CAP6411
Computer Vision Systems
Lecture 7

Alper Yilmaz
Office: CSB 250
Email: yilmaz@cs.ucf.edu
Web: http://www.cs.ucf.edu/courses/**cap6411/cap6411**/spring2006

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 e.t.c) at each pixel
  - Significant change in the properties indicates an interesting change

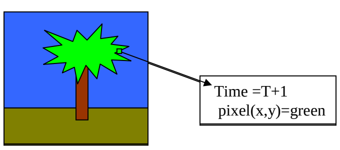
Segmenting Background

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

Background Subtraction Method by Stauffer and Grimson

- In realistic scenarios multiple Processes are generating color '*x*' at each pixel, where $x=[R,G,B]^T$



- A method is required that can incorporate multiple colors in the background model

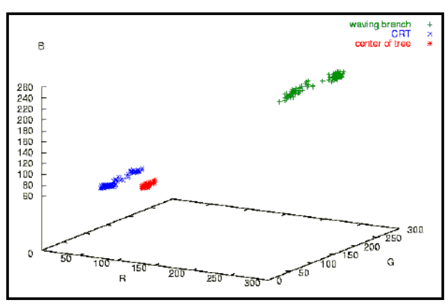
Solution

- For each pixel *(i,j)* at time '*t*' each process is modeled as a Gaussian distribution.
  - Gaussian distribution is described by a mean 'm' and a covariance matrix Σ.

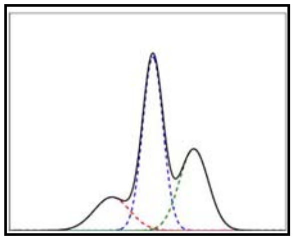
$$N(x_{i,j}^t \mid m_{i,j}^t, \Sigma_{i,j}^t) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \Sigma_{i,j}^t \mid} e^{-\frac{1}{2}(x_{i,j}^t - m_{i,j}^t)^T (\Sigma_{i,j}^t)^{-1} (x_{i,j}^t - m_{i,j}^t)}$$

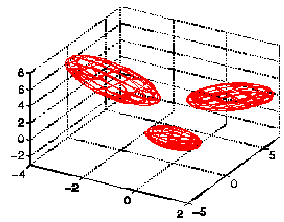
  - Weight ω associated with each distribution signifying relevance in recent time.
- Thus each Pixel is modeled as a mixture of Gaussians.

Multimodality



Mixture of Gaussians

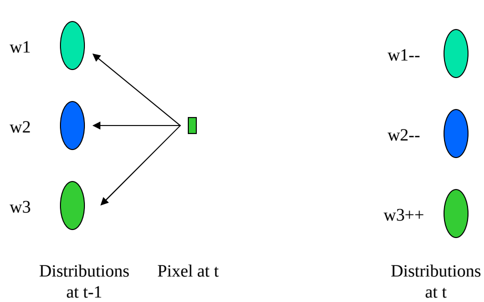




- Finding the mean and variance for one Gaussian is easy
- Much tougher for the Mixture of Gaussians case
- Find
  - Number of Gaussians
  - Mean and variance of each Gaussian

Solution

- At each frame
  - Calculate Mahalanobis distance of pixel's color value from each of the associated K Gaussian distributions



Solution

- If a match is found with the *k*th Gaussian, update parameters

$$m_{i,j}^{t,k} = (1-\rho)m_{i,j}^{t-1,k} + \rho x_{i,j}^{t}$$

$$\Sigma_{i,j}^{t,k} = (1-\rho)\Sigma_{i,j}^{t-1,k} + \rho(x_{i,j}^{t} - m_{i,j}^{t})(x_{i,j}^{t} - m_{i,j}^{t})^{T}$$

where p is a learning parameter

Solution

- If a match is not found
  - Replace lowest weight distribution with a new distribution such that

