Computer Vision II – Lecture 2

Background Modeling

22.04.2014

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de

leibe@vision.rwth-aachen.de

Announcements

- Course webpage
  - http://www.vision.rwth-aachen.de/teaching/
  - Slides will be made available on the webpage
- L2P electronic repository
  - Exercises and supplementary materials will be posted on the L2P
- Please subscribe to the lecture on the Campus system!
  - Important to get email announcements and L2P access!
  - Bachelor students please also subscribe

Course Outline

- Single-Object Tracking
  - Background modeling
  - Template based tracking
  - Color based tracking
  - Contour based tracking
  - Tracking by online classification
  - Tracking-by-detection
- Bayesian Filtering
- Multi-Object Tracking
- Articulated Tracking

Image source: Tobias Jaeggli

Topics of This Lecture

- Motivation: Background Modeling
- Simple Background Models
  - Background Subtraction
  - Frame Differencing
- Statistical Background Models
  - Single Gaussian
  - Mixture of Gaussians
  - Kernel Density Estimation
- Practical Issues and Extensions
  - Background model update
  - False detection suppression
  - Shadow suppression
  - Applications

Motivation: Tracking from Static Cameras

Video source: Wolfgang Mehner

Motivation

- Goals
  - Want to detect and track all kinds of objects in a wide variety of surveillance scenarios.

  $\Rightarrow$ *Need a general algorithm that works for many scenarios*.
  - Video frames come in at 30Hz. There is not much time to process each image.

  $\Rightarrow$ *Real-time algorithms need to be very simple*.
- Assumptions
  - The camera is static.
  - Objects that move are important (people, vehicles, etc.).
- Basic Approach
  - Maintain a model of the static background.
  - Compare the current frame to this model to detect objects.

Slide adapted from Robert Collins

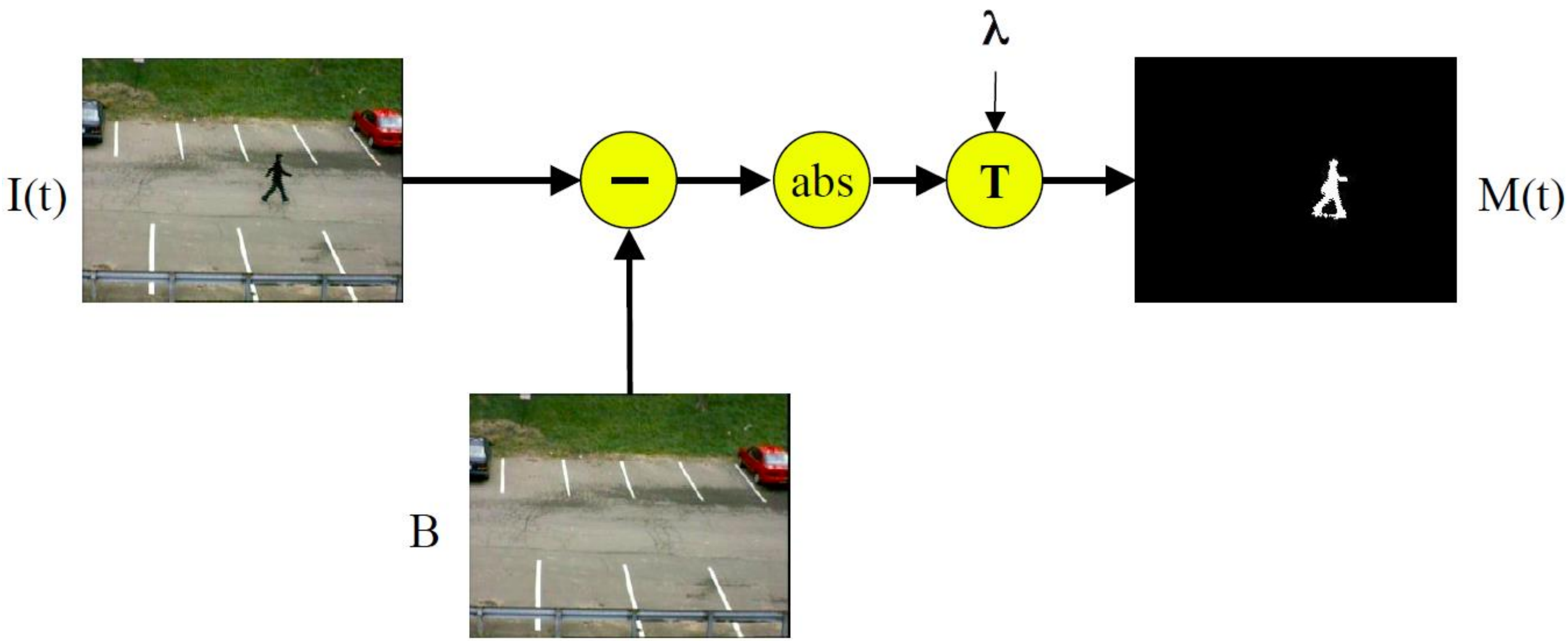
Background Modelling Results

Video source: Wolfgang Mehner

Topics of This Lecture

- Motivation: Background Modeling
- Simple Background Models
  - Background Subtraction
  - Frame Differencing
- Statistical Background Models
  - Single Gaussian
  - Mixture of Gaussians
  - Kernel Density Estimation
- Practical Issues and Extensions
  - Background model update
  - False detection suppression
  - Shadow suppression

Simple Background Subtraction



- **Procedure**
  - Background model is a static image (without any objects).
  - Pixels are labeled based on thresholding the absolute intensity difference between current frame and background.

Background Subtraction Results

- **Observation**
  - **Background subtraction does a reasonable job of extracting the object shape if the object intensity/color is sufficiently different from the background.**
- ***What are the limitations of this simple procedure?***

Background Subtraction: Limitations

- Outdated reference frame
  - Objects that enter the scene and stop continue to be detected...

    ...making it difficult to detect new objects that pass in front of them.

  - If part of the assumed static background starts moving...

    ...both the object and its negative ghost (the revealed background) are detected.

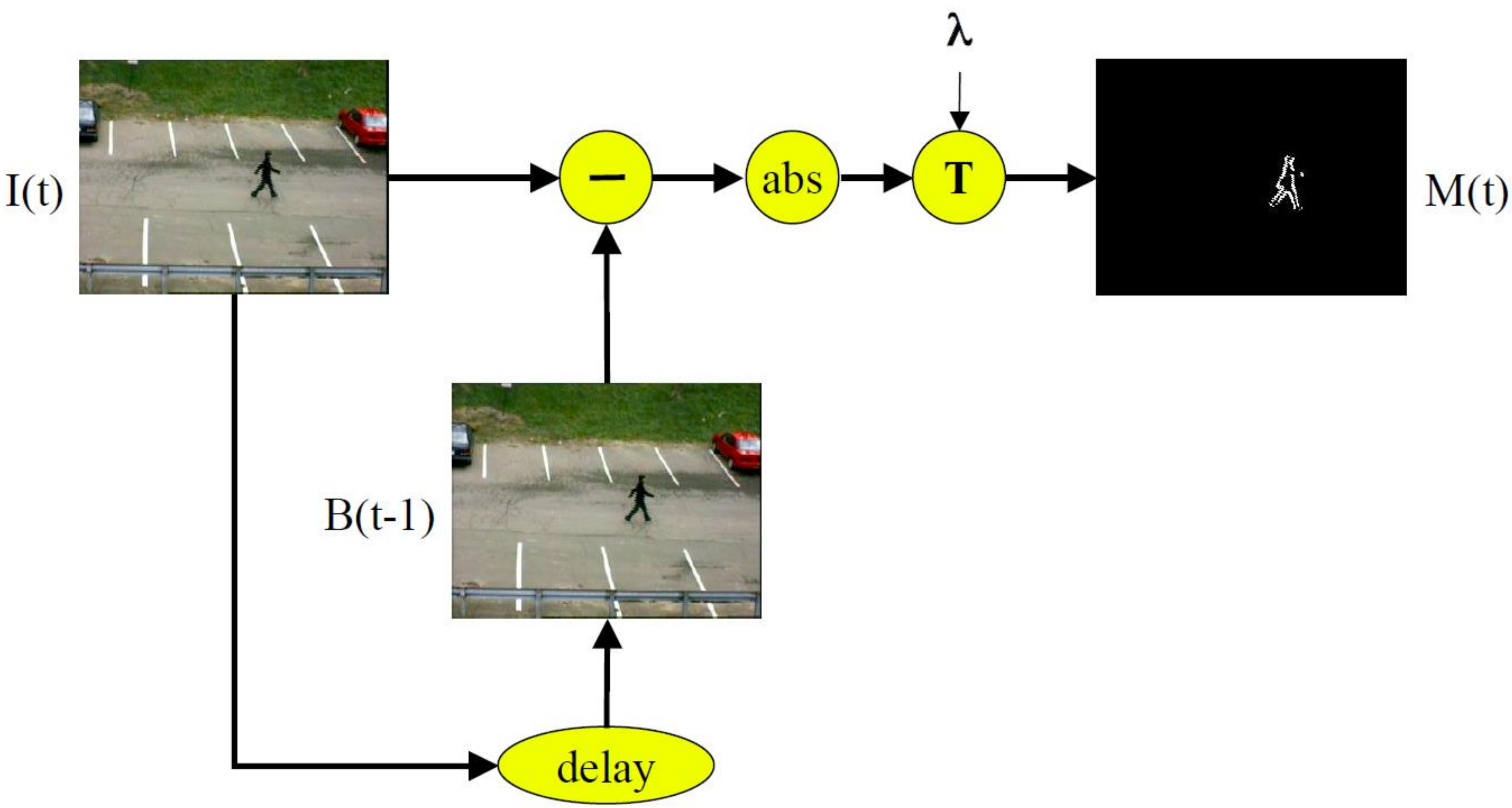
Background Subtraction: Limitations

- **Illumination changes**
  - Background subtraction is sensitive to illumination changes and unimportant scene motion (e.g., tree branches swaying in the wind).

