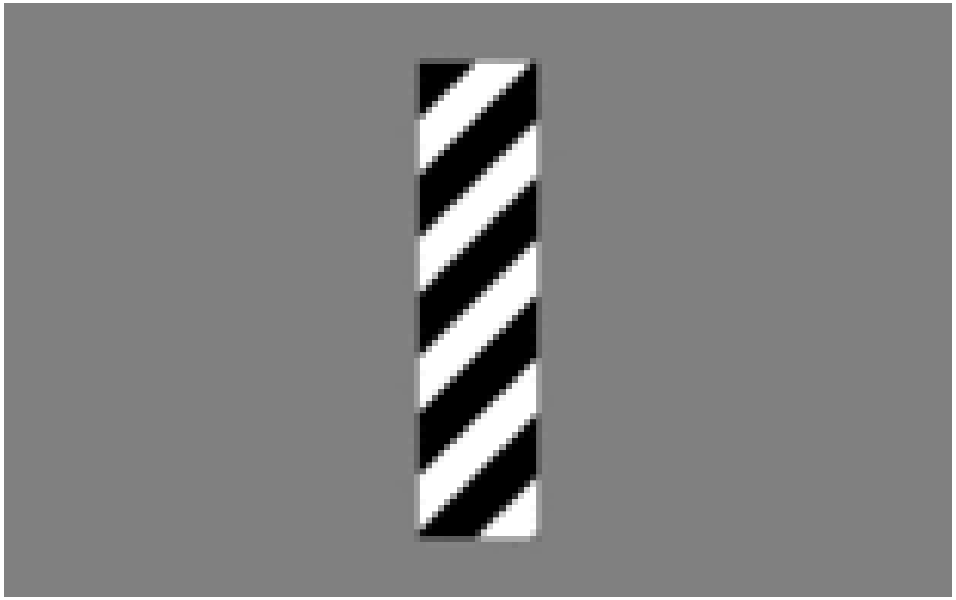


The barber pole illusion



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

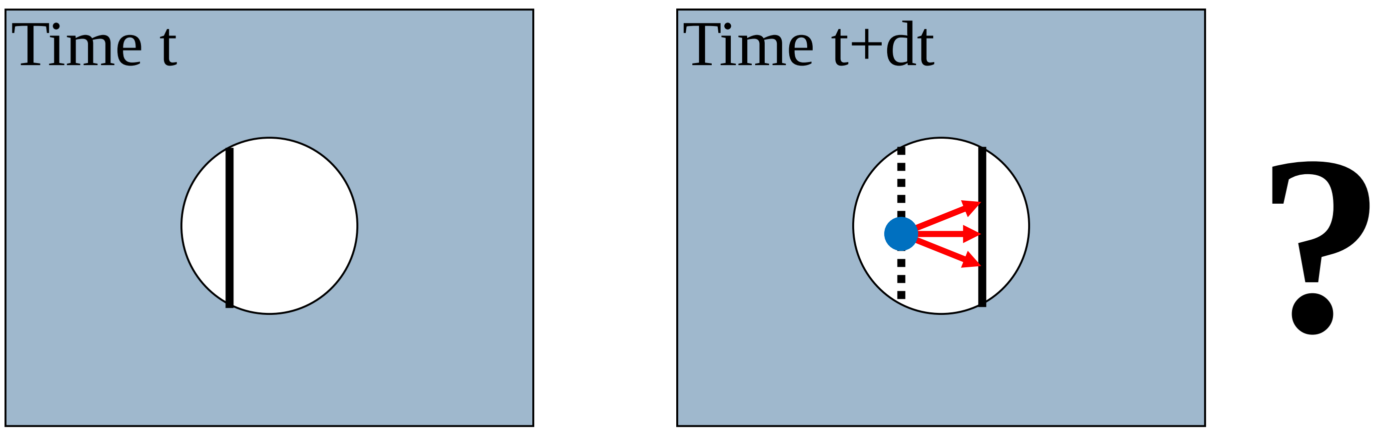
The barber pole illusion



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

The aperture problem

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

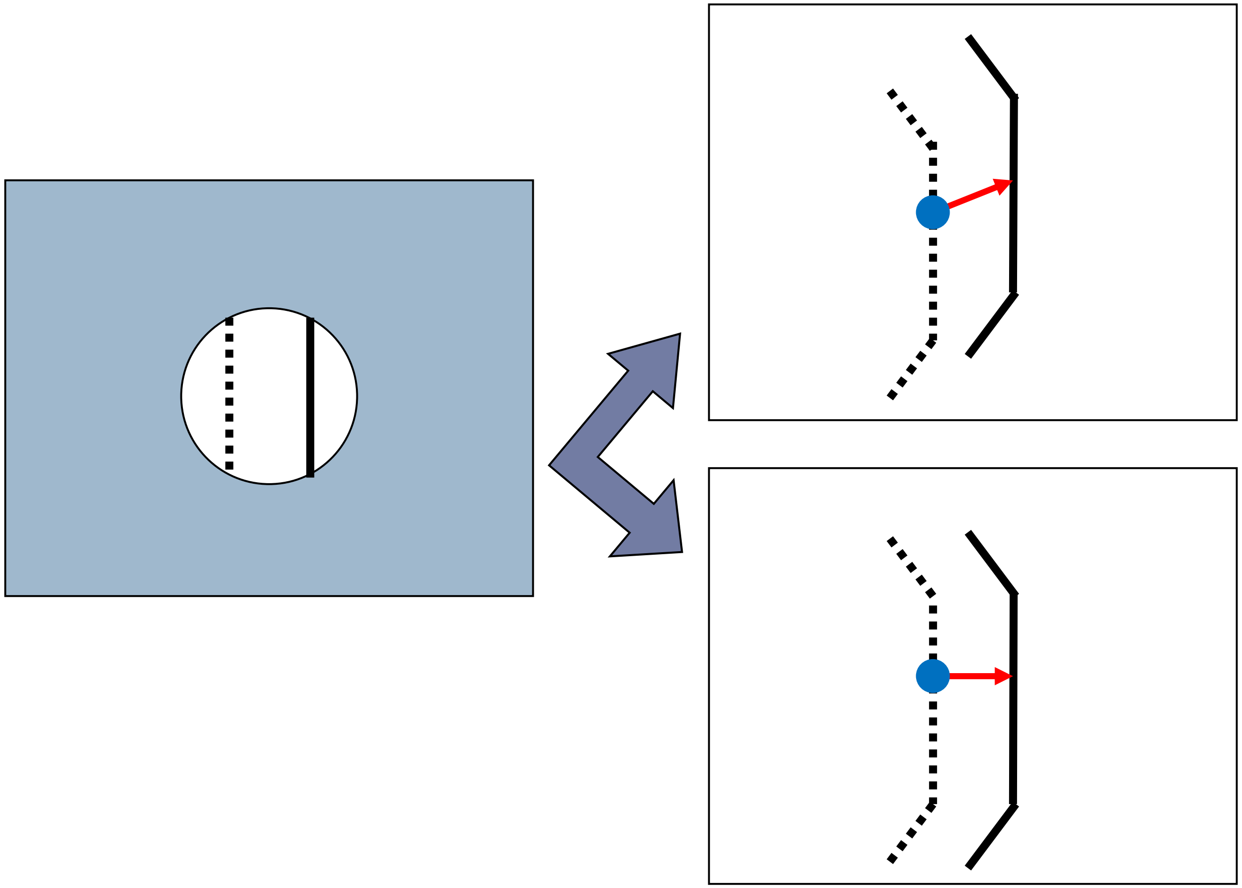


Where did the blue point move to?

We need additional constraints

Solving the ambiguity

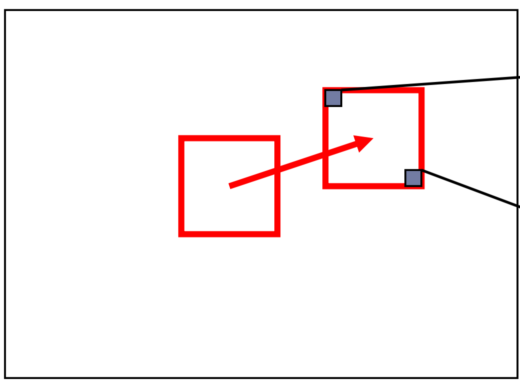
Sometimes enlarging the aperture can help



Spatial coherence assumption

- Assumption 3 [Lucas & Kanade 1981]
  Assume constant (u,v) in small neighborhood

$$I_x u + I_y v = -I_t \quad \Rightarrow \quad \begin{bmatrix} I_x & I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$



$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} I_{t1} \\ I_{t2} \\ \vdots \end{bmatrix}$$

$$\boxed{A\vec{u} = b}$$

Lucas Kanade (1984)

Goal: Minimize $\left\| A\vec{u} - b \right\|^2$

Method: Least-Squares

$$A\vec{u} = b$$

$$\Downarrow$$

$$\underbrace{A^T A}_{2x2} \underbrace{\vec{u}}_{2x1} = \underbrace{A^T b}_{2x1}$$

$$\Downarrow$$

$$\boxed{\vec{u} = \left(A^T A\right)^{-1} A^T b}$$

When is this solvable?

$$\boxed{\vec{\mathrm{u}} = \left(A^T A\right)^{-1} A^T b}$$

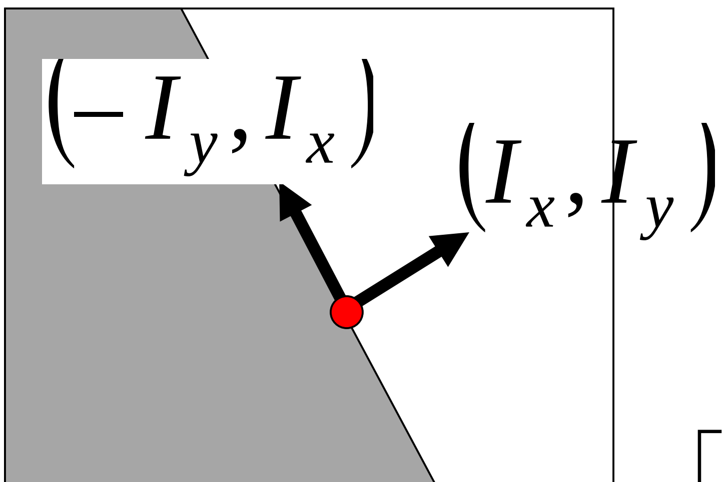
$$A^T A = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

We want this matrix to be invertible ➔

no zero eigenvalues

When is this solvable?

- Edge ➔ $A^T A$ becomes singular



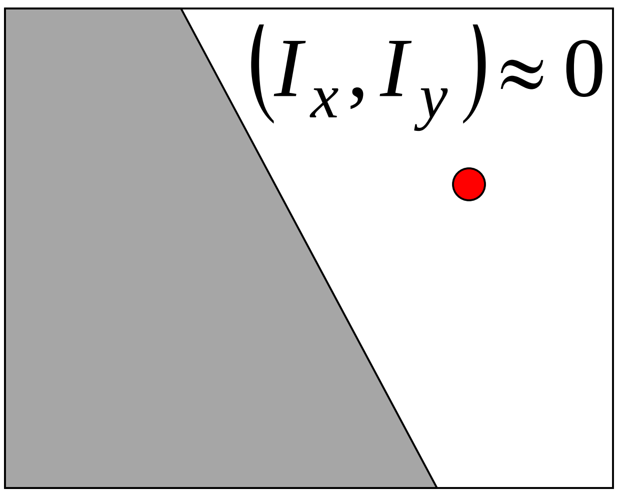
$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} -I_y \\ I_x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Downarrow$$

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

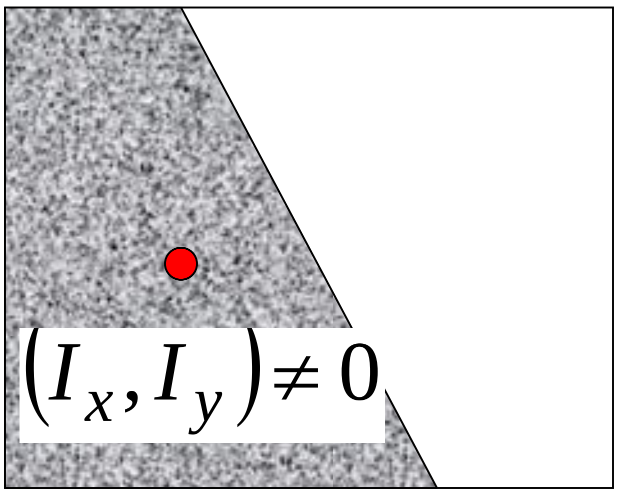
When is this solvable?

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



When is this solvable?

- Textured regions ➔ two high eigenvalues

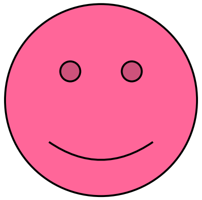


Which features can we track?

- Edge ➔ $A^T A$ becomes singular

- Homogeneous regions ➔ low gradients

$$A^T A \approx 0$$

- High texture ➔ 

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

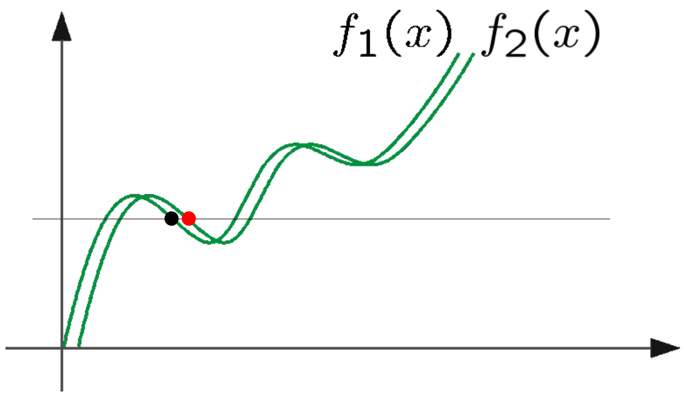
- The motion is **not** small (Taylor expansion doesn't hold)
- Aliasing

Use multi-scale estimation

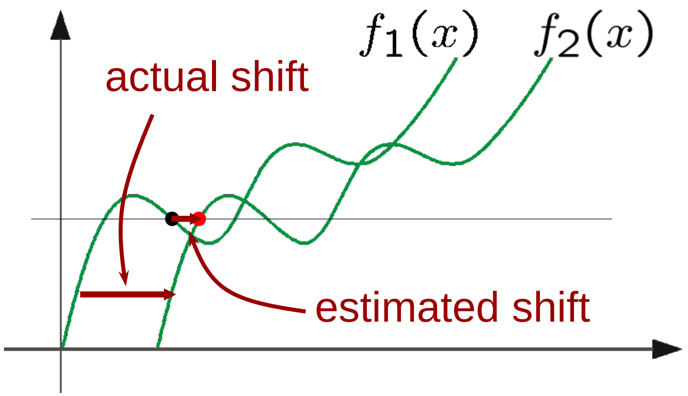
Aliasing

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?



