MSc VIBOT & Computer Vision

Visual Tracking

Practice

Désiré Sidibé
dro-desire.sidibe@u-bourgogne.fr

Tracking a Moving Object I: Background Subtraction

1. Introduction

The goal of this *Visual Tracking* module is to learn about, and, more importantly, to learn how to use basic tracking algorithms and evaluate their performances.
We will start with a very simple and effective technique called *Background Subtraction* which can be used to initialize the tracker, i.e. to find the target's position in the first frame of a sequence, or to track the target through the entire sequence.

Background Subtraction

Background subtraction (BS) is widely used in surveillance and security applications, and serves as a first step in detecting objects or people in videos. BS is based on a model of the scene background, that is the static part of the scene. Each pixel is analyzed and a deviation from the model is uded to classify pixels as being background or foreground.
As an example, we will use the car sequence of figure 1. Say, we want to track the car in this sequence. We first need to detect the car's position in the first frame of the sequence, or provide that location manually. If we have a model $\mathbf{B}$ of the static part of the scene, then moving objects can be detected in an image $\mathbf{I}$, just by taking the difference $\mathbf{I} - \mathbf{B}$.
This is illustrated in in Fig 1, where the background model is shown in Fig 1b), and the detected moving object in Fig 1c).

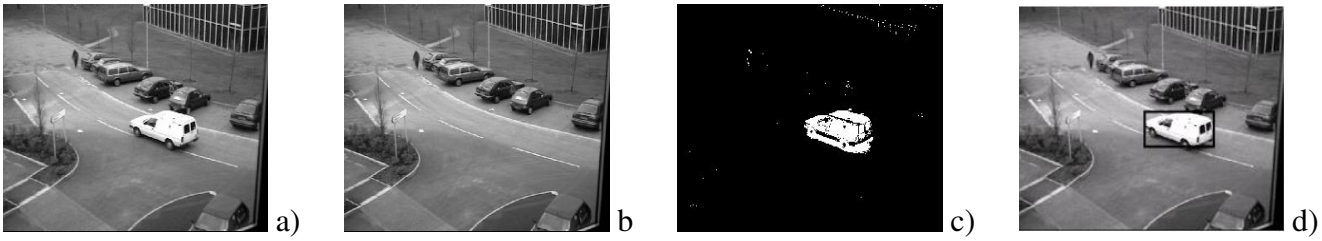

Figure 1: Background subtraction technique: (a) current frame; (b) estimated background; (c) detected moving object; (d) moving object with bounding box.

2. Background modeling

There are different methods to estimate a model of the background, that is the static part of the scene. These include frame differencing (FD), running Gaussian averaging (RGA), mixture of Gaussians (MoG), and eigenbackground (EB) among others.

2.1 Frame differencing

In this method, the background model at each pixel location is based on the pixel's recent history. The

history can be the average or the median of the previous n frames:

$$B_i(x,y) = \text{median}\{I_{i-n+1}(x,y), I_{i-n}(x,y), \ldots, I_{i-2}(x,y), I_{i-1}(x,y)\}.$$

A pixel belongs to the foreground if

$$|I_i(x,y) - B_i(x,y)| > T,$$

where T is a defined threshold.
The estimated background can be updated as follows:

$$B_i(x,y) = \begin{cases} \alpha I_i(x,y) + (1-\alpha)B_{i-1}(x,y) & \text{if } I_{i-1} \text{ is foreground,} \\ B_{i-1}(x,y) & \text{if } I_{i-1} \text{ is background,} \end{cases}$$

where α is the learning rate, usually a small value (0.05).

2.2 Running average Gaussian

In this method, introduced by Wren *et al.*[1], each pixel's recent history is modeled as a Gaussian probability density function (pdf). In order to avoid fitting the pdf from scratch at each new frame time t, a running (or online cumulative) average is computed.
The pdf of every pixel is characterized by a mean μ_t and a variance σ_t^2. To accomodate for changes in the background, illumination variations or non-static background objects, at every frame t, each pixel's mean and variance is updated as follows:

$$\mu_t(x,y) = \alpha I_t(x,y) + (1-\alpha)\mu_{t-1}(x,y)$$

$$\sigma^2(x,y) = d^2\alpha + (1-\alpha)\sigma_{t-1}^2(x,y),$$

where $d = |I_t(x,y) - \mu_t(x,y)|$, and α determines the size of the temporal window that is used to fit the pdf. Usually $\alpha = 0.01$.
A pixel is then classified as background if its current intensity lies within some confidence interval of its distribution's mean:

$$\begin{cases} I_t(x,y) \text{ is foreground} & \text{if } \frac{|I_t(x,y)-\mu_t(x,y)|}{\sigma_t} > T, \\ I_t(x,y) \text{ is background} & \text{if } \frac{|I_t(x,y)-\mu_t(x,y)|}{\sigma_t} \leq T, \end{cases}$$

where T is a free threshold parameter, usually $T = 2.5$. A larger value for T allows for more dynamic background, while a smaller T increases the probability of a transition from background to foreground due to more subtle changes.
This process can be initialized as $\mu_0 = I_0$ and $\sigma_0 = <$ some default value $>$.