- **Global threshold**
  - A single, global threshold for the entire scene is often suboptimal.

⇒ ***Need adaptive model with local decisions***

Simple Frame Differencing



- **Other idea**
  - **Background model is replaced with the previous image.**

Frame Differencing Observations

- **Advantages**
  - Frame differencing is very quick to adapt to changes in lighting or camera motion.
  - Objects that stop are no longer detected.
  - Objects that start up no longer leave behind ghosts.

- **Limitations**
  - Frame differencing only detects the leading and trailing edge of a uniformly colored object.
  - Very few pixels on the object are labeled.
  - Very hard to detect an object moving towards or away from the camera.

Slide credit: Robert Collins

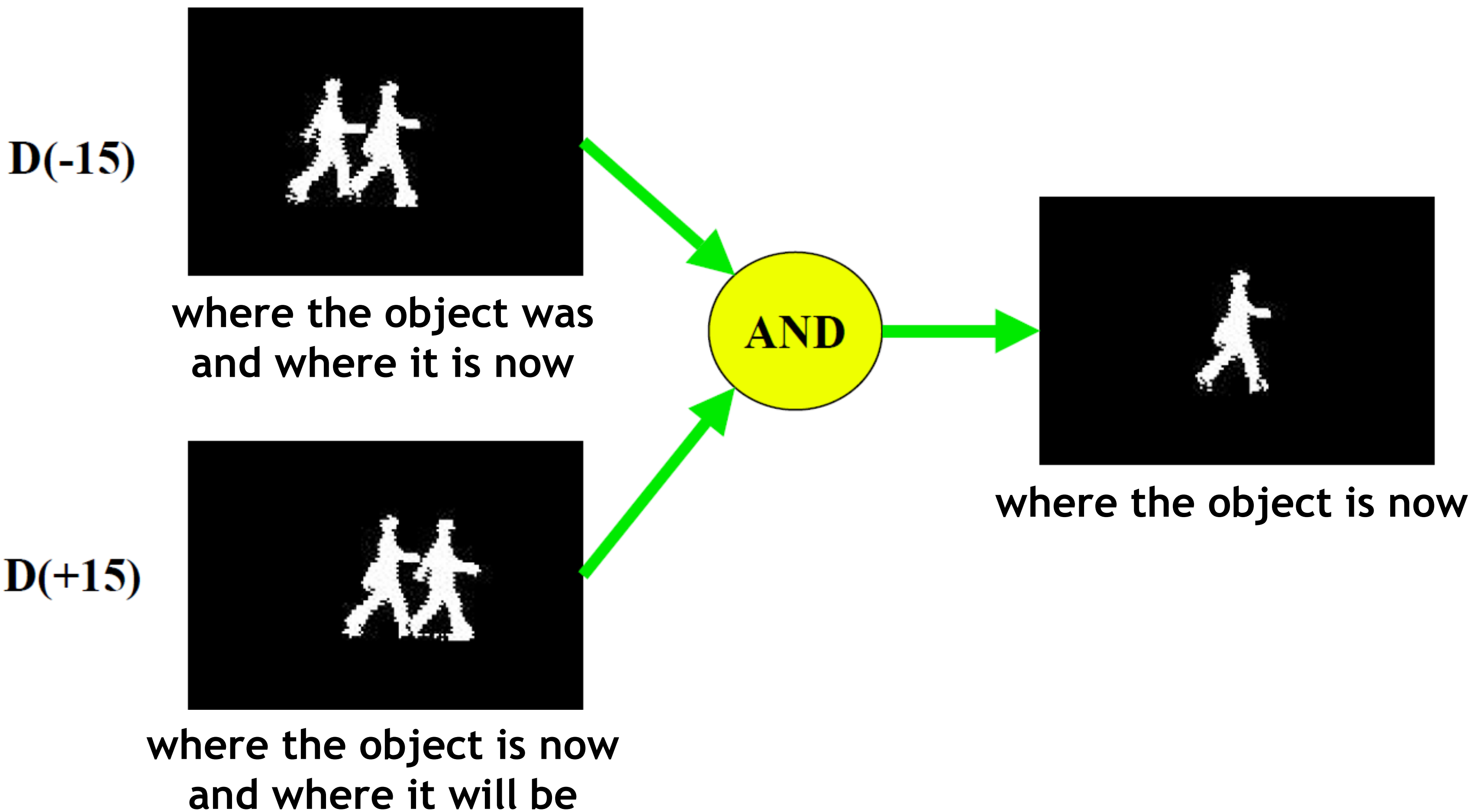
Differencing and Temporal Scale

I(t) D(-1) D(-3) D(-5) D(-9) D(-15)

- More general formulation
  - Define $D(N) = \|I(t) - I(t+N)\|$
- Effect of increasing the temporal scale
  - More complete object silhouette, but two copies of the object (one where it used to be, one where it is now).