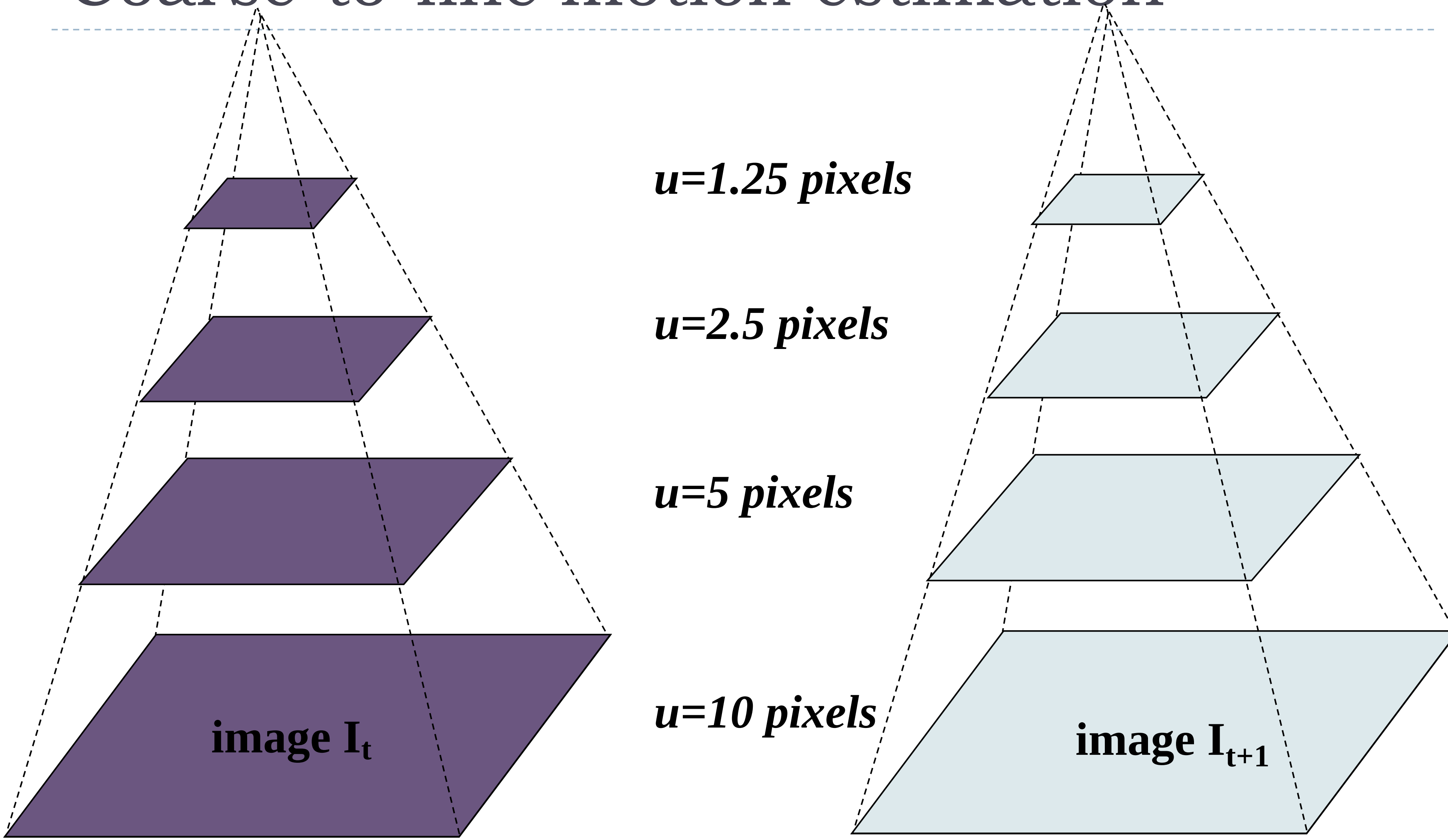
nearest match is correct (no aliasing)



nearest match is incorrect (aliasing)

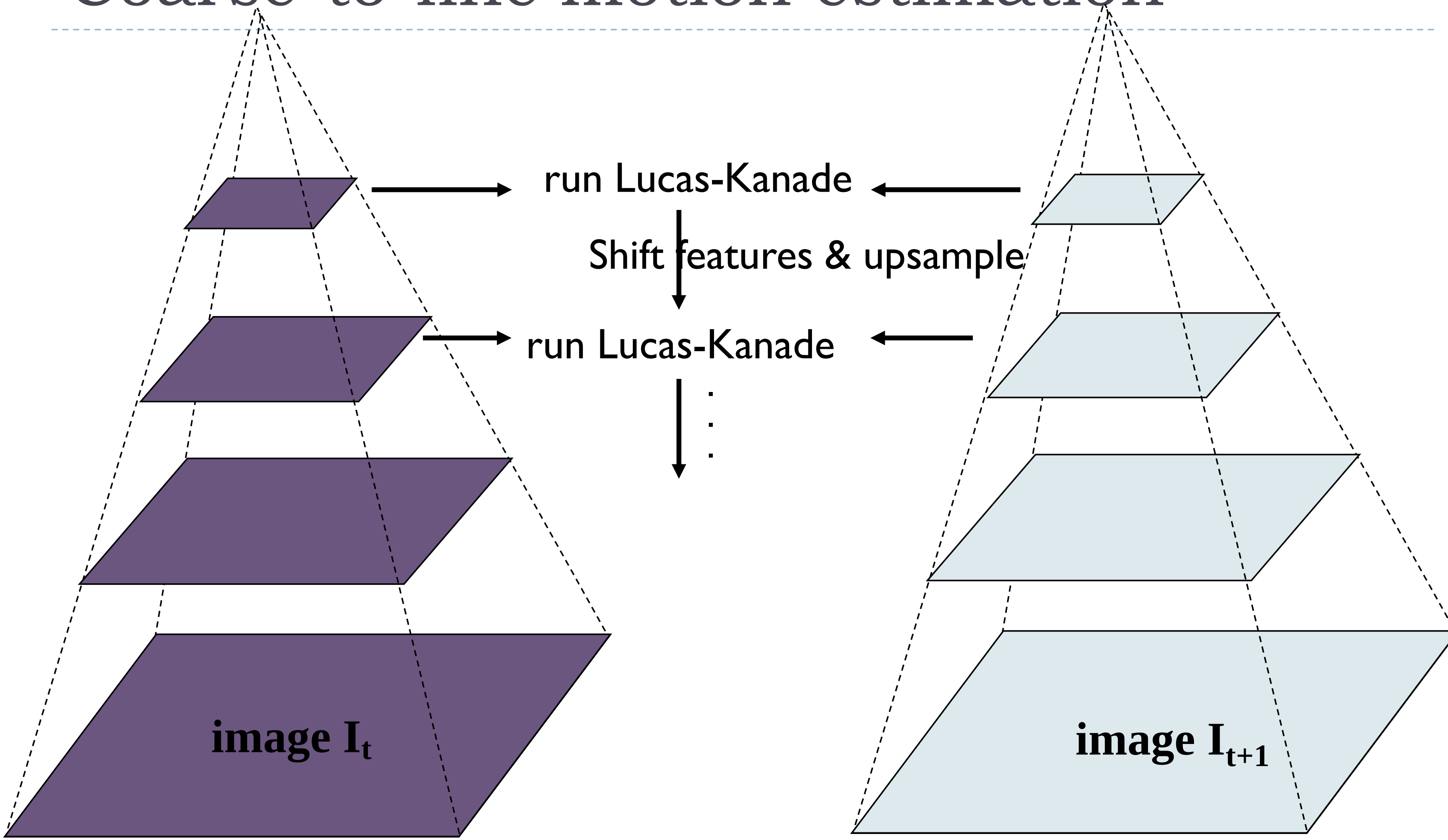
To overcome aliasing: coarse-to-fine estimation.

Coarse-to-fine motion estimation



Gaussian pyramid of image I_t **Gaussian pyramid of image I_{t+1}**

Coarse-to-fine motion estimation



Gaussian pyramid of image I_t **Gaussian pyramid of image I_{t+1}**

Shi-Tomasi feature tracker

1. Find good features (min eigenvalue of 2×2 Hessian)
2. Use Lucas-Kanade to track with pure translation
3. Use affine registration with first feature patch
4. Terminate tracks whose dissimilarity gets too large
5. Start new tracks when needed

[Shi & Tomasi, Good features to track, CVPR'94]
http://www.ces.clemson.edu/~stb/klt/

Tracking example

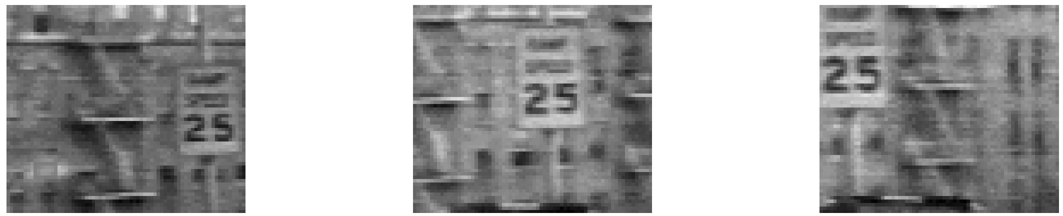


Figure 1: Three frame details from Woody Allen's *Manhattan*. The details are from the 1st, 11th, and 21st frames of a subsequence from the movie.

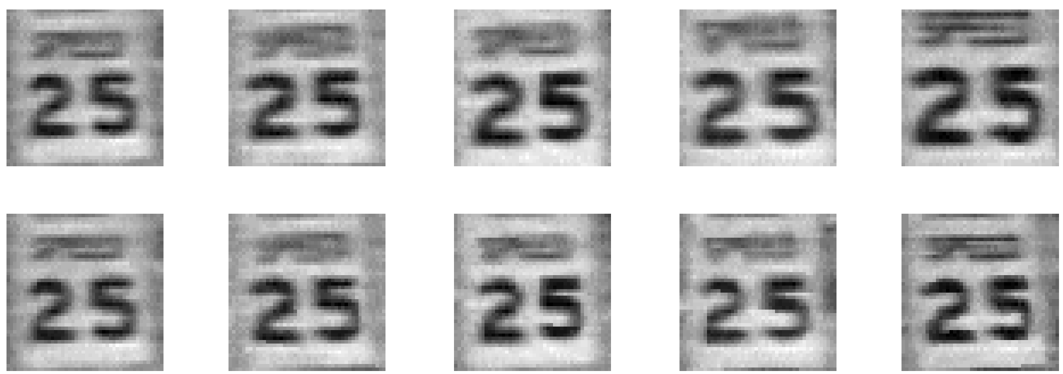


Figure 2: The traffic sign windows from frames 1,6,11,16,21 as tracked (top), and warped by the computed deformation matrices (bottom).

J. Shi and C. Tomasi. Good Features to Track. CVPR 1994.

Tracking example

Implementation issues

- Window size
  - Small window more sensitive to noise and may miss larger motions (without pyramid)
  - Large window more likely to cross an occlusion boundary (and it's slower)
  - 15x15 to 31x31 seems typical
- Weighting the window
  - Common to apply weights so that center matters more (e.g., with Gaussian)

Today

From images to video

- Feature tracking
- Optical flow
- Motion segmentation
- Applications

The Optical Flow Field

What can be done when we need to find the motion of each and every pixel?

Lucas-Kanade Optical Flow

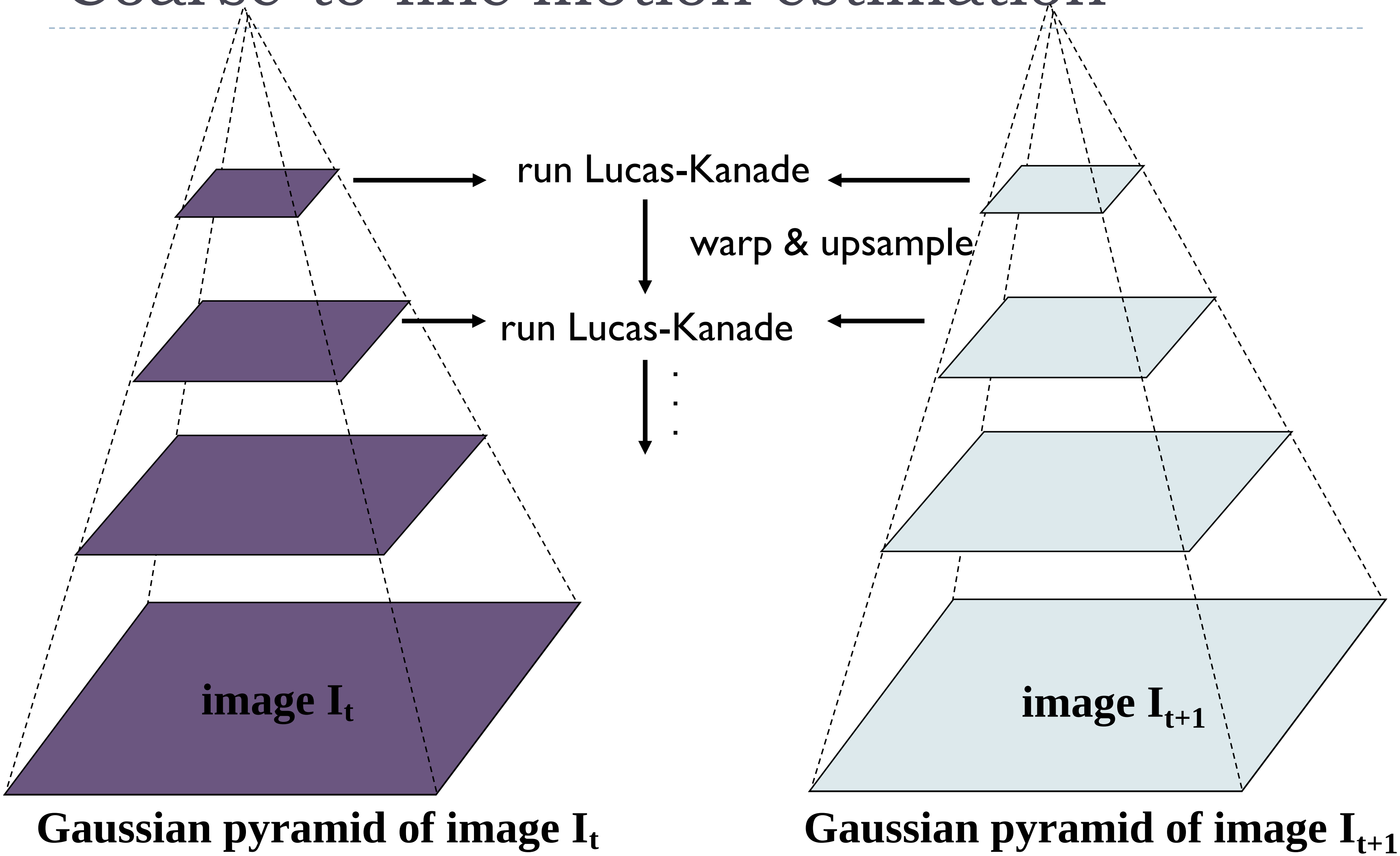
- Same as Lucas-Kanade feature tracking, but for each pixel
  - As we saw, works better for textured pixels
- Operations can be done one frame at a time, rather than pixel by pixel
  - Efficient

