EE795: Computer Vision and Intelligent Systems

Spring 2012
TTh 17:30-18:45 FDH 204

Lecture 16
130321

http://www.ee.unlv.edu/~b1morris/ecg795/

Outline

- Review
  - Optical Flow
- Background Subtraction

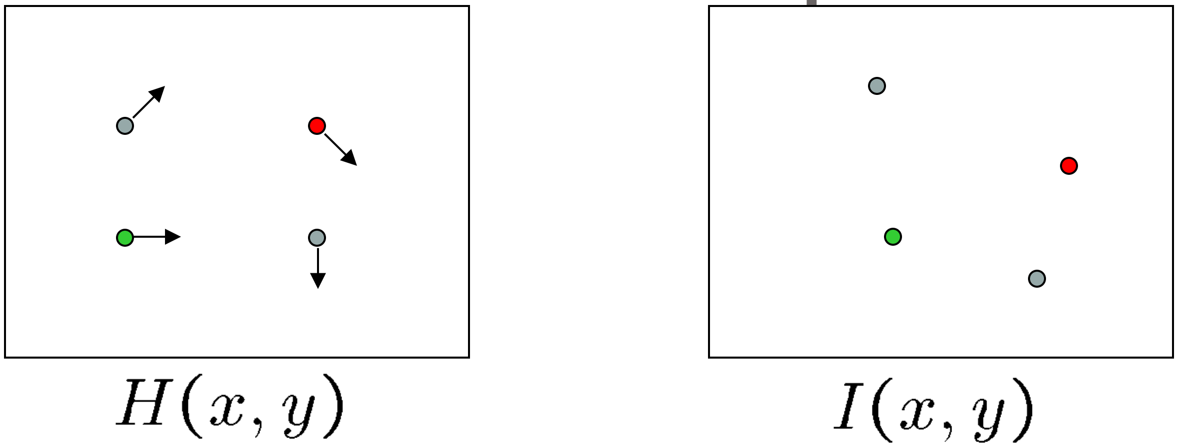
Motion estimation

- Input: sequence of images
- Output: point correspondence

- Feature correspondence: “Feature Tracking”
  - we’ve seen this already (e.g., SIFT)
  - can modify this to be more accurate/efficient if the images are in sequence (e.g., video)

- Pixel (dense) correspondence: “Optical Flow”

Problem definition: optical flow



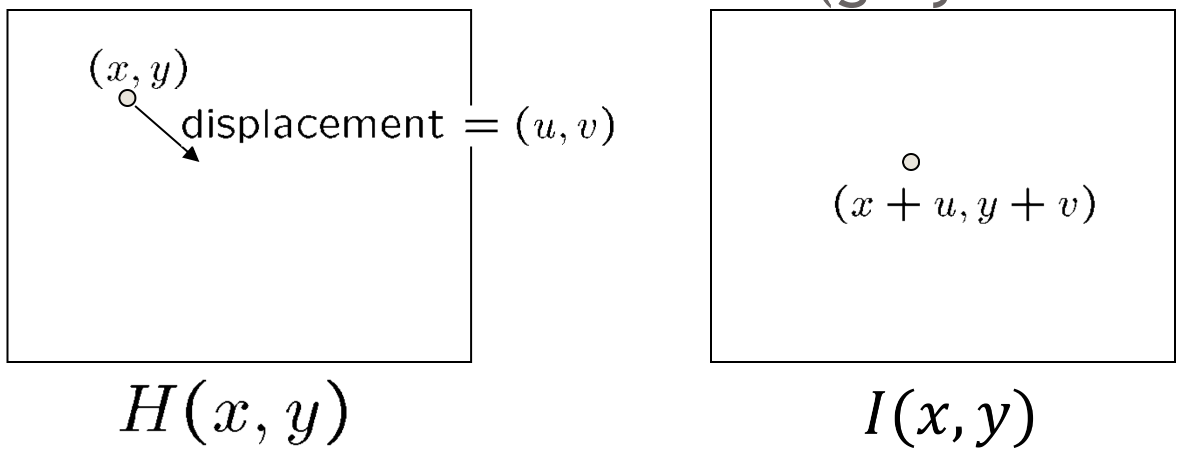
- How to estimate pixel motion from image H to image I?
  - Solve pixel correspondence problem
    - given a pixel in H, look for nearby pixels of the same color in I

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

Optical flow constraints (grayscale images)



- 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: (u and v are less than 1 pixel)
    - suppose we take the Taylor series expansion of I:

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

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

Optical flow equation

- Combining these two equations

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

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

In the limit as u and v go to zero, this becomes exact

$$0 = I_t + \nabla I \cdot [\tfrac{\partial x}{\partial t} \;\; \tfrac{\partial y}{\partial t}]$$

Lucas-Kanade flow

- 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!