Three-Frame Differencing

- Improved approach to handle this problem



Three-Frame Differencing

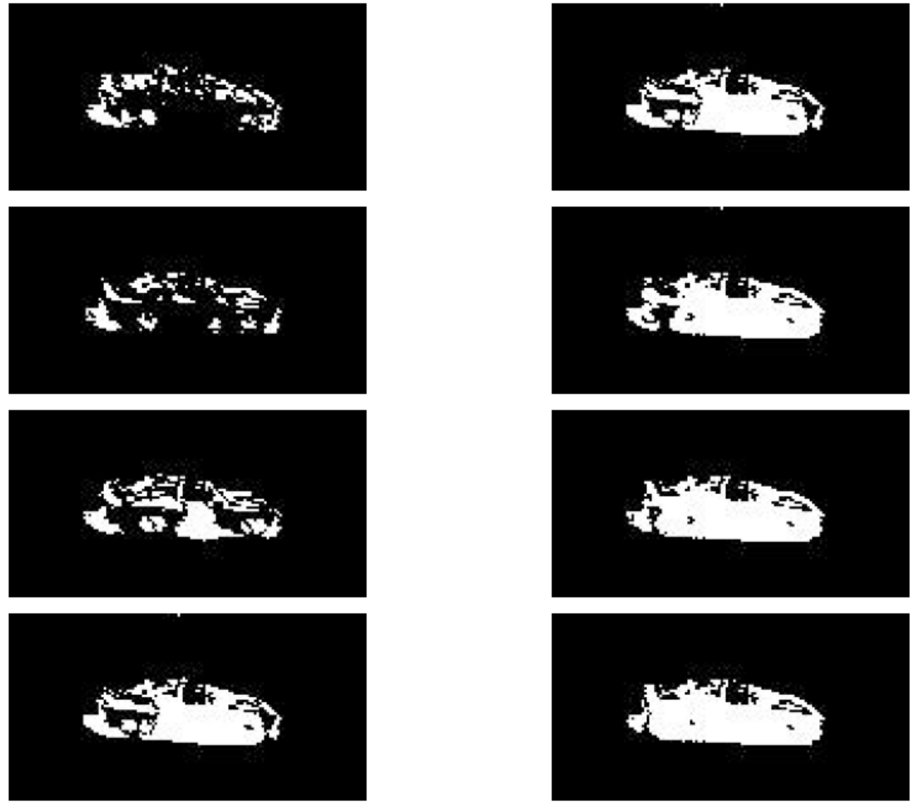
frames skipped

1

5

15

25



35

45

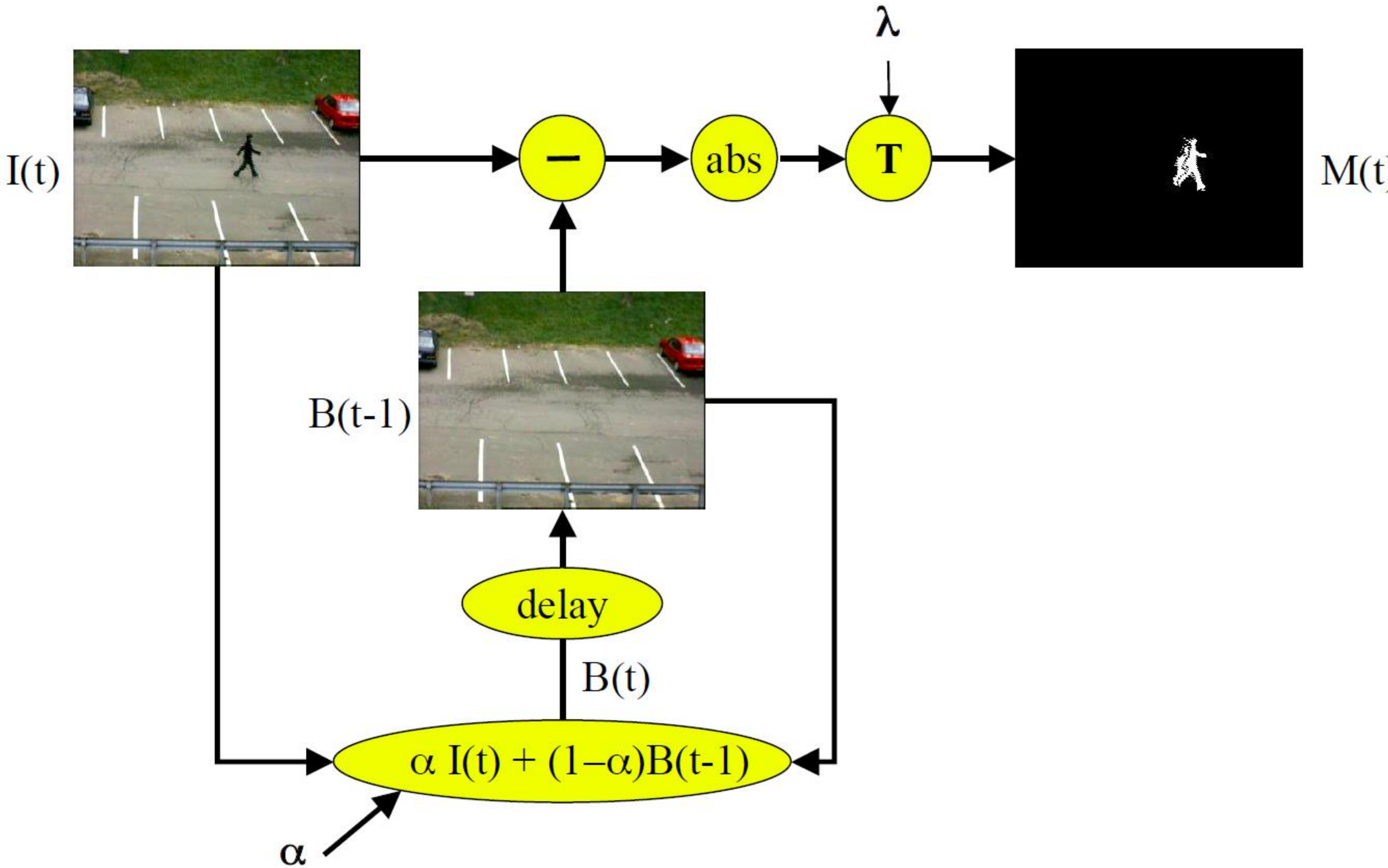
55

65

This worked well for the person → 15

- **Problem**
  - Choice of good frame-rate for three-frame differencing depends on size and speed of object.

Adaptive Background Subtraction



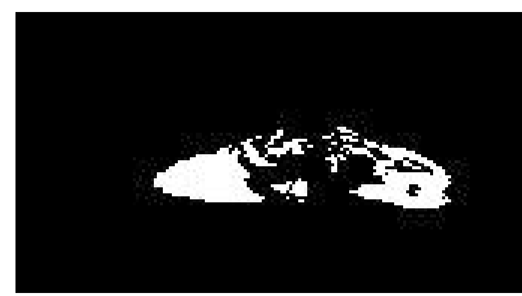
- Current image is "blended" into the background model with α.

Adaptive Background Subtraction

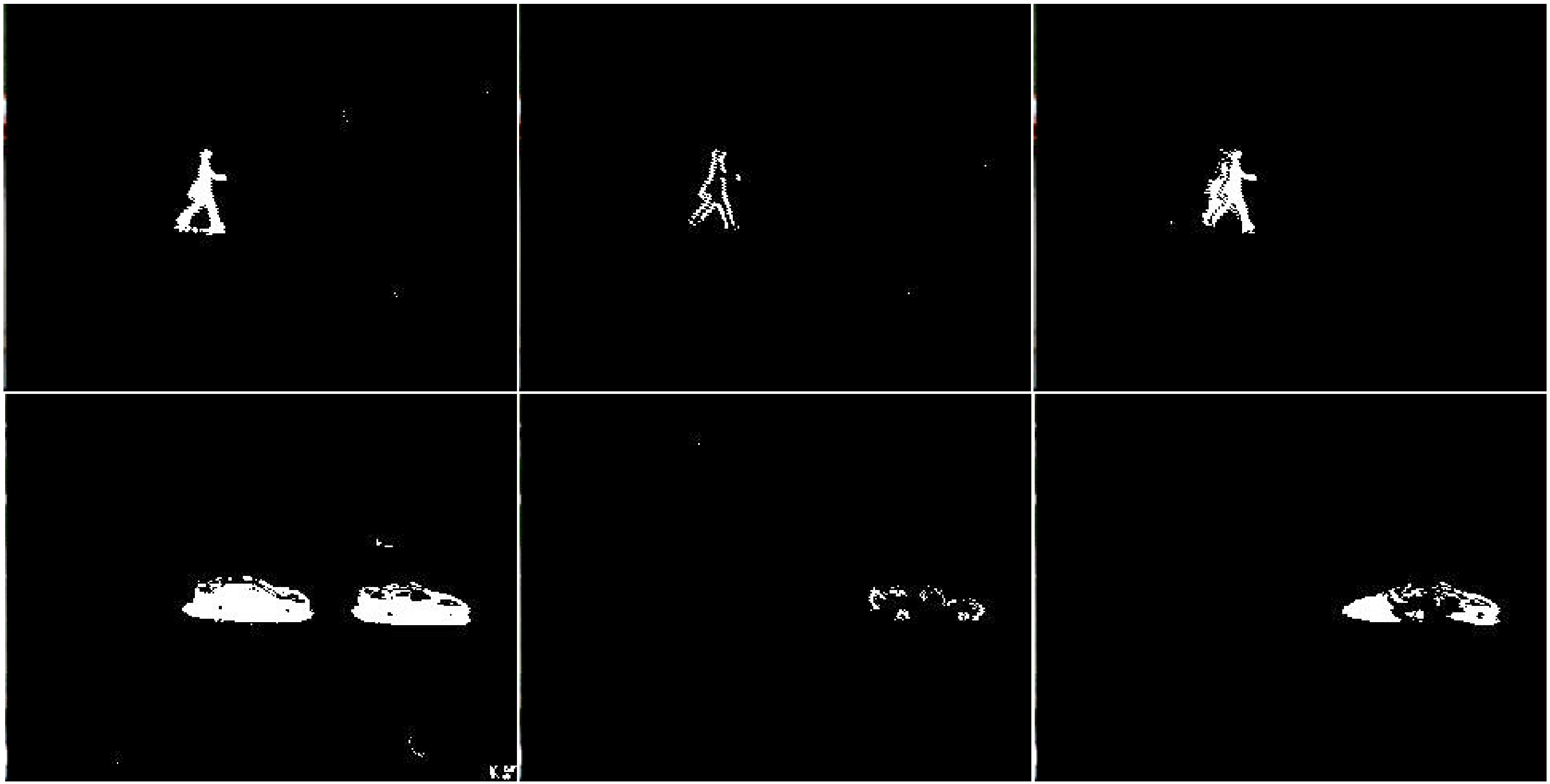
- Properties
  - More responsive to changes in illumination and camera motion.
  - Small, fast-moving objects are well-segmented, but they leave behind short “trails” of pixels.

  - Objects that stop and ghosts left behind by objects that start both gradually fade into the background.

  - The centers of large, slow-moving objects start to fade into the background, too!
  - This can be fixed by decreasing the blend parameter α, but then it takes longer for ghost objects to disappear...



Comparisons



BG Subtraction Frame Differencing Adaptive BG Subtract.

Discussion

- Background subtraction / Frame differencing
  - Very simple techniques, historically among the first.
  - Straight-forward to implement, fast to test out.
  - We've seen some fixes for the most pressing problems.
- Remaining limitations
  - Rather heuristic approach.
  - Leads to relatively poor foreground/background decisions.
  - Optimal temporal scale still depends on object size and speed.
  - Global threshold is often suboptimal for parts of the image.

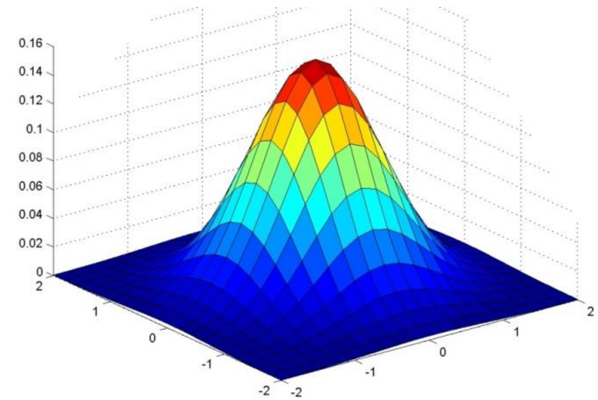
  $\Rightarrow$ *Very fiddly in practice, requires extensive parameter tuning.*
- Let's try to come up with a better founded approach
  - Using a statistical model of background probability...

