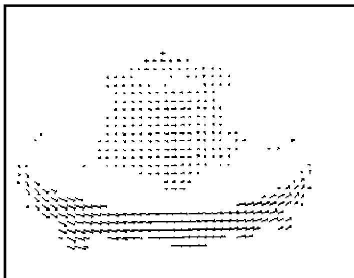


Motion and optical flow

Thurs Feb 2, 2017
Kristen Grauman
UT Austin

Announcements

- A1 due tomorrow, Friday
- Due to AAAI travel
  - Office hours Tues Feb 7 cancelled (by appt)
  - Lecture Tues is ON as normal

Last time

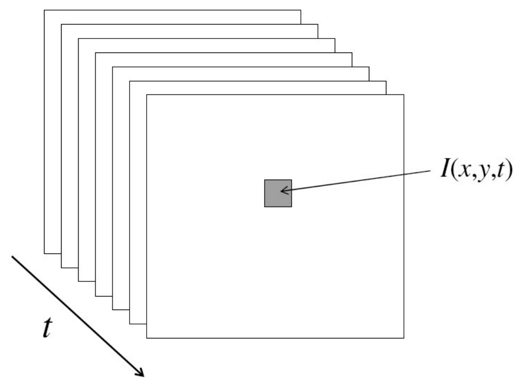
- Texture is a useful property that is often indicative of materials, appearance cues
- **Texture representations** attempt to summarize repeating patterns of local structure
- **Filter banks** useful to measure redundant variety of structures in local neighborhood
  - Feature spaces can be multi-dimensional
- Neighborhood statistics can be exploited to “sample” or **synthesize** new texture regions
  - Example-based technique

Today

- Optical flow: estimating motion in video
- Background subtraction

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)

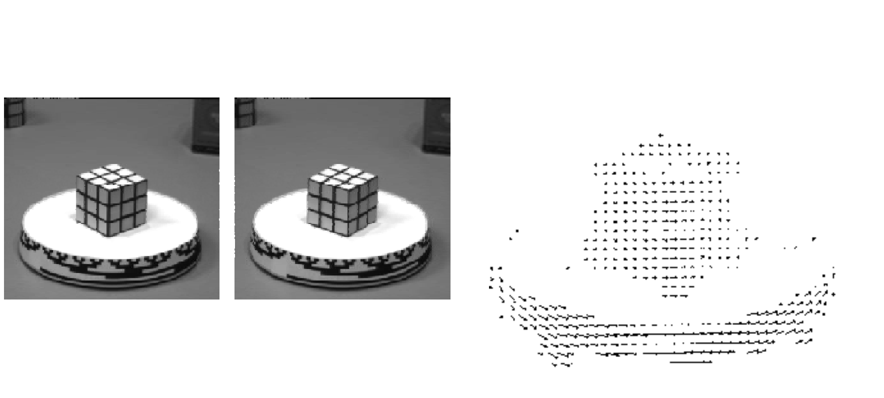


Uses of motion

- Estimating 3D structure
- Segmenting objects based on motion cues
- Learning dynamical models
- Recognizing events and activities
- Improving video quality (motion stabilization)

