Advanced Topics in Computer Vision for Video Surveillance

Lecture 2: Background Modeling

Jan 2009

Tony X. Han
Electrical and Computer Engineering Department
University of Missouri
1. Introduction

Goal:

Given a video from a fixed (may be moving under some circumstances) camera, detecting all the foreground objects.
Definition

**Foreground:**
- If the background is static, the moving objects are defined as foreground.
- Or determined by the user and specific tasks.
  - Using labeling tools such as bounding box.
  - Using semantic information such as people, car, animal. Any objects of interest are specified as foreground.
Definition

Background:
- If the camera is fixed, everything static is regarded as background, unless it is specified by the users and the tasks as foreground.
- If the camera is moving, all area consistent to the camera movement is background component.
  - Motion compensation is needed in order to apply the background subtraction technique.
Key problems:

- How to automatically obtain the image of the whole background? (Especially in an busy place, where empty scene is not available)

- What is the variance of the background? How to select the threshold to determine the foreground pixels?
1. Introduction (cont.)

- Key problems (cont.):
  - How to handle slow or periodic background variations (E.g. tree waving, sea waves)?
    - There is a trade-off between fast-adaption and being robust to short-time static foreground.
  - How to accommodate drastic or fast background change (E.g. turn on/off a desk lamp. Sun occluded by clouds)?
2. Heikkila and Silven method


- A pixel is marked as foreground if

\[ |I_t - B_t| > \tau \]

where \( \tau \) is a predefined threshold.

- The threshold may be learned.
- The threshold need to be tuned for different video.
After thresholding, the morphological operation “closing” is applied.

- The closing kernel is 3X3.
- Small regions are disregarded.

What are morphological operations?

- Erosion
- Dilation
- Opening
- Closing
Erosion

\[ E = B \bigcirc \otimes S = \{(x, y) | S(x, y) \subseteq B\} \]
Dilation

\[ E = B \oplus S = \{(x, y) \mid S_{(x, y)} \cap B \neq \emptyset\} \]
Opening

\[ E = B \circ S = \left( B \bigotimes S \right) \bigoplus S \]
\[ E = B \bullet S = \left( B \oplus S \right) \otimes S \]
2. Heikkila and Silven method (cont.)

- **Update:** \( B_{t+1} = \alpha I_t + (1 - \alpha)B_t \)

  where \( \alpha \) is kept small to prevent artificial “tails” forming behind moving objects.

- **Two background corrections are applied**

  1. If a pixel is marked as foreground for more than \( m \) of the last \( M \) frames, then the background is updated as \( B_{t+1} = I_t \). This correction is designed to compensate for sudden illumination changes and the appearance of static new objects.

  2. If a pixel changes state from foreground to background frequently, it is masked out from inclusion in the foreground. This is designed to compensate for fluctuating illumination, such as swinging branches.
3. Background Modeling by Mixture of Gaussians (MOG)


- Each pixel is modeled *independently* by a mixture of K Gaussians

\[ p(I_t) = \sum_{i=1}^{K} \omega_{i,t} N(I_t; \mu_{i,t}, \Sigma_{i,t}) \]

where,

\[ N(x; \mu, \Sigma) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp\left( -\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) \right) \]
Properties of Gaussian Distribution

1. **Central Limit Theorem:**

Let \( X_1, X_2, \ldots, X_N \) be a set of \( N \) independent random variates and each \( X_i \) have an arbitrary probability distribution \( P(x_1, \ldots, x_N) \) with mean \( \mu_i \) and a finite variance \( \sigma_i^2 \). Then the normal form variable

\[
X_{\text{norm}} = \frac{\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} \mu_i}{\sqrt{\sum_{i=1}^{N} \sigma_i^2}}
\]

has a limiting cumulative distribution function which approaches a normal distribution.

The "fuzzy" central limit theorem says that data which are influenced by many small and unrelated random effects are approximately normally distributed.
Properties of Gaussian Distribution (cont.)

1. If we know a random variable is normally distributed,
Lecture 2 (Continued)

3. Mixture of Gaussian Approach (Grimison, Stauffer, CVPR98)

\[ P(I_t) = \frac{K}{\sum_{k=1}^{K} w_{k,t} N(I_t; \mu_{k,t}, \Sigma_{k,t})} \quad k = 3, 4, 5, \ldots, 8 \]

\[ N(I_t, \mu, \Sigma) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \text{exp}(-\frac{1}{2}(I_t-\mu)^T \Sigma^{-1} (I_t-\mu)) \]

In the paper, it is assumed that \( \Sigma_{k,t} = \sigma_{k,t}^2 I \) (Identity matrix)

