Video Processing

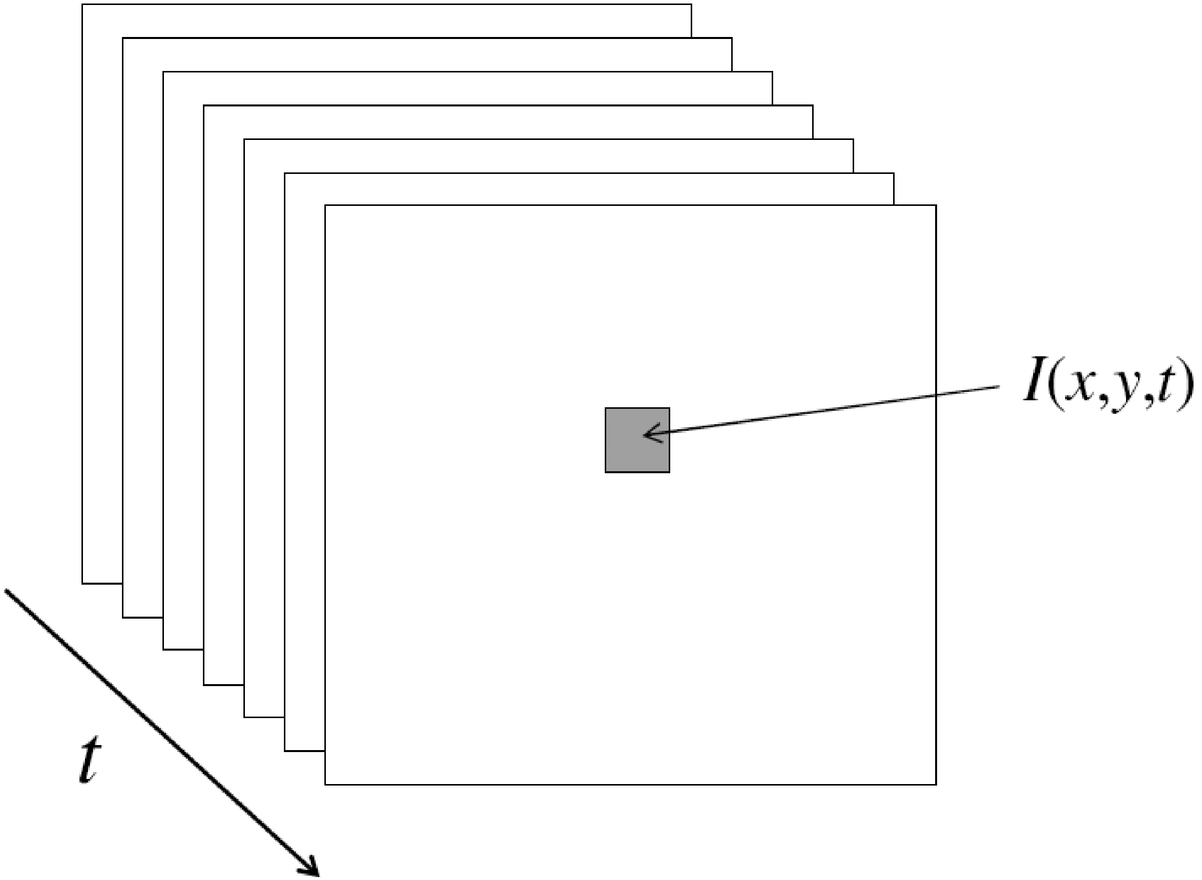
Paulo Dias

Sumary

- Motion
  - Motion analysis
  - Optical Flow
- Background Subtraction
- Tracking
  - Object Tracking
  - Template Matching
  - Detection vs Tracking

Video

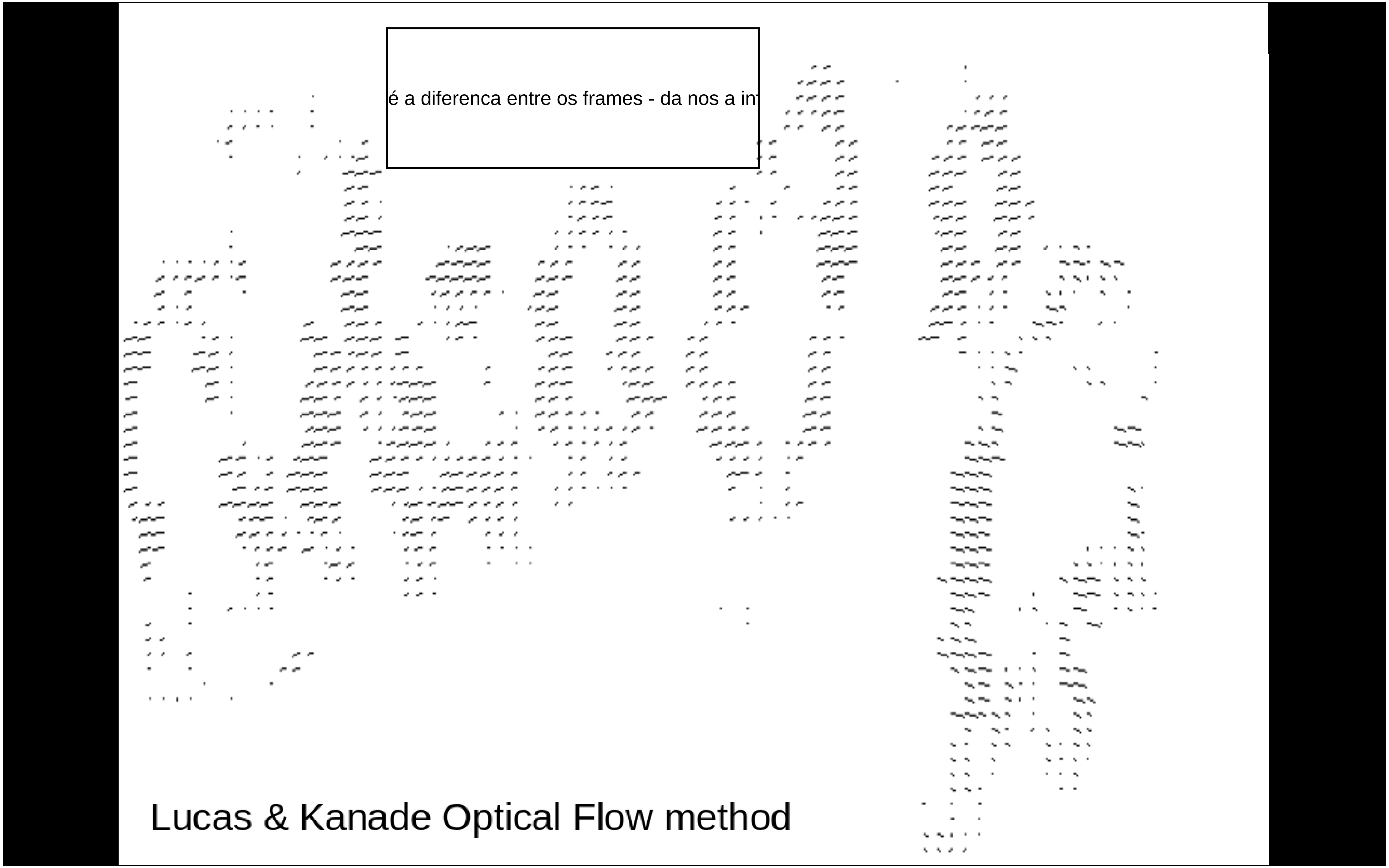
- A video is a sequence of frames captured over time
- The image data is a function of space (x, y) and time (t)



Motion Analysis

- Several information can be extracted from time varying sequences of images:
  - Camouflaged objects easily visible when moving
  - Size and position of objects are more easily determined when the objects move
  - Even simple image differencing provides edge detector for objects moving over any static background

