Change Detection, Skin Detection

Lecture-10

Motivation
- Detection of interesting objects in videos is the first step in the process of automated surveillance.
- Focus of attention method greatly reduces the processing time required for tracking and activity recognition.

Introduction
Objectives:
- Given a sequence of images from a stationary camera identify pixels comprising 'interesting' objects.
  - All independently moving objects are interesting!
- General Solution
  - Model properties of the scene (e.g. color, texture etc.) at each pixel.
  - Significant change in the properties indicates an interesting change.

Introduction
Problems in Realistic situations:
- Moving but uninteresting objects
  - e.g. trees, flags or grass.
- Long term illumination changes
  - e.g. time of day.
- Quick illumination changes
  - e.g. cloudy weather
- Shadows
- Other Physical changes in the background
  - Dropping or picking up of objects
- Initialization

Segmenting Background

Difference Pictures
Background Subtraction

- Problem: Choosing a threshold
  - Pixel is foreground if $I(x,y) - I(x',y') \leq \lambda$
  - What is the correct value of $\lambda$?

Setting a Threshold

MODELING PIXEL INTENSITIES WITH A NORMAL DISTRIBUTION

Each pixel intensity can be modeled by a Normal Distribution, defined in terms of a mean $\mu$ and variance $\sigma^2$, as $N(\mu, \sigma^2)$. $\mu$ and $\sigma$ are called parameters.

Useful when you wish to establish membership of a pixel to one of several models.

$N(\mu, \sigma^2)$ is a probability distribution function defined by:

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

Bi-variate Normal Distribution

- If we were interested in $r-g$, or $g-b$, or $r-b$...

  - The mean can be updated over time simply as
    $$\mu = (1 - \alpha)\mu + \alpha x$$
    $$\sigma^2 = (1 - \alpha)\sigma^2 + \alpha (x - \mu)^2/N$$

Covariance

Tri-variate Normal
Method I: Pfinder

- Pfinder (Person Finder) - Wren et al of MIT (1997)
- Color of each pixel modeled as a three-dimensional Gaussian.
- Big Advantage: adaptivity, pixel-wise 'threshold'.

The Bottom Line

- Model each pixel color as a three dimensional normal distribution
- Adapt the color means and variances over time
- Slowly changing illuminations are handled
- Changes to background are eventually learnt
- Relocation and initialization problems are eventually learnt too.

Limitations

- Unfortunately, this method has limitations
- Due to dynamic nature of real-world scenes modeling pixels with single Gaussian distributions is inaccurate
- Quick illumination changes are not handled
- Good for indoor scenes

Method II: Mixture of Gaussians

- MoG (Mixture of Gaussians) - Stauffer Grimson of MIT (2000)
- Color of each pixel modeled as a mixture of three-dimensional Gaussian
- Big Advantage: Handles Multimodality

Multimodality

In realistic scenarios multiple Processes are generating color \( x \) at each pixel, where \( x=[R,G,B]^T \)

A Background Subtraction Method by Stauffer and Grimson

- A method is required that can incorporate multiple colors in the background model.
Mixture of Gaussians

- Finding the mean and variance for one Gaussian is easy
- Much tougher for the Mixture of Gaussians case
- Find
  - Number of Gaussians
  - Mean and variance of each Gaussian

A Background Subtraction Method by Stauffer and Grimson

For each pixel \((i,j)\) at time 't' each process is modeled as a Gaussian distribution.

- Gaussian distribution is described by a mean \(\bar{m}'\) and a covariance matrix \(\Sigma\).

\[
N(x'_j | \bar{m}'_j, \Sigma_j) = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x'_j - \bar{m}'_j)'\Sigma_j^{-1}(x'_j - \bar{m}'_j)\right)
\]

- Weight \(\omega\) associated with each distribution signifying relevance in recent time.

Thus each Pixel is modeled as a mixture of Gaussians.