Iterative Refinement

- Iterative Lukas-Kanade Algorithm
  1. Estimate displacement at each pixel by solving Lucas-Kanade equations
  2. Warp I(t) towards I(t+1) using the estimated flow field
     - - Basically, just interpolation
  3. Repeat until convergence

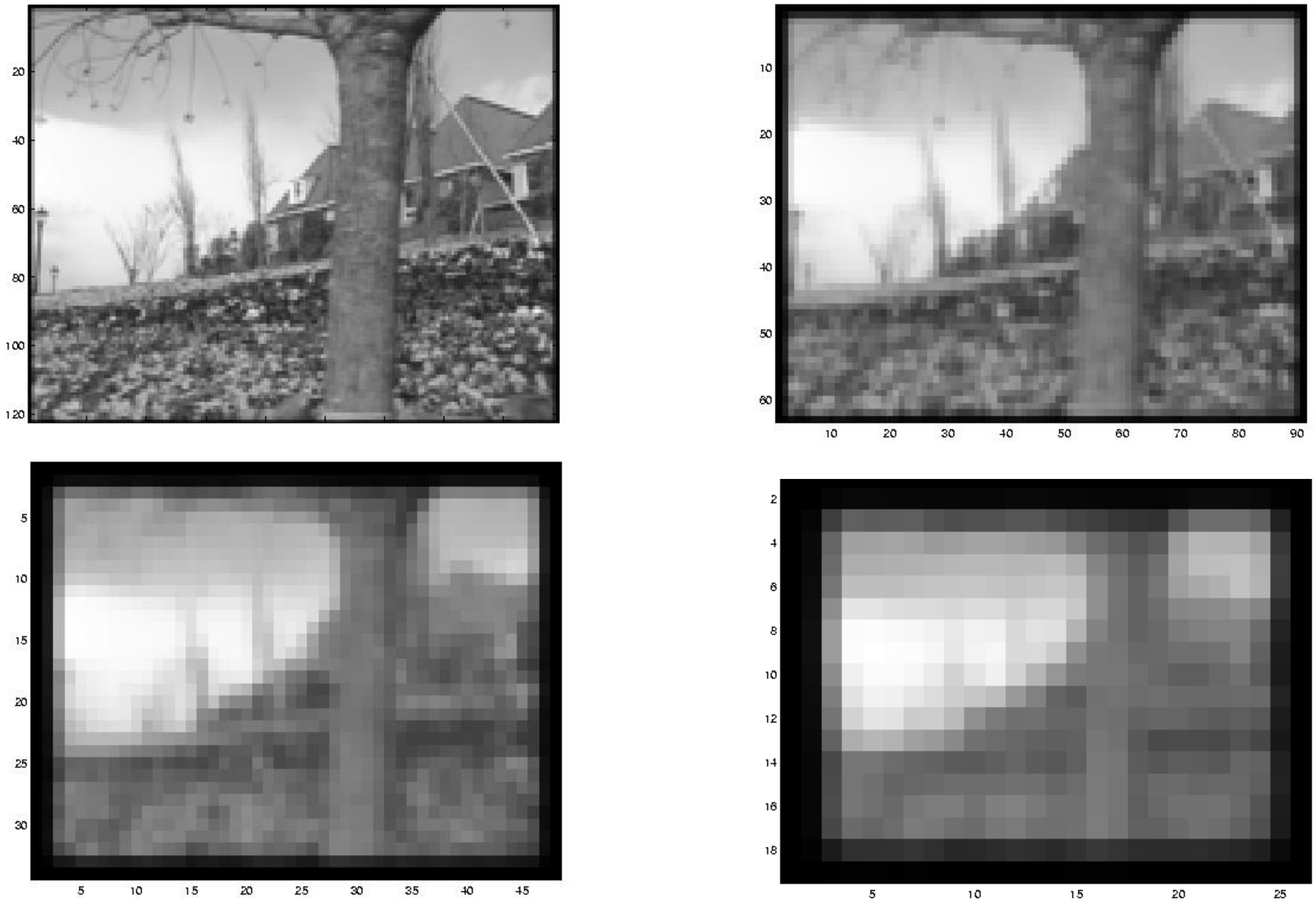
* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Coarse-to-fine motion estimation

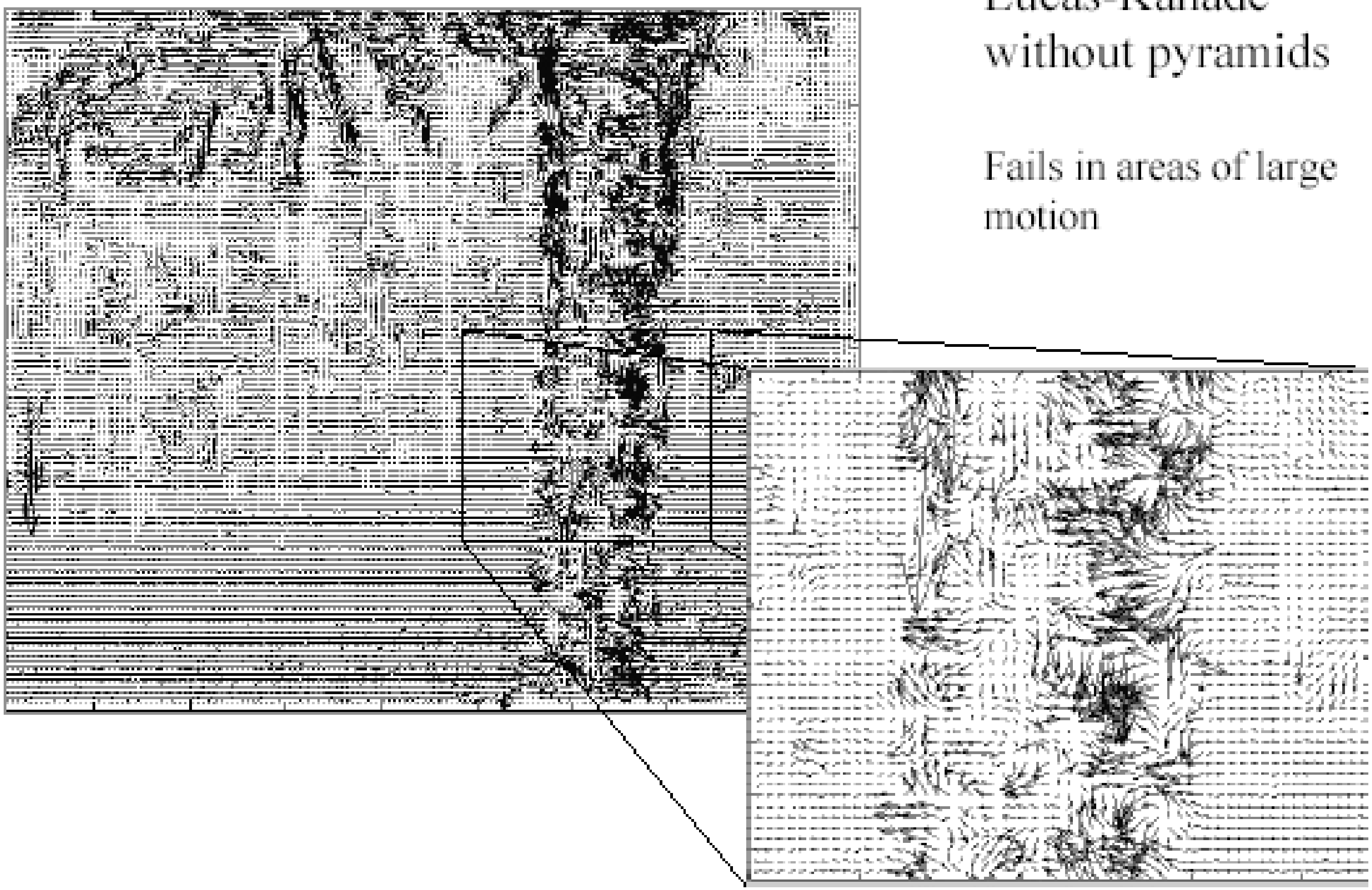


Example

Multi-resolution registration



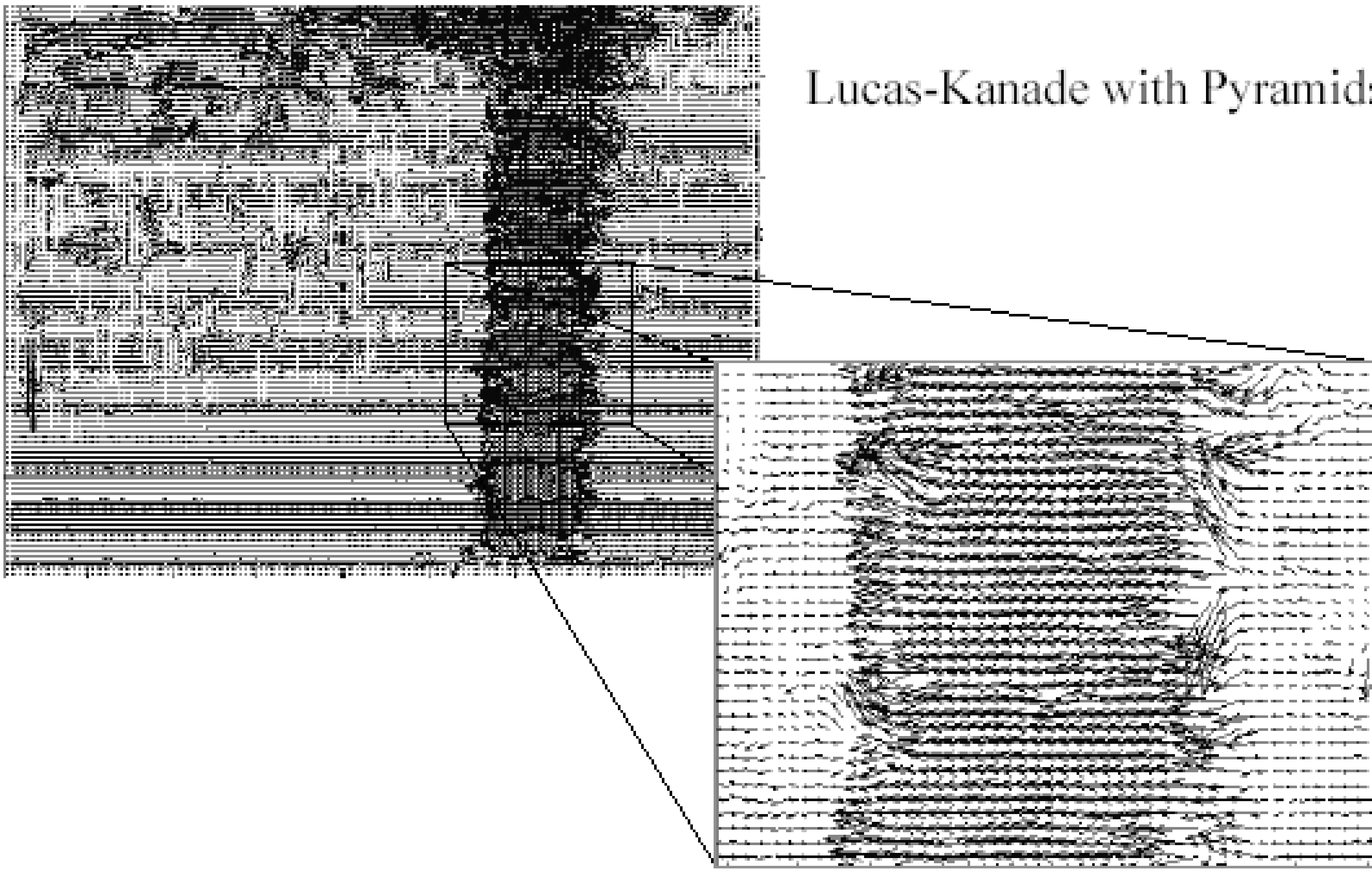
Optical flow results



Lucas-Kanade without pyramids

Fails in areas of large motion

Optical Flow Results



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?