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

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

Conditions for solvability

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

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

- When is This Solvable?
  - **A^TA** should be invertible
  - **A^TA** should not be too small due to noise
    - eigenvalues λ_1 and λ_2 of **A^TA** should not be too small
  - **A^TA** should be well-conditioned
    - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

- Does this look familiar?
  - **A^TA** is the Harris matrix

Background Subtraction

- Motion is an important
  - Indicates an object of interest

- Background subtraction
  - Given an image (usually a video frame), identify the **foreground objects** in that image
    - Assume that foreground objects are moving
    - Typically, moving objects more interesting than the scene
    - Simplifies processing – less processing cost and less room for error

Background Subtraction Example

- Often used in traffic monitoring applications
  - Vehicles are objects of interest (counting vehicles)

$\Longrightarrow$

- Human action recognition (run, walk, jump, …)
- Human-computer interaction ("human as interface")
- Object tracking

Requirements

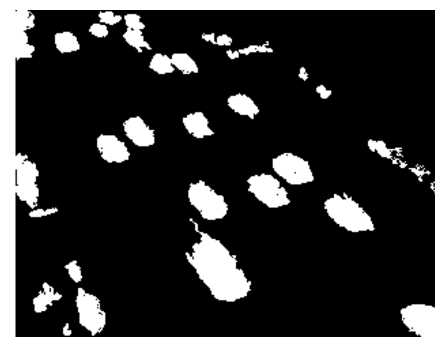
- A reliable and robust background subtraction algorithm should handle:
  - Sudden or gradual illumination changes
    - Light turning on/off, cast shadows through a day
  - High frequency, repetitive motion in the background
    - Tree leaves blowing in the wind, flag, etc.
  - Long-term scene changes
    - A car parks in a parking spot

Basic Approach

- Estimate the background at time t
- Subtract the estimated background from the current input frame
- Apply a threshold, Th, to the absolute difference to get the foreground mask.
  - $I(x, y, t) - B(x, y, t)| > Th = F(x, y, t)$

$|$ 

$-$ 

$| > Th =$ 

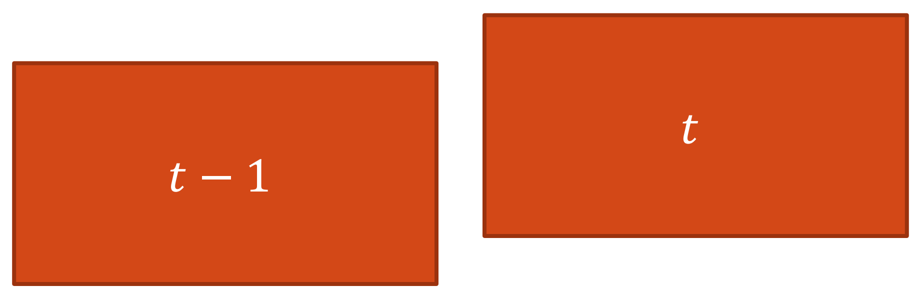
$I(x, y, t)$ $\quad$ $B(x, y, t)$ $\quad$ $F(x, y, t)$

How can we estimate the background?

