MOTION & TRACKING

Readings:

Szeliski: **Chapter 8.1,8.2,8.4**

Some Slides adapted from Univ. of Washington

Additional material by courtesy of Tomas Svoboda
svoboda@cmp.felk.cvut.cz

Why estimate visual motion?

- Visual Motion can be annoying
  - Camera instabilities, jitter
  - Measure it; remove it (stabilize)
- Visual Motion indicates dynamics in the scene
  - Moving objects, behavior
  - Track objects and analyze trajectories
- Visual Motion reveals spatial layout
  - Motion parallax

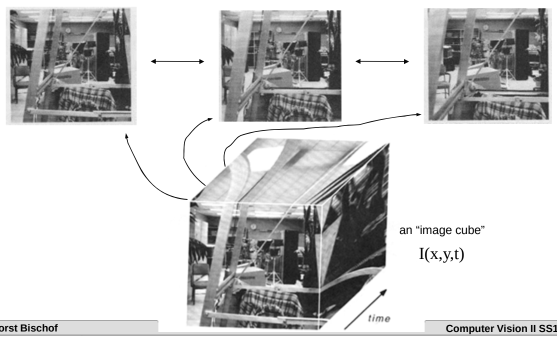
TX-Clip

Motion

- Motion Blur
- Motion Detection
  – Find movement
- Moving Object Detection and Location
  – Motion Based Segmentation
  – Trajectories → Tracking
- Shape from Motion
  – Similar to Stereo
  – Optical Flow

Motion

Information about *scene motion* rather than the *scene.*



Process of images over time.

Image sequence:

– An image sequence is a series of N images, or frames, acquired at discrete time instants $t_k=t_o+k\Delta t$, where Δt is a fixed time interval and $k=0,1,\ldots,N-1$

Motion Blur

Image distortion

$$g(i,j) = s\left[\iint_{(a,b)\in O} f(a,b)h(a,b,i,j)\,da\,db\right] + v(i,j)$$

assuming linearity

g(i,j) ... degraded image
f(a,b) ... original image
h(...) ... distortion function
v(i,j) ... noise

$$g(i,j) = (f * h)(i,j) + v(i,j)$$

f-domain (Fourier transform)

$$G(u,v) = F(u,v)H(u,v) + N(u,v)$$

Inverse filtration

$$F(u,v) = G(u,v)H^{-1}(u,v) - N(u,v)H^{-1}(u,v)$$

Relative motion of camera and object

$$H(u,v) = \frac{\sin(\pi V T u)}{\pi V u}$$

V ... constant speed in x-axes
T ... shutter open time

Poor results in presence of noise ⟹ Wiener filter

Inverse filtration example

Blurred Image

Simple inverse filtration

Wiener filtration

Blurred + noise added

Simple inverse filtration

Wiener filtration

Motion Detection

- Separating background and foreground
  - static scene (background)
  - moving objects (foreground)
- Detect meaningful motions
- Problems:
  - shadows
  - objects that temporarily stopped
  - moving camera / background

Motion Detection with Static Camera

- Frame Differencing:

$$d(x,y)=\begin{cases}0 & |f_t(x,y)-f_{t+1}(x,y)|<\varepsilon \\ 1 & sonst\end{cases}$$

$f_t(x,y)$ $f_{t+1}(x,y)$ $d(x,y)$

Motion Detection with Static Camera

- Frame Differencing (Thresholds)

Motion Detection with Static Camera

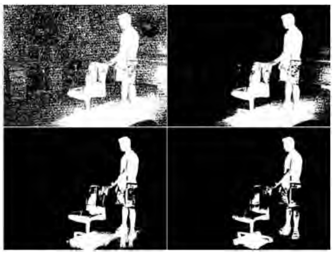
- Background Substraction

$$d(x,y)=\begin{cases}0 & |f_t(x,y)-B(x,y)|<\varepsilon \\ 1 & sonst\end{cases}$$