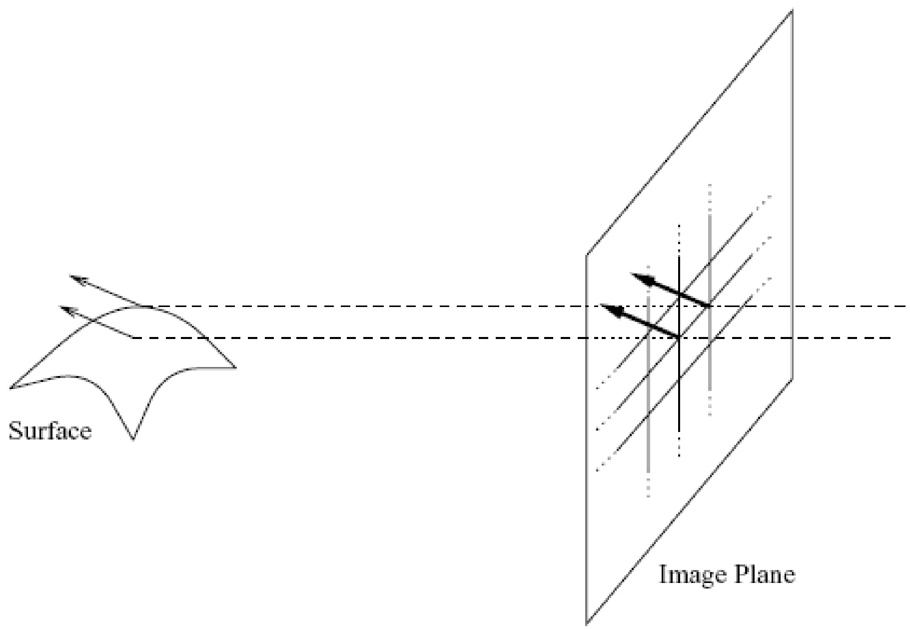
  → Regularization based methods

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

  → Use multi-scale estimation

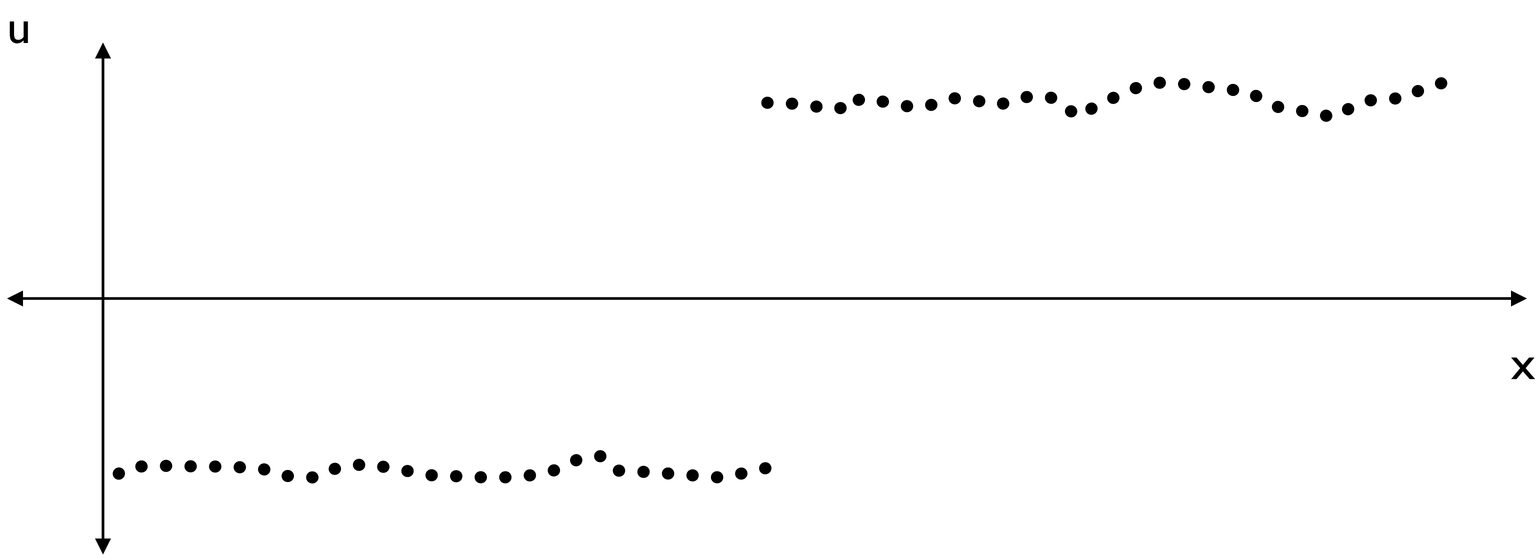
Spatial coherence

- 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

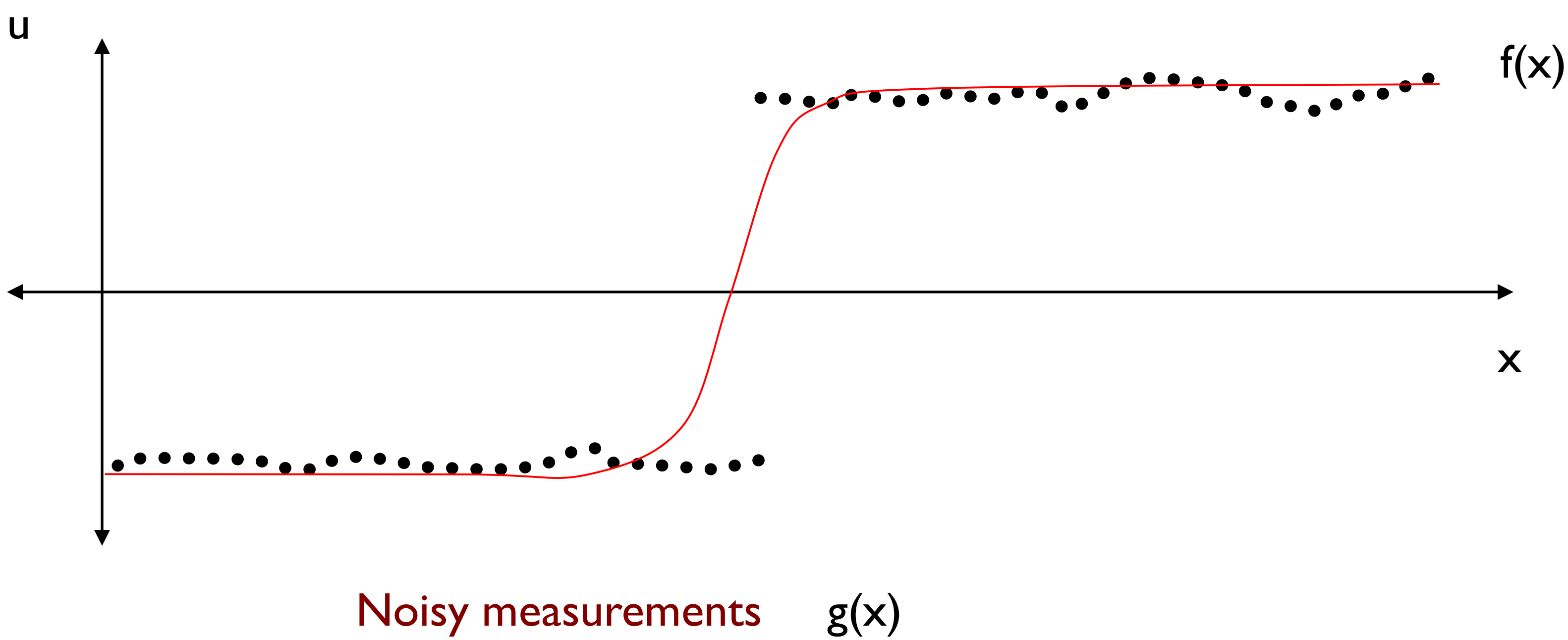
Noisy 1D signal:



Noisy measurements g(x)

