Change Detection

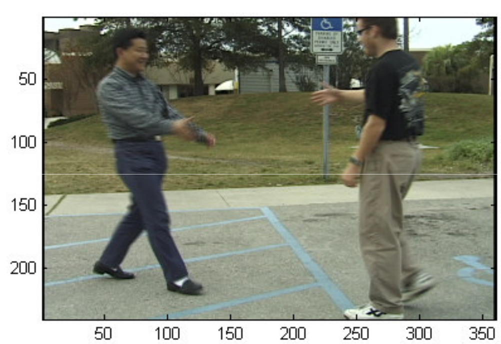
Credits: Some slides taken from Khurram Shafique at UCF

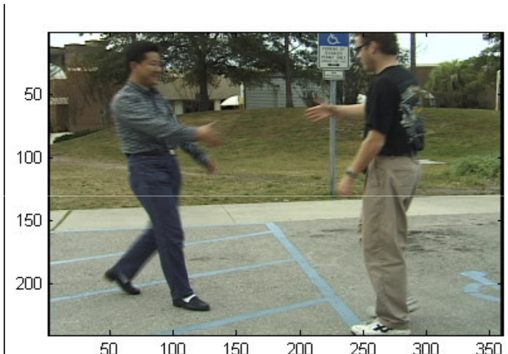
Change Detection

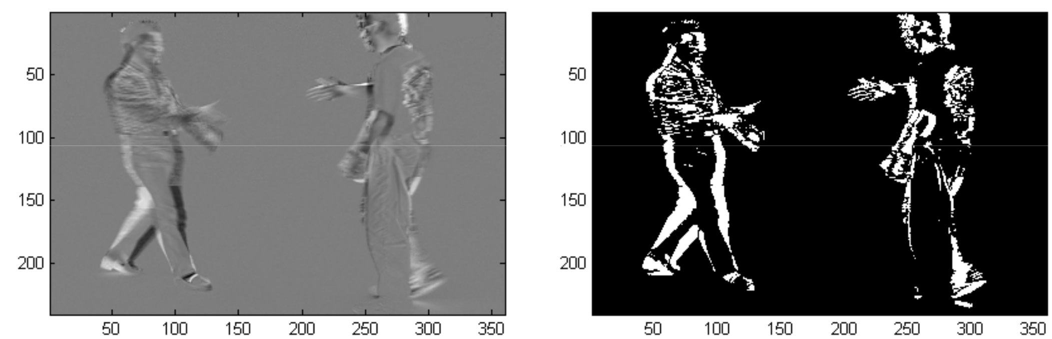
- Motivation:
  - To detect interesting objects in video
  - First step in automated surveillance and tracking
- Objective
  - Given a sequence of images from a **stationary** camera
  - Identify pixels that comprise 'moving' objects
  - Foreground: pixels that comprise moving objects – pixels of interest
  - Background: All other pixels in the image

Simple Form of Change Detection

- Detect how many pixels have changed their color significantly
- HOW?
- Take the time-derivative and threshold