$$m_{i,j}^{t,new} = x_{i,j}^{t}$$

$$\Sigma_{i,j}^{t,new} = \Sigma^{initial}$$

  - The prior weights of K distributions are adjusted as

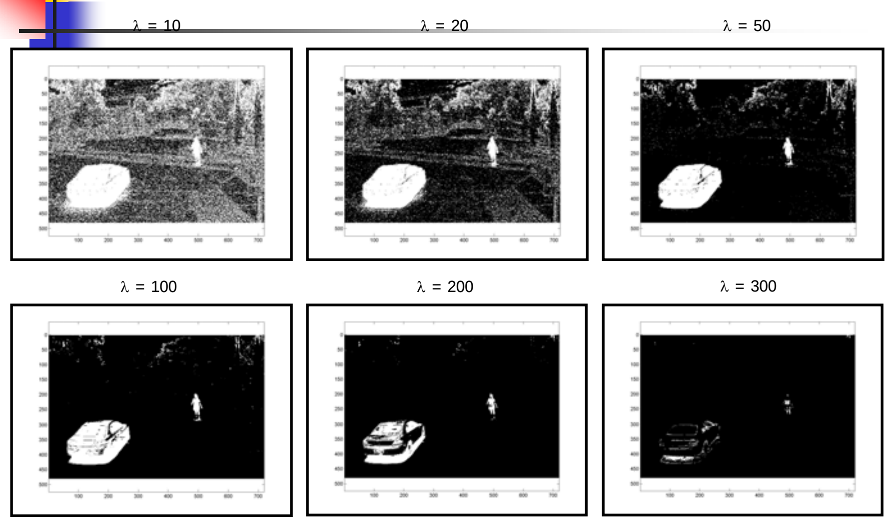
$$\omega_{i,j}^{t-1} = (1-\alpha)\omega_{i,j}^{t-1} + \alpha(M_{i,j}^{t-1})$$

    - M is1 for model that matched and 0 for others

Background Subtraction

- Problem: Choosing a threshold
  - Pixel is foreground if $|I_1(x,y)-I_2(x,y)| \leq \lambda$ otherwise background?
  - What is the correct value of λ ?

Setting a Threshold



Pros

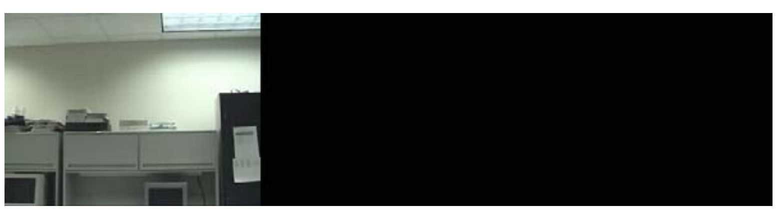
- Handles slow changes in illumination conditions
- Can accommodate physical changes in the background after a certain time interval.
- Initialization with moving objects will correct itself after a certain time interval.

Cons

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

Results

Results



Results

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

Background Models
Camera Motion

- Mobile camera
- Nominal motion

Removing Camera Motion

- Affine Transformation (Anandan)
- Projective Transformation (Mann-Pickard)

Projective Flow (weighted)

$$u\,f_x + v\,f_y + f_t = 0$$

Optical Flow const. equation

$$\mathbf{u}^T \mathbf{f}_{\mathbf{x}} + f_t = 0$$

$$\mathbf{x}' = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathrm{x} + 1}$$

Projective transform

$$\mathbf{u} = \mathbf{x}' - \mathbf{x} = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathbf{T}}\mathrm{x} + 1} - \mathbf{x}$$

Projective Flow (weighted)

$$\varepsilon_{flow} = \sum (\mathbf{u}^{\mathbf{T}} \mathbf{f}_{\mathbf{X}} + f_{\mathbf{t}})^2$$

$$= \sum ((\frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}\mathbf{x}^{\mathbf{T}} + 1} - \mathbf{x})^T \mathbf{f}_{\mathbf{x}} + f_t)^2$$

$$= \sum ((\mathbf{A}\mathbf{x} + \mathbf{b} - (\mathbf{C}^{\mathbf{T}}\mathbf{x} + 1)\mathbf{x})^T \mathbf{f}_x + (\mathbf{C}^T\mathbf{x} + 1)\,f_t)^2$$

$\Downarrow$ **minimize**

Projective Flow (weighted)

$$(\sum \phi\phi^T)\mathbf{a} = \sum(\mathbf{x}^{\mathrm{T}}\mathbf{f}_x - f_t)\phi$$

$$\mathbf{p} = [a_1, a_2, b_1, a_3, a_4, b_2, c_1, c_2]^T$$

$$\phi^t = [\, f_x x,\ f_x y,\ f_x,\ f_y x,\ f_y y,\ f_y,\ x f_t - x^2 f_x - xy f_y,\ y f_t - xy f_x - y^2 f_y\,]$$