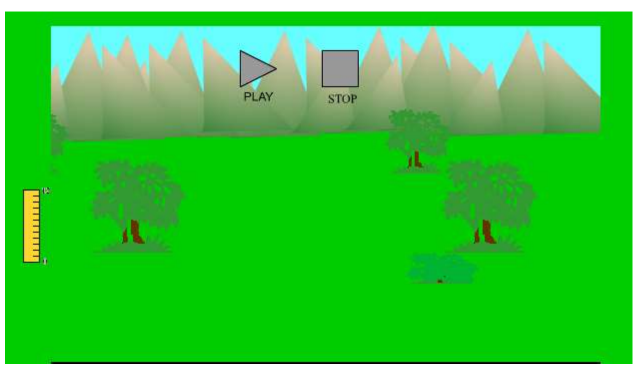
Motion field

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



Motion parallax

http://psych.hanover.edu/KRANTZ/MotionParallax/MotionParallax.html



Motion field + camera motion

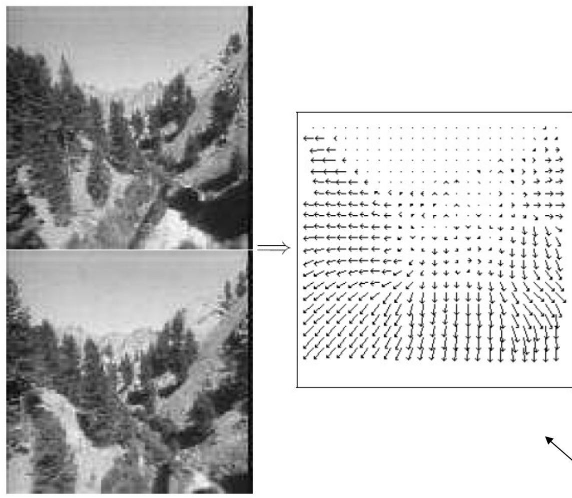


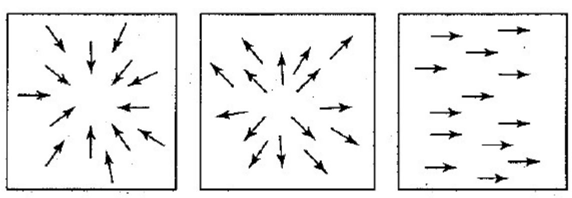
Figure 1.2: Two images taken from a helicopter flying through a canyon and the computed optical flow field.

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 from Michael Black, Ph.D. Thesis

Motion field + camera motion



Zoom out **Zoom in** **Pan right to left**

Motion estimation techniques

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small
- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

Optical flow

- Definition: optical flow is the *apparent* motion of brightness patterns in the image
- Ideally, optical flow would be the same as the motion field
- Have to be careful: apparent motion can be caused by lighting changes without any actual motion

Apparent motion != motion field

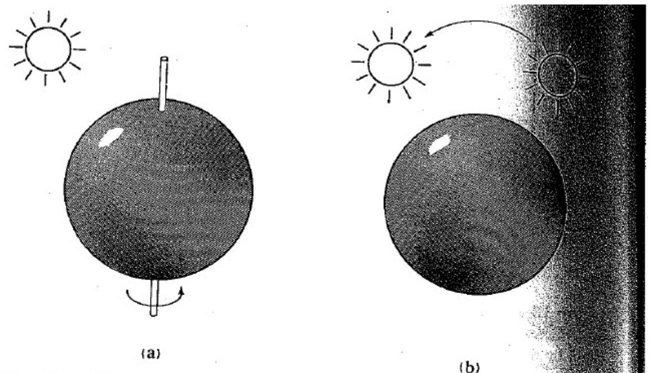
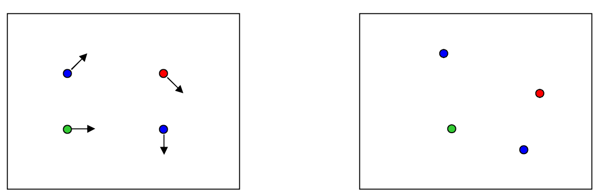


Figure 12-2. The optical flow is not always equal to the motion field. In (a) a smooth sphere is rotating under constant illumination—the image does not change, yet the motion field is nonzero. In (b) a fixed sphere is illuminated by a moving source—the shading in the image changes, yet the motion field is zero.

Figure from Horn book

Problem definition: optical flow



$H(x, y)$ $\quad$ $I(x, y)$

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

- Solve pixel correspondence problem
  - given a pixel in H, look for nearby pixels of the same color in I

Key assumptions

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

This is called the **optical flow** problem

Slide credit: Steve Seitz

Brightness constancy

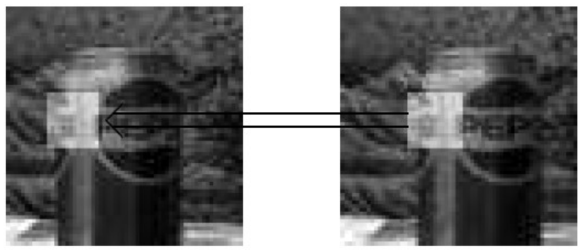
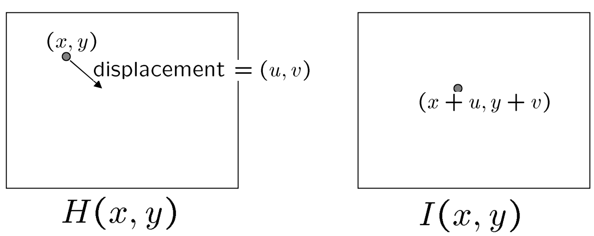


Figure 1.5: Data conservation assumption. The highlighted region in the right image looks roughly the same as the region in the left image, despite the fact that it has moved.