Topics of This Lecture

- Motivation: Background Modeling
- Simple Background Models
  - Background Subtraction
  - Frame Differencing
- **Statistical Background Models**
  - **Single Gaussian**
  - **Mixture of Gaussians**
  - **Kernel Density Estimation**
- Practical Issues and Extensions
  - Background model update
  - False detection suppression
  - Shadow suppression

Gaussian Background Model

- **Statistical model**
  - Value of a pixel represents a measurement of the radiance of the first object intersected by the pixel's optical ray.
  - With a static background and static lighting, this value will be a constant affected by i.i.d. Gaussian noise.



- **Idea**
  - Model the background distribution of each pixel by a single Gaussian centered at the mean pixel value:

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

  - Test if a newly observed pixel value has a high likelihood under this Gaussian model.

$\Rightarrow$ Automatic estimation of a sensitivity threshold for each pixel.

Recap: Maximum Likelihood Approach

- **Computation of the likelihood**
  - Single data point: $p(x_n|\theta)$
  - Assumption: all data points $X = \{x_1, \ldots, x_n\}$ are independent

$$L(\theta) = p(X|\theta) = \prod_{n=1}^{N} p(x_n|\theta)$$

  - Log-likelihood

$$E(\theta) = -\ln L(\theta) = -\sum_{n=1}^{N} \ln p(x_n|\theta)$$

- **Estimation of the parameters θ (Learning)**
  - Maximize the likelihood (=minimize the negative log-likelihood)

$\Rightarrow$ Take the derivative and set it to zero.

$$\frac{\partial}{\partial\theta} E(\theta) = -\sum_{n=1}^{N} \frac{\frac{\partial}{\partial\theta} p(x_n|\theta)}{p(x_n|\theta)} \stackrel{!}{=} 0$$

Recap: Maximum Likelihood Approach

- For a 1D Gaussian, we thus obtain

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

"sample mean"

- In a similar fashion, we get

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - \hat{\mu})^2$$

"sample variance"

- $\hat{\theta} = (\hat{\mu}, \hat{\sigma})$ is the **Maximum Likelihood estimate** for the parameters of a Gaussian distribution.
- Note: the estimate of the sample variance is *biased*. Better use

$$\tilde{\sigma}^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \hat{\mu})^2$$

Online Adaptation (1D Case)

- **Once estimated, adapt the Gaussians over time**
  - **We can compute a running estimate over a time window**

$$\hat{\mu}^{(t+1)} = \hat{\mu}^{(t)} + \frac{1}{N}x^{(t+1)} - \frac{1}{N}x^{(t+1-T)}$$

$$(\tilde{\sigma}^2)^{(t+1)} = (\tilde{\sigma}^2)^{(t)} + \frac{1}{N-1}(x^{(t+1)} - \hat{\mu}^{(t+1)})^2 - \frac{1}{N-1}(x^{(t+1-T)} - \hat{\mu}^{(t+1)})^2$$

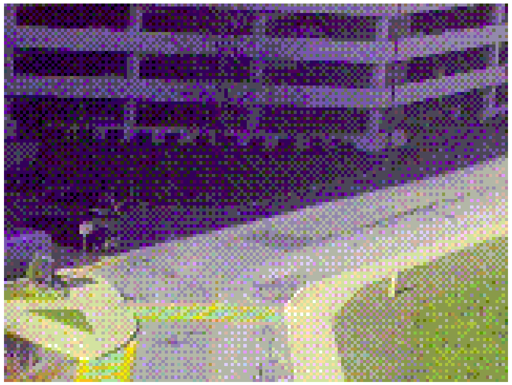
  - **However, distribution is non-stationary (and newer values are more important) ⇒ better use Exponential Moving Average filter**

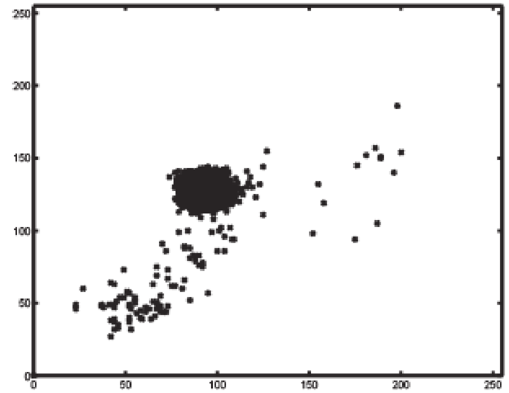
$$\hat{\mu}^{(t+1)} = (1-\alpha)\hat{\mu}^{(t)} + \alpha x^{(t+1)}$$

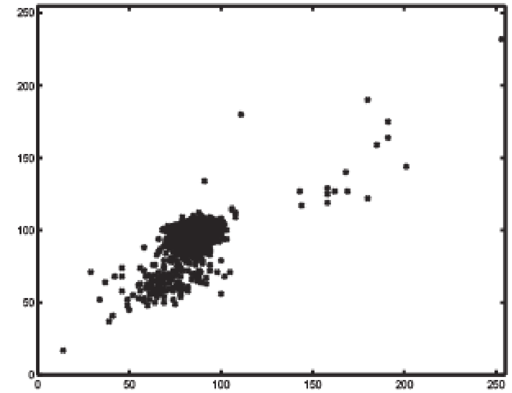
$$(\tilde{\sigma}^2)^{(t+1)} = (1-\alpha)(\tilde{\sigma}^2)^{(t)} + \alpha(x^{(t+1)} - \hat{\mu}^{(t+1)})^2$$

with a fixed learning rate α.

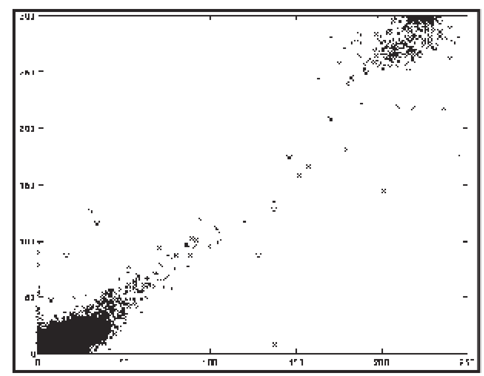
Problem: Complex Distributions







RG scatter plots of the same pixel taken 2 min apart



Bi-modal distribution caused by specularities on the water surface

⇒ ***A single Gaussian is clearly insufficient here...***

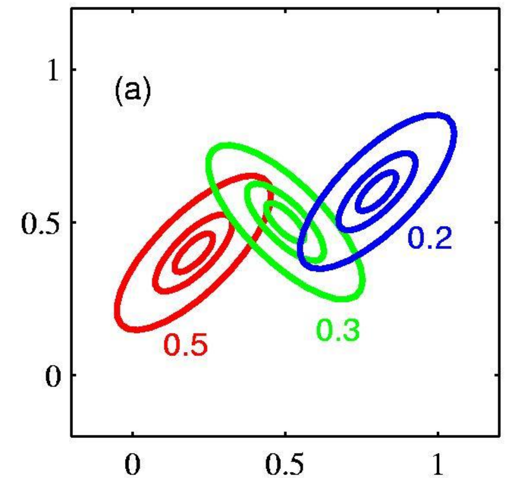
Image source: Chris Stauffer

Problem: Adaptation Speed, Sensitivity

- If the background model adapts too slowly...
  - Will construct a very wide and inaccurate model with low detection sensitivity
- If the model adapts too quickly...
  - Leads to inaccurate estimation of the model parameters
  - The model may adapt to the targets themselves (especially slow-moving ones)
- Design trade-off
  - Model should adapt quickly to changes in the background process *and* detect objects with high sensitivity.
  - *How can we achieve that?*

MoG Background Model

- **Improved statistical model**
  - Large jumps between different pixel values because different objects are projected onto the same pixel at different times.
  - While the same object is projected onto the pixel, small local intensity variations due to Gaussian noise.



- **Idea**
  - Model the color distribution of each pixel by a mixture of K Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

  - Evaluate likelihoods of observed pixel values under this model.
  - Or let entire Gaussian components adapt to foreground objects and classify components as belonging to object or background.