A. Background update

1) If \( I_t \) match to the \( i \)th Gaussian component, i.e. within \( M_{i,t} \)

\[ w_{i,t} = w_{i,t-1} \]

\[ M_{i,t} = (1 - \rho) M_{i,t-1} + \rho I \]

\[ \sigma_{i,t} = (1 - \rho) \sigma_{i,t-1} + \rho (I_t - M_{i,t})^T (I_t - M_{i,t}) \]

\[ \rho = \alpha N(I_t; M_{i,t-1}, \Sigma_{i,t-1}) \]

2) For other components that are not matched to \( I_t \):

\[ w_{i,t} = (1 - \alpha) w_{i,t-1} \]

\[ M_{i,t} = M_{i,t-1} \]

\[ \sigma_{i,t} = \sigma_{i,t-1} \]

3) If \( I_t \) does not match any component in the current MOG, then the least likely component is replaced with a new one:

\[ M_{k,t} = I_t \]

\[ \sigma_{k,t} = \sigma_{max}^2 \quad \text{Large variance} \]

\[ w_{k,t} = \varepsilon \quad \text{Very small weight} \]
4) After the updates, the $w_{i,t}$ are renormalized.

$$\frac{w_{i,t}}{\sum_j w_{j,t}} \rightarrow w_{i,t}$$

B. Foreground Detection

1) All components of the MOG are sorted into the order of decreasing

$$\frac{w_{i,t}}{||z_{i,t}||} \quad ||z_{i,t}|| = \sqrt{\sum_j \alpha_{j,t}^2}$$

2) $B = \arg \min_b \left( \sum_{i=1}^{b} \frac{w_{i,t}}{\sum_{j=1}^{b} w_{j,t}} \rightarrow T \right)$

$T \in (0,1)$

e.g. $T = 0.6$

If it is matched to the $1,...,B$ component, Background

otherwise Foreground

3) Foreground pixels are segmented into regions using connected component labeling
C comments:
1) popular
2) Can be quickly adapted to different camera setup.
3) it is regarded as expensive.
4) don't take the spatial corelation of pixels into account.
   (think about the this problem. May be your final Project!)

W4 system

Minimum of value of certain pixel
Maxmum of value of certain pixel
Largest absolute difference of adjacent frames in background

\[ \text{If } |I_t - M| > D \text{ OR } |I_t - N| > D \]

The pixel is classified as Foreground

The parameters need to be updated.

"FAST SIMPLE, MAY NOT AS GOOD AS OTHER MORE COMPLICATED ALGORITHM."
Using LPC (Linear Prediction Coding) model/ Wiener filtering to do background subtraction.

\[ A_p \cdot B_k = \sum_{k=1}^{p} a_k B_{t-k} \]

\[ I_t = \sum_{k=1}^{p} b_k I_{t-k} B_k \]

The estimation error is:

\[ E[e^2] = E[B_k^2] - \sum_{k=1}^{p} a_k E[B_k \cdot B_{t-k}] \]

The threshold is:

\[ \zeta = 4 \sqrt{\frac{1}{N} E[e^2]} \]

B. The pixel is marked as background if:

\[ |I_t - \hat{I}_t| < \zeta \quad \text{and} \quad |I_t - B_k| < \zeta. \]

C. If more than 70% pixels in the image is classified as foreground, restart the model.
D. How to compute the parameters of the LPC model? i.e. : \( a_k's, b_k's \)

\[ e(n) = b(n) - \sum_{k=1}^{p} a_k \cdot b(n-k) \]

Denote:
- \( e_n(m) = e(n+m) \)
- \( b_n(m) = b(n+m) \)

\[ E_n = \sum_{m} e_n^2(m) \]
\[ = \sum_{m} \left( b_n(m) - \sum_{k=1}^{p} a_k \cdot b_n(m-k) \right)^2 \]

\[ \Rightarrow \sum_{m} \frac{2}{\partial a_k} \left( b_n(m) - \sum_{k=1}^{p} a_k \cdot b_n(m-k) \right) \cdot \left( b_n(m) - \sum_{k=1}^{p} a_k \cdot b_n(m-k) \right) = 0 \]

\[ \Rightarrow \sum_{m} \left[ - b_n(m) \cdot B_n(m-i) + \sum_{k=1}^{p} a_k \cdot B_n(m-i) \cdot B_n(m-k) \right] = 0 \]
\[
\sum_{m} B_n(m-i) B_n(m) = \sum_{m} \sum_{k=1}^{p} a_k \sum_{i} B_n(m-i) B_n(m-k)
\]