Motion Analysis


é a diferenca entre os frames - da nos a in
Lucas & Kanade Optical Flow method

Motion Analysis

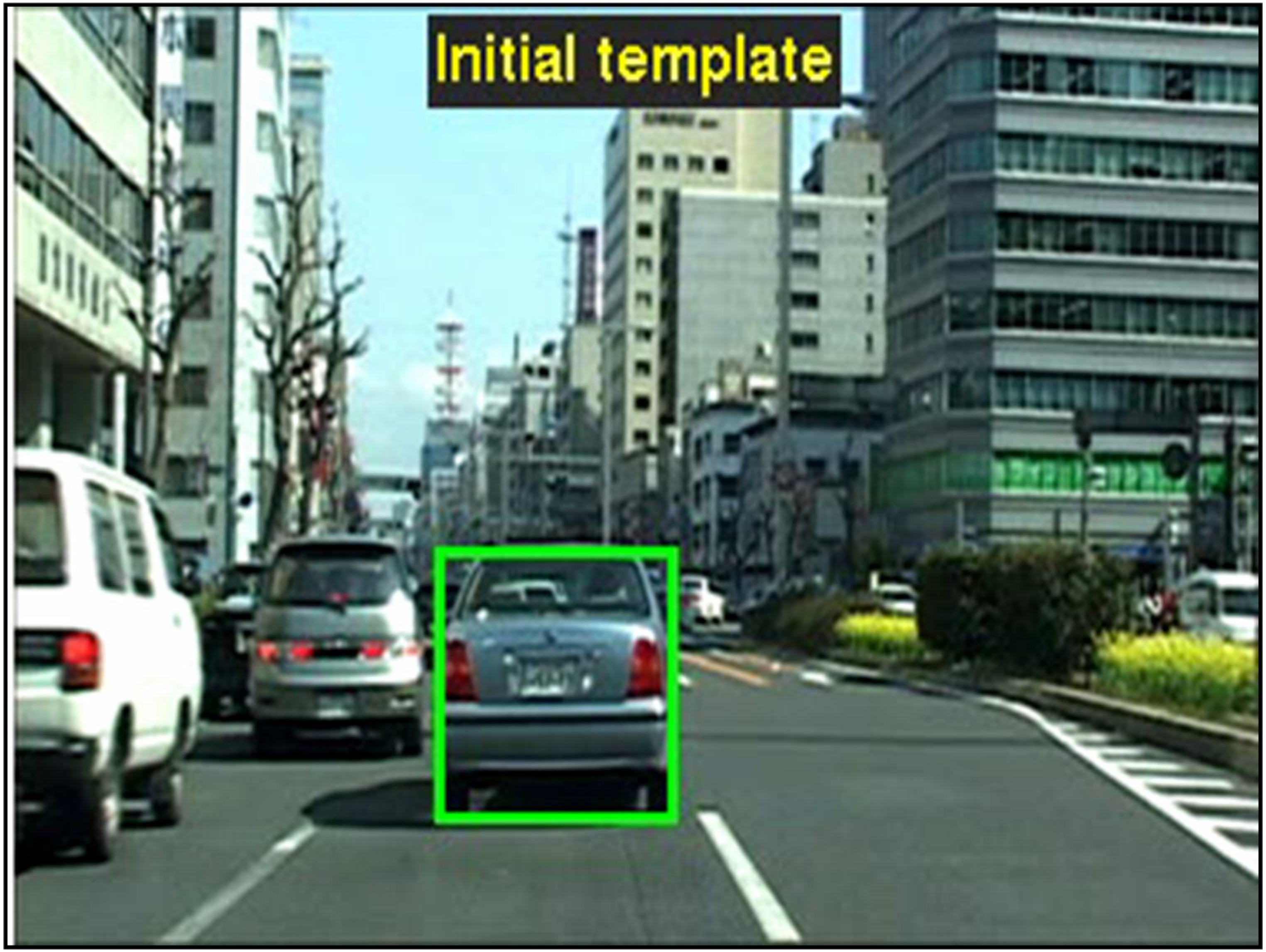
- Analysis of visual motion can be divided into two stages:
  - measurement of the motion
  - use of motion data to segment scene into objects and to extract information about shape and motion.
- There are two types of motion to consider:
  - movement in the scene = static camera,
  - movement of the camera = ego motion.
    - should be the same (motion is relative) but not always true - if scene moves illumination, shadow and specular effects need to be considered

Optical Flow and Motion

- We are interested in finding the movement of scene objects from time-varying images (videos).

- Lots of uses
  - Track object behavior
  - Align images (mosaics)
  - 3D shape reconstruction
  - Correct for camera jitter (stabilization)
  - Special effects

Rigid Object Tracking

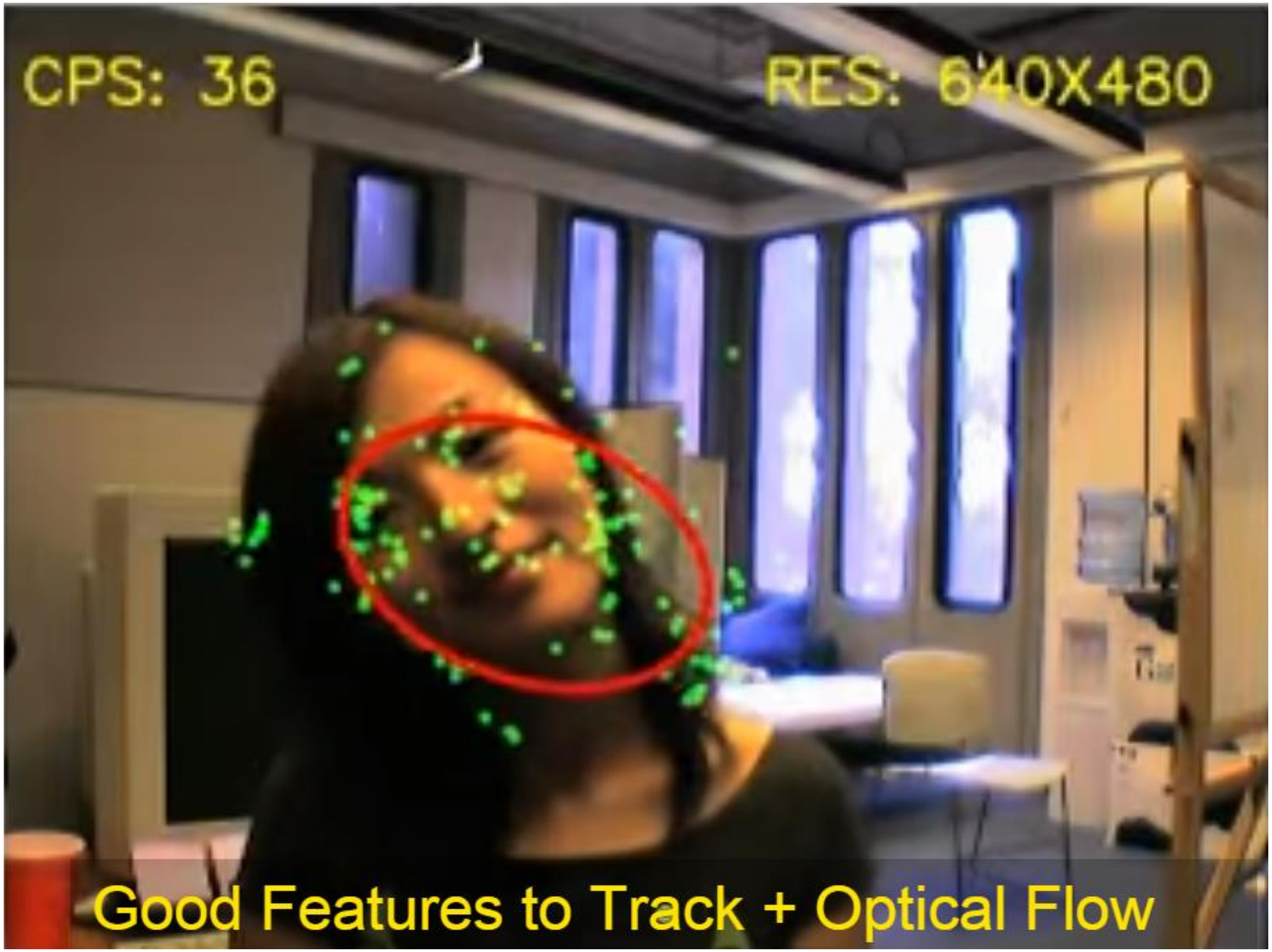


(Simon Baker, CMU)