Figure by Michael Black

Optical flow constraints



Let's look at these constraints more closely

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

$$H(x, y) = I(x+u, y+v)$$

- small motion:

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

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

Slide credit: Steve Seitz

Optical flow equation

Combining these two equations

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

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

Slide credit: Steve Seitz

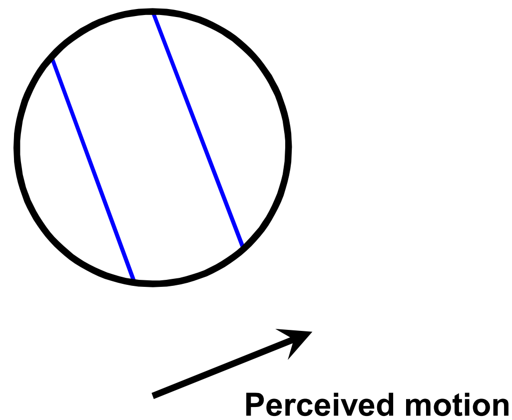
Optical flow equation

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

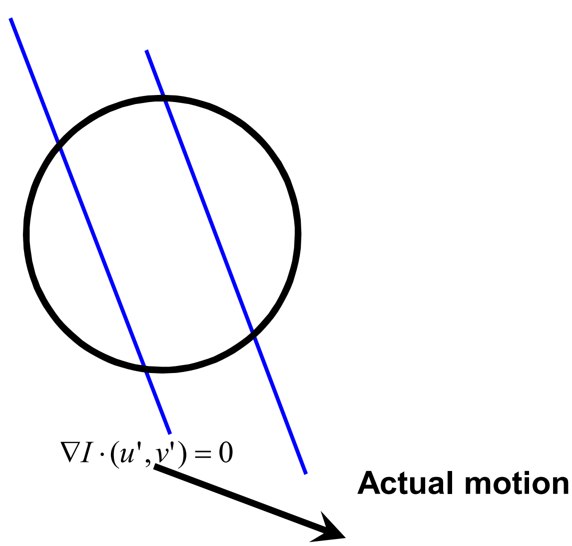
Q: how many unknowns and equations per pixel?

Slide credit: Steve Seitz

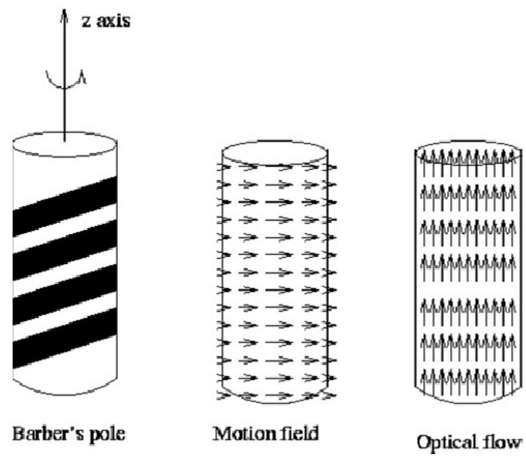
The aperture problem



The aperture problem



The barber pole illusion



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

Solving the aperture problem

- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)

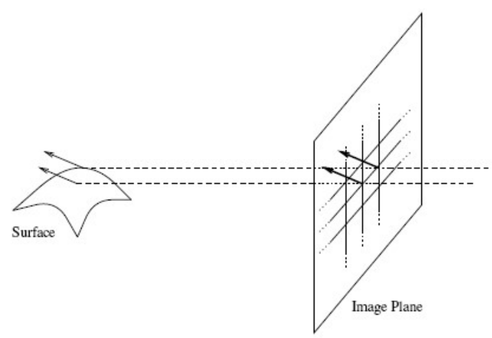


Figure 1.7: Spatial coherence assumption. Neighboring points in the image are assumed to belong to the same surface in the scene.

