Foreground Estimation and Hidden Markov Models for Tracking

Ardö, Håkan; Berthilsson, Rikard; Åström, Karl

2005

Foreground Estimation and Hidden Markov Models for Tracking

Håkan Ardö and Rikard Berthilsson and Kalle Åström

1st February 2005

Abstract

We will give a short introduction to foreground/background estimation and Hidden Markov for tracking. More information about the topics can be found in the papers listed at the end.

1 Foreground estimation

The objective is to extract the foreground and consequently also the background from a sequence of images. Problems facing us includes for example

- long execution time,
- slowly varying lighting conditions,
- rapidly varying lighting conditions, and
- what should be considered background.

The image sequence may come from a video camera with 352×288 resolution color images running 20 frames per second.

Let $I_t : \mathbb{R}^2 \to \mathbb{R}^3$, $t = 0, \ldots n-1$ be a sequence of n color images. We use the notation $I_t = (I_t^1, I_t^2, I_t^3)$ to denote the different color channels when needed.

In order to compute a feature at each location we can use convolution

$$f_t^j(x,y) = I_t^j * h(x,y) = \iint I_t^j(x-a, y-b)h(a,b)dadb,$$

where $h : \mathbb{R}^2 \to \mathbb{R}$ is the filter mask. This gives a filter response at every point $(x,y) \in \mathbb{R}^2$ and the statistical properties of these can be used to classify background and foreground.

1.1 Pixel based foreground estimation

The Stauffer–Grimson [15] estimator is obtained by letting $h = \delta_{0,0}$ be the Dirac measure at the origin in which case $I_t^j * \delta_{0,0} = I_t^j$, i.e. that is the estimator is based on the individual pixel data. A simple solution would be to define a probability function at each point $(x,y) \in \mathbb{R}$. Note that for digital images there are only a finite set of points in the definition set giving a finite set of probability functions. Thus, for a gray level image we will need to define probability functions $p_{x,y}(a)$ like for example

$$p_{x,y}(a) = \frac{1}{\sqrt{2\pi\sigma_{x,y}}} e^{(a-m_{x,y})^2/(2\sigma_{x,y}^2)} \tag{1}$$

which gives a normal probability function at each $(x, y) \in \mathbb{R}^2$. The parameters $m_{x,y}$ and $\sigma_{x,y}$ can be estimated from a sequence of real images as

$$m_{x,y} = \frac{1}{N} \sum_{t=1}^{N} I_t(x, y)$$

and similar for the standard deviation $\sigma_{x,y}$. When these parameters are computed a new image I_t is transformed into a binary image according to

$$\tilde{I}_t(x, y) = \begin{cases} 1 & p_{x,y}(I_t(x, y)) < \epsilon \\ 0 & \text{otherwise} \end{cases}$$

where $\epsilon > 0$ is a constant. Points having $\tilde{I}_t(x, y) = 1$ is considered foreground and $\tilde{I}_t(x, y) = 0$ is considered background. A more advanced probability function can be obtained by setting

$$p_{x,y}(a) = \sum_{k=1}^{n} c_{x,y,k} e^{(a-m_{x,y,k})^2/(2\sigma^2_{x,y,k})}$$

that is a linear combination of normal probability functions. The constant $c_{x,y,k} \geq 0$ is computed such that $\int p_{x,y}(a)da = 1$. It is however not fully trivial how the parameters should be estimated. The interested reader can have a look at for example [15] for more details.

In order to cope with a background that is not totally static we have to let the parameters of the probability function be allowed to change over time. To update the probability function (1) we could for example assume that the standard deviation is constant and update the mean according to

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

where the constant α gives how fast old values should be forgotten.

1.2 Feature based foreground estimation

It is a well know problem that many background estimators are sensitive to rapid changes in lighting. Such rapid changes are often present and can occur for example when a cloud suddenly occludes the sun, moving objects casting shadows or fast changes in indoor lighting.

In order to deal with this problem we would like to use features that are independent to changes that behaves at least locally in a nice manner. For this reasons to model the background changes due to varying lighting we assume that there exists some constant c such that

$$I^j_t(x + \Delta x, y + \Delta y) = cI^j_{t+1}(x + \Delta x, y + \Delta y), \quad (\Delta x, \Delta y) \in \Omega_{(x,y)}, \tag{2}$$

where $\Omega_{(x,y)}$ defines a neighborhood of (x, y). An image sequence that fulfills condition (2) is called **locally proportional**. Local proportionality is usually fulfilled, at least approximately, for most points (x, y). To take advantage of this assumption we introduce two filters $h_1 : \mathbb{R}^2 \to \mathbb{R}$ and $h_2 : \mathbb{R}^2 \to \mathbb{R}$. Furthermore, we usually choose these filters such that $\operatorname{supp} h_1 \subseteq \operatorname{supp} h_2$, i.e. $h_1 \neq 0$ only on the set where $h_2 \neq 0$. We can now use

$$f^j_t(x, y) = \frac{I^j_t * h_1(x, y)}{I^j_t * h_2(x, y)}, \qquad j = 1, 2, 3 \tag{3}$$

as features for each $(x, y) \in \mathbb{R}^2$. It follows from the locally proportional assumption that I_t^j and cI_t^j, $j = 1, 2, 3$ gives the same feature values, i.e. they are independent of rapid changes in lighting as long as $\operatorname{supp} h_1 \subseteq \operatorname{supp} h_2 \subseteq \Omega_{(x,y)}$.