$B(x,y)$ $f_t(x,y)$ $d(x,y)$

Motion Detection with Static Camera

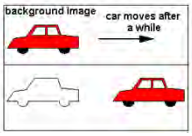
- Frame Differencing (Thresholds)

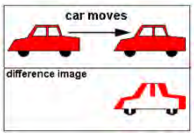


Motion Detection Summary

Problems:

- **Background subtraction:**
  - How to get background model?
  - When to update
  - Holes
- **Frame differencing:**
  - Parts of object
  - Timescale of frames
  - Threshold





Motion Detection

Surveillance:

Motion Analysis Problems

Correspondence Problem
- Track corresponding elements across frames

Reconstruction Problem
- Given a number of corresponding elements, and camera parameters, what can we say about the 3D motion and structure of the observed scene?

Segmentation Problem
- What are the regions of the image plane which correspond to *different* moving objects?

Motion field

Vector field where each vector represents motion of point across image at time interval dt

Optical Flow

Image sequence:
- Spatial- and Temporal sampling
- Temporal sampling needs to be high
- Correspondence problem (occlusions, etc)

Assuming that illumination does not change:

Image changes are due to the RELATIVE MOTION between the scene and the camera.

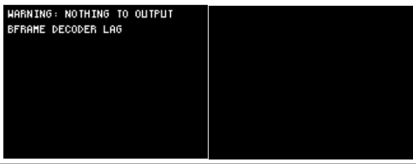
There are 3 possibilities:
- Camera still, moving scene
- Moving camera, still scene
- Moving camera, moving scene

Optical Flow Examples

Static Camera / Moving Scene

Color-coded

Moving Camera / Moving Scene



Motion Field (MF)

The MF assigns a velocity vector to each pixel in the image.

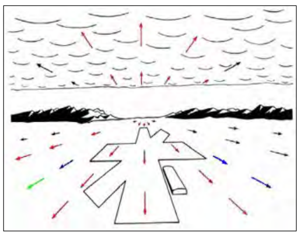
These velocities are INDUCED by the RELATIVE MOTION btw the camera and the 3D scene

The MF can be thought as the *projection* of the 3D velocities on the image plane.

Motion Field

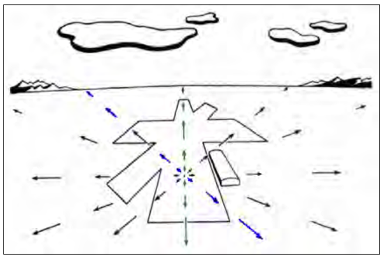
Non rotating Camera:

- Vectors are radial to Focus
- **FOE**: Focus Of **Expansion**
- **FOC**: Focus Of **Contraction**
- Point where Motion vectors intersect with the image plane.



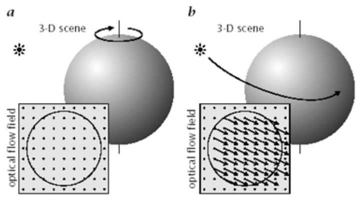
Motion Field

Length of vector:
- inversely proportional to distance of point
- proportional to sine of point and movement direction of camera



Motion Field

Attention: Motion Vector ≠ Motion



Moving Ball Moving Light source

Optical Flow estimation

Correspondence problem
- sparse vector field
- same problem as stereo

To reliably establish correspondences
- High temporal sampling → small changes
- Constant intensities/edges

