Probabilistic Classification Between Foreground Objects and Background

Paul Withagen[1,2], Klamer Schutte[1]
[1]TNO Physics and Electronics Laboratory
P.O. Box 96864
2509 JG The Hague, The Netherlands
{withagen,schutte}@fel.tno.nl

Frans Groen[2]
[2] IAS group, University of Amsterdam
Kruislaan 403
1098 SJ Amsterdam, The Netherlands
{paulw,groen}@science.uva.nl

Abstract

Tracking of deformable objects like humans is a basic operation in many surveillance applications. Objects are detected as they enter the field of view of the camera and they are then tracked during the time they are visible. A problem with tracking deformable objects is that the shape of the object should be re-estimated for each frame.

We propose a probabilistic framework combining object detection, tracking and shape deformation. We make use of the probabilities that a pixel belongs to the background, a new object or any of the known objects. Instead of using arbitrary thresholds for deciding to which class the pixel should be assigned we assign the pixel based on the Bayes criterion.

Preliminary experiments show the classification error drops to about half the error of traditional approaches.

1. Introduction and previous work

A surveillance application (see for example [2, 4, 6, 7]) usually consists of moving object detection, object tracking, and higher order processing like the detection of persons entering a prohibited area, face recognition and/or gesture recognition to determine what a person is doing. This paper concentrates on the initial steps, moving object detection and tracking. Its main subject is the introduction of a new method for the classification of pixels between foreground and background.

Often, background modeling is used to detect moving objects. Toyama compares such algorithms in [6]. A popular algorithm is Expectation Maximization [1]. It is feasible to realize a real-time implementation using the online version [3] of this algorithm. The EM algorithm models the background by maintaining for each pixel a model of the appearance of colors over time. This model consists of a mixture of Gaussian kernels. For each frame, the probability that a pixel is background is calculated by comparing its color to this mixture model.

The advantage of such an adaptive method is that it learns the background from the images. This eliminates the need of initialization with an empty background scene. It also provides a background model that is always up to date. An important disadvantage is the trade-off between two conflicting demands. On the one hand updating should be performed fast to deal with changes in illumination and changes in the background (Time of Day, Light Switch, Walking Person and Moved Objects problems in [6]). On the other hand, updating should be performed slow to avoid classifying slowly moving objects as background (Bootstrapping and Sleeping Person problems in [6]).

In earlier work [**?**] we proposed to solve the problem of having to choose between fast and slow updating. That paper proposes to solve this by only updating the background model for those pixels that depict background. This prohibits slowly moving objects from being learned into the background model, regardless of the update speed. Changes in the background (Moved Object, Bootstrapping, Walking Person) can now be dealt with at the object level instead of the pixel level. See [**?**] for a full description and evaluation of this update model.

Once a reliable model of the background is available, classification between foreground and background can be performed. A frequently used approach is that of Stauffer and Grimson [5] where the Gaussian kernels in the mixture model are classified as depicting either foreground or background. Pixels for which their color value is described well enough (i.e. within a certain threshold) by any of the kernels depicting background classified as background, all other pixels as foreground.

A disadvantage of the Stauffer and Grimson classification approach is that it tries to solve a two-class classification problem (i.e. foreground, background) with a model of only one of the classes (the model of the background).

In this paper we propose to do classification using optimal (lowest cost) classification between: background, any of the known foreground objects and new foreground object. This is performed by calculating the probabilities that a pixel belongs to each of the classes mentioned above and using these to classify the pixel. This approach is similar to the background vs foreground segmentation described in [4]. Important improvements of our approach are that we do not need to do the classification in two steps and we incorporate the probability that an object is occluded directly in the calculated probabilities for each of the objects.

This paper is structured as follows. In section 2 we will present our approach for pixel classification between foreground and background, based on lowest cost classification. An experimental evaluation of the proposed algorithm will be given in section 3. Finally, conclusions will be drawn in section 4.

2. Probability based pixel classification

The main idea of our approach is that the pixel classification between objects (foreground) and background should be based on the statistical comparison between all relevant classes. Therefore the first thing to determine is which classes are relevant for a certain pixel.

We observe that the objects being tracked (humans in our case) are not rigid but deformable. In order to allow for object deformation we split the object in two parts: its core and its shell. The shell is the outer boundary of the object with a certain width. The core is the remainder of the object, located in the center. It is created using an erosion of size d_{erode} on the object. This allows us to incorporate the following assumptions about the objects:

- We assume that the color of an object (both core and shell) can be described by the color of its core.
- We assume that the deformation between subsequent frames is bounded by an upper value. Existing objects therefore have a limited ”reach”, outside which the probability for these objects is zero.
- For a known object at least the core of the object will be present in the subsequent frame.
- Objects are more likely to occur and disappear near the boundary of the image then in the center of the image.
- Objects have a minimum size (i.e. a minimum number of pixels).