Frame Differencing

- Background is estimated to be the previous frame
  - $B(x, y, t) = I(x, y, t-1)$
- Depending on the object structure, speed, frame rate, and global threshold, may or may not be useful
  - Usually not useful – generates impartial objects and ghosts

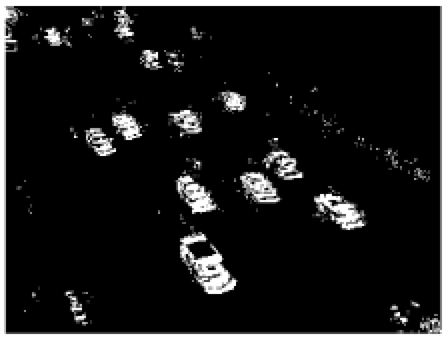
Incomplete object



ghosts

Frame Differencing Example

$Th = 25$



$Th = 50$

$Th = 100$

$Th = 200$

Mean Filter

- Background is the mean of the previous N frames
  - $B(x,y,t) = \frac{1}{N}\sum_{i=0}^{N-1} I(x,y,t-i)$
  - Produces a background that is a temporal smoothing or “blur”
- $N = 10$

Estimated Background

Foreground Mask

Mean Filter

- $N = 20$

Estimated Background

Foreground Mask

- $N = 50$

Estimated Background

Foreground Mask

Median Filter

- Assume the background is more likely to appear than foreground objects
  - $B(x,y,t) = median\big(I(x,y,t-i)\big),\ i \in \{0, N-1\}$

- $N = 10$

Estimated Background

Foreground Mask

Median Filter

- $N = 20$

Estimated Background

Foreground Mask

- $N = 50$

Estimated Background

Foreground Mask

Frame Difference Advantages

- Extremely easy to implement and use
- All the described variants are pretty fast
- The background models are not constant
  - Background changes over time

Frame Differencing Shortcomings

- Accuracy depends on object speed/frame rate
- Mean and median require large memory
  - Can use a running average
  - $B(x, y, t) = (1 - \alpha)B(x, y, t - 1) + \alpha I(x, y, t)$
    - α – is the learning rate
- Use of a global threshold
  - Same for all pixels and does not change with time
  - Will give poor results when the:
    - Background is bimodal
    - Scene has many slow moving objects (mean, median)
    - Objects are fast and low frame rate (frame diff)
    - Lighting conditions change with time

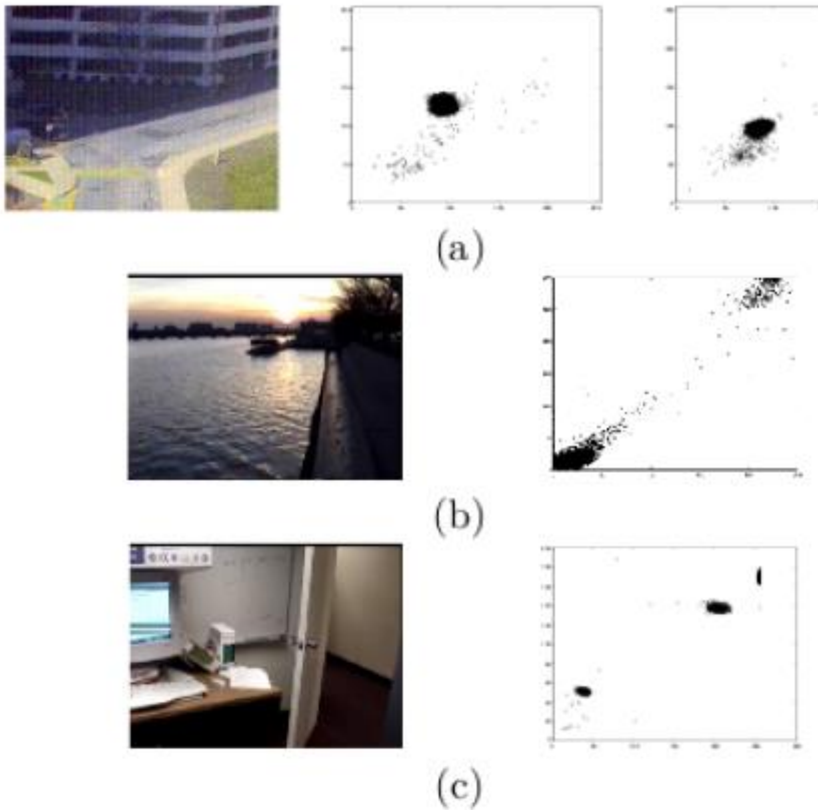
Improving Background Subtraction

- Adaptive Background Mixture Models for Real-Time Tracking
  - Chris Stauffer and W.E.L. Grimson