Optical Flow estimation

Two assumptions

1. Observed brightness is constant
2. Nearby points move in similar manner (velocity smoothness)

Brightness Constraint

Brightness Constancy Equation:

$$J(x,y) \approx I(x+u(x,y), y+v(x,y))$$

Or, minimize :

$$E(u,v) = (J(x,y) - I(x+u, y+v))^2$$

Linearizing (assuming small *(u,v)*):

$$J(x,y) \approx I(x,y) + I_x(x,y) \cdot u(x,y) + I_y(x,y) \cdot v(x,y)$$

Gradient Constraint (or the Optical Flow Constraint)

$$E(u,v) = (I_x \cdot u + I_y \cdot v + I_t)^2$$

Minimizing:

$$\frac{\partial E}{du} = \frac{\partial E}{dv} = 0$$

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

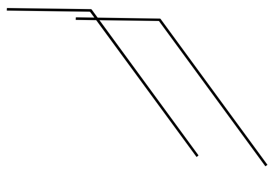
$$I_y(I_x u + I_y v + I_t) = 0$$

In general $I_x, I_y \neq 0$

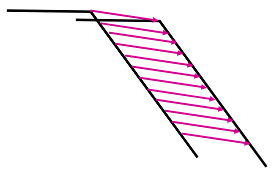
Hence, $I_x \cdot u + I_y \cdot v + I_t \approx 0$

The gradient constraint – *only one constraint for each pixel*

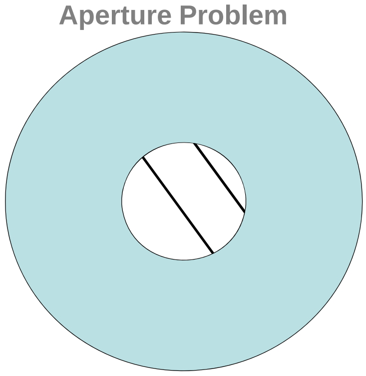
Aperture Problem



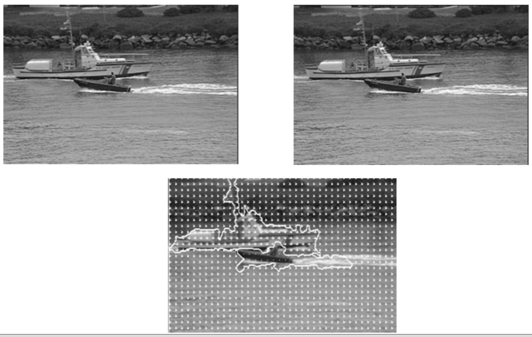
Aperture Problem



Aperture Problem



Motion Based Segmentation



Horst Bischof — Computer Vision II SS11

34

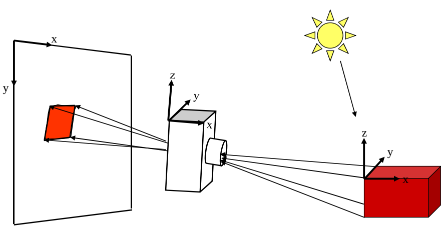
Motion Based Segmentation

- **Top-Down:**
  - Global Motion
  - All regions that are significantly different
- **Joint estimation / segmentation**
  - Pixel as Mixture of Motions
  - Estimate Motion and Segmentation simultaneously
- **Grouping of elementary regions**
  - Region Growing Split & Merge, etc.
  - Use also Gray value information

Horst Bischof — Computer Vision II SS11

35

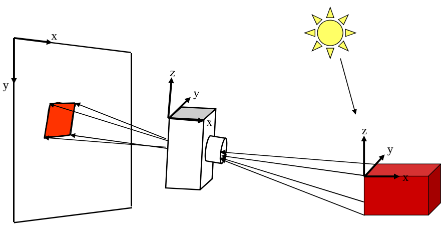
Vision = Inverse Graphics



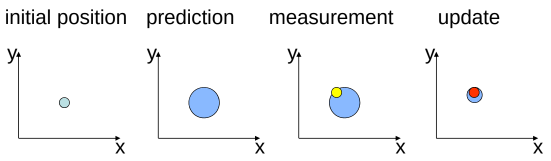
Horst Bischof — Computer Vision II SS11

36

Tracking = Inverse Animation

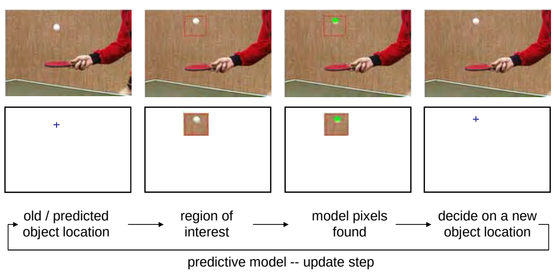


Tracking: First Idea!