Non Rigid Object Tracking

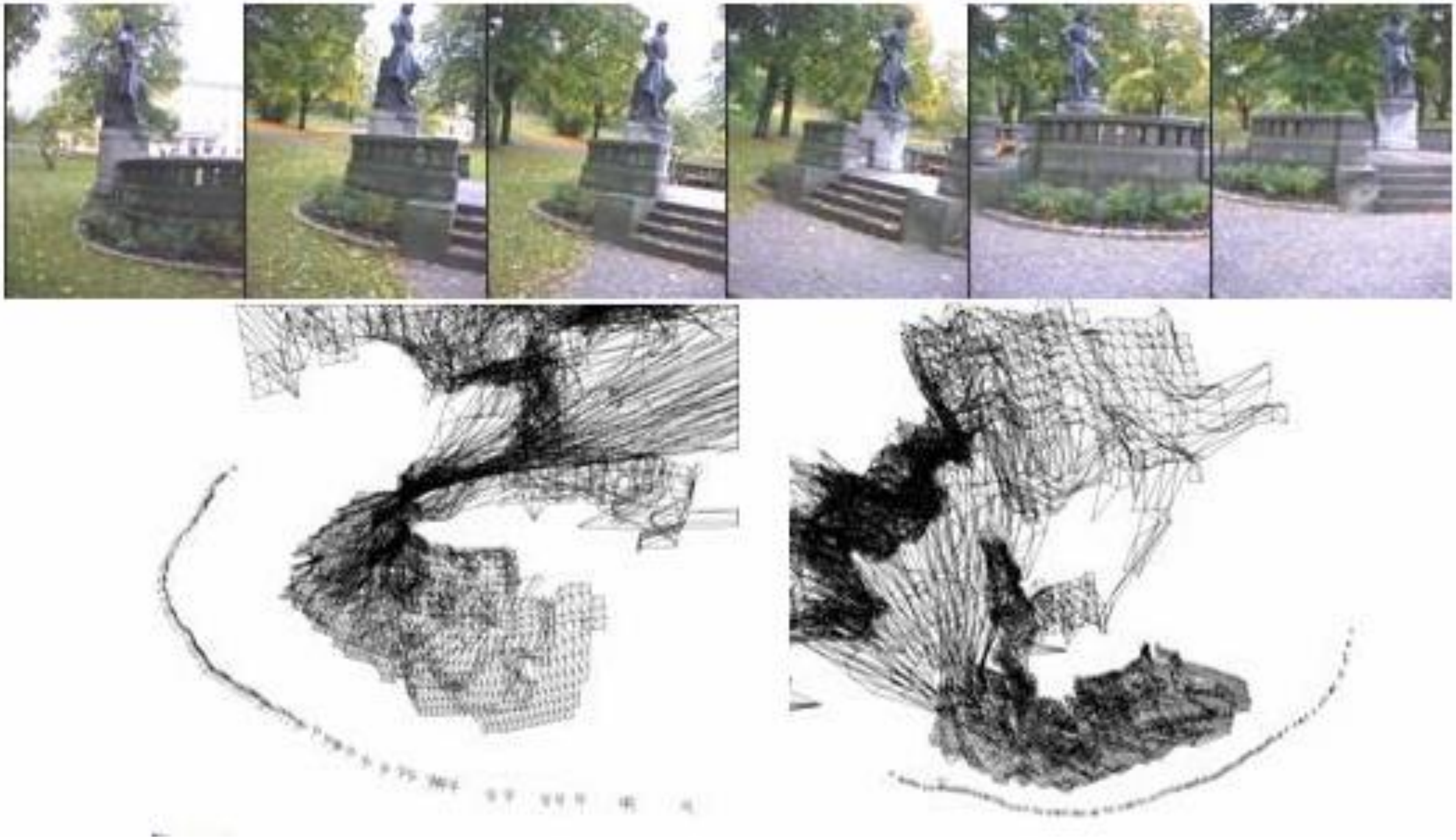
(Comaniciu et al, Siemens)

Face Tracking



https://youtu.be/Yw_zkLaZNsQ

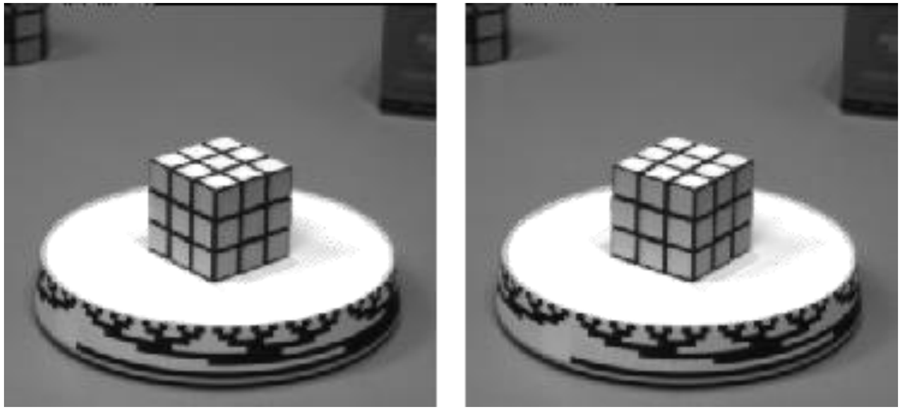
Structure from motion

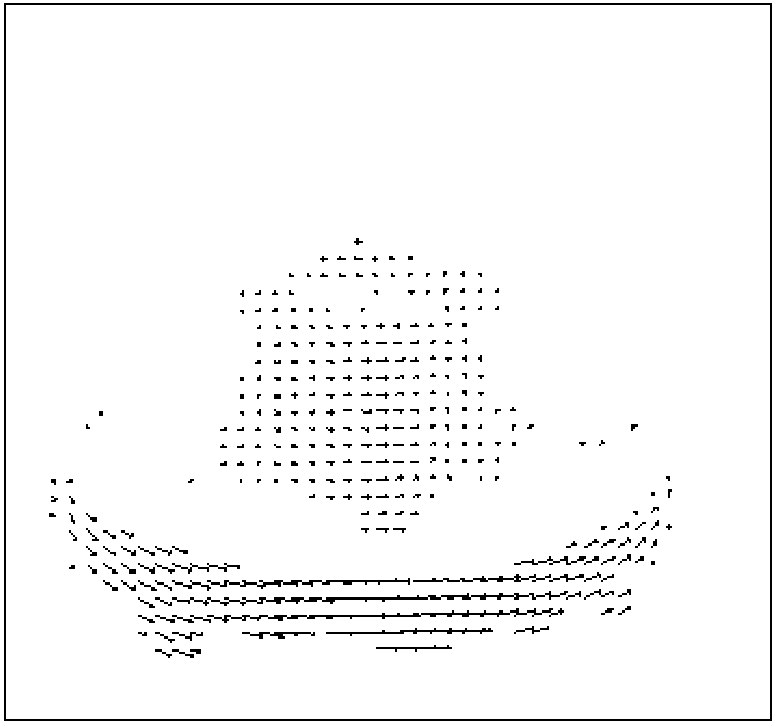


(David Nister, Kentucky)

Motion field

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





Motion field

Motion field and Camera Motion

- Length of flow vectors inversely proportional to depth Z of 3D point - points closer to the camera move more quickly across the image plane.