Denote: \( \phi_n(i, k) = \sum_{m} B_n(m-i) B_n(m-k) \)

\[
\Rightarrow \phi(i, 0) = \sum_{k=1}^{p} a_k \phi(i, k) \quad \text{for } i=1, \ldots, p
\]

\[
\begin{bmatrix}
\phi(1, 1) & \cdots & \phi(1, p) \\
\vdots & \ddots & \vdots \\
\phi(p, 1) & \cdots & \phi(p, p)
\end{bmatrix}
\begin{bmatrix}
a_1 \\
\vdots \\
a_p
\end{bmatrix}
= 
\begin{bmatrix}
\phi(1, 0) \\
\vdots \\
\phi(p, 0)
\end{bmatrix}
\]

\[
W \quad \hat{a} \quad P
\]

\[
\Rightarrow \hat{a} = W^{-1} P
\]

\[
E_n = \sum_{m} (B_n(m) - \sum_{k=1}^{p} a_k B_n(m-k))^2
\]

\[
= \sum_{m} B_n^2(m) - \sum_{k=1}^{p} \sum_{m} B_n(m) \cdot B_n(m-k)
\]

\[
= \phi_n(0, 0) - \sum_{k=1}^{p} a_k \phi(0, k)
\]

\[
2 = 4 \sqrt{n} \sqrt{E_n}
\]
E. Comments:
This is a complicated model to capture the background. The results are fairly good (refer to the original paper).
Chap 1. Background Modeling (Cont.)

6. Eigenbackgrounds


Background mean image, blob segmentation image, and input image with blob bounding boxes.

A. Principal components analysis (PCA):
is a technique used to reduce multidimensional data sets to lower dimensions for analysis. Depending on the field of application, it is also named the discrete Karhunen-Loève transform, the Hotelling transform or proper orthogonal decomposition (POD).
PCA has the distinction of being the optimal linear transformation for keeping the subspace that has largest variance. This advantage, however, comes at the price of greater computational requirement if compared, for example, to the discrete cosine transform.

Unlike other linear transforms, PCA does not have a fixed set of basis vectors. Its basis vectors depend on the data set.

Always use PCA (KL Transform) as the baseline for comparison

Graphical representation of a PCA transformation in only two dimensions. The variance of the tow dimension data in the original cartesian space (x, y) is best captured by the basis vectors v1, if the data are reduced to 1 dimension.
B. The procedure of Eigenbackground:

1) The nframes are re-arranged as the columns of a matrix \( \tilde{A} \)

\[
\begin{pmatrix}
19 \\
42 \\
37 \\
251 \\
250 \\
\vdots
\end{pmatrix}
\Rightarrow
\begin{pmatrix}
19 \\
42 \\
37 \\
251 \\
250 \\
\vdots
\end{pmatrix}
= I_1
\]

\( K : \# \text{ of Training Image} \)
\[
\tilde{A} = [I_1 \ I_2 \ \cdots \ I_K]
\]

(e.g. image is 200x300 
N = 60000)

2) Demean

\[
\bar{I} = \frac{1}{k} \sum_{i=1}^{N} I_i
\]

\[
\bar{A} = \bar{I} \cdot [1 \ 1 \ \cdots \ 1] = [\bar{I} \ \bar{I} \ \cdots \ \bar{I}]
\]

\[
\tilde{A} = \bar{A} - \bar{A}
\]
3) The covariance matrix, is computed:

\[ C = A A^T \]

so \( C \) is a \( N \) by \( N \) matrix for \( N = 60000 \), which is normal so \( C \) is a huge matrix.

4) From \( C \), the diagonal matrix of its eigenvalues, \( L \), and the eigenvector matrix, \( \Phi \), are computed

\[
\begin{bmatrix}
\phi_1 & \phi_2 & \cdots & \phi_N
\end{bmatrix}
= \text{eig}(C);
\]

\[
C \phi = \phi L = \begin{bmatrix}
\phi_1 & \phi_2 & \cdots & \phi_N
\end{bmatrix}
\begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \lambda_N
\end{bmatrix}
\]

Matlab command: \( \text{eig}(C) \)

5) Only the first \( M \) eigenvectors (eigenbackgrounds) are retained

\[
W^T = \begin{bmatrix}
\phi_1 & \phi_2 & \cdots & \phi_M
\end{bmatrix} \rightarrow M \times N
\]

\[
W = \begin{bmatrix}
\phi_1^T \\
\phi_2^T \\
\vdots \\
\phi_M^T
\end{bmatrix} \rightarrow M \times N \text{ matrix}
\]

6) Once a new image, \( I \), is available, it is first projected in the \( M \) eigenvectors sub-space and then reconstructed as \( I' \)

\[
W \text{ is a projection matrix from } W \text{ Dim space}

to \( M \) Dim space. \hat{I} \text{ original image vector}
\]

\[
I = \hat{I} - \bar{I} \quad \text{demean version of Image}
\]
C Comments:

1) The authors state that it works well and is faster than a Mixture of Gaussians approach

2) Hard to update the background.

\[
\text{projection: } I_p = W^T I
\]
\[
\text{reconstruction: } I' = W^T I_p
\]
\[
\text{residue: } I - I' = I - W^T I_p = I - W^T W I
\]

7) The difference \( I - I' \) is computed: since the sub-space well represents only the static parts of the scene, the outcome of this difference are the foreground objects.
3) The online update maybe achieve by incremental PCA.

4) The eigendecomposition can be computed very fast by SVD (Singular Value Decomposition)