2.3 Mixture of Gaussians

This model is an extension of the simple Gaussian model to account for muli-modal backgrounds, i.e. non-stionary backgrounds, proposed dy Stauffer and Grimson[2].
Let $\mathbf{x}_t = I_t(x,y)$ be the value of a pixel in frame t. Note that $\mathbf{x}_t$ is assumed to be a vector in $\mathcal{R}^3$.
The history of pixel at location (x,y) at any given time t, is therefore given by the ordered set $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$.
This history is modeled by a mixture of K Gaussian pdfs, so that the probability of observing the current pixel value is

$$P(\mathbf{x}_t) = \sum_{i=1}^{K} w_{i,t}\mathcal{N}(\mathbf{x}_t|\mu_{i,t}, \Sigma_{i,t}),$$

[1] C.R. Wren; A. Azarbayejani; T. Darrell; A.P. Pentland. "Pfinder: real-time tracking of the human body". IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7): 780-785, 1997.

[2] C. Stauffer, W. E. L. Grimson. "Adaptive background mixture models for real-time tracking". IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2. pp. 246-252, 1999.

where K is the number of distributions, $w_{i,t}$ is an estimate of the weight, $\mu_{i,t}$ is the mean, and $\Sigma_{i,t}$ is the covariance matrix of the ith Gaussian in the mixture at time t.
The Gaussian pdf is defined as

$$\mathcal{N}(\mathbf{x}_t|\mu_{i,t},\Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)},$$

with d the dimension of $\mathbf{x}$, here $d = 3$.
For computational reasons, the covariance matrix is assumed to be diagonal, i.e. $\Sigma_{k,t} = \sigma_{k,t}^2\mathbf{I}$, where $\mathbf{I}$ is the $d \times d$ identity matrix.
Every new pixel value, $\mathbf{x}_t$, is checked against the existing K Gaussian distributions until a match is found: there is a match if the pixel value is within 2.5 standard deviations of the mean of that distribution. If none of the K pdfs match the current pixel value, the least probable distribution is replaced that has a distribution with the current value as its mean. The distribution initially has high variance and low prior weight.
The weights of of the K distributions at time t are updated as follows:

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

where α is the learning rate, and $M_{k,t}$ is 1 for the model that matched and 0 for the rest. Note that the weights are normalized to sum to one.
The parameters of the unmatched distributions remain unchanged, but those of the distribution that matches the new observation are updated as follows:

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

$$\sigma_{k,t}^2 = \rho(\mathbf{x}_{k,t}-\mu_{k,t})^T(\mathbf{x}_{k,t}-\mu_{k,t}) + (1-\rho)\sigma_{t-1}^2,$$

and

$$\rho = \alpha\mathcal{N}(\mathbf{x}_{k,t}|\mu_{k,t-1},\Sigma_{k,t-1}).$$

2.4 Eigen Background

The eigen-background model describes the range of variations in intensity values that have been observed by building an eigenspace that models the background.
First, each frame I_t of the sequence is represented as a column vector $\mathbf{x}_t$ of dimension $d = w \times h$, where w and h are the size of the images. The model is formed by taking a sample of N images. The mean image is computed as:

$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i.$$

The mean-normalized image vectors are then put as column of a matrix X:

$$X = [\mathbf{x}_1 - \mathbf{m} \quad \mathbf{x}_2 - \mathbf{m} \quad \cdots \quad \mathbf{x}_N - \mathbf{m}].$$

The columns of X corresponding to frames in the video, all lie in a low-dimensional subspace of $\mathcal{R}^d$. This lower dimension space can be recovered by performing a singular value decomposition (SVD) of the matrix X:

$$X = U\Sigma V^T,$$

where U and V are orthogonal matrices and Σ is diagonal matrix with singular values, in decreasing order, in its diagonal.

We can approximates the subspace spanned by the columns of X, quite well, by considering as a basis only the first k columns of U, where $k \ll N$. This is also known as principal component analysis (PCA).
The matrix U_k obtained by keeping the first k columns of U is called the eigen-background. A new image $\mathbf{y}$ can be projected onto the reduced subspace as

$$\hat{\mathbf{y}} = U_k \mathbf{p} + \mathbf{m}.$$

Since U_k is orthogonal, $\mathbf{p}$ is easily obtained as $\mathbf{p} = U_k^T(\mathbf{y} - \mathbf{m})$.
Finally, by computing and thresholding the absolute difference between the input image and the projected image, we can detect moving objects in the scene:

$$|\hat{\mathbf{y}} - \mathbf{y}| > T,$$

where T is a given threshold.

WHAT TO DO?

Q1: Implement the frame differencing approach for background subtraction and apply it to the `car` sequence.
Note that our region of interest (ROI) is the white car. So, you might need to use some morphological operators, and find the center of the ROI, and the bounding box of the car as shown in Fig 1d).

Q2: Implement the running Gaussian average method.

Q3: Implement the eigen-background method.

Q4: Apply all your methods to the `highway` sequence.
This is a sequence of 1700 color images of a highway, thus showing several moving cars. The sequence comes with groundtruth segmentation of the moving objects in eac frame. You can use the 470 first frame to initialize your background model. Then detect moving objects from frames 470 to 1700.
For each method, you can compute the following quantities:

$$\texttt{precision} = \frac{True\ positive}{True\ positive\ +\ False\ positive}$$

$$\texttt{recall} = \frac{True\ positive}{True\ positive\ +\ False\ negative}$$

Finally, you also compute the F-score as

$$F = 2 * \frac{\texttt{precision} * \texttt{recall}}{\texttt{precision} + \texttt{recall}}$$

WHAT TO TURN IN?
A `zip` file with a report showing a commenting your results, and with a copy of your Matlab code.