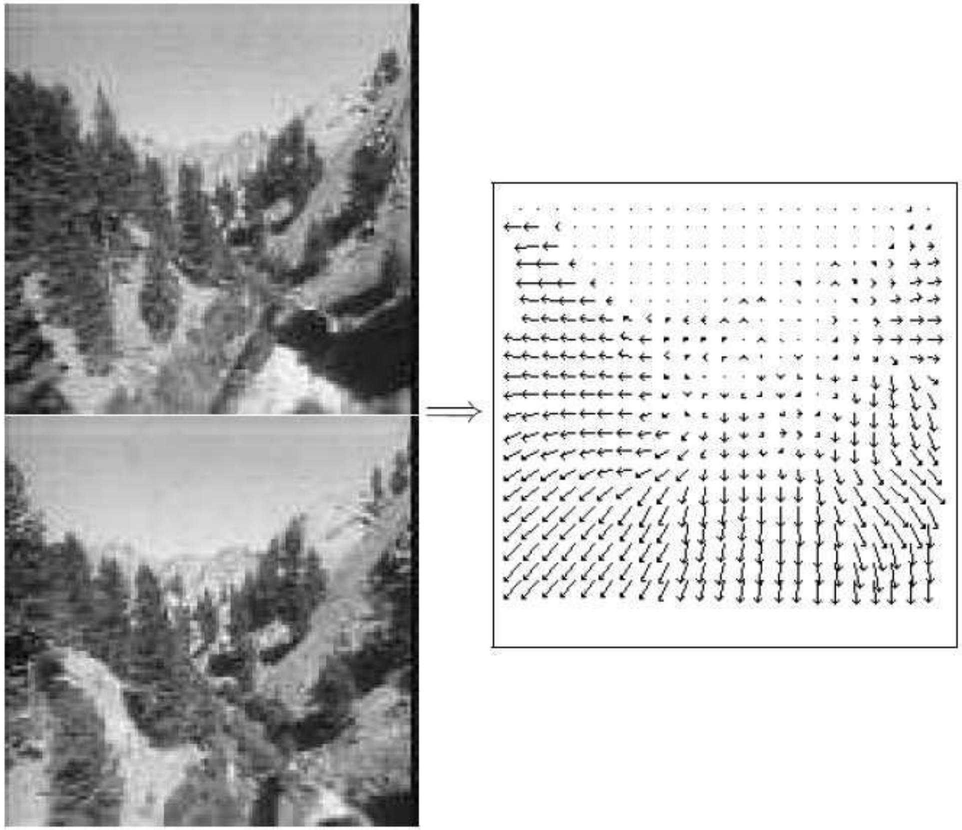


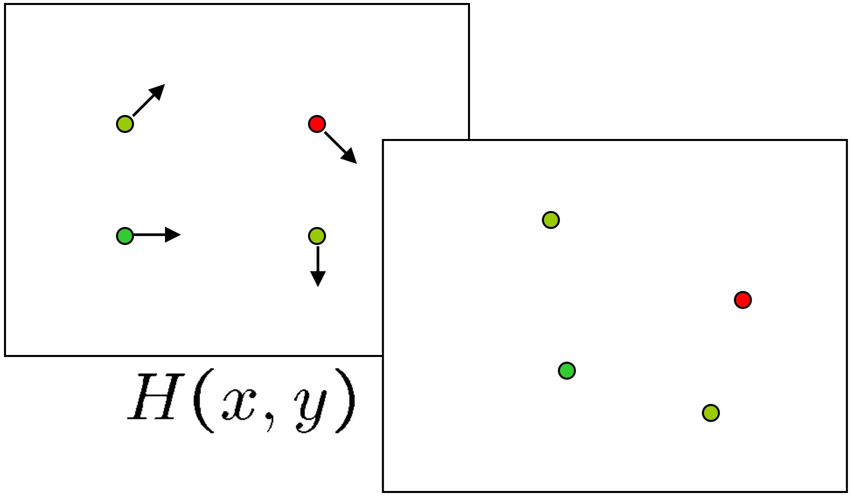
Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

Optical flow

- Optical flow is the apparent motion of brightness patterns (or colours) in the image.
- Ideally, optical flow should be the same as the motion field.
- Careful: apparent motion can be caused by lighting changes without any actual motion.
- To estimate pixel motion from image we have to solve the pixel correspondence problem.
- Given a pixel in frame t, look for nearby pixels with same characteristics (colour, brightness, ...) in frame $t-1$.

Problem Definition: Optical Flow

- How to estimate pixel motion from image H to image I?
  - Find pixel correspondences
    - Given a pixel in H, look for nearby pixels of the same color in I



$H(x, y)$

$I(x, y)$

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

Lucas-Kanade method

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

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

color constancy

$$\underset{\substack{A \\ 25\times 2}}{\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix}} \underset{\substack{d \\ 2\times 1}}{\begin{bmatrix} u \\ v \end{bmatrix}} = - \underset{\substack{b \\ 25\times 1}}{\begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}}$$

small motion constant within neighborhood

Lucas-Kanade method

- Prob: we have more equations than unknowns

$$\underset{25\times2}{A}\ \underset{2\times1}{d} = \underset{25\times1}{b} \longrightarrow \text{minimize } \|Ad - b\|^2$$

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

$$\underset{2\times2}{(A^T A)}\ \underset{2\times1}{d} = \underset{2\times1}{A^T b}$$

$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = -\underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

  - The sum are over all pixels in the K x K window
  - This technique was first proposed by Lukas & Kanade (1981)
    - described in Trucco & Verri reading

Lucas-Kanade method – Conditions for solvability

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

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

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

When is This Solvable?

- **$A^T A$** should be invertible
- **$A^T A$** should not be too small due to noise
  – eigenvalues λ_1 and λ_2 of **$A^T A$** should not be too small
- **$A^T A$** should be well-conditioned
  – λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Summary

- Optical flow:
  - Algorithms try to approximate the true motion field of the image plane.
  - The Optical Flow Constraint Equation needs additional constraints (e.g. smoothness, constant local flow).
  - The Lucas Kanade method is the most popular Optical Flow Algorithm.