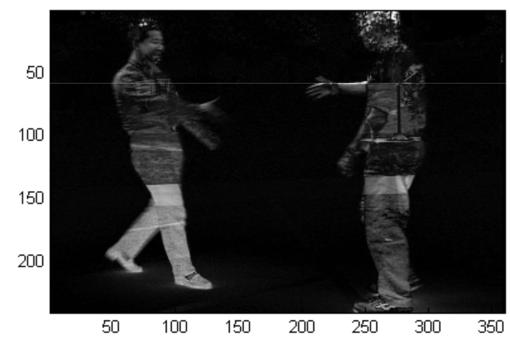


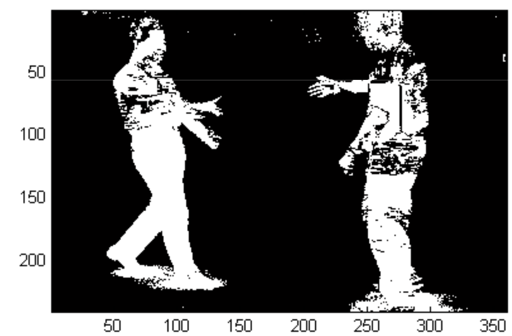


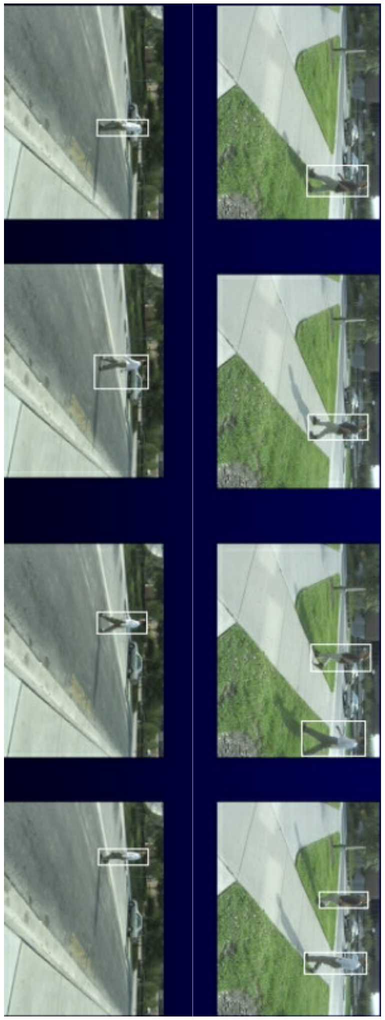


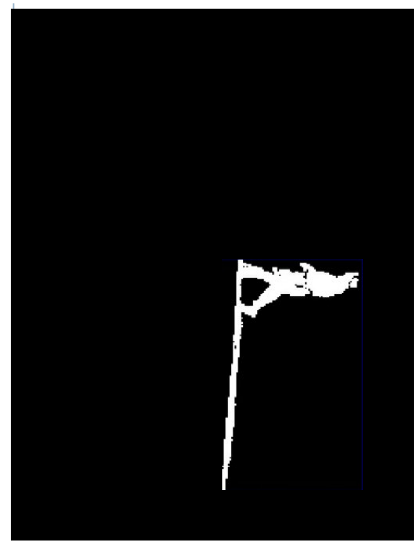
Background Subtraction

- Instead of subtracting from previous image, subtract from the first image
- Assuming first image was only background


50
100
150
200
50
100
150
200
250
300
350


50
100
150
200
50
100
150
200
250
300
350





Change Detection

- 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.

Basic Background Subtraction

- Acquire a background model
  - Simplest case: Take first image as background
- Take the difference of each new image from background image
- Take the absolute value and threshold
- Difference will be low where no change has happened. High for foreground regions

Improved Background Subtraction

- Every pixel has a ‘normal’ variation
- If 10 images are taken of the same scene, pixel values in all 10 images will not be the same
- Pixels at edges may exhibit more variation
- Model variation with a Gaussian distribution
- Estimate its mean and variance

Background Model

- Simple background model assuming Gaussian distribution of pixel color:
  - for a set of k frames that contain only background:
  - for every pixel (x, y) with color x
    - mean: $\boldsymbol{\mu} = \frac{1}{k}\sum_k \mathbf{x}_k$
    - covariance: $\Sigma = \frac{1}{k}\sum_k (\mathbf{x}_k - \boldsymbol{\mu}_k)(\mathbf{x}_k - \boldsymbol{\mu}_k)^T$

Mean, Variance and Covariance

Let two features x and y and n observations of each feature be $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$ respectively.

Mean: $$m = [m_x \quad m_y]^T = \frac{1}{n}\left[\sum_{i=1}^{n} x_i \quad \sum_{i=1}^{n} y_i\right]^T$$

Variance: $$\sigma_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - m_x)^2 \quad \sigma_y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - m_y)^2$$

Covariance: $$\sigma_{xy}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)$$

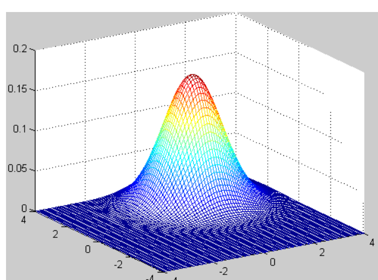
Covariance Matrix: $$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy}^2 \\ \sigma_{xy}^2 & \sigma_y^2 \end{bmatrix}$$