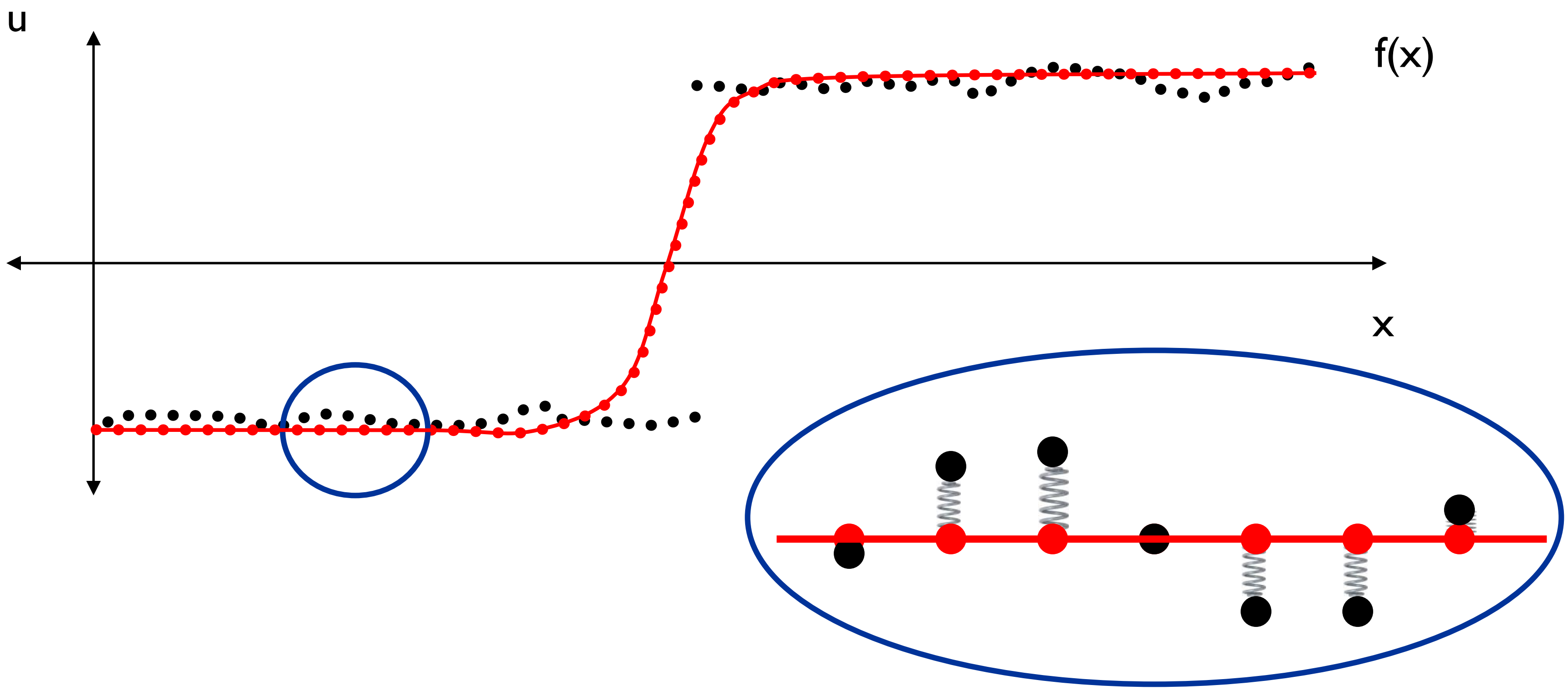
Regularization

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



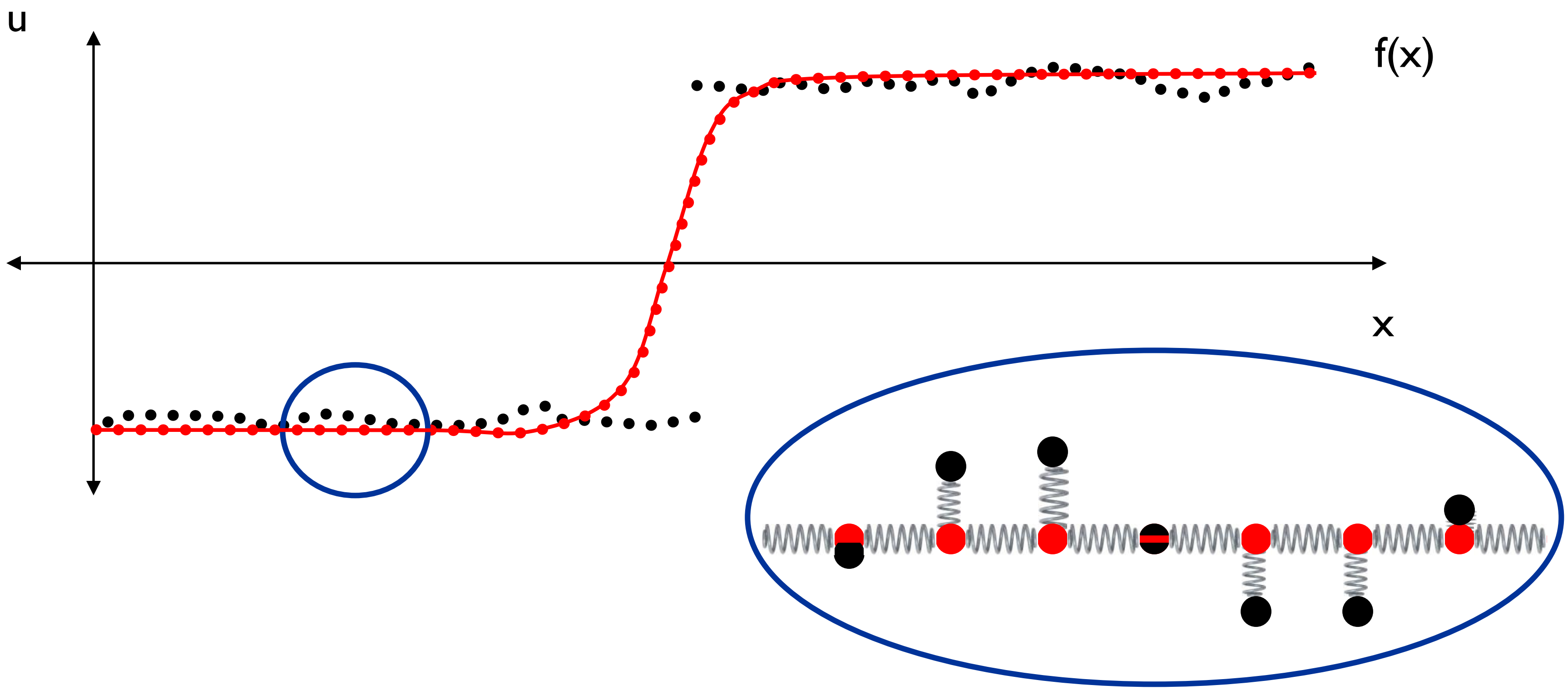
Membrane model

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



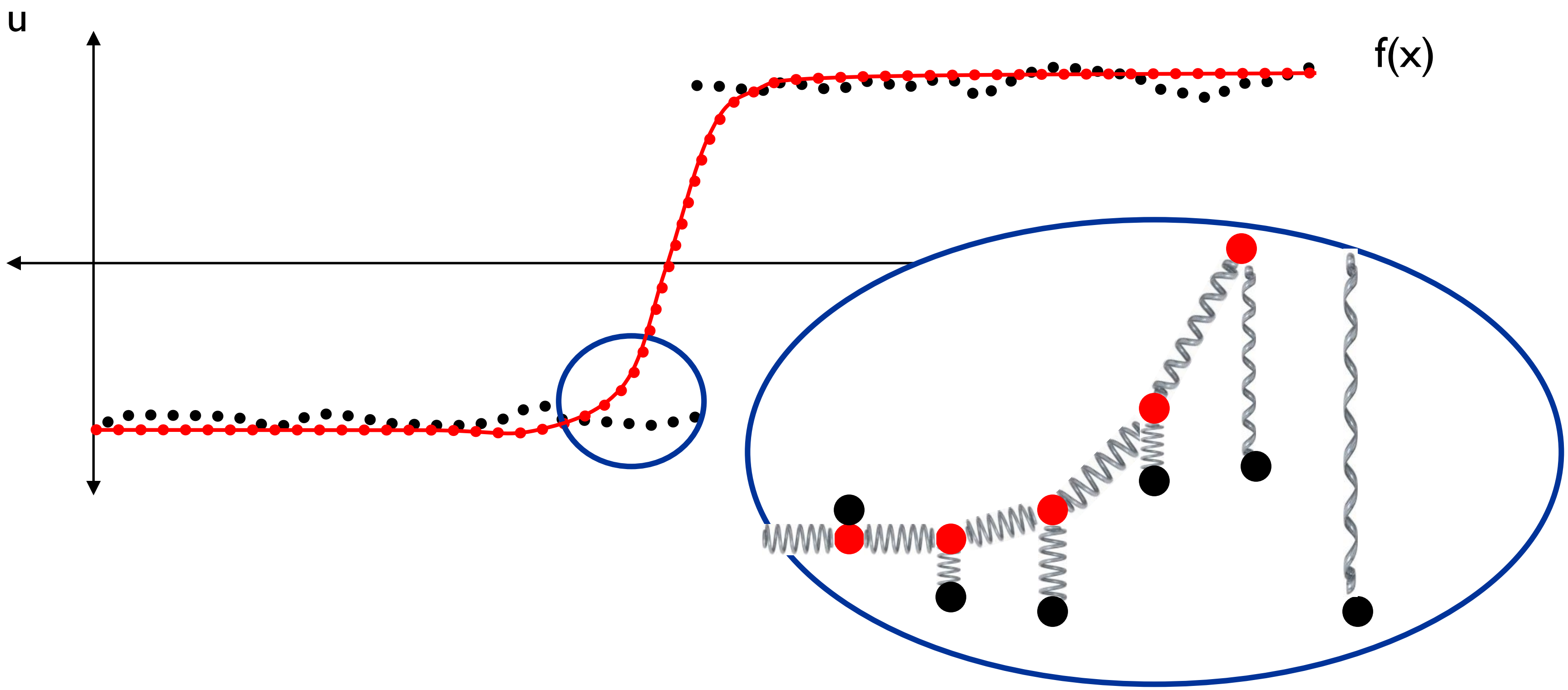
Membrane model

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

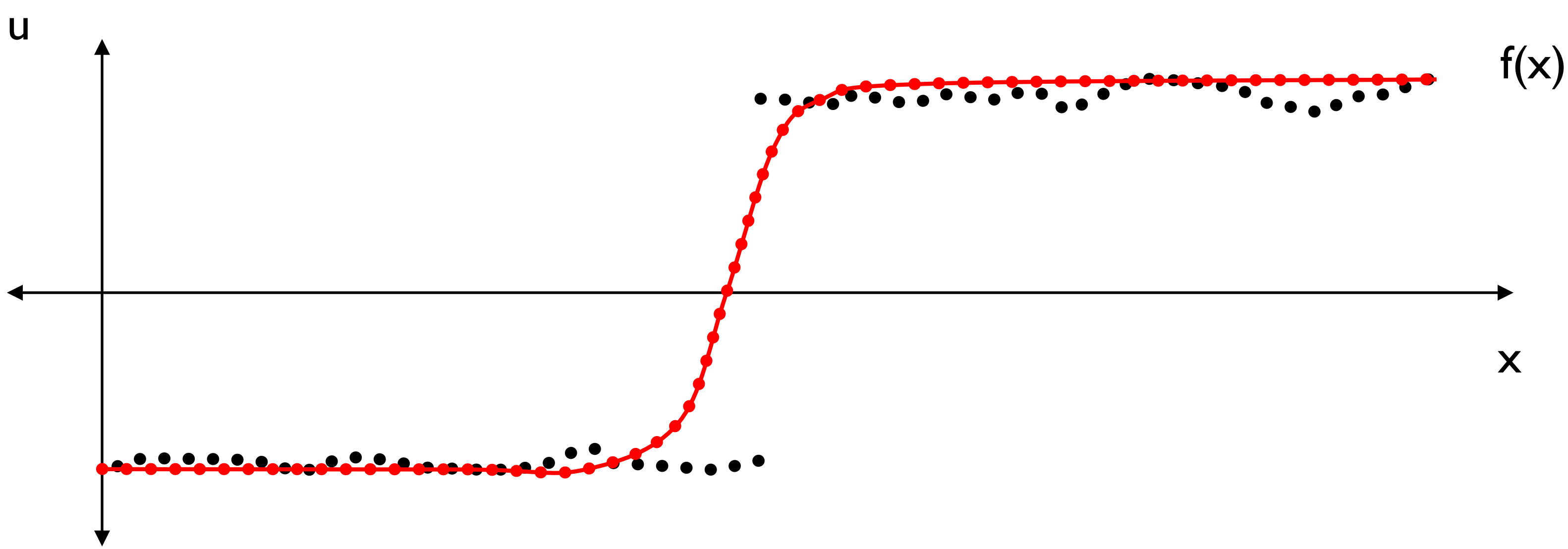


Membrane model

Find the “best fitting” smoothed function f(x)



Regularization



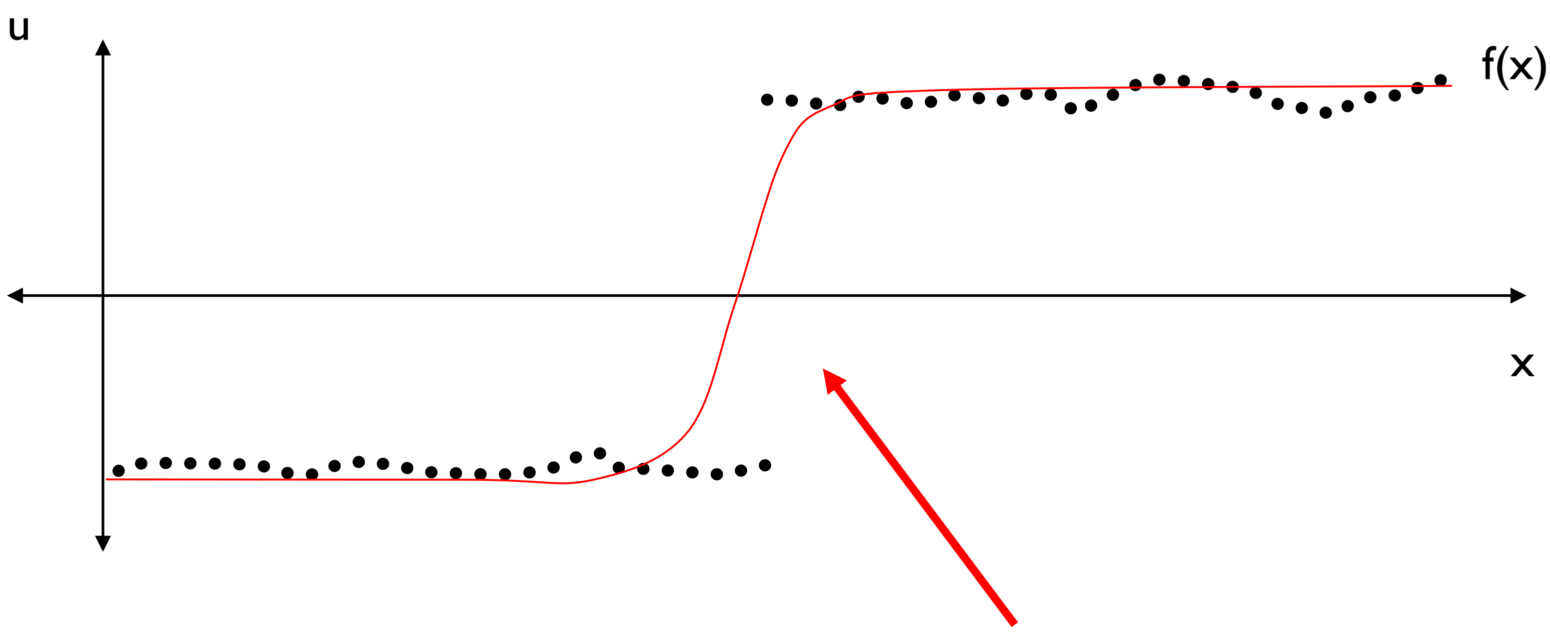
Minimize:

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

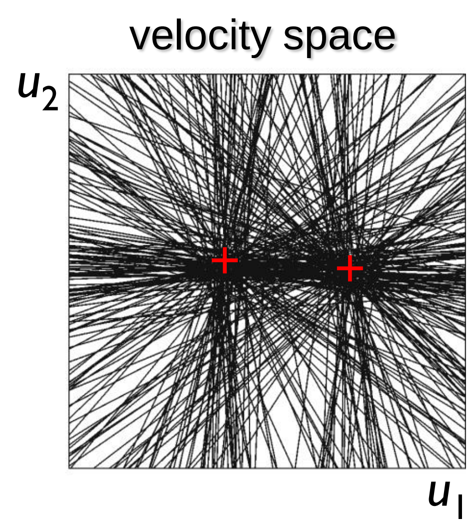


What about this discontinuity?
What is happening here?
What can we do?

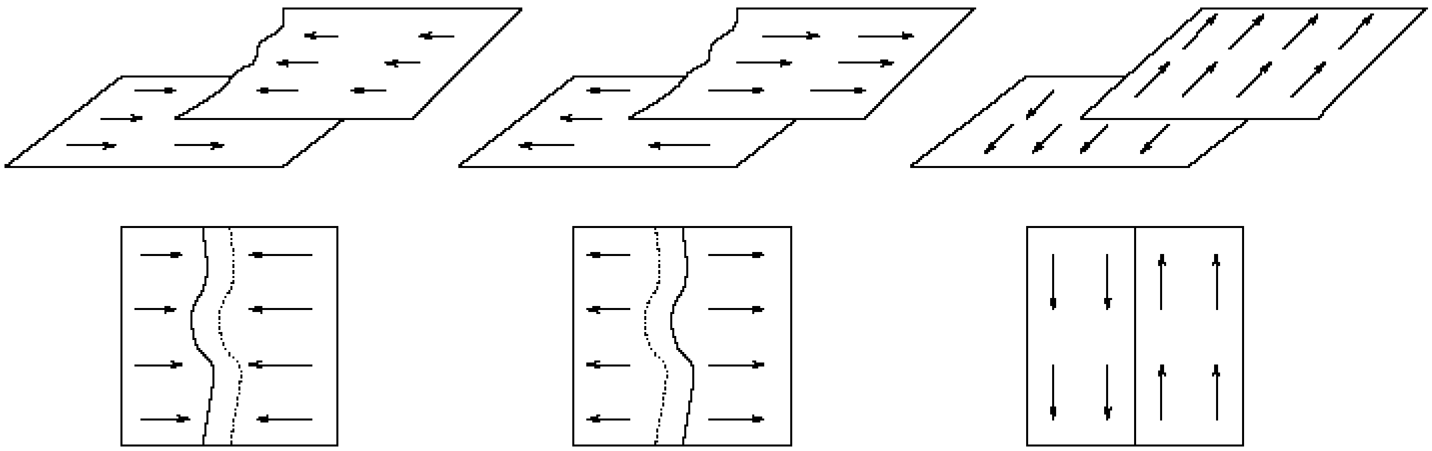
Robust estimation

Noise distributions are often non-Gaussian, having much heavier tails. Noise samples from the tails are called outliers.

- Sources of outliers (multiple motions):
  - specularities / highlights
  - jpeg artifacts / interlacing / motion blur
  - multiple motions (occlusion boundaries, transparency)



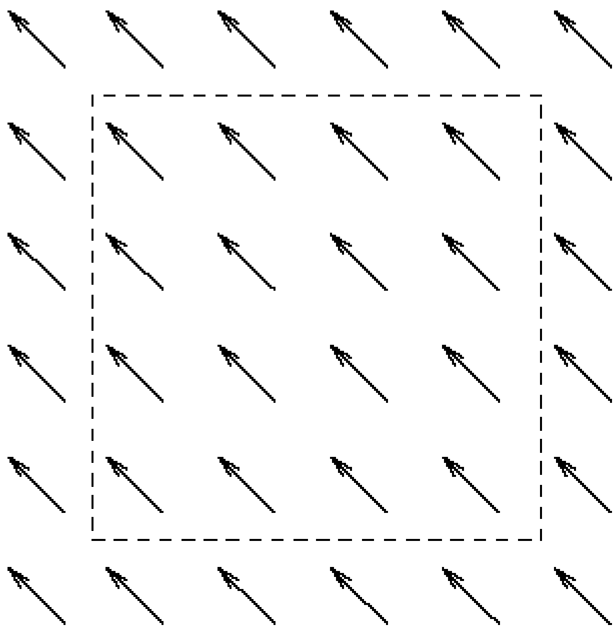
Occlusion

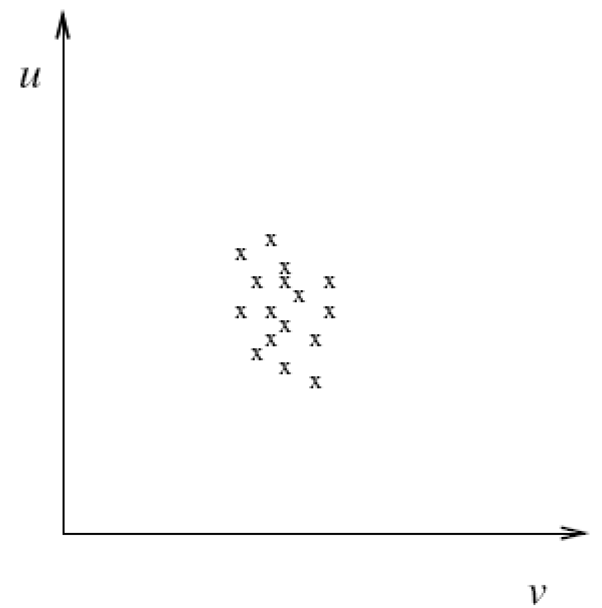


occlusion disocclusion shear

Multiple motions within a finite region.