Image source: Chris Bischop

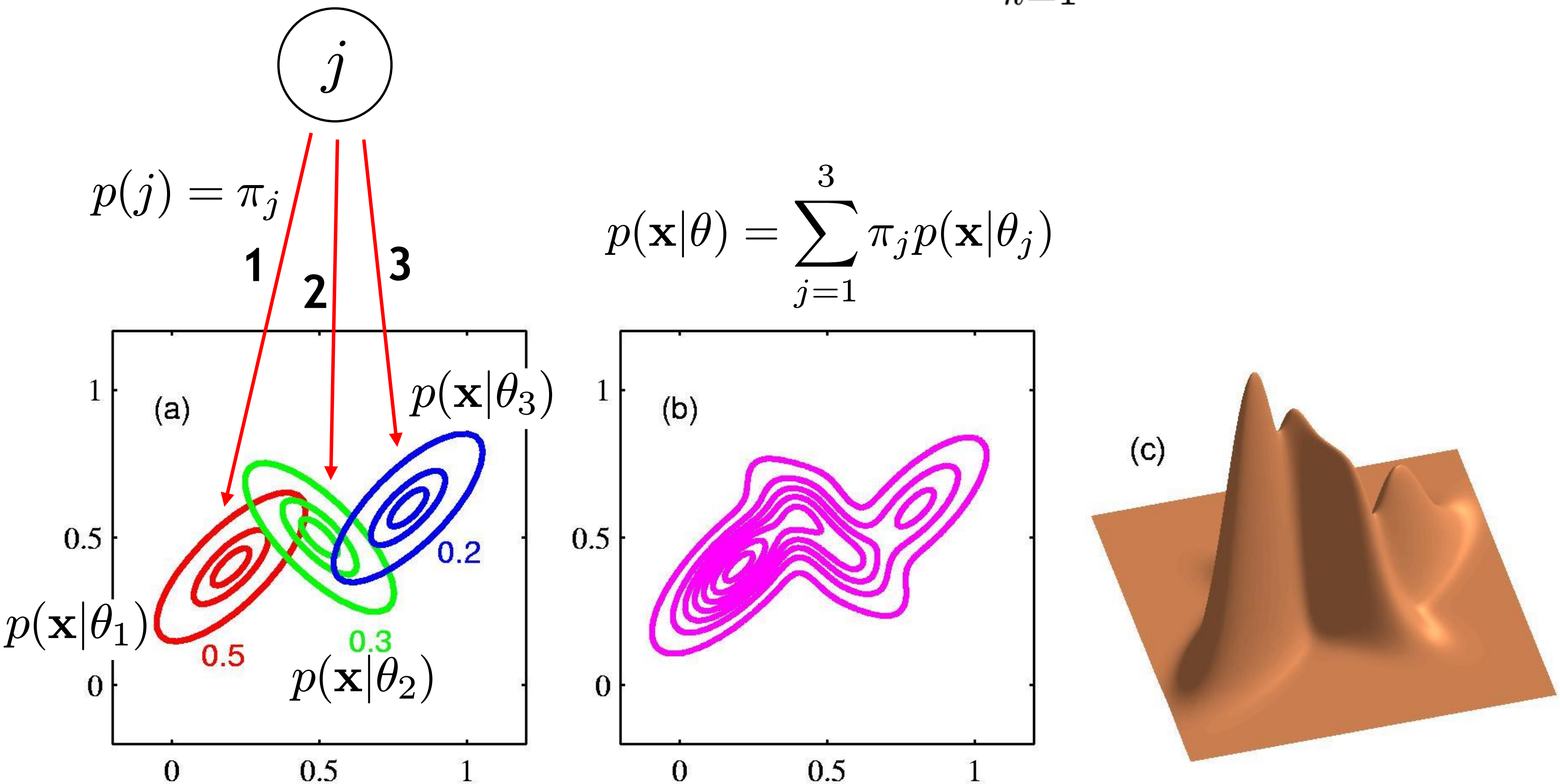
Recap: Mixture of Gaussians

- **"Generative model"**

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(j) = \pi_j$$

$$p(\mathbf{x}|\theta) = \sum_{j=1}^{3} \pi_j p(\mathbf{x}|\theta_j)$$



Recap: EM Algorithm

- **Expectation-Maximization (EM) Algorithm**
  - **E-Step**: softly assign samples to mixture components

$$\gamma_j(\mathbf{x}_n) \leftarrow \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{N} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \qquad \forall j = 1, \ldots, K, \quad n = 1, \ldots, N$$

  - **M-Step**: re-estimate the parameters (separately for each mixture component) based on the soft assignments

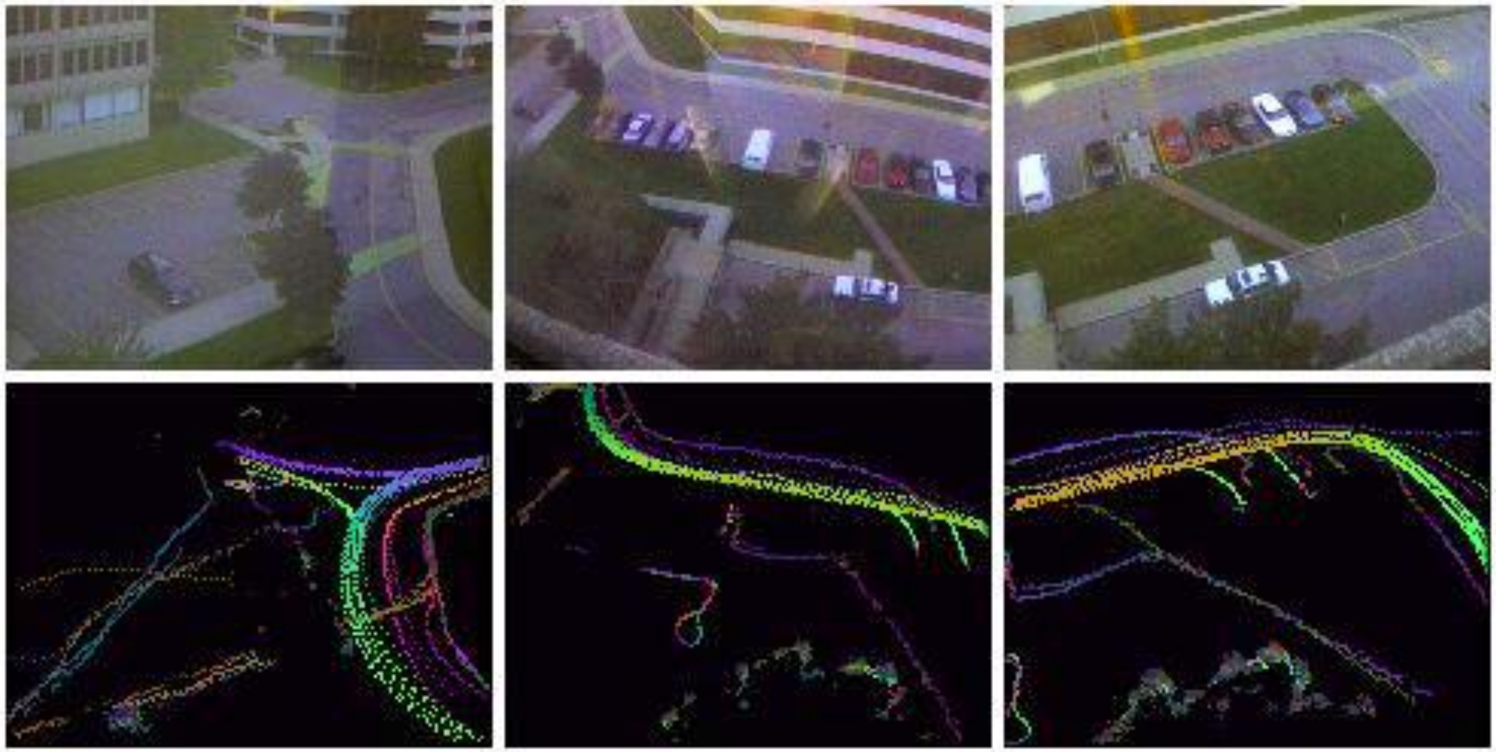
$$\hat{N}_j \leftarrow \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n) \text{ = soft number of samples labeled } j$$

$$\hat{\pi}_j^{\text{new}} \leftarrow \frac{\hat{N}_j}{N}$$

$$\hat{\boldsymbol{\mu}}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n$$

$$\hat{\boldsymbol{\Sigma}}_j^{\text{new}} \leftarrow \frac{1}{\hat{N}_j} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_j^{\text{new}})(\mathbf{x}_n - \hat{\boldsymbol{\mu}}_j^{\text{new}})^{\text{T}}$$

Stauffer-Grimson Background Model



- Very popular model
  - Used in many tracking approaches
  - Suitable for long-term observations (finding patterns of activity)

C. Stauffer, W.E.L. Grimson, Adaptive Background Mixture Models for Real-Time Tracking, CVPR 1998.

Image source: C. Stauffer

Stauffer-Grimson Background Model

- **Idea**
  - Model the distribution of each pixel by a mixture of K Gaussians

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \text{where} \quad \boldsymbol{\Sigma}_k = \sigma_k^2 \mathbf{I}$$

  - Check every new pixel value against the existing K components until a match is found (pixel value within $2.5\ \sigma_k$ of μ_k).
  - If a match is found, adapt the corresponding component.
  - Else, replace the least probable component by a distribution with the new value as its mean and an initially high variance and low prior weight.
  - Order the components by the value of w_k/σ_k and select the best B components as the background model, where

$$B = \arg\min_b \left(\sum_{k=1}^{b} \frac{w_k}{\sigma_k} > T \right)$$

Stauffer-Grimson Background Model

- **Online adaptation**
  - **Instead of estimating the MoG using EM, use a simpler online adaptation, assigning each new value only to the matching component.**
  - **Let $M_{k,t} = 1$ iff component k is the model that matched, else 0.**

$$\pi_k^{(t+1)} = (1-\alpha)\pi_k^{(t)} + \alpha M_{k,t}$$

  - **Adapt only the parameters for the matching component**

$$\boldsymbol{\mu}_k^{(t+1)} = (1-\rho)\boldsymbol{\mu}_k^{(t)} + \rho x^{(t+1)}$$

$$\boldsymbol{\Sigma}_k^{(t+1)} = (1-\rho)\boldsymbol{\Sigma}_k^{(t)} + \rho(x^{(t+1)} - \boldsymbol{\mu}_k^{(t+1)})(x^{(t+1)} - \boldsymbol{\mu}_k^{(t+1)})^T$$

where

$$\rho = \alpha \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

(i.e., the update is weighted by the component likelihood)