The probability functions $p_{x,y}$ are now probability functions for the values of the features $f_t^j(x, y)$ and can be approximated as normal probability function as above and updated in a similar fashion. It is also possible to work directly with computed histograms that are updated with each new frame. This usually leads to high accuracy and also high computational speed. The binary foreground and background image is computed as above using some $\epsilon > 0$.

1.3 Implementation

The convolution for computing (3) can of course be done using FFT with a computational cost of $\mathcal{O}(MN \log(MN))$ for $M \times N$ images. However, if we let h_k, $k = 1, 2$, be simple functions like for example Haar wavelets or scale functions then we can use the well known integral image to speed up the computation. Let

$$\tilde{I}_t^j(x, y) = \int_{-\infty}^{x} \int_{-\infty}^{y} I_t^j(a, b) da db, \qquad j = 1, 2, 3,$$

be the integral image with a computational cost of about $4MN$ additions for $M \times N$ images. Furthermore, for simplicity we let

$$h_k(x, y) = \begin{cases} 1, & |x| \leq c_k, |y| \leq c_k \\ 0, & \text{otherwise} \end{cases}, \qquad k = 1, 2,$$

be the two filter functions. In order to fulfill $\operatorname{supp} h_1 \subseteq \operatorname{supp} h_2$ we choose $0 < c_1 < c_2$. It follows that

$$\begin{aligned} I_t^j * h_k(x, y) = \tilde{I}_t^j(x - c_k, y - c_k) + \tilde{I}_t^j(x + c_k, y + c_k) \\ - \tilde{I}_t^j(x - c_k, y + c_k) - \tilde{I}_t^j(x + c_k, y - c_k), \qquad k = 1, 2, \end{aligned}$$

requiring only four additions for each (x, y).

2 Hidden Markov models

A hidden Markov model is defined as a discrete time stochastic process with a finite number $(N + 1)$ of states, $S = S_0, ..., S_N$ and a constant transitional probability distribution $a_{i,j} = p(q_{t+1} = S_j | q_t = S_i)$, where $Q = (q_0, ..., q_T)$ is a state sequence for the time $t = 0, 1, ..., T$ [14]. The initial state distribution is denoted $\pi = (\pi_0, ..., \pi_N)$, where $\pi_i = p(q_0 = S_i)$. The state of the process cannot be directly observed, instead some sequence of observation symbols, $O = (O_0, ..., O_T)$ are measured, and the observation probability distribution, $b_j(O_t) = b_{j,t} = p(O_t | q_t = S_j)$, depends on the current state. We make the following assumption about the conditional probabilities

$$p(q_{t+1} \mid q_t, q_{t-1}, \ldots q_0, O_t, \ldots, O_0) = p(q_{t+1} \mid q_t),$$

$$p(O_t \mid q_t, q_{t-1}, \ldots q_0, O_{t-1}, \ldots, O_0) = p(O_t \mid q_t).$$

2.1 Viterbi Optimization

From a hidden Markov model $\lambda = (a_{i,j}, b_j, \pi)$ and and an observation sequence, O, the most likely state sequence, $Q^* = \text{argmax}_Q\, p(Q|\lambda, O) = \text{argmax}_Q\, p(Q, O|\lambda)$, to produce O can be determined using Viterbi optimization [14] by defining

$$\delta_t(i) = \max_{q_0 \cdots q_{t-1}} p(q_0, ..., q_{t-1}, q_t = S_i, O_1, ..., O_t). \tag{4}$$

Note that the assumption about the conditional probabilities implies that

$$p(q_0, ..., q_{t-1}, q_t, O_1, ..., O_t) = p(O_t \mid q_t)p(q_t \mid q_{t-1})p(q_0, ..., q_{t-1}, O_1, ..., O_{t-1}).$$

Thus, for $t = 0$, $\delta_0(i)$ becomes $p(q_1 = S_i, O_1)$, which can be calculated as $\delta_0(i) = \pi_i b_{i,0}$, and for $t > 0$ it follows that $\delta_t(i) = \max_j(\delta_{t-1}(j)a_{j,i}) \cdot b_{i,t}$. By also keeping track of $\psi_t(i) = \text{argmax}_j(\delta_{t-1}(j)a_{j,i})$ the optimal state sequence can be found by backtracking from $q_T^* = \text{argmax}_i\, \delta_T(i)$, and letting $q_t^* = \psi_{t+1}(q_{t+1}^*)$.