Region tracking

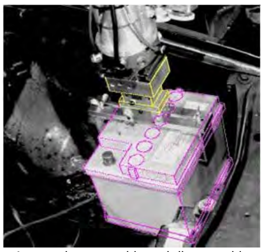
or "BLOB" tracking



recognizing a ping-pong ball by pixel intensity

Contour Tracking

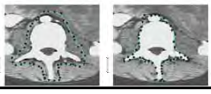
Contour-tracking, applied



Automotive assembly and disassembly

Welding

Gesture recognition



Medical image processing

Optical Tracking - Animation

multi-camera tracking system:

- real-time motion capturing of actors
- real-time visual feedback
- added redundancy by integrating single view tracking as fallback

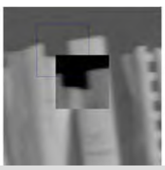
Tracking for MOCAP

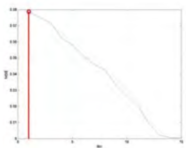
Tracking landmarks to control avatars facial expressions (VIDEO)

Tracking f. Augmented Reality

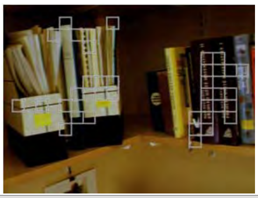
Lucas-Kanade Tracking

Find correspondes using a template
Estimate „optimal" Translation
Minimize SSD
Gradient based Optimization





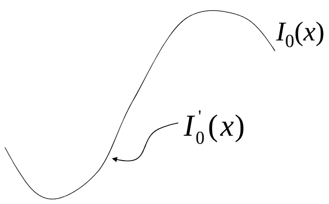
- Computer Vision Applications:
  - Optical Flow
  - Feature Tracker
    KLT (Kanade-Lucas-Tomasi Tracking)
  - Widely used