Discussion: Stauffer-Grimson Model

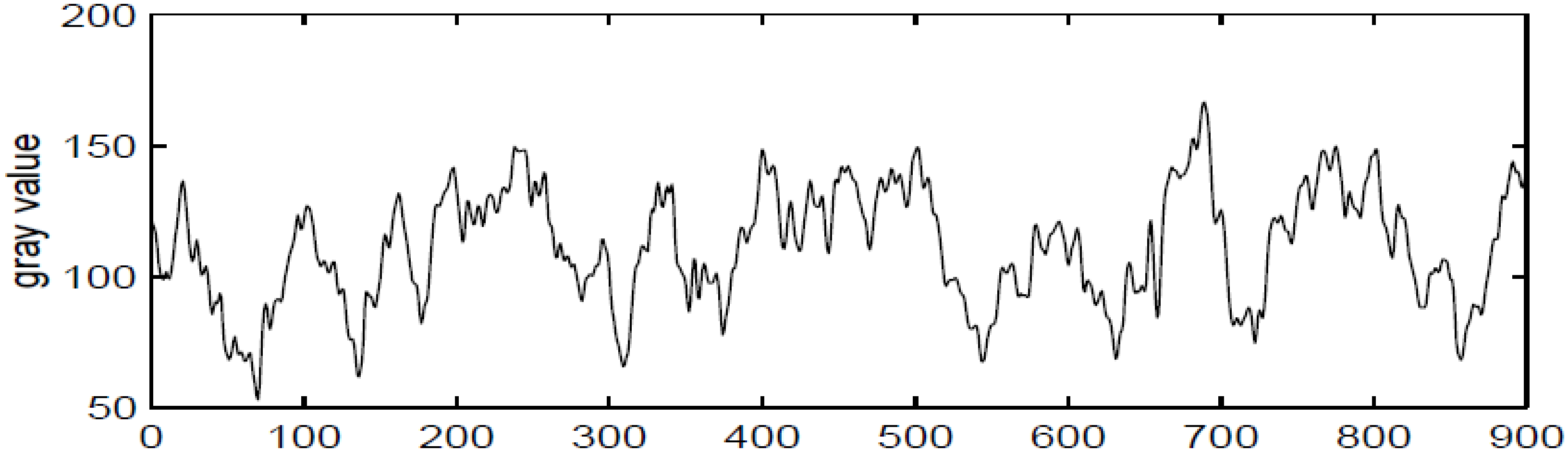
- Properties
  - Static foreground objects can be integrated into the mixture
    - Advantage: This doesn't destroy the existing background model.
    - If an object is stationary for some time and then moves again, the distribution for the background still exists

    ⇒ Quick recovery from such situations.
  - Ordering of components by w_k/σ_k
    - Favors components that have more evidence (higher w_k) and a smaller variance (lower σ_k).

    ⇒ Those are typically the best candidates for background.
  - Model can adapt to the complexity of the observed distribution.
    - If the distribution is unimodal, only a single component will be selected for the background.

    ⇒ This can be used to save memory and computation.

Problem: Outdoor Scenes



- **Dynamic areas**
  - **Waving trees, rippling water, ...**
  - **Fast variations**

⇒ ***More flexible representation needed here.***

Image & Video source: A. Elgammal

Recap: Kernel Density Estimation

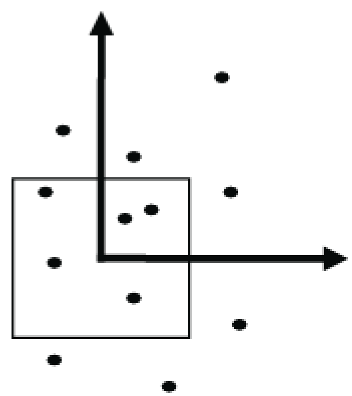
- Estimating the probability density from discrete samples
  - Approximation: $$p(\mathbf{x}) \approx \frac{K}{NV}$$

fixed V
determine K

fixed K
determine V

Kernel Methods | **K-Nearest Neighbor**

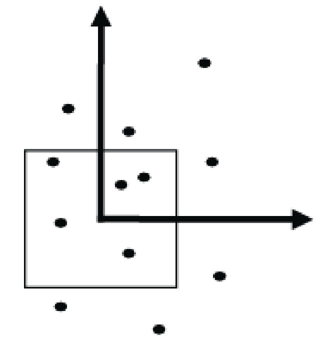
- Kernel methods
  - Example: Determine the number K of data points inside a fixed hypercube...



Recap: Kernel Density Estimation

- **Parzen Window**
  - **Hypercube of dimension D with edge length h:**

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i \cdot \frac{1}{2}, \quad i = 1, \ldots, D \\ 0, & else \end{cases}$$



"Kernel function"

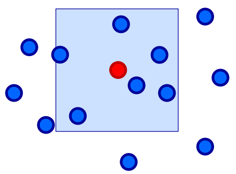
$$K = \sum_{n=1}^{N} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right) \qquad V = \int k(\mathbf{u}) d\mathbf{u} = h^d$$

  - **Probability density estimate:**

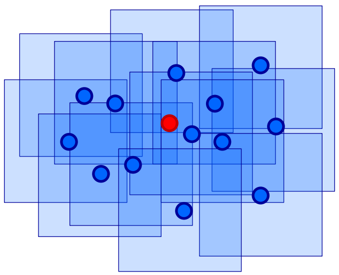
$$p(\mathbf{x}) \approx \frac{K}{NV} = \frac{1}{Nh^D} \sum_{n=1}^{N} k\left(\frac{\mathbf{x} - \mathbf{x}_n}{h}\right)$$

Recap: Parzen Window

- Interpretations
  1. We place a *kernel window* k *at location* $\mathbf{x}$ and count how many data points fall inside it.



  2. We place a *kernel window* k *around each data point* $\mathbf{x}_n$ and sum up their influences at location $\mathbf{x}$.



  $\Rightarrow$ Direct visualization of the density.

- Still, we have artificial discontinuities at the cube boundaries…
  - We can obtain a smoother density model if we choose a smoother kernel function, e.g. a Gaussian

Kernel Background Modeling

- Nonparametric model of background appearance
  - Very flexible approach, can deal with large amounts of background motion and scene clutter

A. Elgammal, D. Harwood, L.S. Davis, Non-parametric Model for Background Subtraction, ECCV 2000.

Video source: Ahmed Elgammal

Kernel Background Modeling

- **Nonparametric density estimation**
  - Estimate a pixel's background distribution using the kernel density estimator $K(\cdot)$ as

$$p(\mathbf{x}^{(t)}) = \frac{1}{N}\sum_{i=1}^{N} K\left(\mathbf{x}^{(t)} - \mathbf{x}^{(i)}\right)$$

  - Choose K to be a Gaussian $\mathcal{N}(0,\ \mathbf{\Sigma})$ with $\mathbf{\Sigma} = \mathrm{diag}\{\sigma_j\}$. Then

$$p(\mathbf{x}^{(t)}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{1}{2}\frac{(x_j^{(t)}-x_j^{(i)})^2}{\sigma_j^2}}$$

  - A pixel is considered foreground if $p(\mathbf{x}^{(t)}) < \theta$ for a threshold θ.
    - This can be computed very fast using lookup tables for the kernel function values, since all inputs are discrete values.
    - Additional speedup: partial evaluation of the sum usually sufficient

Results Kernel Background Modeling

- **Performance in heavy rain**

Video source: Ahmed Elgammal

Results Kernel Background Modeling

- Results for color images

- Practical issues with color images
  - *Which color space to use?*

Video source: Ahmed Elgammal

Topics of This Lecture

- Motivation: Background Modeling
- Simple Background Models
  - Background Subtraction
  - Frame Differencing
- Statistical Background Models
  - Single Gaussian
  - Mixture of Gaussians
  - Kernel Density Estimation
- **Practical Issues and Extensions**
  - **Background model update**
  - **False detection suppression**
  - **Shadow suppression**
  - **Applications**

Practical Issues: Background Model Update

- Kernel background model
  - Sample N intensity values taken over a window of W frames.
- FIFO update mechanism
  - Discard oldest sample.
  - Choose new sample randomly from each interval of length W/N frames.
- When should we update the distribution?
  - Selective update: add new sample only if it is classified as a background sample
  - Blind update: always add the new sample to the model.

Updating Strategies

- Selective update
  - Add new sample only if it is classified as a background sample.
  - Enhances detection of new objects, since the background model remains uncontaminated.
  - But: Any incorrect detection decision will result in persistent incorrect detections later.

  ⇒ Deadlock situation.

- Blind update
  - Always add the new sample to the model.
  - Does not suffer from deadlock situations, since it does not involve any update decisions.
  - But: Allows intensity values that do not belong to the background to be added to the model.

  ⇒ Leads to bad detection of the targets (more false negatives).