To use HMMs for problems like tracking, where the state sequences are very long, e.g. $T \to \infty$, and results have to be produced before the entire observation sequence have been measured, some modifications to the Viterbi optimization algorithm described above are needed. We do not go into this problem in this presentation.

2.2 Single object tracking

A HMM such as described above can be used for tracking objects in a video sequence produced by a stationary camera. We begin by assuming that the world only contains one mobile object and that this object sometimes is visible in the video sequence and sometimes located outside the part of the world viewed by the camera. The assumption of only one car will later be removed.

The state space of the HMM is a finite set of points $S_i \in \mathbb{R}^2$, $j = 1, \ldots, N$ representing that the mass center of the object is at position S_i in the camera coordinate system. The points $\{S_i\}$ are typically spread in a homogeneous grid over the image. A special state S_0, representing the state when the object is not visible, is also needed. The $a_{i,j}$ constants, representing probabilities of the object appearing, disappearing or moving from one position to another, can be measured from training data.

The observation symbols of this model will be a binary background/foreground image, $O_t : \mathbb{R}^2 \to \{0, 1\}$, as produced by for example [15]. By analyzing the background/foreground segmentation algorithm, the probabilities

$$p_{fg} = p(\text{x is a foreground pixel}|O_t(x) = 1), \tag{5}$$

and

$$p_{bg} = p(\text{x is a background pixel}|O_t(x) = 0) \tag{6}$$

can be calculated. Typically these are well above $1/2$. Furthermore, for the segmentation described in [15] the inequality $p_{bg} > p_{fg}$ is usually fulfilled, i.e. it is more likely for a foreground pixel to be erroneously detected as background than it is for a background pixel to be erroneously detected as foreground.

The shape of the object when located in state S_i, can be defined as the set of pixels, C_{S_i}, that the object covers when centered in at this position. To track circular objects of some radius r, let $C_{S_i} = \{x \in \mathbb{R}^2; |x - S_i| < r\}$. As there is only one object in the world, when the HMM is in state S_i, the pixels in C_{S_i} are foreground pixels and all other pixels are background pixels. The probability, $b_{i,t} =$

$p(O_t|q_t = S_i)$, of this is

$$b_{i,t} = \prod_{x \in C_{S_i}} [O_t(x)p_{fg} + (1 - O_t(x))(1 - p_{bg})] \cdot \prod_{x \notin C_{S_i}} [(1 - O_t(x))p_{bg} + (O_t(x))(1 - p_{fg})], \quad (7)$$

and thereby all parts of the HMM is defined.

2.3 Multi object HMMs

To generalize the one object model in the previous section into two or several objects is straight forward. For the two object case the states becomes $S_{i,j} \in S^2 = S \times S$ and the shapes, $C_{S_{i,j}} = C_{S_i} \cup C_{S_j}$. The transitional probabilities becomes $a_{i_1 j_1 i_2 j_2} = a_{i_1 i_2} \cdot a_{j_1 j_2}$ and an additional term can be be added to the observation probability making $p(O_t|q_t = S_{i,j}) \approx 0$ if $C_{S_i} \cap C_{S_j} \neq \emptyset$.

There is one problem with this though. The number of states increases exponentially with the number of objects and in practice this approach is only plausible for a small number och object within a small region of space.

To track objects traveling over the entire image several small HMMs could be used, each not much bigger than the size of the objects being tracked. Within these small models a grid of state points as small as 5×5 can be used and only one or two object has to be considered. Each model would be evaluated separately and then the results are combined.

2.4 Footfall

By mounting a camera above an entrance looking straight down, the number of people entering and leaving the building can be counted using the Markov models described above. Figure 2.4 shows the camera view from such a setup. A number of 5×5 models were placed along one fixed row in the images. The the overlapping tracks produced by each model were combined. Also, all tracks laying entirely on the border of its model were removed as these could represent a maxima outside the current model and would thus show up in another model.

The test sequence consist of 14 minutes video where 249 persons passes the line. 7 are missed and 2 are counted twice. This gives an error rate of 3.6%.