Lucas & Kanade Derivation

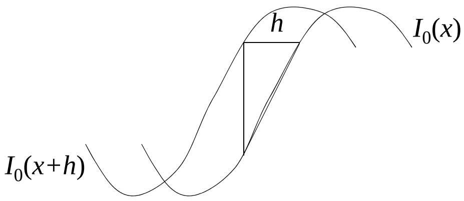
#1

L&K Derivation 1



$$I_0'(x) = \lim_{h \to 0} \frac{I_0(x+h) - I_0(x)}{h}$$

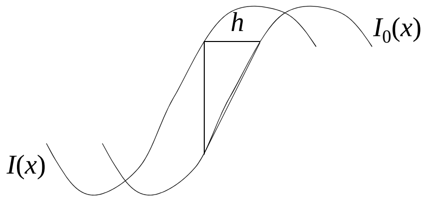
L&K Derivation 1



$$I_0'(x) \approx \frac{I_0(x+h) - I_0(x)}{h}$$

Horst Bischof | Computer Vision II SS11 | 49

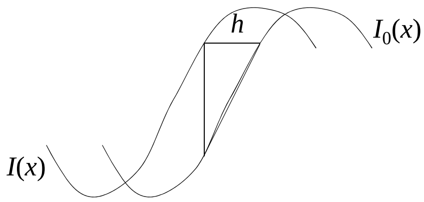
L&K Derivation 1



$$I_0'(x) \approx \frac{I(x) - I_0(x)}{h}$$

Horst Bischof | Computer Vision II SS11 | 50

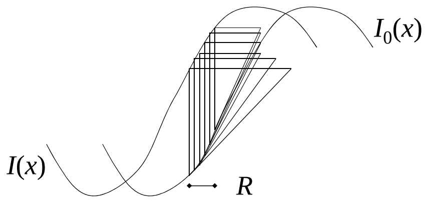
L&K Derivation 1



$$h \approx \frac{I(x) - I_0(x)}{I_0'(x)}$$

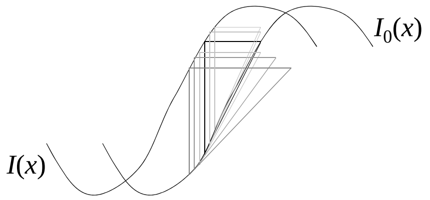
Horst Bischof | Computer Vision II SS11 | 51

L&K Derivation 1



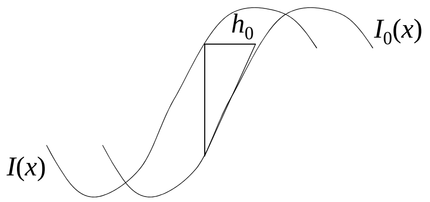
$$h \approx \frac{1}{|R|}\sum_{x \in R} \frac{I(x) - I_0(x)}{I_0^{'}(x)}$$

L&K Derivation 1



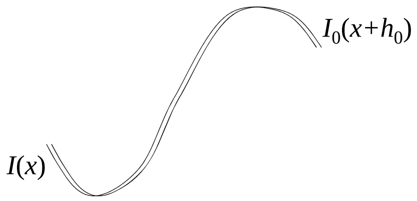
$$h \approx \frac{1}{\sum_x w(x)}\sum_{x \in R} \frac{w(x)[I(x) - I_0(x)]}{I_0^{'}(x)}$$

L&K Derivation 1



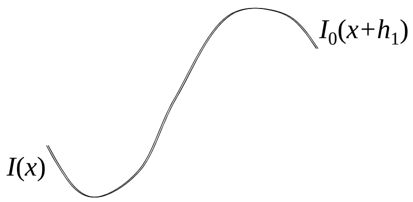
$$h_0 \leftarrow \frac{1}{\sum_x w(x)}\sum_{x \in R} \frac{w(x)[I(x) - I_0(x)]}{I_0^{'}(x)}$$

L&K Derivation 1



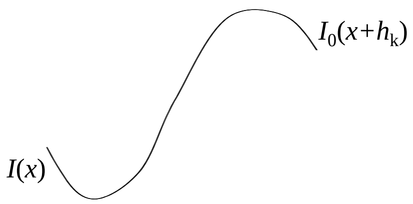
$$h_1 \leftarrow h_0 + \frac{1}{\sum_x w(x)} \sum_{x \in R} \frac{w(x)[I(x) - I_0(x+h_0)]}{I_0'(x+h_0)}$$

L&K Derivation 1



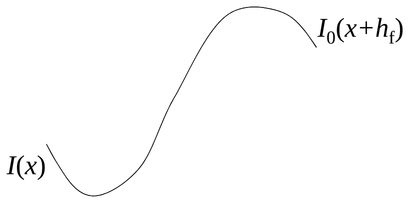
$$h_2 \leftarrow h_1 + \frac{1}{\sum_x w(x)} \sum_{x \in R} \frac{w(x)[I(x) - I_0(x+h_1)]}{I_0'(x+h_1)}$$

L&K Derivation 1



$$h_{k+1} \leftarrow h_k + \frac{1}{\sum_x w(x)} \sum_{x \in R} \frac{w(x)[I(x) - I_0(x+h_k)]}{I_0'(x+h_k)}$$

L&K Derivation 1



$$h_{k+1} \leftarrow h_k + \frac{1}{\sum_x w(x)} \sum_{x \in R} \frac{w(x)[I(x) - I_0(x+h_k)]}{I_0'(x+h_k)}$$