Solution: Combining the Two Models

- Short-term model
  - Recent model, adapts to changes quickly to allow very sensitive detection
  - Consists of the most recent N background sample values.
  - Updated using a selective update mechanism based on the detection mask from the final combination result.
- Long-term model
  - Captures a more stable representation of the scene background and adapts to changes slowly.
  - Consists of N samples taken from a much larger time window.
  - Updated using a blind update mechanism.
- Combination
  - Intersection of the two model outputs.

[A. Elgammal, D. Harwood, L. Davis, ECCV'00]

Extension: False Detection Suppression

- Problem
  - Small camera motion (e.g., due to wind swaying) may still result in false detections.

- Workaround
  - Consider a small circular neighborhood (e.g., 5×5) $\mathrm{Ne}(\mathbf{x})$ and evaluate the pixel under each neighbor's background model $B_{\mathbf{y}}$:

$$p_{\mathrm{Ne}}(\mathbf{x}^{(t)}) = \max_{\mathbf{y}\in\mathrm{Ne}(\mathbf{x})} p(\mathbf{x}^{(t)}|B_{\mathbf{y}})$$

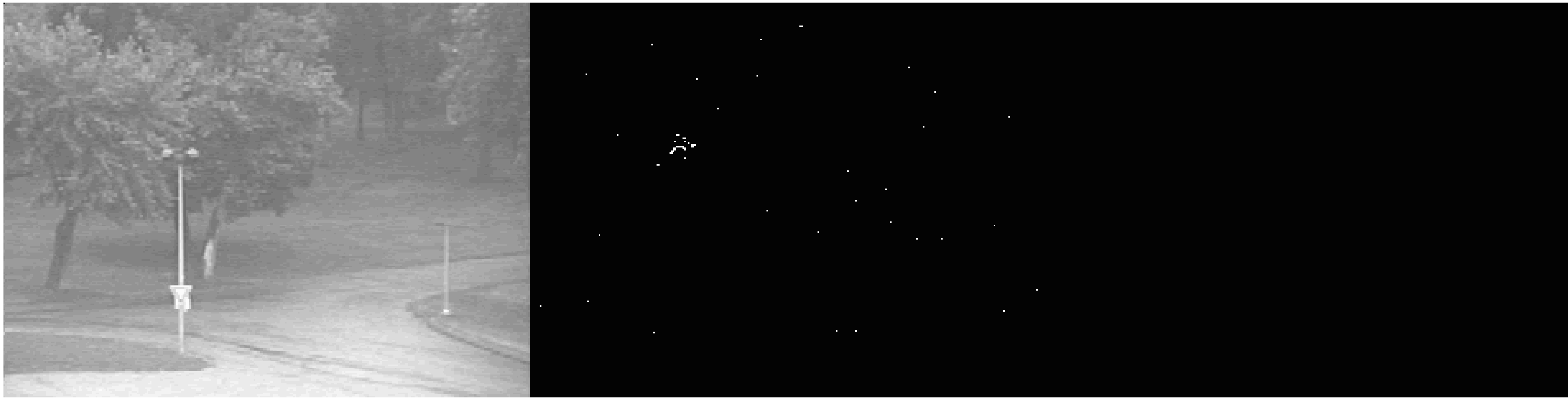
  - Threshold p_{Ne} to determine the foreground pixels.

  $\Rightarrow$ *Eliminates many false detections, but also some true ones.*

  - To avoid losing true detections, add the constraint that an entire connected component must have moved from a nearby location, not only some of its pixels.

[A. Elgammal, D. Harwood, L. Davis, ECCV'00]

Effect of False Detection Suppression

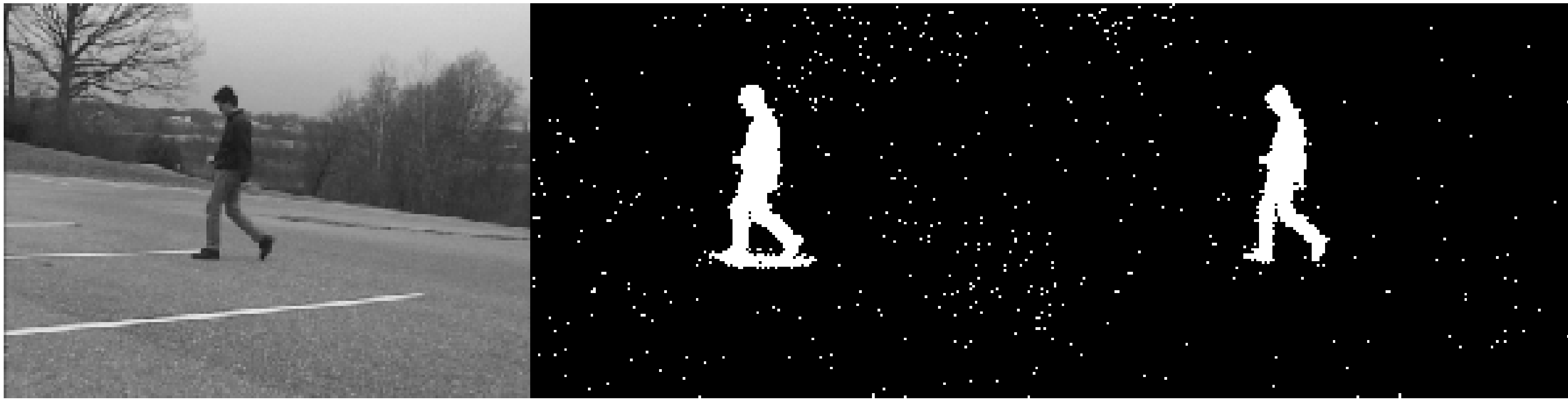


Original video | Without false detection suppr. | With false detection suppr.

- **Results**
  - Effects of camera wind shaking are almost entirely suppressed

Video source: Ahmed Elgammal

Extension: Shadow Suppression



- **Shadows are often detected together with the objects**
  - **Leads to poor localization, should be avoided.**
  - **Idea: Shadowed regions should have the same color as the neighboring background, only the intensity is lower.**

  $\Rightarrow$ Use chromaticity coordinates to remove shadows.

Color Normalization

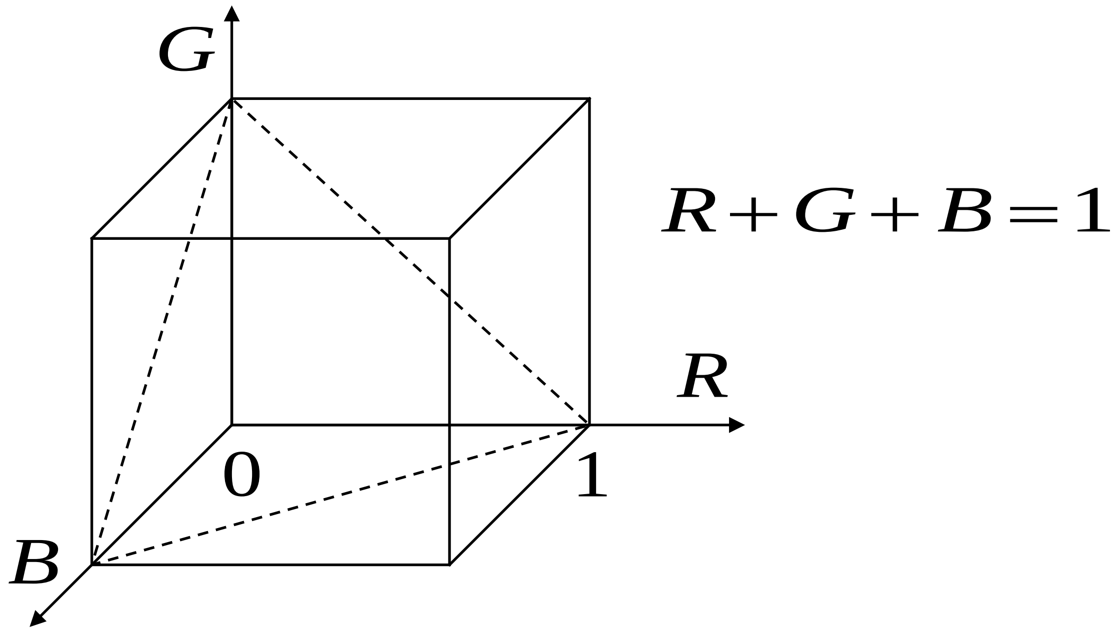
- **One component of the 3D color space is intensity**
  - If a color vector is multiplied by a scalar, the intensity changes, but not the color itself.
  - This means colors can be normalized by the intensity.
    - Intensity is given by $I = R + G + B$:
  - „Chromatic representation“

$$r = \frac{R}{R+G+B} \qquad g = \frac{G}{R+G+B}$$

$$b = \frac{B}{R+G+B}$$

Chromaticity Coordinates

- **Observation:**
  - Since $R + G + B = 1$, only 2 parameters are necessary
  - E.g., one can use R and G and obtains $B = 1 - R - G$



  - Caveat: cannot distinguish between white and gray anymore!

$\Rightarrow$ Use the normalized (r,g) coordinates, but keep the lightness $s = R + B + G$ as third coordinate $\Rightarrow (r,g,s)$

Shadow Removal Procedure

- **Idea**
  - Let $<r, g, s>$ be the expected background pixel color and $<r_t, g_t, s_t>$ be the observed one.
  - Shadows or highlights affect the expected pixel lightness within certain bounds $\alpha \leq s_t/s \leq \beta$.