2D Gaussian

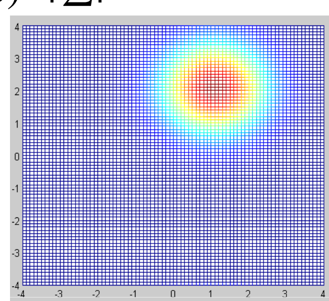
$$g(\mathbf{x} \mid \mathbf{m}, \Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \mid \Sigma \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathbf{T}}(\Sigma)^{-1}(\mathbf{x}-\mathbf{m})}$$

$$m = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

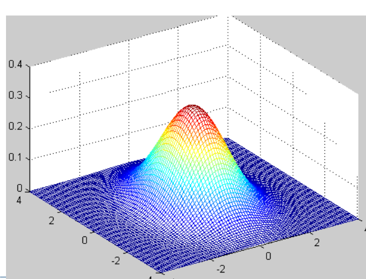


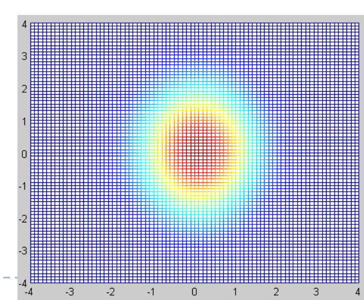
$g($



$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

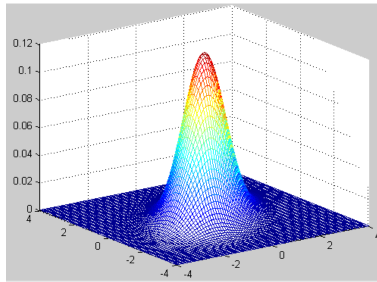


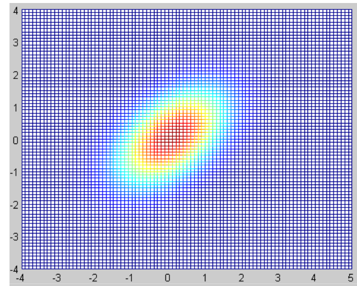


2D Gaussian

$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

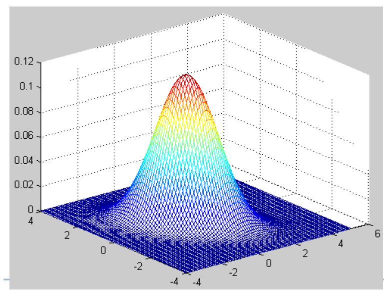
$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

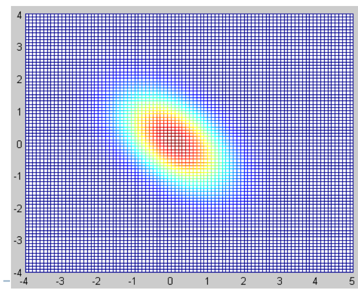




$$m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$





Gaussian Background Subtraction

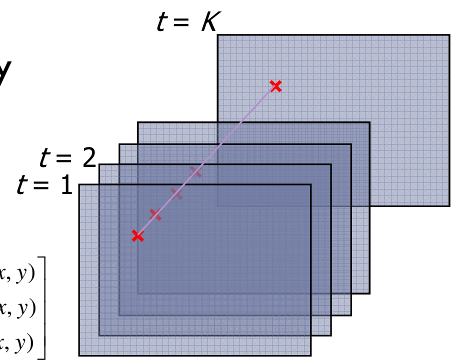
- Training
  - For all training images
  - Find mean color vector **m** of each pixel (x,y) in all images
  - Find color covariance matrix **Σ** of each pixel (x,y) in all images
- Detection
  - For each pixel (x,y) with color $\mathbf{x} = [R,G,B]^T$
  - Compute

$$P(\mathbf{x} \mid \mathbf{m}, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T (\Sigma)^{-1} (\mathbf{x}-\mathbf{m})}$$

  - Threshold

Gaussian Background Subtraction

- **m** and **Σ** are computed for every pixel using *K* training images



$$\mathbf{m}(x,y)=\begin{bmatrix}\frac{1}{K}\sum_{t=1}^{K} r(x,y,t)\\ \frac{1}{K}\sum_{t=1}^{K} g(x,y,t)\\ \frac{1}{K}\sum_{t=1}^{K} b(x,y,t)\end{bmatrix}\qquad \mathbf{\Sigma}(x,y)=\begin{bmatrix}\sigma_r^2(x,y) & \sigma_{rg}(x,y) & \sigma_{rb}(x,y)\\ \sigma_{rg}(x,y) & \sigma_g^2(x,y) & \sigma_{gb}(x,y)\\ \sigma_{rb}(x,y) & \sigma_{gb}(x,y) & \sigma_b^2(x,y)\end{bmatrix}$$