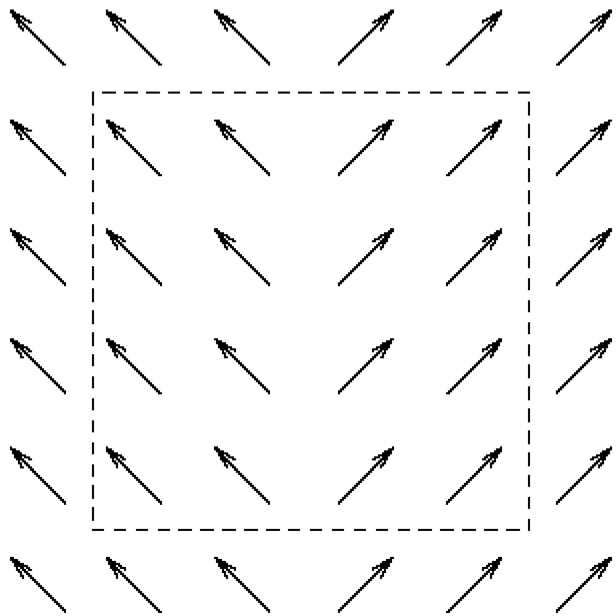
Coherent motion

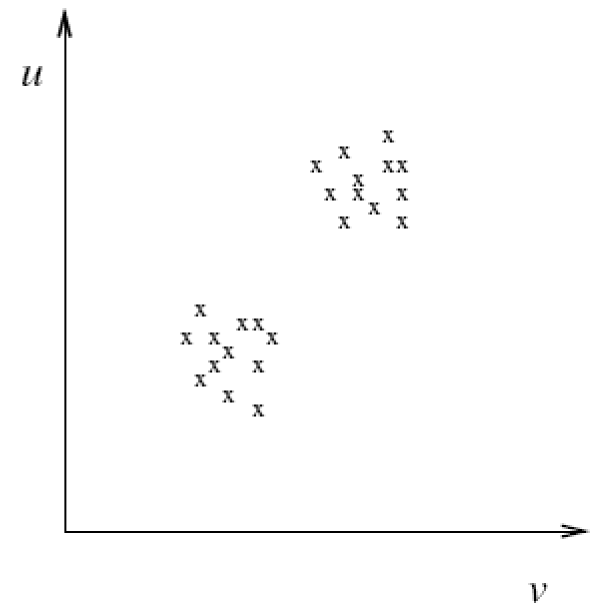




Possibly Gaussian.

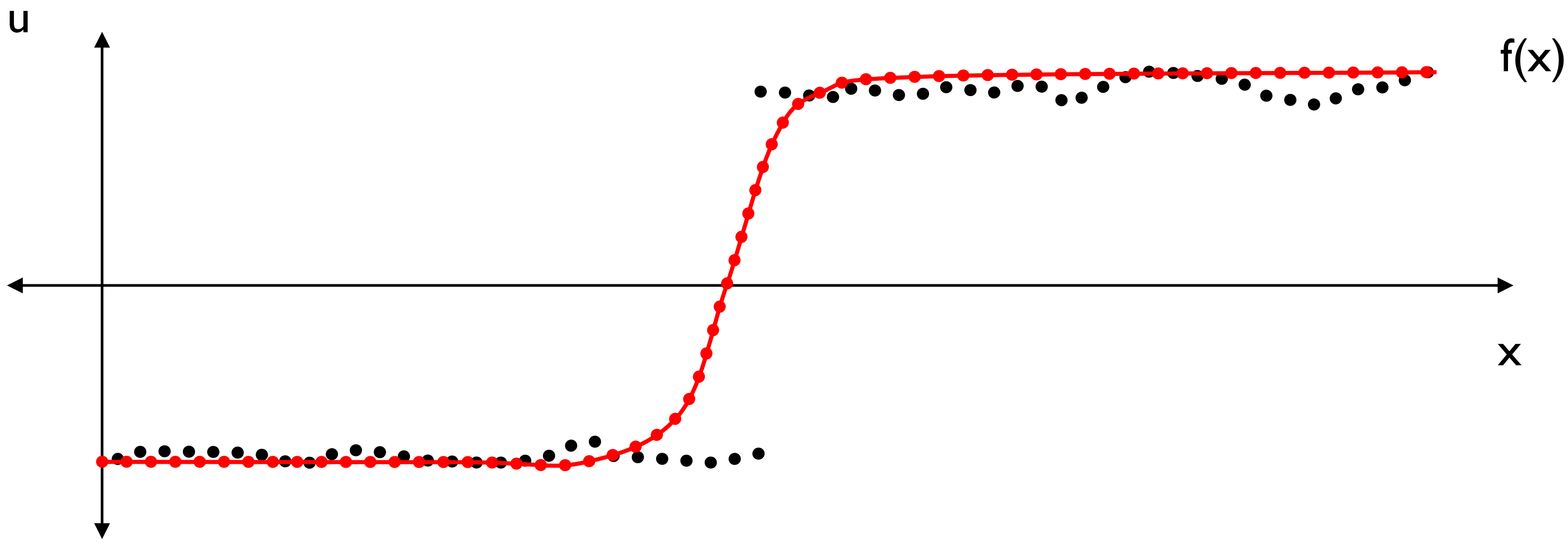
Multiple motions





Definitely not Gaussian.

Regularization

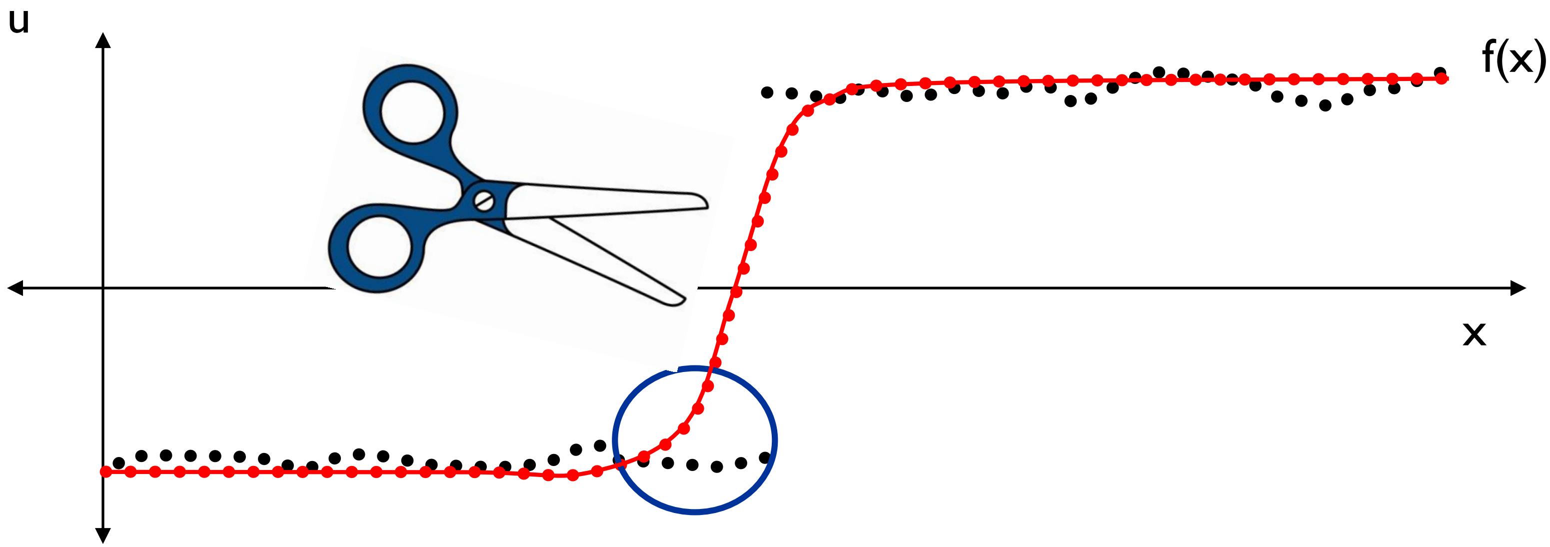


Faithful to the data

Spatial smoothness assumption

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

Weak membrane model



Faithful to the data

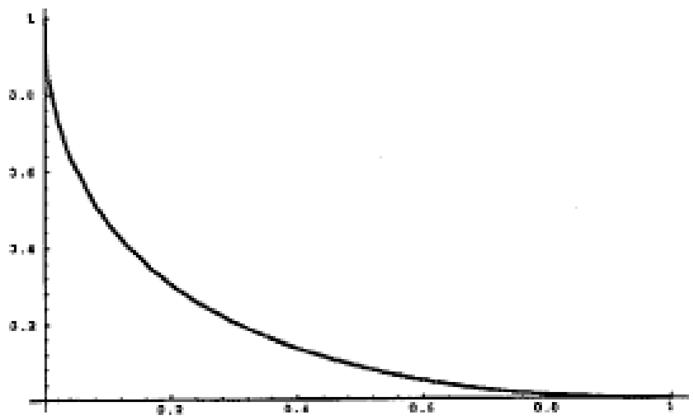
Spatial smoothness assumption

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

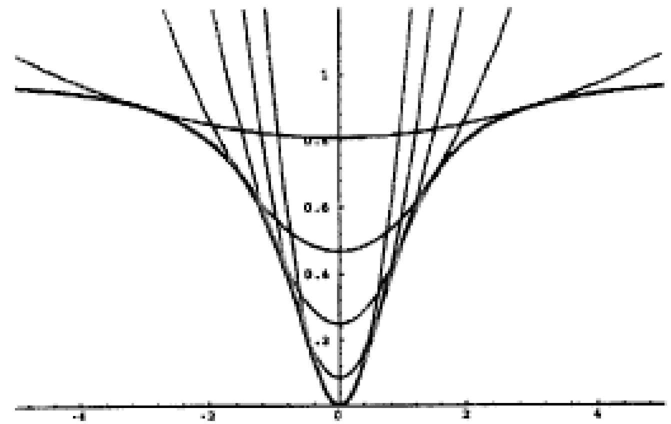
Robustness $l(x)\in\{0,1\}$

Analog line process

Penalty function



Family of quadratics



Faithful to the data

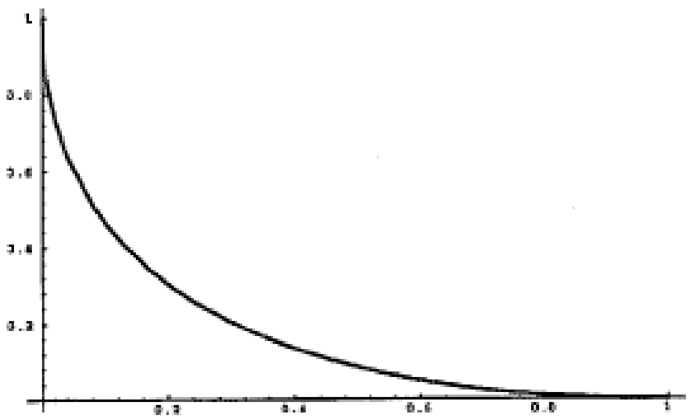
Spatial smoothness assumption

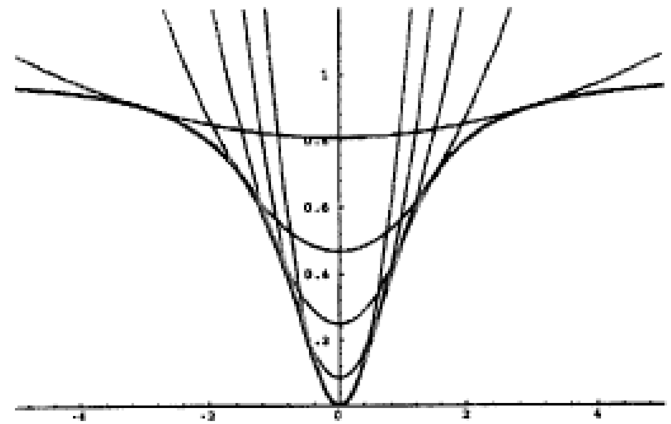
$$E(f,l)=\sum_{x=1}^{N}(f(x)-g(x))^2+\lambda\sum_{x=1}^{N-1}\left[l(x)(f(x+1)-f(x))^2+\Psi(l(x))\right]$$

Robustness $0\le l(x)\le 1$

Analog line process

Infimum defines a robust error function.



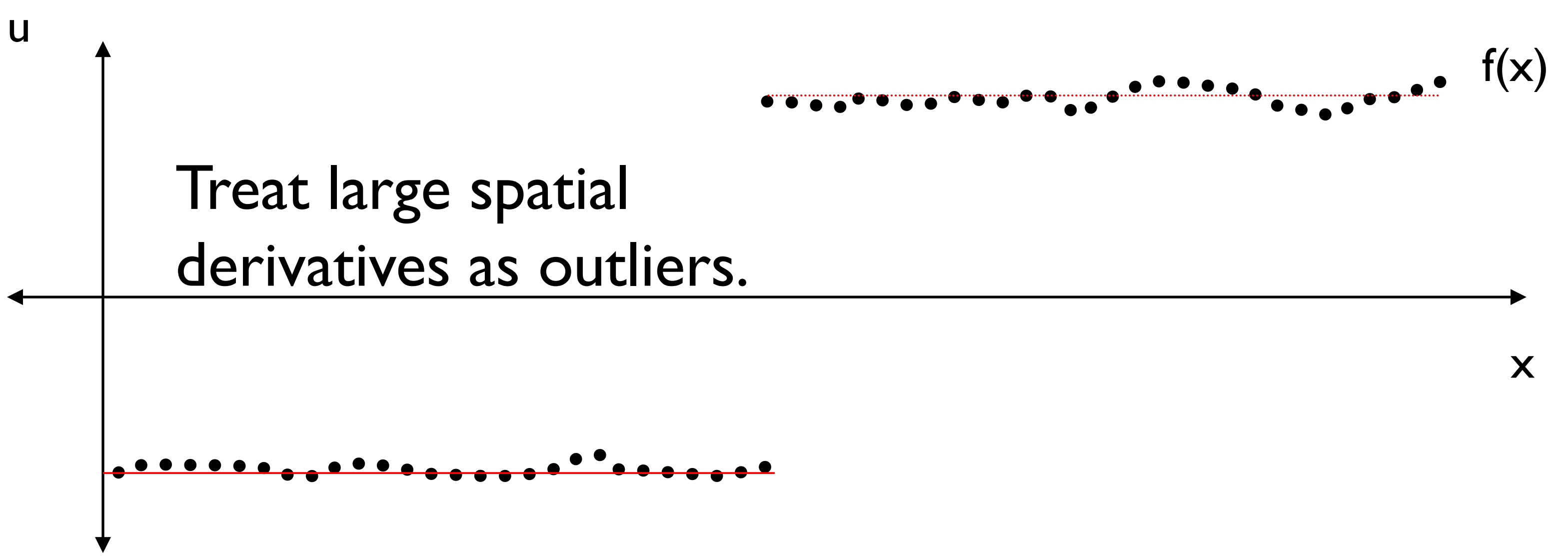


Minima are the same:

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

$$E(f)=\sum_{x=1}^{N}(f(x)-g(x))^2+\lambda\sum_{x=1}^{N-1}\rho(f(x+1)-f(x),\sigma_2)$$

Robust regularization



Faithful to the data

Spatial smoothness assumption

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

Robustness

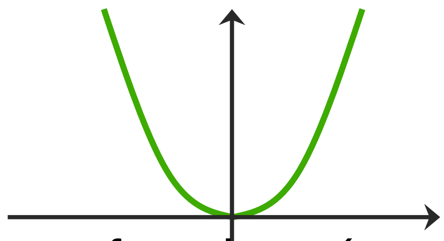
Robust estimation

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

error penalty function

influence function

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



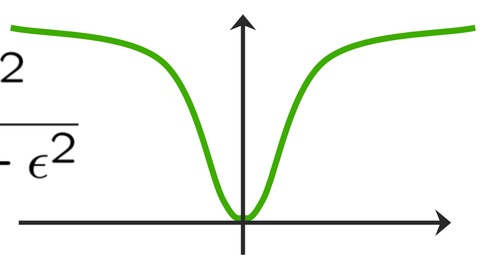
$$\psi(\epsilon) = \frac{\partial \rho(\epsilon)}{\partial \epsilon} = 2\,\epsilon$$

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

error penalty function

influence function

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



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

Robust regularization

Faithful to the data

Spatial smoothness assumption

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

$$\boxed{\text{Minimize: } \quad E = E_c + \lambda E_s}$$

What are E_c and E_s for optical flow estimation?