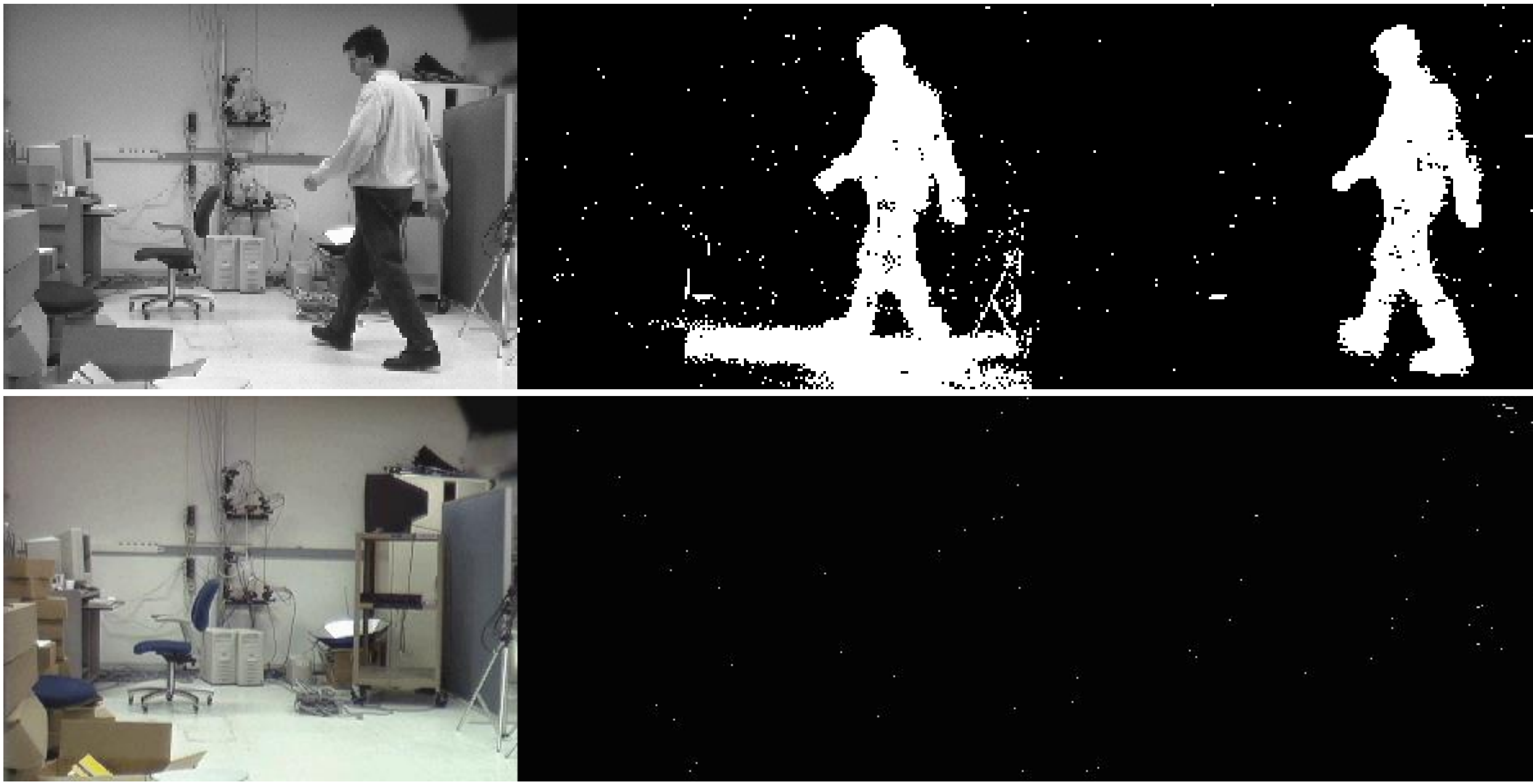
- **Procedure**
  - Select the subset B of relevant sample points for each pixel from the stored set A, i.e. those samples that could produce the observed lightness if affected by shadows:

$$B = \left\{ x_i | x_i \in A \wedge \alpha \leq \frac{s_t}{s_i} \leq \beta \right\}$$

  - Apply the regular kernel background model based on this subset B using only the (r,g) color components.

[A. Elgammal, D. Harwood, L. Davis, ECCV'00]

Effect of Shadow Suppression



Original video | Without shadow suppr. | With shadow suppr.

Video source: Ahmed Elgammal

Topics of This Lecture

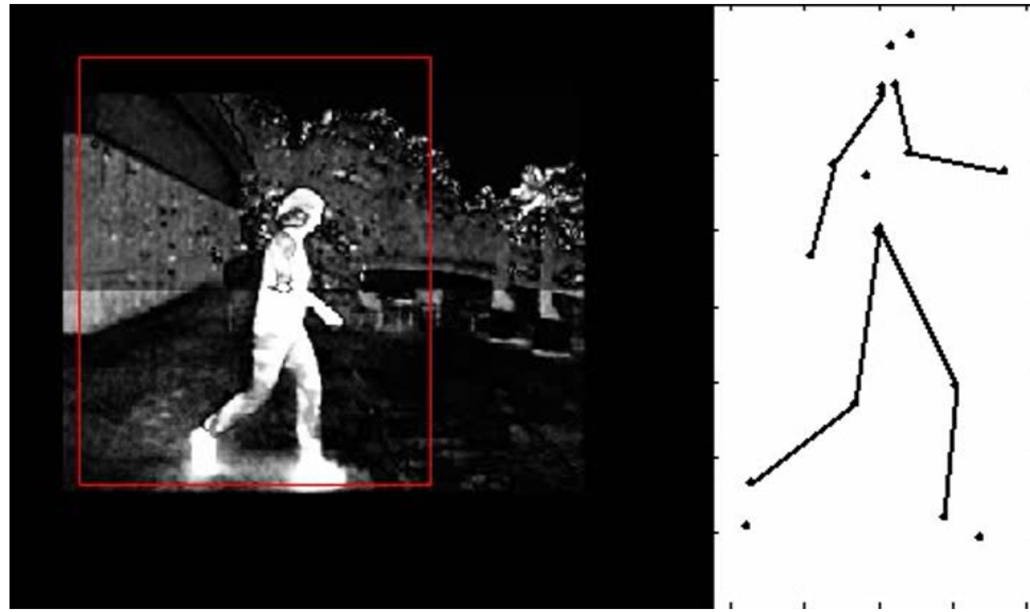
- Motivation: Background Modeling
- Simple Background Models
  - Background Subtraction
  - Frame Differencing
- Statistical Background Models
  - Single Gaussian
  - Mixture of Gaussians
  - Kernel Density Estimation
- Practical Issues and Extensions
  - Background model update
  - False detection suppression
  - Shadow suppression
  - Applications

Applications: Visual Surveillance

- **Background modeling to detect objects for tracking**
  - **Extension: Learning a foreground model for each object.**

Video source: Ian Reid, Univ. of Oxford

Applications: Articulated Tracking



- Background modeling as preprocessing step
  - Track a person's location through the scene
  - Extract silhouette information from the foreground mask.
  - Perform body pose estimation based on this mask.

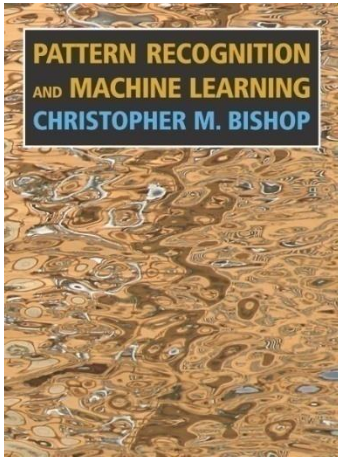
Video source: Hedvik Kjellstroem, Tobias Jaeggli

Summary

- Background Modeling
  - Fast and simple procedure to detect moving object in static camera footage.
  - Makes subsequent tracking *much* easier!
  - $\Rightarrow$ *If applicable, always make use of this information source!*
- We've looked at two models in detail
  - Adaptive MoG model (Stauffer-Grimson model)
  - Kernel background model (Elgammal et al.)
  - Both perform well in practice, have been used extensively.
- Many extensions available
  - Learning object-specific foreground color models
  - Background modeling for moving cameras
  - ...

References and Further Reading

- More information on density estimation in Bishop's book
  - Gaussian distribution and ML: Ch. 1.2.4 and 2.3.1-2.3.4.
  - Mixture of Gaussians: Ch. 2.3.9 and 9
  - Nonparametric methods: Ch. 2.5.
- More information on background modeling:
  - Visual Analysis of Humans: Ch. 3
  - C. Stauffer et al., Adaptive Background Models for Real-Time Tracking, CVPR'98
  - A. Elgammal et al., Non-parametric Model for Background Subtraction, ECCV'00



Christopher M. Bishop
Pattern Recognition and Machine Learning
Springer, 2006

T. Moeslund, A. Hilton, V. Krueger, L. Sigal
Visual Analysis of Humans: Looking at People
Springer, 2011