Projective Flow (unweighted)

Pseudo-Perspective

$$\mathbf{x}' = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathrm{T}}\mathrm{x} + 1}$$

$\Downarrow$ **Taylor Series**

$$\begin{aligned} x + u &= a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy \\ y + v &= a_6 + a_7 x + a_8 y + a_4 xy + a_5 y^2 \end{aligned}$$

Bilinear

$$\mathbf{x}' = \frac{A\mathbf{x} + \mathbf{b}}{\mathbf{C}^{\mathrm{T}}\mathrm{x} + 1}$$

$\Downarrow$ Taylor Series & remove Square terms

$$\begin{aligned} u + x &= a_1 + a_2 x + a_3 y + a_4 xy \\ v + y &= a_5 + a_6 x + a_7 y + a_8 xy \end{aligned}$$

Projective Flow (unweighted)

$$\varepsilon_{flow} = \sum (\mathbf{u}^{\mathrm{T}}\mathbf{f}_{\mathrm{X}} + f_{\mathrm{t}})^2$$

Minimize

Bilinear and Pseudo-Perspective

Homework: Derive this equation Due Feb 8

$$(\sum \Phi\Phi^T)\mathbf{q} = -\sum f_t\Phi$$

$$\Phi^T = [f_x(xy, x, y, 1),\ f_y(xy, x, y, 1)]$$

bilinear

$$\Phi^T = [f_x(x, y, 1) \quad f_y(x, y, 1) \quad c_1 \quad c_2]$$

$$c_1 = x^2 f_x + xyf_x$$

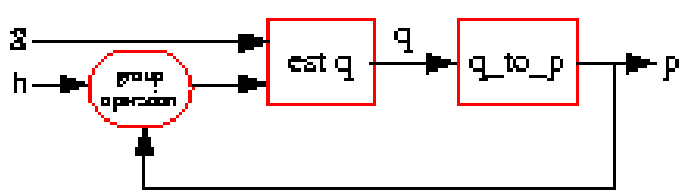
$$c_2 = xyf_x + y^2 f_y$$

Pseudo perspective

Algorithm-1

- Estimate “q” (using approximate model, e.g. bilinear model).
- Relate “q” to “p”
  - select four points S1, S2, S3, S4
  - apply approximate model using “q” to compute (x'_k, y'_k)
  - estimate exact “p”:

Flow



True Projective

$$x' = \frac{a_1 x + a_2 y + b_1}{c_1 x + c_2 y + 1} \qquad y' = \frac{a_3 x + a_4 y + b_1}{c_1 x + c_2 y + 1}$$

$$\begin{bmatrix} x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} x_k & y_k & 1 & 0 & 0 & 0 & -x_k x'_k & -y_k x'_k \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y'_k & -y_k y'_k \end{bmatrix} \mathbf{a}$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & b_1 & a_3 & a_4 & b_2 & c_1 & c_1 \end{bmatrix}^T$$

$$\begin{bmatrix} x'_1 \\ y'_1 \\ \\ \\ x'_k \\ y'_k \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1 x'_1 & -y_1 x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1 y'_1 & -y_1 y'_1 \\ \\ \\ x_k & y_k & 1 & 0 & 0 & 0 & -x_k x' & -y_k x'_k \\ 0 & 0 & 0 & x_k & y_k & 1 & -x_k y'_k & -y_k y' \end{bmatrix} \mathbf{a}$$

$$\mathbf{P} = \mathbf{A}\mathbf{a}$$

Perform least squares fit to compute a.

Final Algorithm

- A Gaussian pyramid of three or four levels is constructed for each frame in the sequence.
- The parameters "p" are estimated at the top level of the pyramid, between the two lowest resolution images, "g" and "h", using algorithm-1.

Final Algorithm

- The estimated "p" is applied to the next higher resolution image in the pyramid, to make images at that level nearly congruent.
- The process continues down the pyramid until the highest resolution image in the pyramid is reached.

Presentations

- Support Vector Machines: **Vladimir** (February 8, 2006)
- Graph-Cut: **Paul** (February 13, 2006)
- Adaboost: **Alex** (February 15, 2006)