$$\mathbf{\Sigma}(x,y)=\begin{bmatrix}\frac{1}{K-1}\sum_{t=1}^{K}(r(x,y,t)-\mathbf{m}_r(x,y))^2 & \frac{1}{K-1}\sum_{t=1}^{K}(r(x,y,t)-\mathbf{m}_r(x,y))(g(x,y,t)-\mathbf{m}_g(x,y)) & \frac{1}{K-1}\sum_{t=1}^{K}(r(x,y,t)-\mathbf{m}_r(x,y))(b(x,y,t)-\mathbf{m}_b(x,y))\\ \frac{1}{K-1}\sum_{t=1}^{K}(r(x,y,t)-\mathbf{m}_r(x,y))(g(x,y,t)-\mathbf{m}_g(x,y)) & \frac{1}{K-1}\sum_{t=1}^{K}(g(x,y,t)-\mathbf{m}_g(x,y))^2 & \frac{1}{K-1}\sum_{t=1}^{K}(g(x,y,t)-\mathbf{m}_g(x,y))(b(x,y,t)-\mathbf{m}_b(x,y))\\ \frac{1}{K-1}\sum_{t=1}^{K}(r(x,y,t)-\mathbf{m}_r(x,y))(b(x,y,t)-\mathbf{m}_b(x,y)) & \frac{1}{K-1}\sum_{t=1}^{K}(g(x,y,t)-\mathbf{m}_g(x,y))(b(x,y,t)-\mathbf{m}_b(x,y)) & \frac{1}{K-1}\sum_{t=1}^{K}(b(x,y,t)-\mathbf{m}_b(x,y))^2\end{bmatrix}$$

Simplifications

1. Instead of thresholding $P(\mathbf{x}|\mathbf{m},\Sigma)=\frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathrm{T}}(\Sigma)^{-1}(\mathbf{x}-\mathbf{m})}$

   you may just threshold the exponent

$$d^2=(\mathbf{x}-\mathbf{m})^{\mathrm{T}}(\Sigma)^{-1}(\mathbf{x}-\mathbf{m})$$

d called *Mahalanobis Distance*

Simplifications

2. Assume **Σ** to be diagonal
   Then

$$d^2 = (\mathbf{x}-\mathbf{m})^{\mathrm{T}}(\Sigma)^{-1}(\mathbf{x}-\mathbf{m})$$

$$= \begin{bmatrix} \mathbf{x}_r - \mathbf{m}_r \\ \mathbf{x}_g - \mathbf{m}_g \\ \mathbf{x}_b - \mathbf{m}_b \end{bmatrix}^T \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_g^2 & 0 \\ 0 & 0 & \sigma_b^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{x}_r - \mathbf{m}_r \\ \mathbf{x}_g - \mathbf{m}_g \\ \mathbf{x}_b - \mathbf{m}_b \end{bmatrix}$$

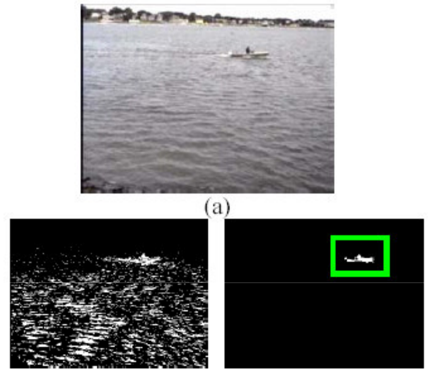
$$= \frac{(\mathbf{x}_r - \mathbf{m}_r)^2}{\sigma_r^2} + \frac{(\mathbf{x}_g - \mathbf{m}_g)^2}{\sigma_g^2} + \frac{(\mathbf{x}_b - \mathbf{m}_b)^2}{\sigma_b^2}$$

Simplifications

3. Instead of reading in all *K* images into memory to compute the mean and the variances, they can be read in one by one

- Running mean: $E(\mathbf{x}) = 1/N * \mathbf{x}_N + (N-1)/N * \mathbf{m}_{N-1}$
- Running Variance: ?
- $E((\mathbf{x} - \mathbf{m})^2) = E(\mathbf{x}^2) - \mathbf{m}^2$

- 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
    - e.g. dropping or picking up of objects
  - Initialization



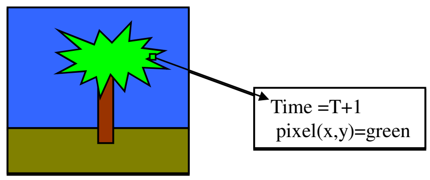
(a)

Issues

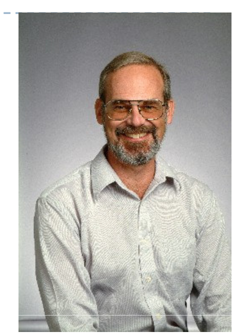
- Adaptivity
  - Background model must be adaptive to changes in background.
- Multiple Models
  - Multiple processes generate color at every pixel. The background model should be able to account for these processes.
- Weighting the observations (models)
  - The system must be able to weight the observation to make decisions. For example, the observations made a long time back should have less weight than the recent observations. Similarly, the frequent observations are more important than the ones with less occurrence.