- The paper on background subtraction
  - Over 4000 citations since 1999

Motivation

- Robust background subtraction should handle lighting changes, repetitive motion from clutter and long term scene changes



Differing threshold over time

RG plots of a single pixel

Bimodal distribution over time

Algorithm Overview

- Pixel value is modeled as a mixture of adaptive Gaussian distributions
  - Why a mixture?
    - Multiple surfaces appear in a pixel (mean background assumes a single pixel distribution)
  - Why adaptive?
    - Lighting conditions change
- Gaussians are evaluated to determine which ones are most likely to correspond to the background
- Pixels that do not match the background Gaussians are classified as foreground

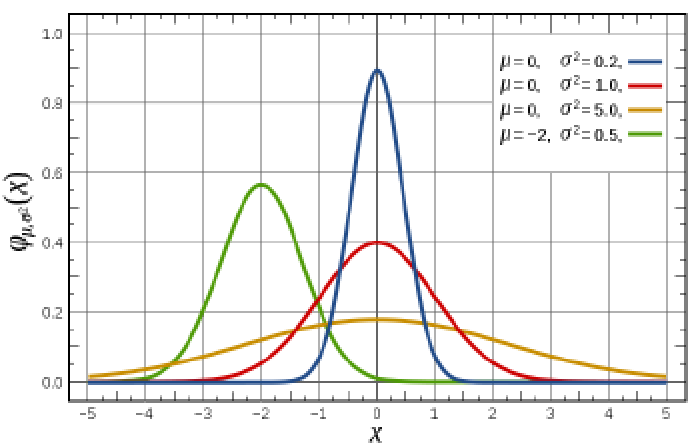
Gaussian (Normal) Distribution

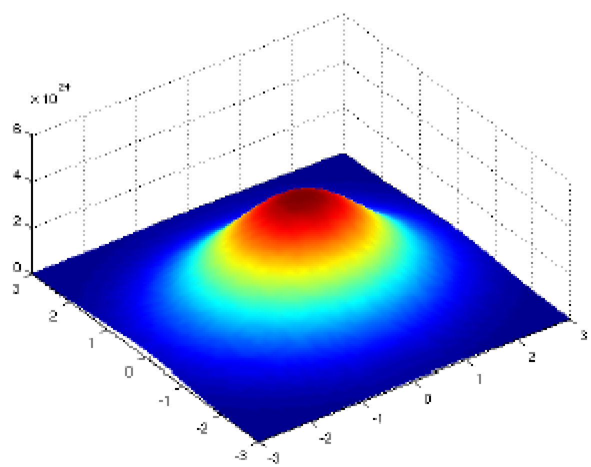
- Univariate

$$\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multivariate

$$\mathcal{N}(\mathbf{x}|\mu,\mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}$$