Regularization for optical flow

Add global smoothness term [Horn and Schunk 1981]

Error in brightness constancy equation

$$E_c = \iint_D \left(I_x u + I_y v + I_t\right)^2 dx\,dy$$

Smoothness error:

$$E_s = \iint_D \left(u_x^2 + u_y^2\right) + \left(v_x^2 + v_y^2\right) dx\,dy$$

Minimize: $E_c + \lambda E_s$

Solve by calculus of variations

Robust regularization for optical flow

[Black & Anandan 1993]

Regularization can over-smooth across edges

⇩

Use "smarter" regularization

Robust regularization for optical flow

[Black & Anandan 1993]

Regularization can over-smooth across edges

Use "smarter" regularization

Minimize:

$$\iint_D \underbrace{\rho_1\left(I_x u + I_y v + I_t\right)}_{\text{Brightness constancy}} + \lambda \left[\underbrace{\rho_2\left(u_x, u_y\right) \rho_2\left(v_x, v_y\right)}_{\text{Smoothness}} \right] dx\, dy$$

Optimization

- Gradient descent
- Coarse-to-fine (pyramid)
- Deterministic annealing

Recent GPU Implementation

- http://gpu4vision.icg.tugraz.at/
- Real time flow exploiting robust norm + regularized mapping

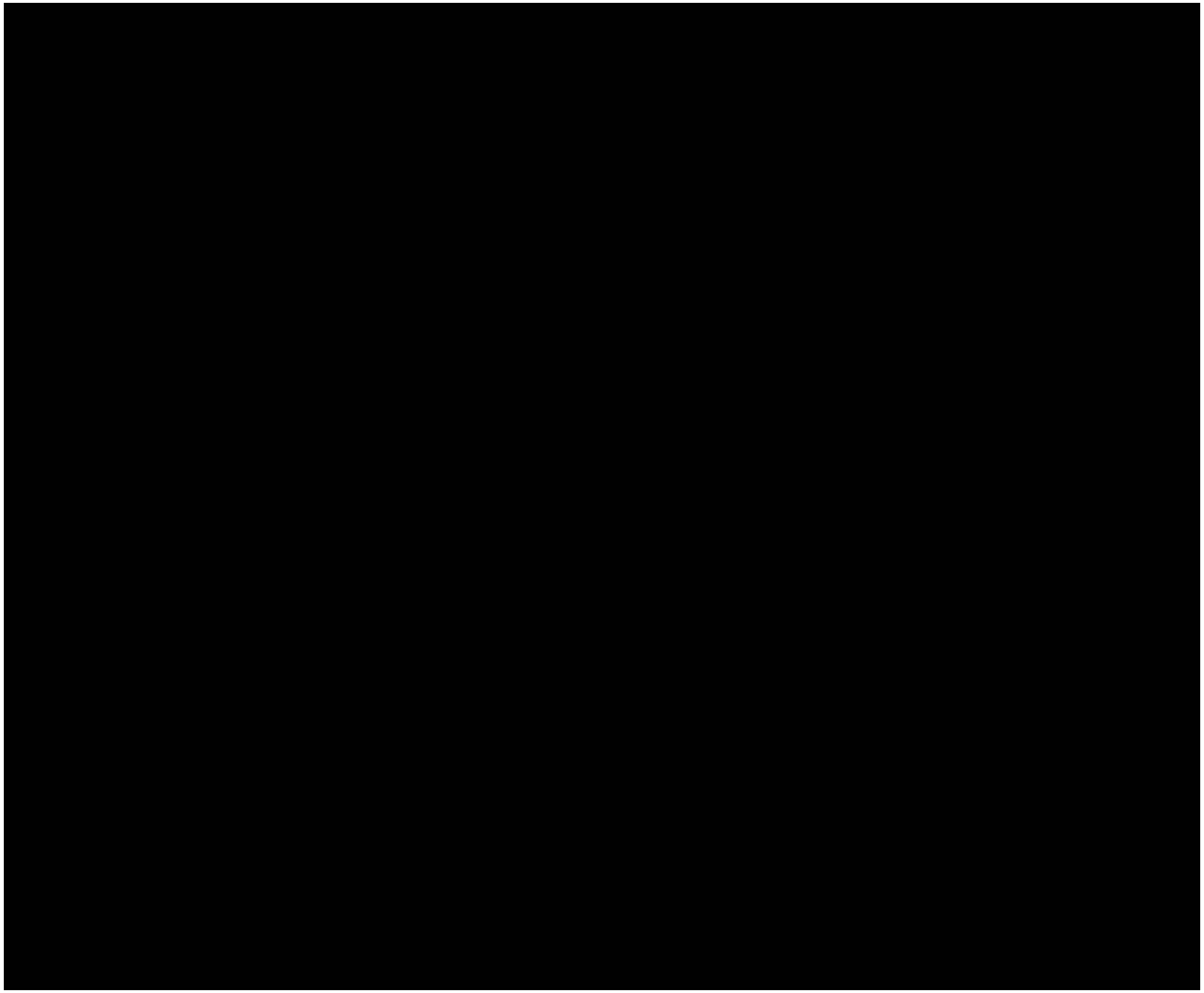
https://www.youtube.com/watch?v=ssINeWRb58M

A Duality Based Approach for Realtime TV-L^1 Optical Flow

C. Zach[1], T. Pock[2], and H. Bischof[2]

[1] VRVis Research Center
[2] Institute for Computer Graphics and Vision, TU Graz

Using optical flow



State-of-the-art optical flow

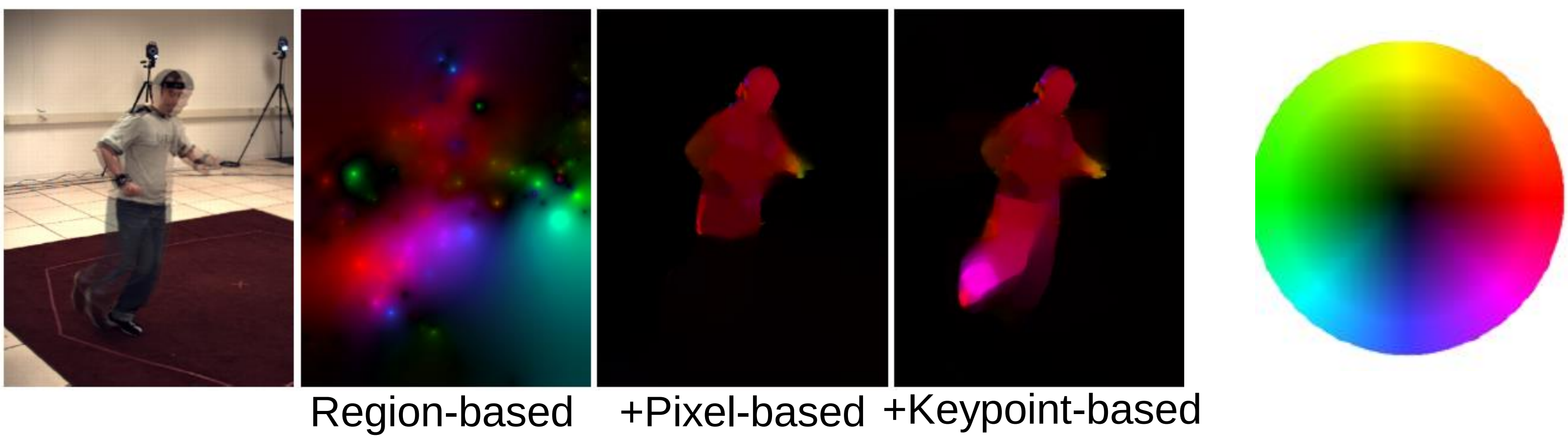
Start with something similar to Lucas-Kanade

+ gradient constancy

+ energy minimization with smoothing term

+ region matching

+ keypoint matching (long-range)



Large displacement optical flow, Brox et al., CVPR 2009

Today

From images to video

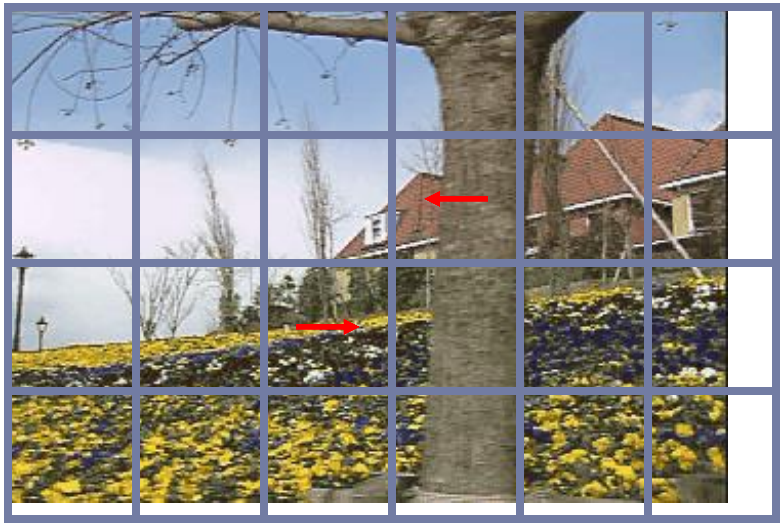
- Optical flow
- Feature tracking
- Motion segmentation
  - Layered representation
- Applications

Motion representations

- How can we describe the motion in the scene?

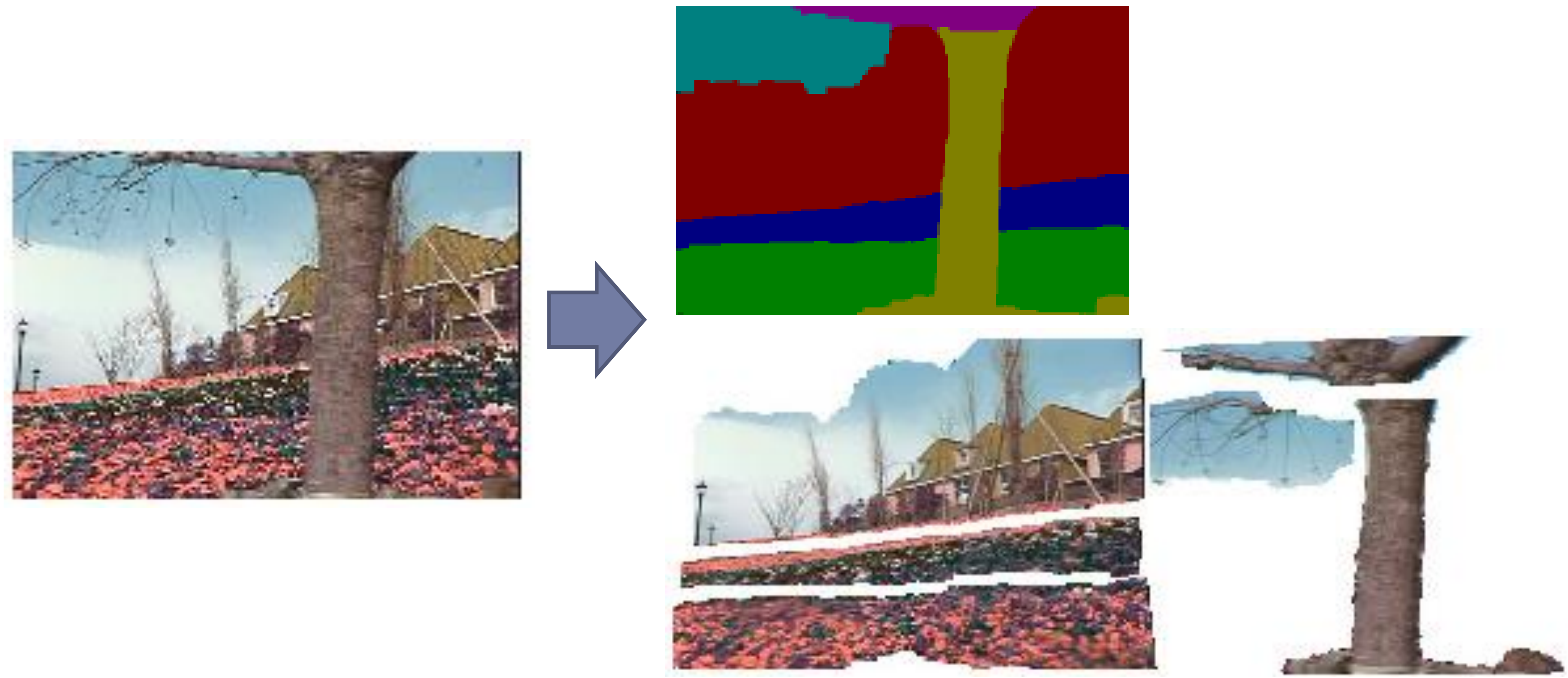
Block-based motion prediction

- 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

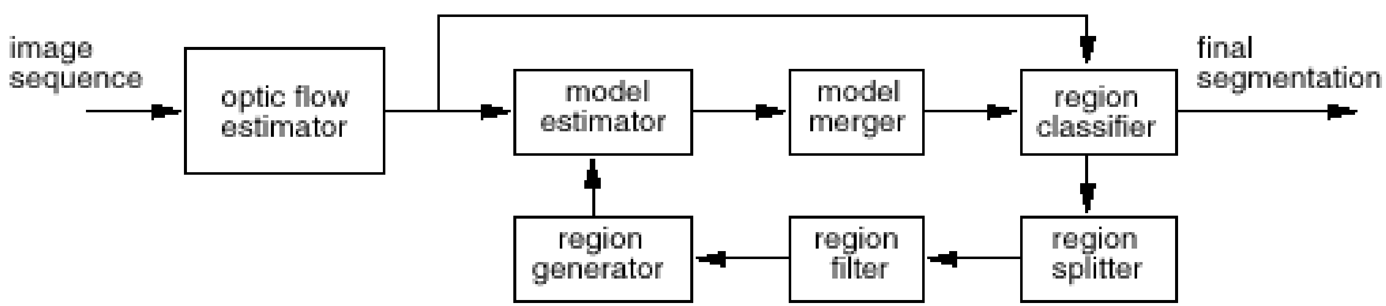
Layered motion

- Advantages
  - can represent occlusions / disocclusions
  - each layer's motion can be smooth
  - video segmentation for semantic processing