Figure by Michael Black

Solving the aperture problem

- How to get more equations for a pixel?
- **Spatial coherence constraint:** pretend the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

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

$$\begin{bmatrix} I_x(\mathbf{p_1}) & I_y(\mathbf{p_1}) \\ I_x(\mathbf{p_2}) & I_y(\mathbf{p_2}) \\ \vdots & \vdots \\ I_x(\mathbf{p_{25}}) & I_y(\mathbf{p_{25}}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p_1}) \\ I_t(\mathbf{p_2}) \\ \vdots \\ I_t(\mathbf{p_{25}}) \end{bmatrix}$$

$$\underset{25\text{x}2}{A} \ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Slide credit: Steve Seitz

Solving the aperture problem

Prob: we have more equations than unknowns

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

Solution: solve least squares problem

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

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

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

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

Slide credit: Steve Seitz

Conditions for solvability

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

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

When is this solvable?

- $\mathbf{A^T A}$ should be invertible
- $\mathbf{A^T A}$ should not be very small
  – eigenvalues λ_1 and λ_2 of $\mathbf{A^T A}$ should not be very small
- $\mathbf{A^T A}$ should be well-conditioned
  – λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Slide by Steve Seitz, UW

Edge

– gradients very large or very small
– large λ_1, small λ_2

Slide credit: Steve Seitz

Low-texture region

– gradients have small magnitude
– small λ_1, small λ_2

Slide credit: Steve Seitz

High-texture region

– gradients are different, large magnitudes
– large λ_1, large λ_2

Slide credit: Steve Seitz

Example use of optical flow: facial animation

http://www.fxguide.com/article333.html

Example use of optical flow: Motion Paint

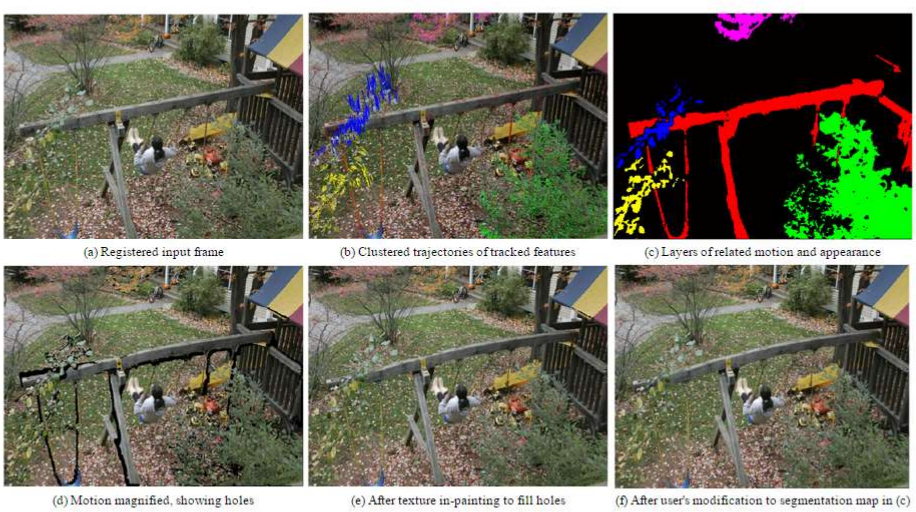
Use optical flow to track brush strokes, in order to animate them to follow underlying scene motion.

http://www.fxguide.com/article333.html

Motion estimation techniques

- Direct methods
  - Directly recover image motion at each pixel from spatio-temporal image brightness variations
  - Dense motion fields, but sensitive to appearance variations
  - Suitable for video and when image motion is small
- Feature-based methods
  - Extract visual features (corners, textured areas) and track them over multiple frames
  - Sparse motion fields, but more robust tracking
  - Suitable when image motion is large (10s of pixels)

Motion magnification



(a) Registered input frame (b) Clustered trajectories of tracked features (c) Layers of related motion and appearance

(d) Motion magnified, showing holes (e) After texture in-painting to fill holes (f) After user's modification to segmentation map in (c)

Liu et al. SIGGRAPH 2005

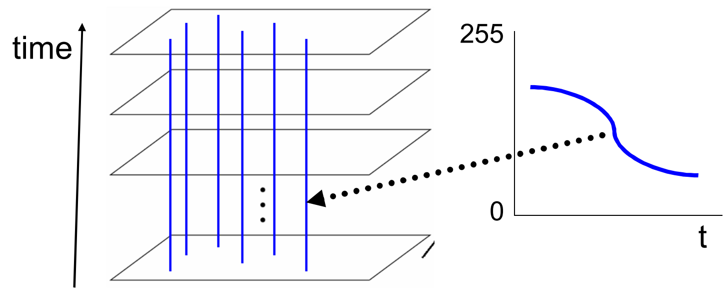
Fun with flow

- https://www.youtube.com/watch?v=3YE5tff8pqg
- http://www.youtube.com/watch?v=TbJrc6QCeU0&feature=related
- http://www.youtube.com/watch?v=pckFacsIWg4