A Background Subtraction Method by Stauffer and Grimson

At each frame

- Calculate mahalanobis distance of pixel's color value from each of the associated \(K\) Gaussian distributions

\[
D^2_{ij} = (x'_j - \bar{m}'_j)'\Sigma_j^{-1}(x'_j - \bar{m}'_j)
\]

- If a match is found with the \(k^{th}\) Gaussian, update parameters

\[
m_{ik} = (1-\rho)m_{ik}^{t-1} + \rho x'_{ij}
\]

\[
\Sigma_{ik} = (1-\rho)\Sigma_{ik}^{t-1} + \rho (x'_j - m_{ik}^{t-1})(x'_j - m_{ik}^{t-1})'
\]

where \(\rho\) is a learning parameter

- If a match is not found

  - Replace lowest weight distribution with a new distribution such that

\[
m_{i,new} = x'_{ij}
\]

\[
\Sigma_{i,new} = \Sigma_{i,initial}
\]

- The prior weights of \(K\) distributions are adjusted as

\[
\alpha_{i,new} = (1-\alpha)\omega_{i,new} + \alpha(M_{i,new})
\]

where \(M = 1\) for model that matched and 0 for others

- Foreground= Matched distributions with weight< \(T\)+ Unmatched pixels

Background Modeling using Mixture of Gaussians

Pros

- Handles slow changes in illumination conditions
- Can accommodate physical changes in the background after a certain time interval.
- Initialization with moving objects will correct itself after a certain time interval.
Background Modeling using Mixture of Gaussians

**Cons**
- Can't handle quick changes in illumination conditions, e.g., cloudy weather.
  - Initialization with moving objects
  - Physical changes in background
  - Shadows

Results

Stauffer & Grimson, Javed, Khurram & Shah

Summary of Algorithm

- **Learn** background model by watching 30 second video
- **Detect** moving object by measuring deviations from background model, and applying connected component to foreground pixels.
- **Update** background and blob statistics

Kanade
Summary

- Very similar to k-Gaussian with following differences:
  - uses only single Gaussian
  - uses gray level images, the mean and variance are scalar values

Algorithm

- Learn background model by watching 30 second video
- Detect moving object by measuring deviations from background model, and applying connected component to foreground pixels.
- Update background and region statistics

Detection

- During detection if intensity value is more than two sigma away from the background it is considered foreground:
  - keep original mean and variance
  - track the object with new mean and variance
  - if new mean and variance persists for sometime, then substitute the new mean and variance as the background model
  - If object is no longer visible, it is incorporated as part of background

W4

- Compute “minimum” (M(x)), “maximum” (N(x)), and “largest absolute difference” (L(x)).

\[ D_i(x, y) = \begin{cases} 1 & \text{if } |M(x, y) - f(x, y)| > L(x, y) \text{or} \\ N(x, y) - f(x, y) > L(x, y) \\ 0 & \text{otherwise} \end{cases} \]

- Theoretically, the performance of this tracker should be worse than others.
- Even if one value is far away from the mean, then that value will result in an abnormally high value of \( L \).
- Having short training time is better for this tracker.
Limitations

- Occlusion
- Shadows
- Slow moving people
- Multiple processes (swaying of trees..)
- Quick Illumination Changes

Webpage

- Http://www.cs.cmu.edu/~vsam

Skin Detection

Kjeldsen and Kender

Training

- Crop skin regions in the training images.
- Build histogram of training images.
- Ideally this histogram should be bi-modal, one peak corresponding to the skin pixels, other to the non-skin pixels.
- Practically there may be several peaks corresponding to skin, and non-skin pixels.

freq

Gray levels: 0 - 255

Training

- Apply threshold to skin peaks to remove small peaks.
- Label all gray levels (colors) under skin peaks as "skin", and the remaining gray levels as "non-skin".
- Generate a look-up table for all possible colors in the image, and assign "skin" or "non-skin" label.
Detection

- For each pixel in the image, determine its label from the "look-up table" generated during training.

Building Histogram

- Instead of incrementing the pixel counts in a particular histogram bin:
  - for skin pixel increment the bins centered around the given value by a Gaussian function.
  - For non-skin pixels decrement the bins centered around the given value by a smaller Gaussian function.

Example training images

Results of skin detection
Results of skin detection

Detecting Fire