Issues

- Pixel level Color Modeling
  - Multiple Processes are generating color '*x*' at each pixel
    - Where $x=[R,G,B]^T$



Background Subtraction – Grimson's Method



- Need:
  - Background pixels are often composed of multiple processes.
  - Therefore unimodal pdf is not enough
- Problem:
  - It is hard to estimate the parameters of a multi-modal pdf (weights, mean, covariance)
  - The parameters need to be kept updated if the system is to work for a longer period of time

Ref: Chris Stauffer and Eric Grimson, 'Learning Patterns of Activity using real-time tracking', *PAMI* Aug 2000

Grimson’s Method

- Model each pixel with a mixture of Gaussians
- A mixture of Gaussians can be used to approximate any arbitrary pdf

Gaussian pdf (Multivariate)

$$g(\mathbf{x} \mid \mathbf{m},\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}} \mid\Sigma\mid} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathbf{T}}(\Sigma)^{-1}(\mathbf{x}-\mathbf{m})}$$

Mixture of Gaussians pdf

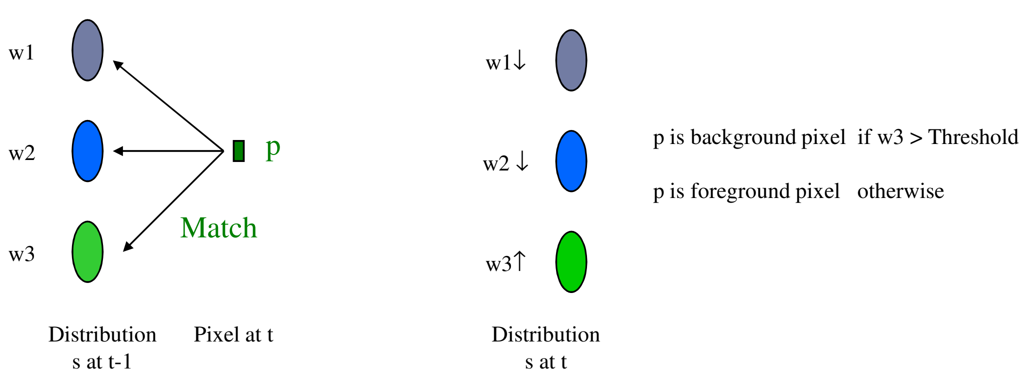
$$f(\mathbf{x} \mid w_1,\mathbf{m}_1,w_2,\Sigma_1,\mathbf{m}_2,\Sigma_2,\ldots,w_K,\mathbf{m}_K,\Sigma_K,) = \sum_{i=1}^{K} \frac{w_i}{(2\pi)^{\frac{d}{2}} \mid\Sigma_i\mid} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^{\mathbf{T}}(\Sigma_i)^{-1}(\mathbf{x}-\mathbf{m}_i)}$$

Grimson’s Method

At each frame

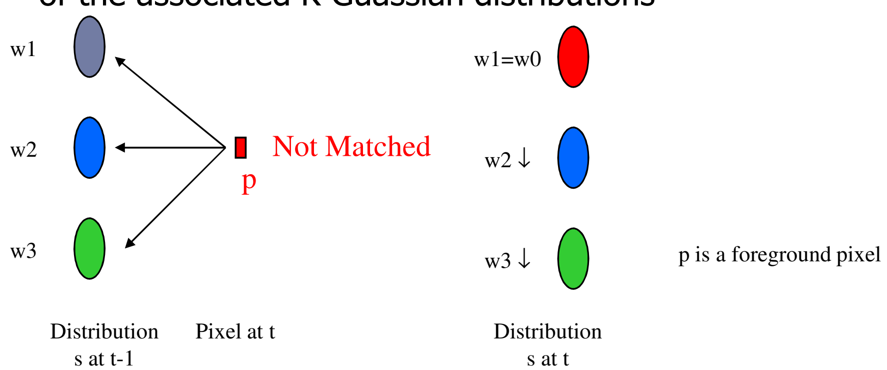
For each pixel

Calculate distance of pixel’s color value from each of the associated *K* Guassian distributions