Today

- Optical flow: estimating motion in video
- Background subtraction

Video as an “Image Stack”



Can look at video data as a spatio-temporal volume

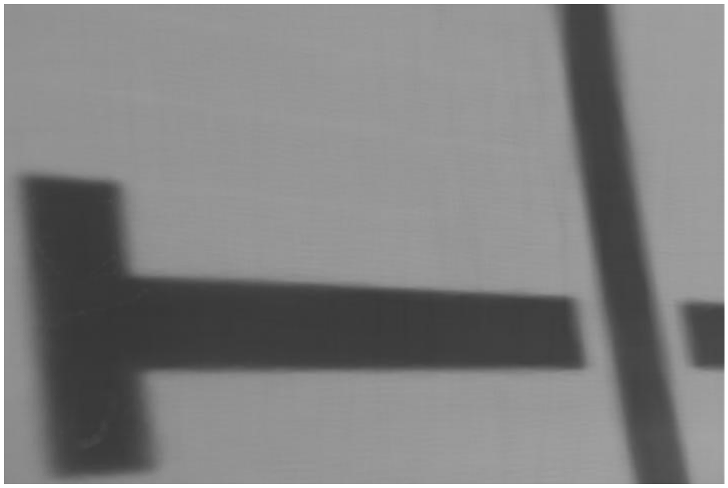
- If camera is stationary, each line through time corresponds to a single ray in space

Alyosha Efros, CMU

Input Video

Alyosha Efros, CMU

Average Image



Alyosha Efros, CMU

Background Subtraction

- Given an image (mostly likely to be a video frame), we want to identify the **foreground objects** in that image!

$\Rightarrow$

Motivation

- In most cases, objects are of interest, not the scene.
- Makes our life easier: less processing costs, and less room for error.

Slide credit: Birgi Tamersoy

Background subtraction

- Simple techniques can do ok with static camera
- …But hard to do perfectly

- Widely used:
  - Traffic monitoring (counting vehicles, detecting & tracking vehicles, pedestrians),
  - Human action recognition (run, walk, jump, squat),
  - Human-computer interaction
  - Object tracking

Simple Approach

Image at time t: $I(x, y, t)$ $\Downarrow$

Background at time t: $B(x, y, t)$ $\Downarrow$

$|$ 

$-$ 

$| > Th$

1. Estimate the background for time t.
2. Subtract the estimated background from the input frame.
3. Apply a threshold, Th, to the absolute difference to get the **foreground mask**.

Slide credit: Birgi Tamersoy

Frame Differencing

- Background is estimated to be the previous frame. Background subtraction equation then becomes:

$$B(x, y, t) = I(x, y, t-1)$$
$$\Downarrow$$
$$|I(x, y, t) - I(x, y, t-1)| > Th$$

- Depending on the object structure, speed, frame rate and global threshold, this approach may or may **not** be useful (usually **not**).

$|$

$-$

$| > Th$

Slide credit: Birgi Tamersoy

Frame Differencing

$Th = 25$

$Th = 50$

$Th = 100$

$Th = 200$

Slide credit: Birgi Tamersoy

Mean Filter

- In this case the background is the mean of the previous n frames:

$$B(x, y, t) = \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t-i)$$
$$\Downarrow$$
$$|I(x, y, t) - \frac{1}{n} \sum_{i=0}^{n-1} I(x, y, t-i)| > Th$$

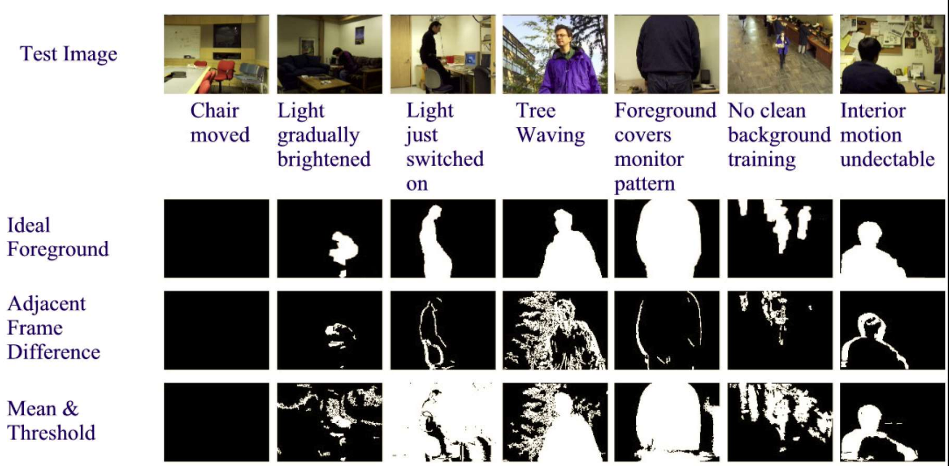
- For $n = 10$:

Estimated Background

Foreground Mask

Slide credit: Birgi Tamersoy

Frame differences vs. background subtraction



- Toyama et al. 1999

Median Filter

- Assuming that the background is more likely to appear in a scene, we can use the median of the previous n frames as the background model:

$$B(x, y, t) = median\{I(x, y, t - i)\}$$
$$\Downarrow$$
$$|I(x, y, t) - median\{I(x, y, t - i)\}| > Th \text{ where } i \in \{0, \dots, n - 1\}.$$

- For $n = 10$:

Estimated Background

Foreground Mask

Slide credit: Birgi Tamersoy

Average/Median Image

Alyosha Efros, CMU

Background Subtraction

-

=

Alyosha Efros, CMU

Pros and cons

Advantages:

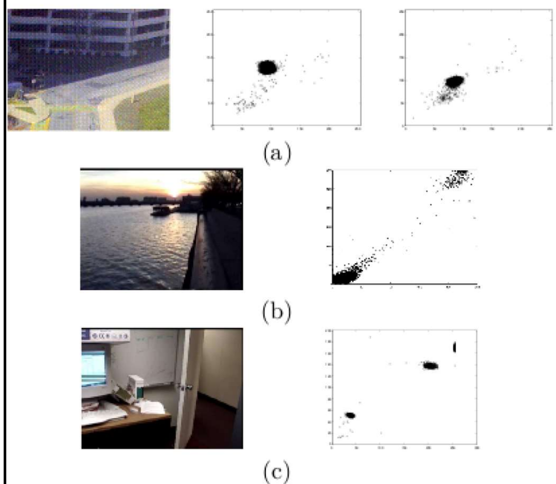
- Extremely easy to implement and use!
- All pretty fast.
- Corresponding background models need not be constant, they change over time.

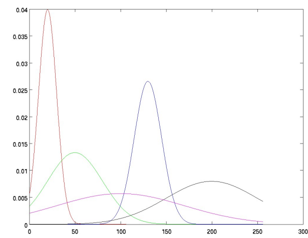
Disadvantages:

- Accuracy of frame differencing depends on object speed and frame rate
- Median background model: relatively high memory requirements.
- Setting global threshold Th…

When will this basic approach fail?

Background mixture models

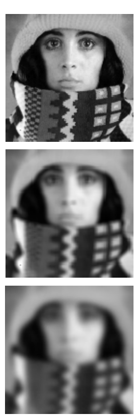


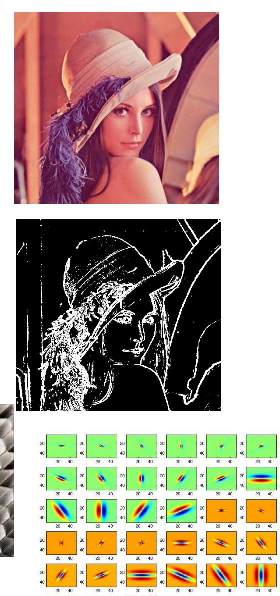


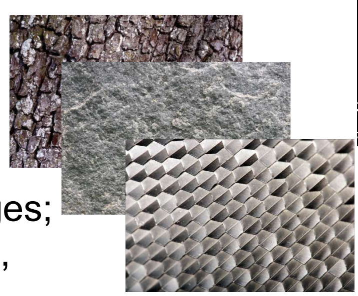
Idea: model each background pixel with a *mixture* of Gaussians; update its parameters over time.

Adaptive Background Mixture Models for Real-Time Tracking, 1999,
Chris Stauer & W.E.L. Grimson

So far: features and filters







Transforming images; gradients, textures, edges, flow

Slide credit: Kristen Grauman