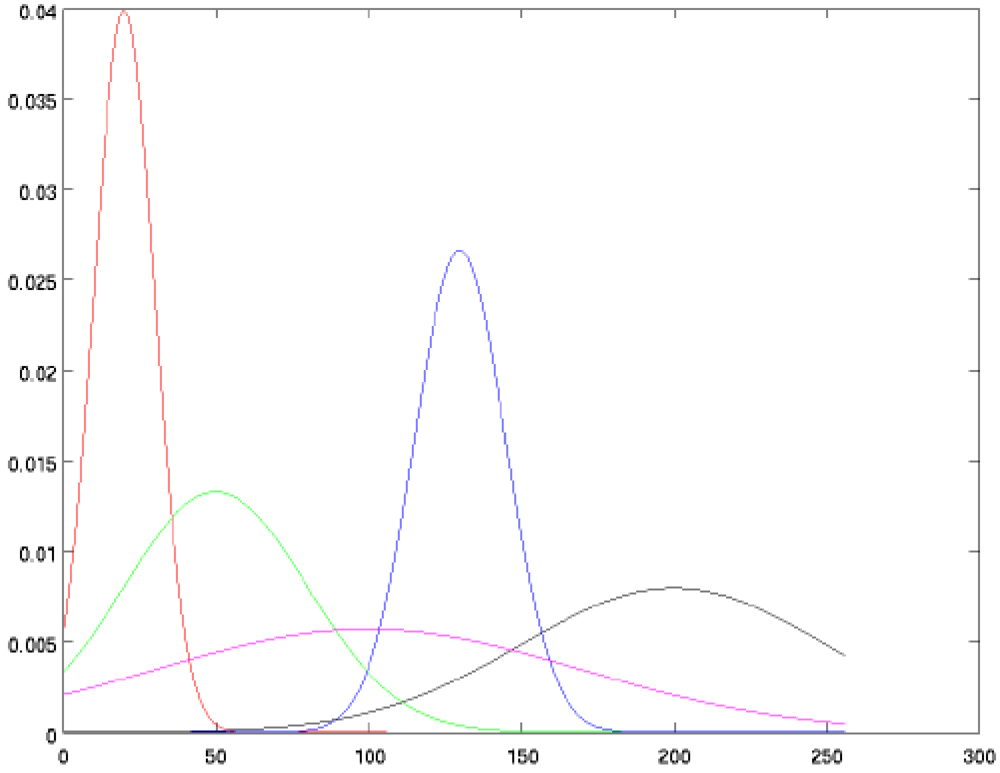
Online Mixture Model

- History of a pixel is known up to current time t
  - $\{X_1, \ldots, X_t\} = \{I(x_o, y_o, i): 1 \leq i \leq t\}$
- Model the history as a mixture of K Gaussian Distributions
  - $P(X_t) = \sum_{i=1}^{K} w_{i,t} \mathcal{N}(X_t | u_{i,t}, \Sigma_{i,t})$
    - $w_{i,t}$ - prior probability (weight) of Gaussians i

  - What is the dimensionality of the Gaussian?

Mixture Model Example

- For a grayscale image with $K = 5$



Model Adaption

- Online K-means approximation is used to update the Gaussians
- Match a new pixel X_{t+1} to an existing Gaussian and update
  - Must be within 2.5σ
  - $\mu_{i,t+1} = (1-\rho)\mu_{i,t} + \rho X_{t+1}$
  - $\sigma^2_{i,t+1} = (1-\rho)\sigma^2_{i,t} + \rho\left(X_{t+1} - \mu_{i,t}\right)^2$
    - $\rho = \alpha\mathcal{N}\left(\mathrm{X}_{\mathrm{t+1}} \middle| \mu_{i,t}, \sigma^2_{i,t}\right)$
    - α – is a learning rate
- Prior weights of Gaussians are updated
  - $w_{i,t+1} = (1-\alpha)w_{i,t} + \alpha\left(M_{i,t+1}\right)$
  - $M_{i,t+1} = 1$ for matching Guassian or $M_{i,t+1} = 0$ for all others

Model Adaption

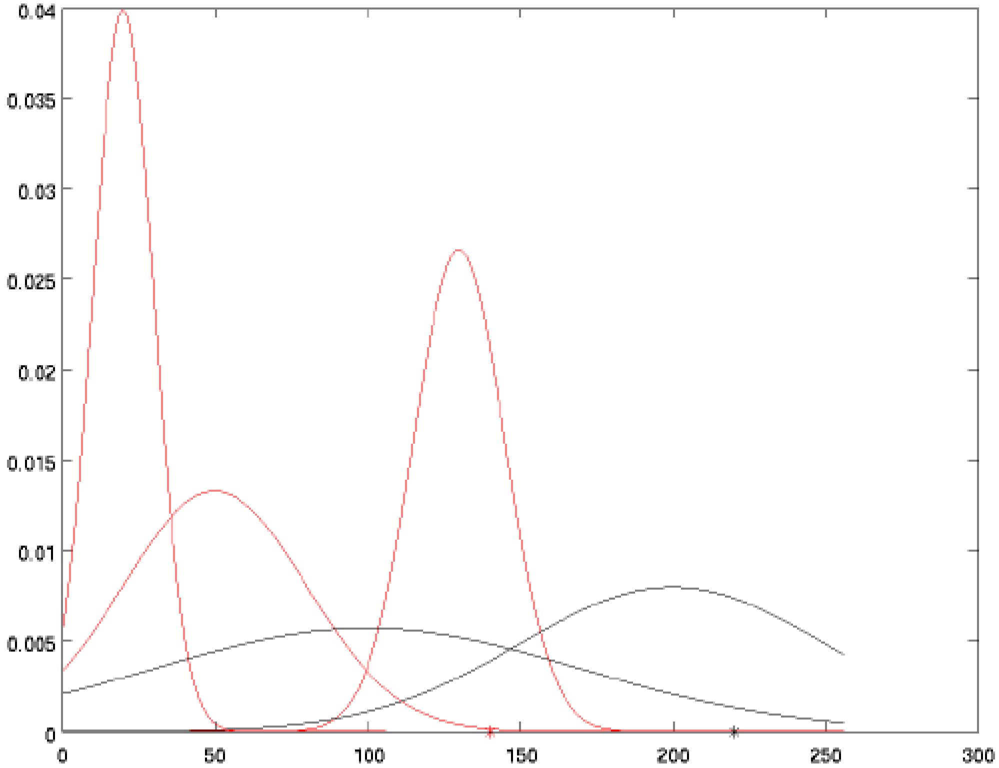
- If X_{t+1} do not match and of the K Gaussians, there is no matching mixture
- Replace the least probable distribution with a new one
  - Least probable in the ω/σ sense (to be explained)
  - The newly created distribution has
    - $\mu_{t+1} = X_{t+1}$
    - Has high variance and low prior weight