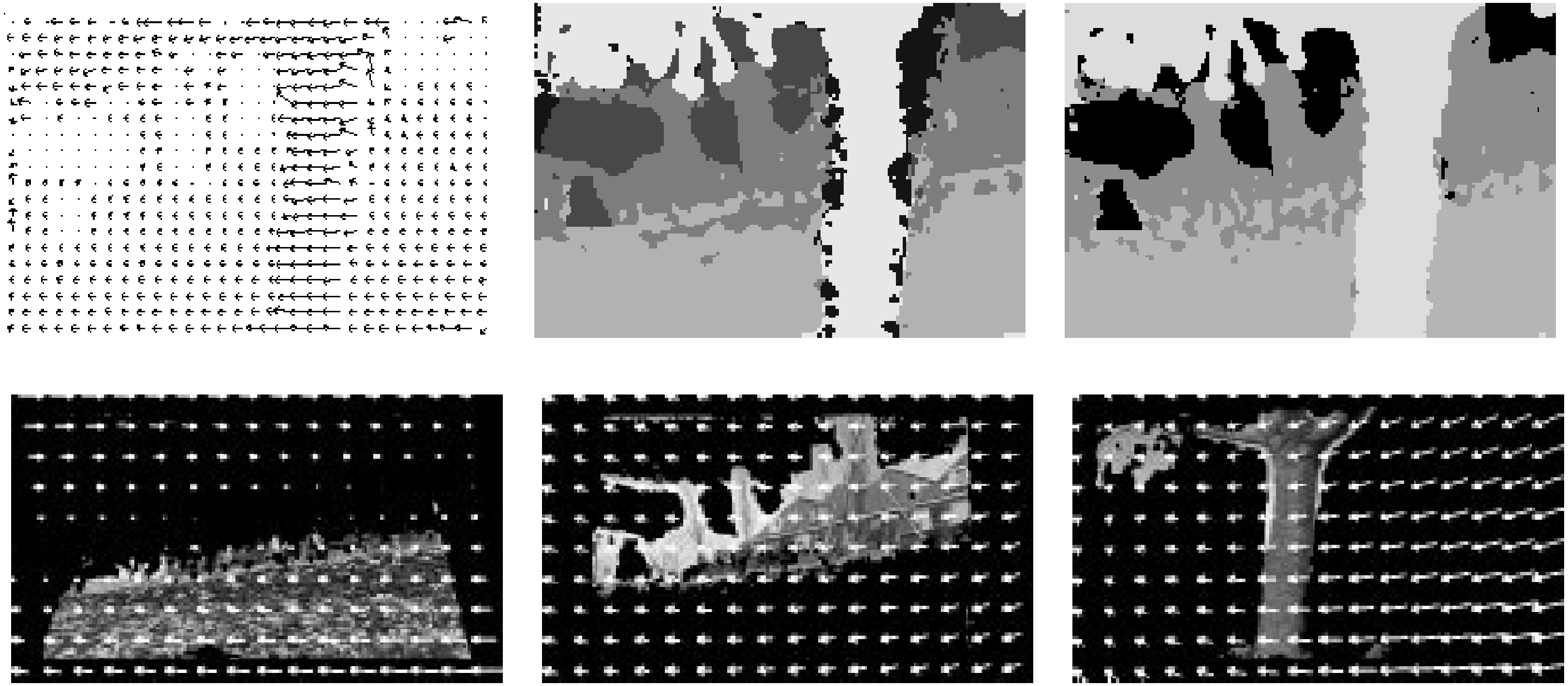
- Difficulties:
  - how do we determine the correct number?
  - how do we assign pixels?
  - how do we model the motion?

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

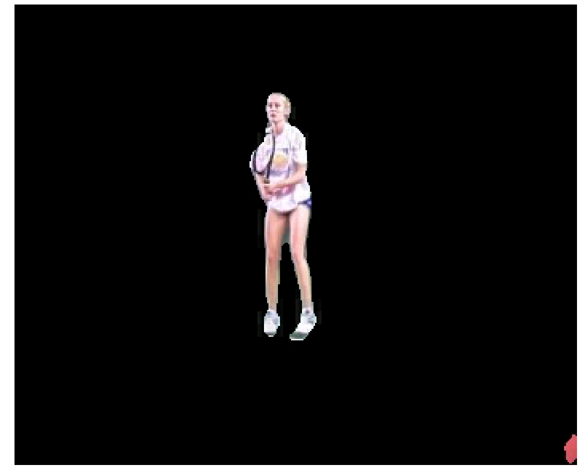
Convergence

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

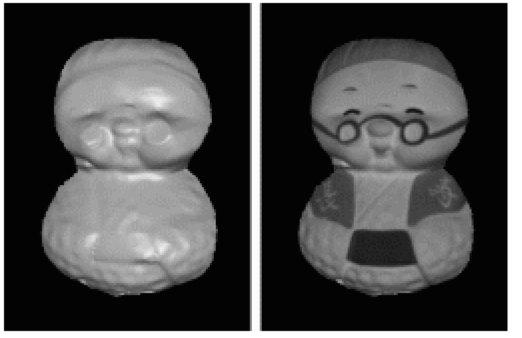
Input

Camera ego-motion

Result by MobilEye (www.mobileye.com)

Structure from Motion

Input



Reconstructed shape

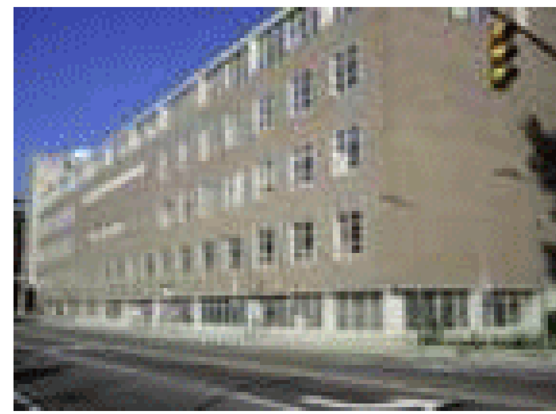
[Zhang, et al. ICCV'03]

Stabilization

[Zelnik-Manor & Irani, PAMI 2000]

SIFT Flow

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



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

Today

From images to video

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

- How do we evaluate success? [Baker et al. ICCV'07]

Synthetic video sequence

- Synthetic sequences can be used for quantitative evaluation

Yosemite

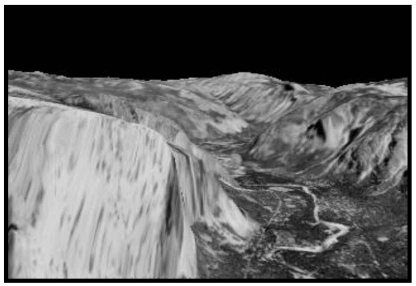


Image 7

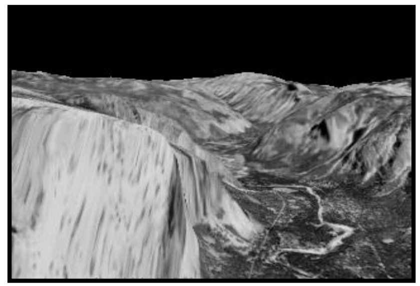
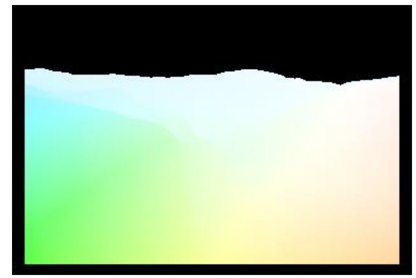
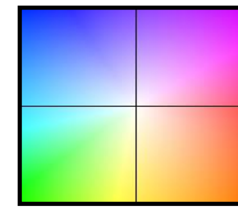


Image 8



Ground-Truth Flow

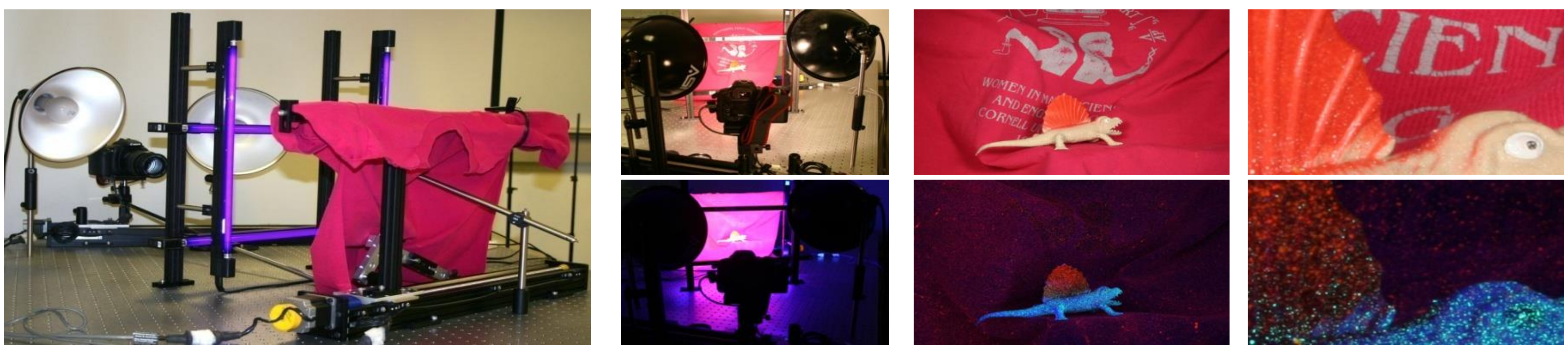


Flow Color Coding

- 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

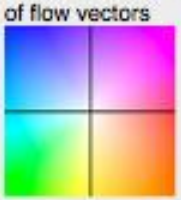
Middlebury dataset

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			Seashell (Hidden texture) GT im0 im1			Rock (Synthetic) GT im0 im1			Grove (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Venus (Stereo) GT im0 im1			Moebius (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	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	**6.32** 1	**12.41** 1	**10.98** 1	**1.69** 1	**2.86** 1	**1.05** 1	8.73 2	31.46 3	8.15 2	**5.85** 1	10.12 2	8.80 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	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	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	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™	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	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.	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	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

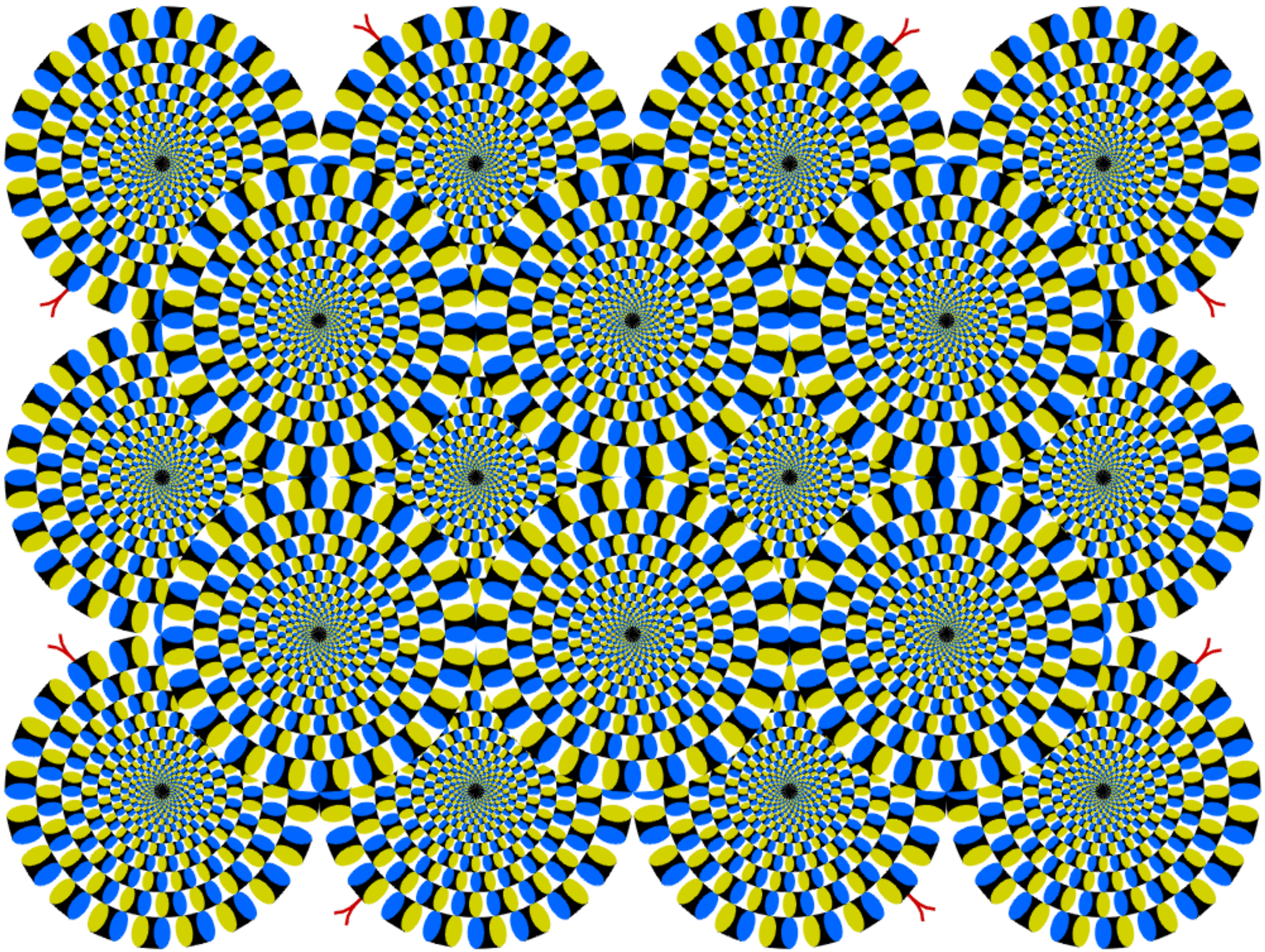
Color encoding of flow vectors



Flow image

Error image

Optical flow without motion



End – Optical flow

Now you know how it works