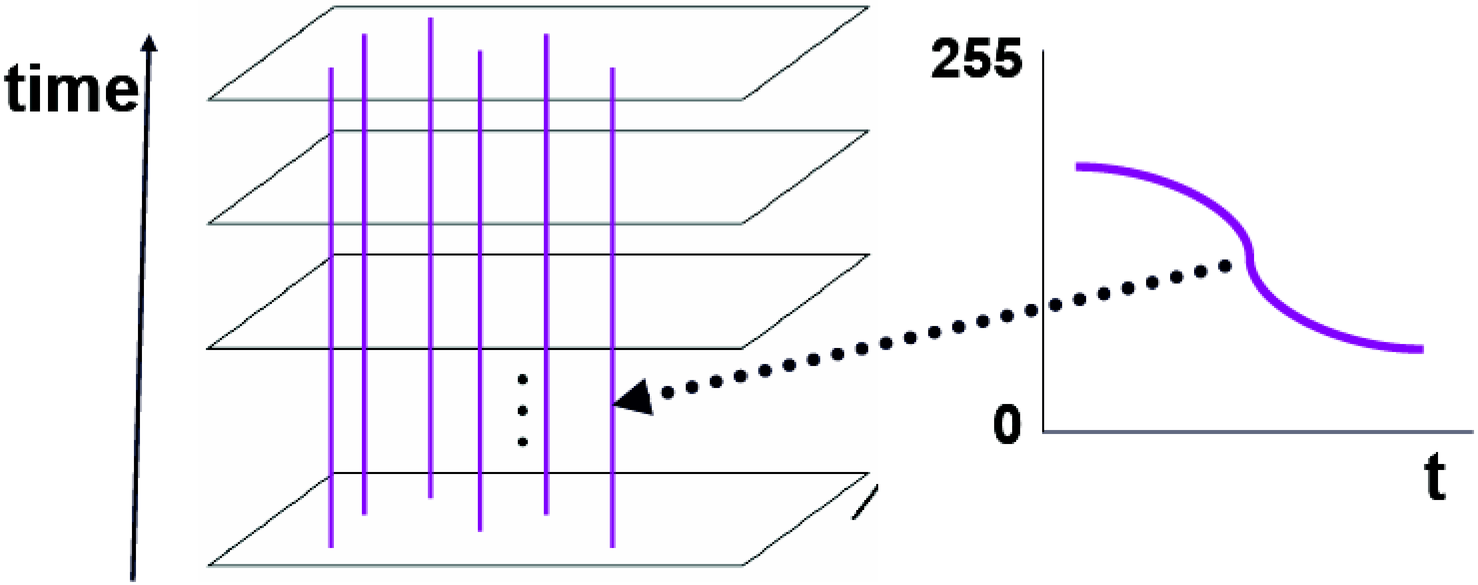
- Lucas Kanade Optical Flow in OpenCV
  - https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html

Sumary

- Motion
  - Motion analysis
  - Optical Flow
- Background Subtraction
- Tracking
  - Object Tracking
  - Template Matching
  - Detection vs Tracking

Background Subtraction

- It is possible to look at video data as a spatio-temporal volume.
- If camera is stationary, each line through time corresponds to a single ray in space.

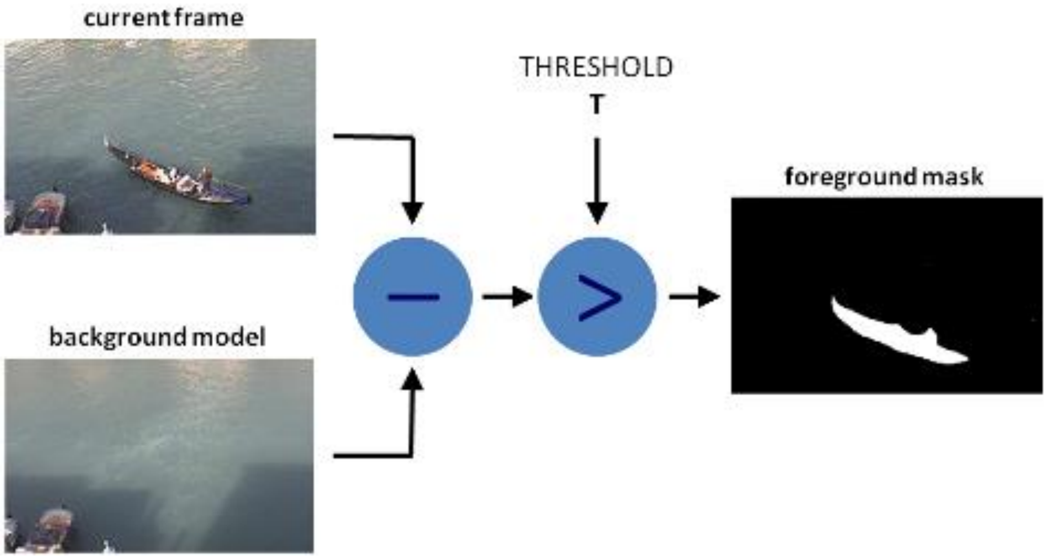


Background Substraction

- Background subtraction techniques are commonly used for segmenting out objects of interest in a static camera scene for applications such as:
    - Surveillance
    - Robot vision
    - Object tracking
    - Traffic applications
    - Human motion capture
    - Augmented reality

Background Subtraction

- Used for generating a foreground mask (binary image) containing moving objects in static camera setups.
- Name comes from the simple technique of subtracting the observed image from the estimated image and thresholding the result to generate the objects of interest.

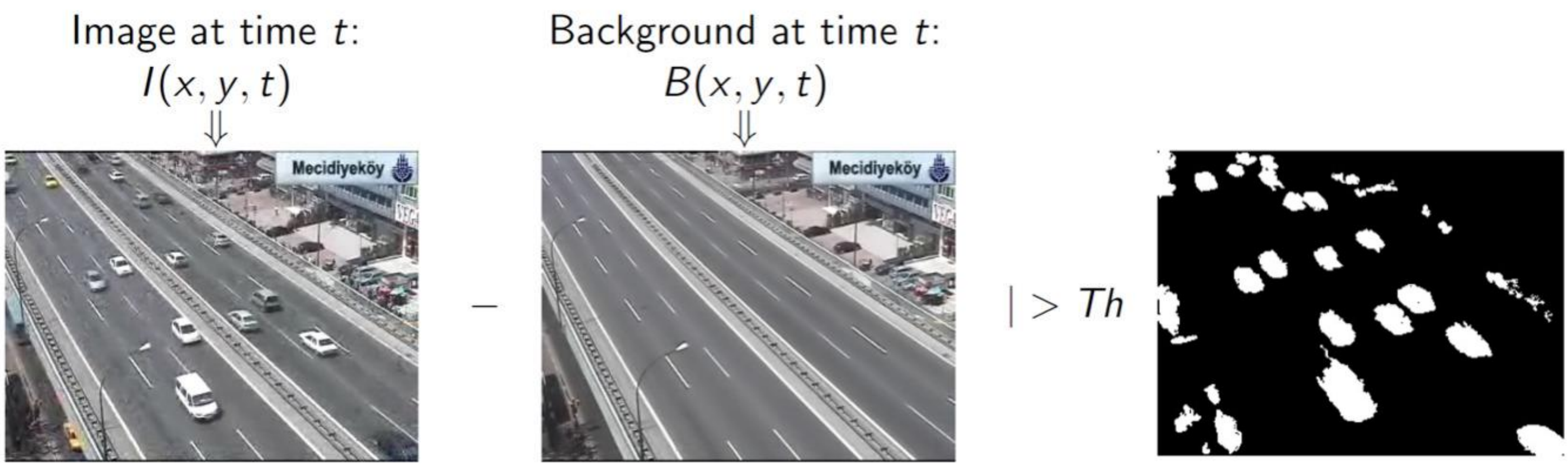


Background Subtraction

- Foreground detection – how the object areas are distinguished from the background
- Background maintenance – how the background is maintained over time
- Post-processing – how the segmented object areas are detected

Background Subtraction

- Generic algorithm:
  - Create an image of the stationary background by averaging a long sequence.
  - Subtract current frame and known background frame
  - Motion detection algorithms such as these only work if the camera is stationary and objects are moving against a fixed background



Background Subtraction

- Generic algorithm:
  - With frame differencing, background is estimated to be the previous frame. Background subtraction equation becomes $\boldsymbol{B(x,y,T) = I(x,y,t-1)}$ and $\boldsymbol{|I(x,y,t) - I(x,y,t-1)| > Th}$
  - Depending on the object structure, speed, frame rate and global threshold may or may not be useful (usually not).
  - Another approach is to model the background using a running average. A pixel is marked as foreground if

    $$\boldsymbol{|I_t - B_t| > th}$$

    where ***th*** is a predefined threshold.