References

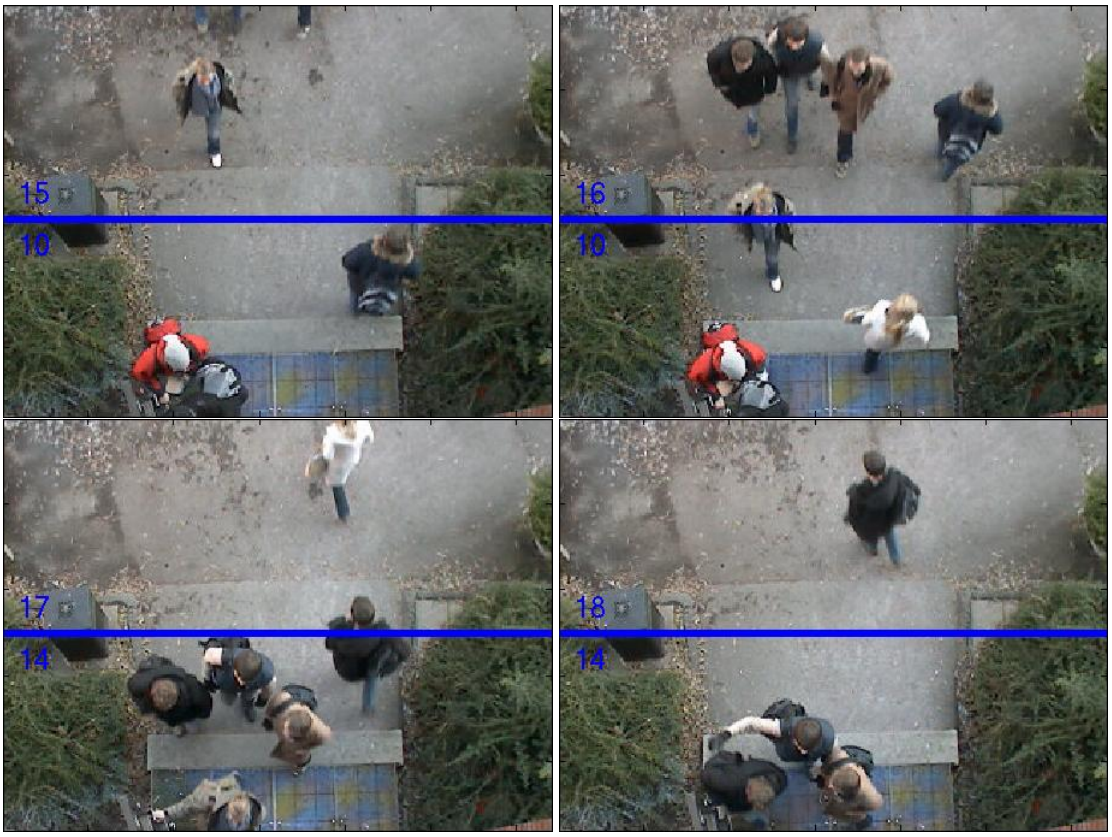
[1] M. Bonnisegna and A. Bozzoli. A tunable algorithm to update a reference image. *Signal Processing: Image Communication*, 1998.

[2] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *6th European Conference on Computer Vision*, 2000.

[3] D. Farin et al. Robust background estimation for complex video sequences. In *International Conference on Image Processing*, pages 145–148, 2003.

[4] D.S. Gao, J. Zhou, and L.P. Xin. A novel algorithm of adaptive background estimation. In *International Conference on Image Processing*, 2001.

[5] G. Gordon, T. Darrell, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.



People entering and exiting entrance. Number above line gives number of people crossing the line going out. Number below line gives number of people crossing the line going in.

[6] E. Hayman and J-O. Eklundh. Background subtraction for a mobile observer. In *Proc. 9th Int. Conf. on Computer Vision, Nice, France*, 2003.

[7] J. Heikkila and O. Silven. A real-time system for monitoring of cyclists and pedestrians. In *Second IEEE Workshop on Visual Surveillance*, pages 74–81, 1999.

[8] C. Hydén. *The development of a method for traffic safety evaluation: The Swedish traffic conflicts technique*. PhD thesis, Institutionen för trafikteknik, LTH, Lund, 1987.

[9] P. Kumar, K. Sengupta, and A. Lee. A comparative study of different color spaces for foreground and shadow detection for traffic monitoring system. In *The IEEE 5th International Conference on Intelligent Transportation Systems*, pages 100– 105, 2002.

[10] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 873 – 889, 2001.

[11] J. Orwell, P. Remagnino, and G.A. Jones. Multi-camera colour tracking. In *Second IEEE Workshop on Visual Surveillance*, pages 14–21, 1999.

[12] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of dynamic backgrounds. In *Conference on Computer Vision and Pattern Recognition*, page 73, 2003.

[13] A. Prati, I. Mikic, C. Grana, and M. M. Trivedi. Shadow detection algorithms for traffic flow analysis: a comparative study. In *IEEE Intelligent Transportation Systems*, pages 340 – 345, 2001.

[14] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, 1989.

[15] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747 – 757, 2004.

[16] R Wren and et al. Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 780–785, 1997.