Lucas & Kanade Derivation

#2

L&K Derivation 2

Sum-of-squared-difference (SSD) error

$$E(h) = \sum_{x \in R} [\, I(x) - I_0(x+h) \,]^2$$

$$E(h) \approx \sum_{x \in R} [\, I(x) - I_0(x) - hI_0'(x) \,]^2$$

L&K Derivation 2

$$\frac{\partial E}{\partial h} \approx \sum_{x \in R} 2[I_0'(x)(I(x) - I_0(x)) - hI_0'(x)^2]$$

$$= 0$$

$$h \approx \frac{\sum_{x \in R} I_0'(x)(I(x) - I_0(x))}{\sum_{x \in R} I_0'(x)^2}$$

Comparison

$$h \approx \frac{\sum_x \frac{w(x)[I(x) - I_0(x)]}{I_0'(x)}}{\sum_x w(x)}$$

$$h \approx \frac{\sum_x I_0'(x)[I(x) - I_0(x)]}{\sum_x I_0'(x)^2}$$

Comparison

$$h \approx \frac{\sum_x \frac{w(x)[I(x) - I_0(x)]}{I_0'(x)}}{\sum_x w(x)}$$

$$h \approx \frac{\sum_x I_0'(x)[I(x) - I_0(x)]}{\sum_x I_0'(x)^2}$$

Generalizations

Original

$$E(h) = \sum_{x \in R} [I(x+h) - I_0(x)]^2$$

Original

Dimension of image

$$E(h) = \sum_{x \in R} [I(x+h) - I_0(x)]^2$$

1-dimensional

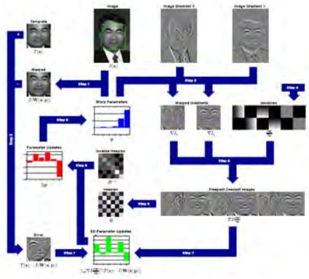
Generalization 1a

Dimension of image

$$E(\mathbf{h}) = \sum_{\mathbf{x} \in R} [I(\mathbf{x}+\mathbf{h}) - I_0(\mathbf{x})]^2$$

2D: $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$

Lucas-Kanade Algorithm



Iterate:

(1) Warp I with $\mathbf{W}(\mathbf{x};\mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
(2) Compute the error image $T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))$
(3) Warp the gradient ∇I with $\mathbf{W}(\mathbf{x};\mathbf{p})$
(4) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x};\mathbf{p})$
(5) Compute the steepest descent images $\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$
(6) Compute the Hessian matrix using Equation (11)
(7) Compute $\sum_{\mathbf{x}} [\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}}]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]$
(8) Compute $\Delta \mathbf{p}$ using Equation (10)
(9) Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}$

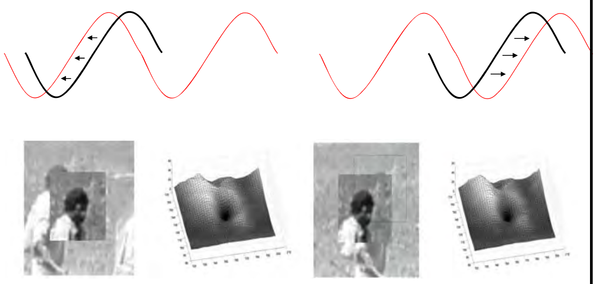
$\mathbf{p}\| \leq \epsilon$

Problem A

Does the iteration converge?