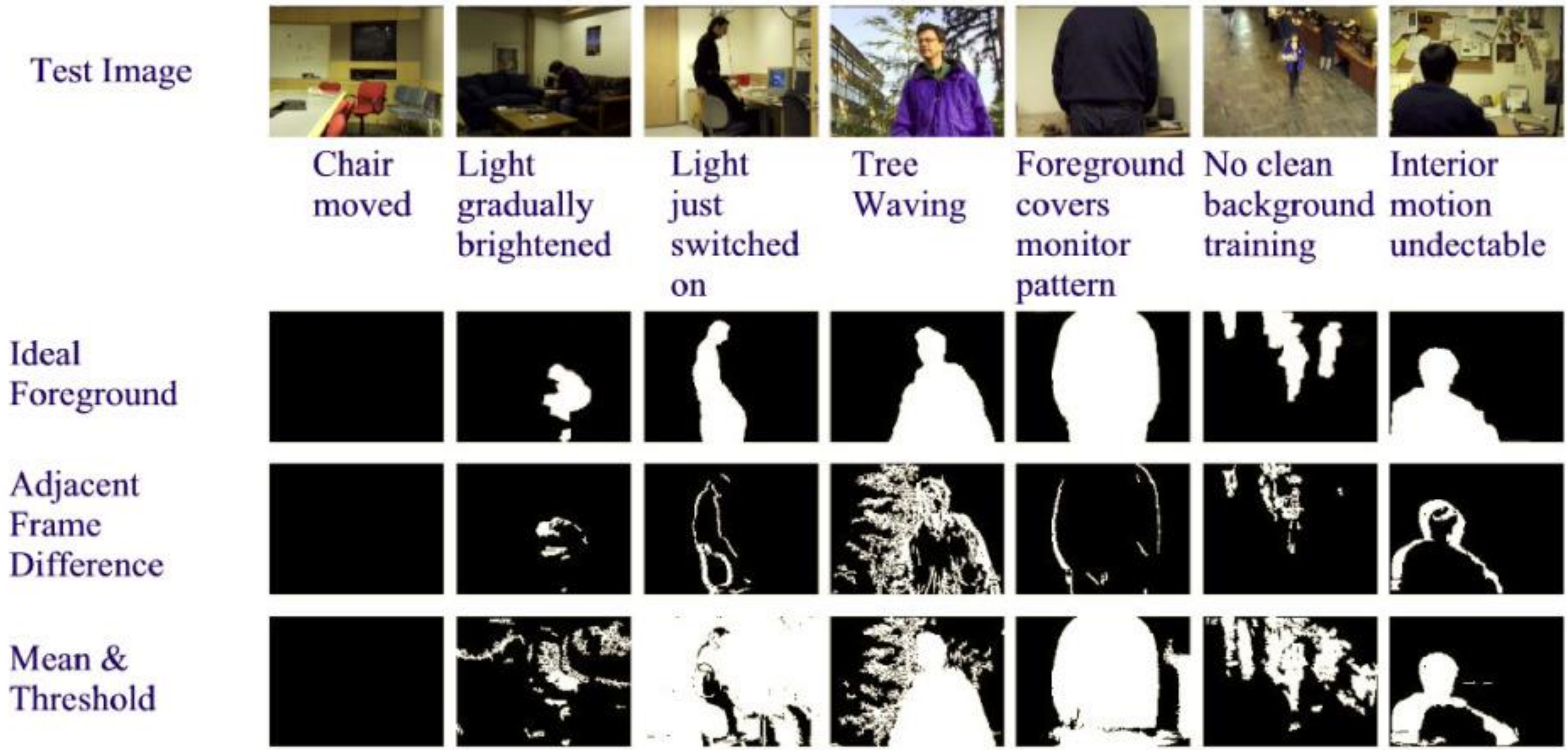
  The thresholding is often followed by morphological closing with a 3x3 kernel and the discarding of small regions
  - The background update is

    $$\boldsymbol{B_{t-1} = \alpha I_t + (1-\alpha) B_t}$$

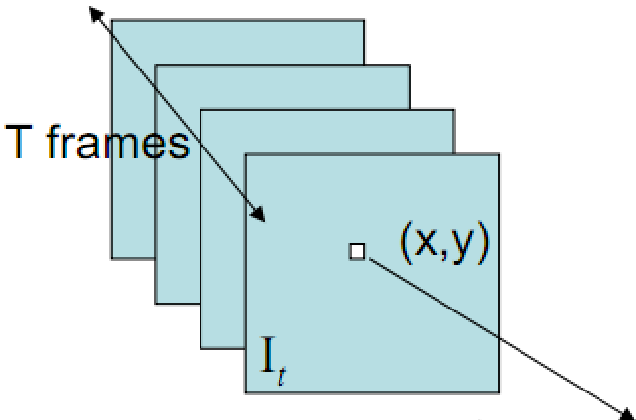
    where $\boldsymbol{\alpha}$ is kept small to prevent artificial tails forming behind moving object

Background Subtraction

- Examples



- Adaptive Mixture of Gaussians



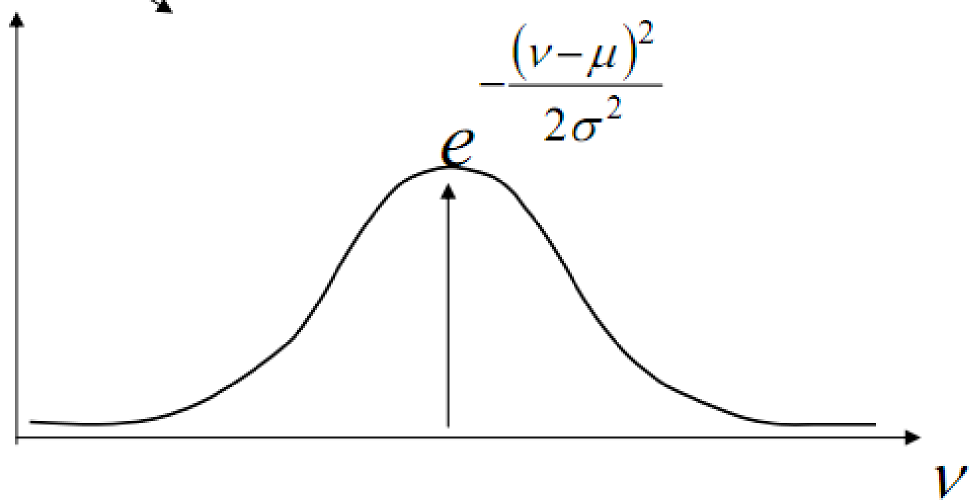
Background model

$$\mu(x,y) = \frac{1}{T}\sum_{t} I_t(x,y)$$

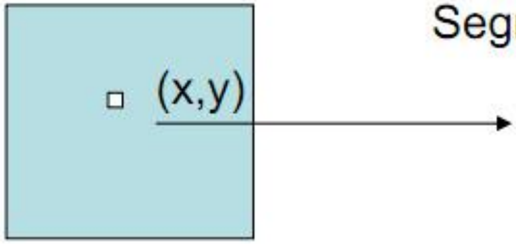
$$\sigma^2(x,y) = \frac{1}{T}\sum_{t} (I_t(x,y) - \mu(x,y))^2$$

$$P_b(I(x,y) = \nu)$$

$$e^{-\frac{(\nu-\mu)^2}{2\sigma^2}}$$



New frame



Segmentation - moving objects:

$$\left|I(x,y) - \mu(x,y)\right| > 2\sigma$$

Estimated background

The most probable background image

dominant Gaussian mean for each pixel's mixture model

Sumary

- Motion
  - Motion analysis
  - Optical Flow

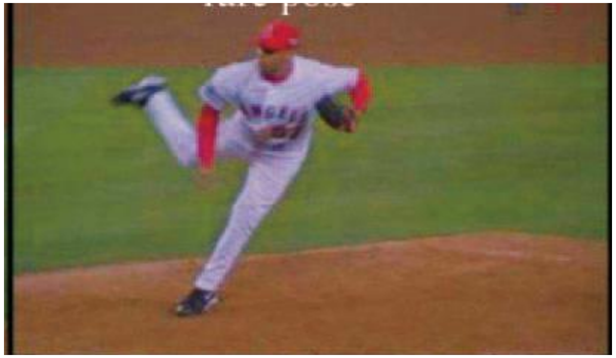
- Background Subtraction

- Tracking
  - Object Tracking
  - Template Matching
  - Detection vs Tracking

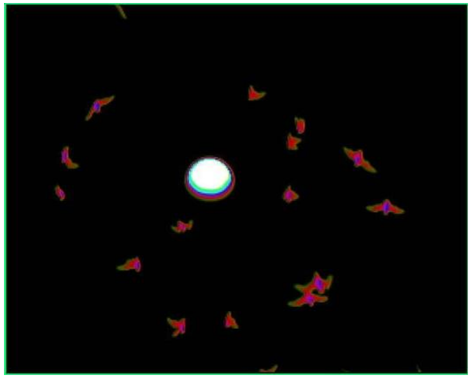
Object Tracking

- Object tracking is a crucial issue in computer vision, especially for the applications where the environment is in continuous changing:
  - Robot Vision - mobile robot navigation, applications that must deal with unstable grasps, . . .
  - Surveillance
  - Traffic applications
  - Human motion capture

Tracking: some applications

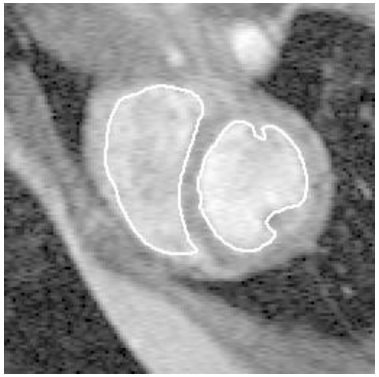


Body pose tracking, activity recognition



Censusing a bat population

Video-based interfaces

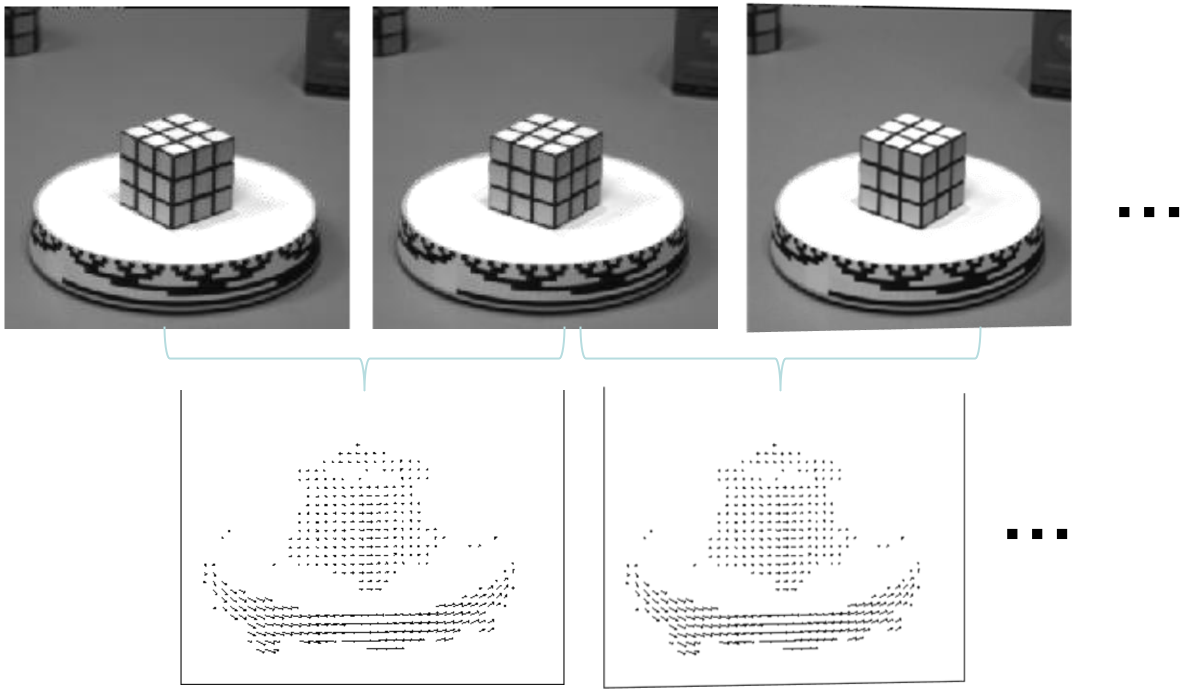


Medical apps

Surveillance

Optical flow for tracking?

If we have more than just a pair of frames, we could compute optical flow from one to the other next



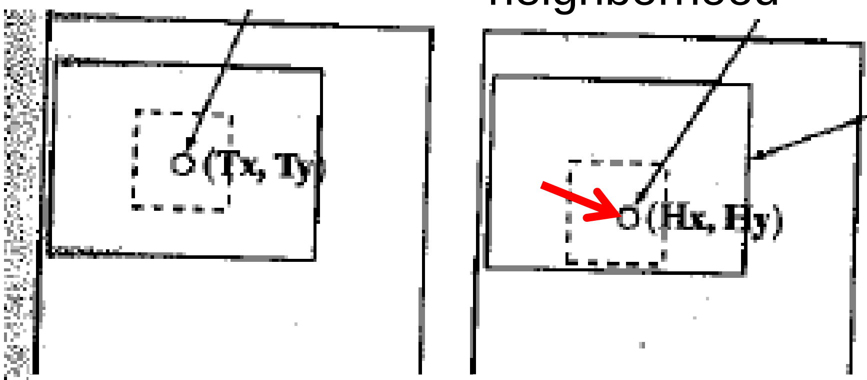
But flow only reliable for small motions, and we may have occlusions, texture less regions that yield bad estimates anyway…

Feature-based matching for motion

Block matching or template matching (see previous class)

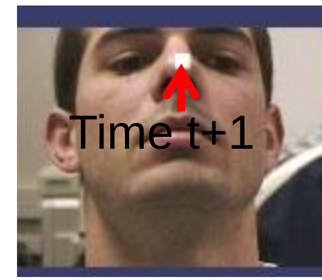
Interesting point

Best matching neighborhood



Search window

Time t



Search window is centered at the point where we last saw the feature, in image I1.

Best match = position where we have the highest normalized cross-correlation value.