Grimson’s Method

- At each frame
- For each pixel
  - Calculate distance of pixel’s color value from each of the associated K Guassian distributions



w1 ≤ w2 ≤ w3

Summary – Grimson

- Each pixel is an independent statistical process, which may be combination of several processes.
  - Swaying branches of tree result in a bimodal behavior of pixel intensity.
- The intensity is fit with a mixture of K Gaussians.

$$f(\mathbf{x} \mid w_1, \mathbf{m}_1, w_2, \Sigma_1, \mathbf{m}_2, \Sigma_2, \ldots, w_K, \mathbf{m}_K, \Sigma_K,) = \sum_{i=1}^{K} \frac{w_i}{(2\pi)^{\frac{d}{2}} \mid \Sigma_i \mid} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^{\mathrm{T}}(\Sigma_i)^{-1}(\mathbf{x}-\mathbf{m}_i)}$$

- **x** is the observation $[R, G, B]^T$ of the current pixel
- w_i , $\mathbf{m}_i$ and Σ_i are the parameters of the i^{th} Gaussian distribution
- For simplicity, it may be assumed that RGB color channels are independent and have the same variance σ^2.
- In this case $\Sigma = \sigma^2 I$, where I is a 3x3 unit matrix.

Summary - Grimson

- Every new pixel is checked against all existing distributions. The match is the distribution with Mahalanobis distance less than a threshold.

$$d = \sqrt{(\mathbf{x}-\mathbf{m})^{\mathbf{T}}(\sum)^{-1}(\mathbf{x}-\mathbf{m})}$$

- The mean and variance of unmatched distributions remain unchanged. For the matched distributions they are updated as

$$\mathbf{m}^{t,k} = (1-\rho)\mathbf{m}^{t-1,k} + \rho\mathbf{x}^{t}$$

$$\mathbf{\Sigma}^{t,k} = (1-\rho)\mathbf{\Sigma}^{t-1,k} + \rho(\mathbf{x}^{t}-\mathbf{m}^{t})(\mathbf{x}^{t}-\mathbf{m}^{t})^{T}$$

Summary - Grimson

- For the unmatched pixel, replace the lowest weight Gaussian with the new Gaussian with mean at the new pixel and an initial estimate of covariance matrix.

- The weights are adjusted:

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

$$M_{ij}^{t-1} = \begin{cases} 1 & \text{if distribution matches} \\ 0 & \text{otherwise} \end{cases}$$

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

Results - Grimson

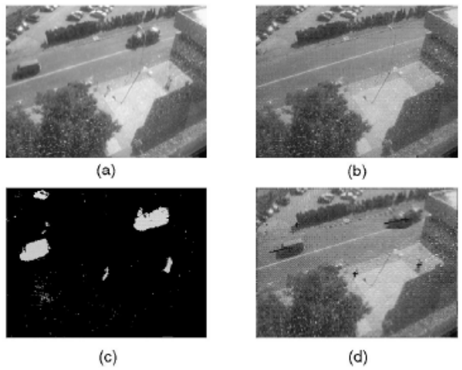


Fig. 1. The execution of the program. (a) the current image, (b) an image composed of the means of the most probable Gaussians in the background model, (c) the foreground pixels, (d) the current image with tracking information superimposed. Note: While the shadows are foreground in this case, if the surface was covered by shadows a significant portion of the time, a Gaussian representing those pixel values may be significant enough to be considered background.

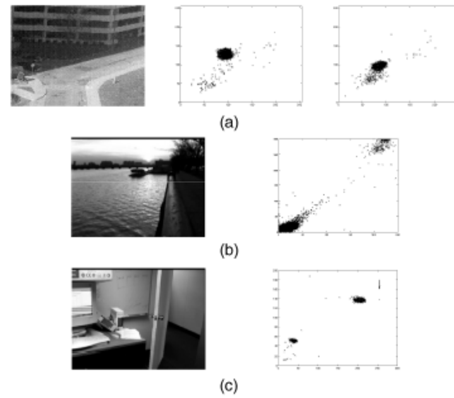


Fig. 2. This figure contains images and scatter plots of the red and green values of a single pixel from the image over time. It illustrates some of the difficulties involved in real environments. (a) shows two scatter plots from the same pixel taken 2 minutes apart. This would require two thresholds. (b) shows a bi-model distribution of a pixel values resulting from specularities on the surface of water. (c) shows another bi-modality resulting from monitor flicker.