Problem A

Local minima:



Problem B

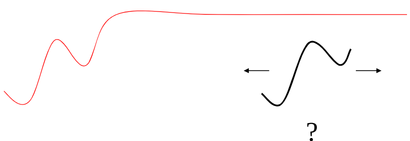
Zero gradient:

$$h \approx \frac{-\sum_{x \in R} I_0'(x)(I(x) - I_0(x))}{\sum_{x \in R} I_0'(x)^2}$$

h is undefined if $\sum_{x \in R} I_0'(x)^2$ is zero

Problem B

Zero gradient:



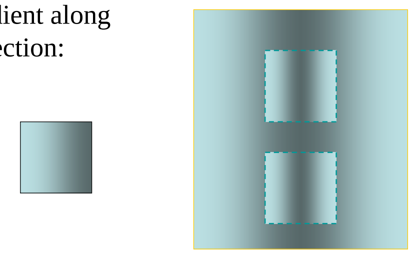
Problem B'

Aperture problem:

$$h_y \approx \frac{-\sum_{x \in R} \frac{\partial I_0(\mathbf{x})}{\partial y}(\mathbf{x})(I(\mathbf{x}) - I_0(\mathbf{x}))}{\sum_{x \in R} \left(\frac{\partial I_0(\mathbf{x})}{\partial y} \right)^2}$$

Problem B'

No gradient along one direction:



?

Good Features to Track?

- Textured Regions
  otherwise Aperture problem

- Harris corners
  - Edge: one eigenvector with eigenvalue zero
  - Homogenous: zero eigenvalues

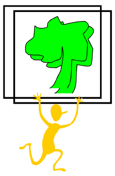
- Other break-downs:
  - No brightness constancy
  - Nearby points move different (which window size)
  - Too much motion (multi-scale estimation)
  - Occlusions

KLT Examples


Solutions to A & B

Possible solutions:

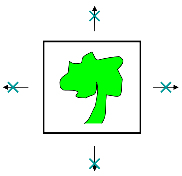
- Manual intervention



Solutions to A & B

Possible solutions:

- Manual intervention
- Zero motion default



Solutions to A & B

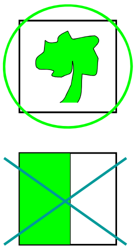
Possible solutions:

- Manual intervention
- Zero motion default
- Coefficient “dampening”

Solutions to A & B

Possible solutions:

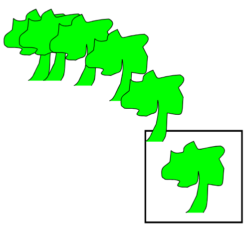
- Manual intervention
- Zero motion default
- Coefficient “dampening”
- Reliance on good features



Solutions to A & B

Possible solutions:

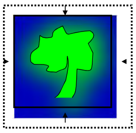
- Manual intervention
- Zero motion default
- Coefficient “dampening”
- Reliance on good features
- Temporal filtering



Solutions to A & B

Possible solutions:

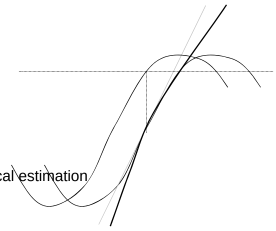
- Manual intervention
- Zero motion default
- Coefficient “dampening”
- Reliance on good features
- Temporal filtering
- Spatial interpolation / hierarchical estimation



Solutions to A & B

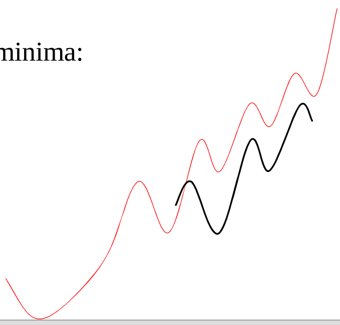
Possible solutions:

- Manual intervention
- Zero motion default
- Coefficient “dampening”
- Reliance on good features
- Temporal filtering
- Spatial interpolation / hierarchical estimation
- Higher-order terms



Problem A

Local minima:



Summary

- Motion Detection Methods
  - Background Subtraction / Frame difference
  - Assumption and Problems
- Motion Field Estimation
  - Assumptions
  - Aperture Problem
- Tracking
  - Main Loop
- Lucas Kanade
  - Principal Idea
  - Problems, Limitations and Solution