Example: A Camera Mouse

- Video interface: use feature tracking as mouse replacement

- User clicks on the feature to be tracked
- Take the 15x15 pixel square of the feature
- In the next image do a search to find the 15x15 region with the highest correlation
- Move the mouse pointer accordingly
- Repeat in the background every 1/30th of a second

James Gips and Margrit Betke

Detection vs. tracking

t=1

t=2

…

t=20

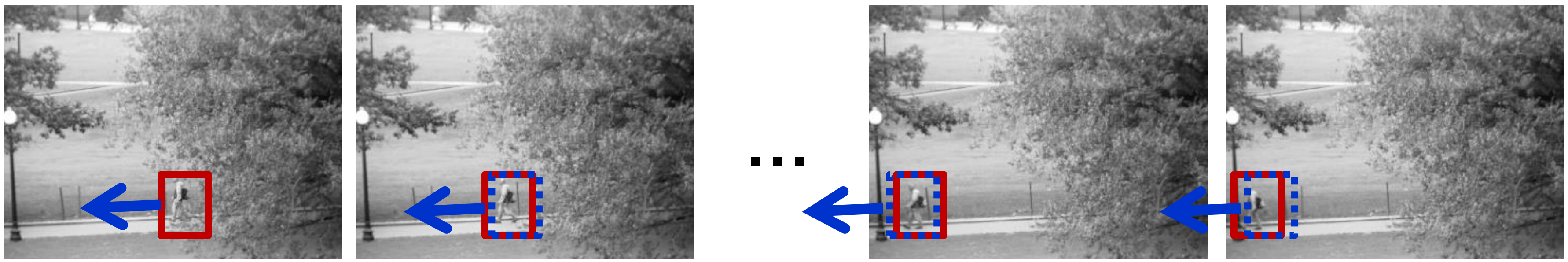
t=21

Detection vs. tracking

Detection: We detect the object independently in each frame and can record its position over time, e.g., based on blob’s centroid or detection window coordinates

Kristen Grauman

Detection vs. tracking



Tracking with *dynamics*: We use image measurements to estimate position of object, but also incorporate position predicted by dynamics, i.e., our expectation of object's motion pattern.

Kristen Grauman

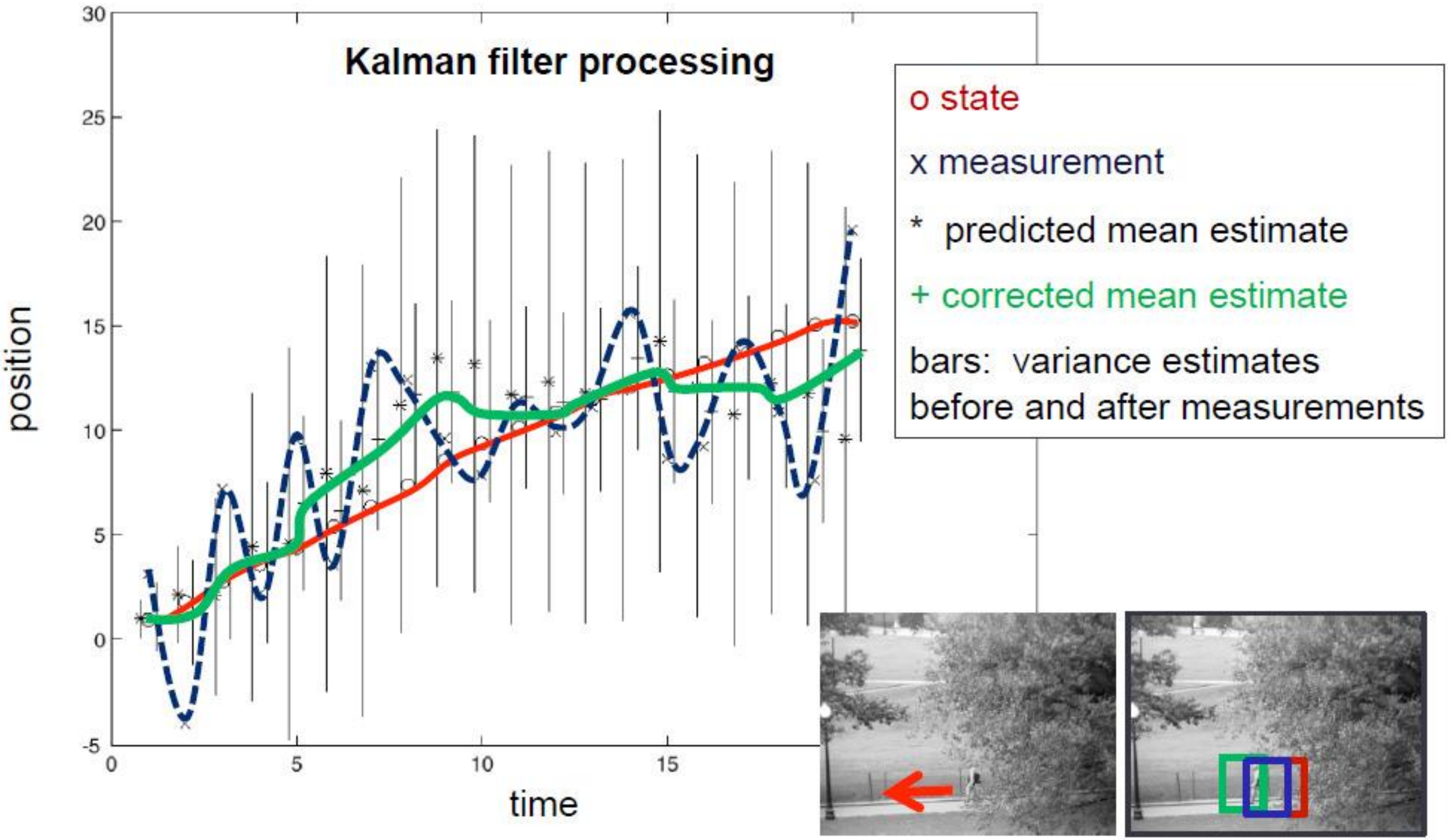
Tracking: Example Kalman Filter

- The hidden state consists of the true parameters we care about, denoted X.
- The measurement is our noisy observation that result from the underlying state, denoted Y.
- At each time step, state changes (from X_{t-1} o X_t) and we get a new observation Yt .

Tracking: Kalman Filter

- Method for tracking linear dynamical models in Gaussian noise
- The predicted/corrected state distributions are Gaussian
- Only need to maintain the mean and covariance

Tracking: Constant velocity model



Time t

Time t+1

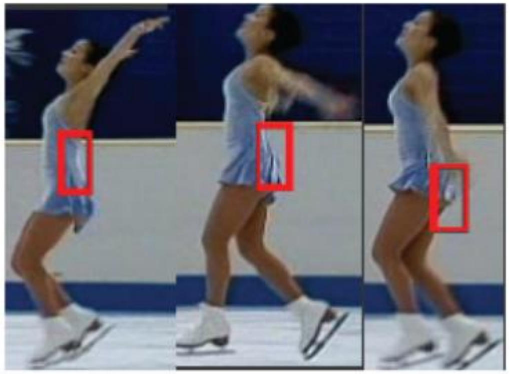
Kristen Grauman

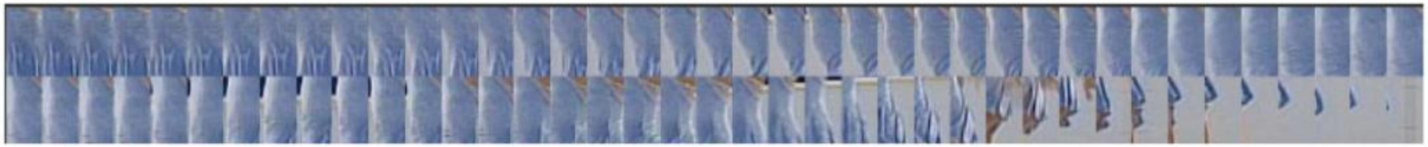
Tracking: issues

- **Initialization**
  - Often done manually
  - Background subtraction, detection can also be used
- **Data association**, multiple tracked objects
  - Occlusions, clutter

Tracking: issues

- **Deformable** and articulated objects
- Constructing **accurate models** of dynamics - (example: Fitting parameters for a linear dynamics model)
- **Drift** - accumulation of errors over time





Resources

- Barron, “Tutorial: Computing 2D and 3D Optical Flow.”, http://www.tina-vision.net/docs/memos/2004-012.pdf
- CVonline: Optical Flow
- Fast Image Motion Estimation Demo

  http://extra.cmis.csiro.au/IA/changs/motion/
- Part of this lecture is from the courses offered by Prof. Shree Nayar at Columbia University, USA and by Prof. Srinivasa Narasimhan at CMU, USA.