Background Model Estimation

- Heuristic: Gaussians with the most **supporting evidence** and **least variance** should correspond to the background
  - Why?
- Gaussians are ordered by the value of ω/σ
  - High support and smaller variance give larger value
- First B distributions are selected as the background model
  - $B = argmin_b(\sum_{i=1}^{b} w_i > T)$
    - T minimum portion of image expected to be background

Background Estimation Example

- After background estimation, red are the background and black are foreground

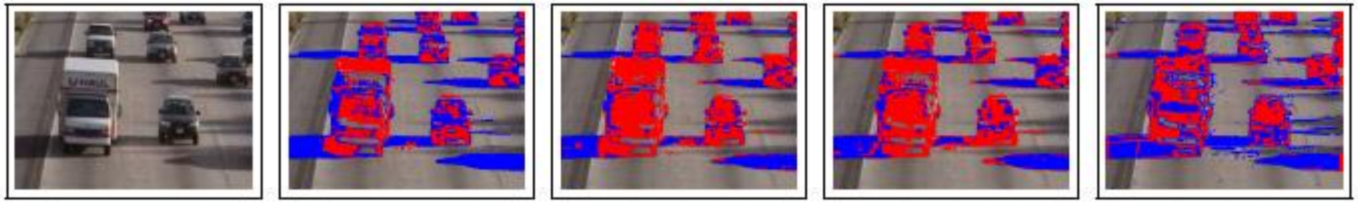


Discussion

- Advantages
  - Different threshold for each pixel
  - Pixel-wise thresholds adapt over time
  - Objects are allowed to become part of the background without destroying the existing background model
  - Provides fast recovery
- Disadvantages
  - Cannot handle sudden, drastic lighting changes
  - Must have good Gaussian initialization (median filtering)
  - There are a number of parameters to tune

More Issues?

- Shadows detection
  - [Prati, Mikic, Trivedi, Cucchiara 2003]



(a) Raw image (b) SNP result (c) SP result (d) DNM1 result (e) DNM2 result

- Chen & Aggarwal: The likelihood of a pixel being covered or uncovered is decided by the relative coordinates of optical flow vector vertices in its neighborhood.
- Oliver et al.: "Eigenbackgrounds" and its variations.
- Seki et al.: Image variations at neighboring image blocks have strong correlation.

Simple Improvement

- Adaptive background mixture model + 3D connected component analysis [Goo et al.]
  - 3rd dimension is time
- Incorporate both spatial and temporal information into the background model

Summary

- Simple background subtraction approaches such as fame diff, mean, and median filtering are fast
  - Constant thresholds make them ill-suited for challenging real-world problems
- Adaptive background mixture model approach can handle challenging situations
  - Bimodal backgrounds, long-term scene changes, and repetitive motion
- Improvements include upgrade the approach with temporal information or using region-based techniques