Locating the core objects will be described in subsection 2.1. Once the location of the core objects is known, probabilities for each of the objects are calculated. As described in subsection 2.2.

2.1. Histogram based tracking of core objects

Given our assumptions about the objects, only the shell of an object will change due to deformation and occlusion. Therefore we can use template matching to locate the core-object. Histogram-based template matching is used because it is computationally efficient and it results in an accurate object location, even if the assumption that the core object does not change is only partially true.

A color histogram of the core object is maintained as object model. In the new frame in a certain region around the old position of the object, histograms are created of groups of pixels with the same shape as the core object The location where the absolute difference between the histogram in the new frame and the histogram of the core object is minimal is the new position of the core object. This histogram based template matching can be implemented efficiently by remembering the difference histogram at the first position: $D_1 = H_1 - C$, with D_1 the difference histogram at the first position, H_1 the histogram at the first position and C the histogram of the core object. The absolute error for this position is $e_1 = \sum_{\text{all bins}} |D_1|$. For all subsequent positions, only those pixels that are added or removed by shifting the position need to be considered. The difference histogram is updated: $D_{n+1} = D_n - H_{\text{removed}} + H_{\text{added}}$ with H_{removed} and H_{added} the histograms of the removed and added pixels. The error at this position is calculated by $e_{n+1} = e_n + \sum_{\text{added}} \text{sign}(D_n) - \sum_{\text{removed}} \text{sign}(D_n)$.

After pixel classification (see subsection 2.2) a new core object is constructed by eroding the full object. Then the histogram of the core object is updated using $C_{t+1} = (1 - \gamma_F)C_t + (\gamma_F)N$. with N the histogram of the new core object and γ_F a parameter which regulates the update speed. All histograms we use are normalized by letting the sum over all bins equal one.

2.2. Classification of pixels in the object shell

Now that we have the location of all core objects, we know which objects are relevant for each pixel. We use the Bayes decision rule to do lowest cost pixel classification:

$$\min_i \big(\sum_{j=1}^{N} C(\omega_i, \omega_j) P(\omega_j|\vec{x}) \big) \tag{1}$$

with C the cost function for wrong classification, ω_i class i and N the number of classes.

To classify a pixel we need the probability for each of the N classes: B: background, F_i: known foreground object i and F_{new}: new foreground object. The posterior probability for class ω_i, given the pixel color $\vec{x}$ is given by

$$P(\omega_i|\vec{x}) = \frac{P(\vec{x}|\omega_i)P(\omega_i)}{P(\vec{x})} \tag{2}$$

with $P(\vec{x}) = \sum_i P(\vec{x}|\omega_i)$. The posterior probabilities sum to one: $\sum_i P(\omega_i|\vec{x}) = 1$ and the same is true for the prior probabilities: $\sum_i P(\omega_i) = 1$. The posterior probability is given by either the model of the background or the model of the foreground objects (the core histogram in our case). When no other objects are relevant, the prior probability $P(\omega)$ depends only on the distance to the core object. In subsection 2.4 we will give the function we used in our experiments.

2.3. Occlusion modelling

Consider a scene with multiple objects. We know the objects cannot be inside each other, therefore if multiple objects overlap in view, one must be in front the others. We have to solve the occlusion problem. Assume that the ordering of foreground objects Ω is known. The first element in this ordering $\Omega(1)$ denotes the relevant object closest to the camera and the last element denotes the object with the largest distance to the camera.

The probability that we observe object $F_{\Omega(m)}$ in a certain pixel is given by the probability that we do not observe any object in front of this object i.e. objects $F_{\Omega(1)}, \cdots, F_{\Omega(m-1)}$, multiplied by the probability that we observe this object, given that no other objects are relevant:

$$P(F_{\Omega(m)}|\vec{s}, \Omega) = (1 - \sum_{n=1}^{m-1} P(F_{\Omega(n)}|\vec{s}, \Omega)) \\ P(F_{\Omega(m)}|R_{\Omega(m)}, \vec{s}) \quad (3)$$

with $\vec{s}$ the position of the pixel and R_i meaning that only object i is relevant at this position.

Equation 3 assumed a known ordering of the objects. For the general case we should sum over all possible orderings:

$$P(F_i|\vec{s}) = \sum_{k=1}^{N_{\text{orderings}}} P(\Omega_k) P(F_i|\vec{s}, \Omega_k) \quad (4)$$

The probability that the pixel at location $\vec{s}$ depicts background or a new object is given by

$$P(B|\vec{s}) + P(F_{new}|\vec{s}) = 1 - \sum_{i=1}^{N_R} P(F_i|\vec{s}) \ , \quad (5)$$

with N_R the number of relevant objects. The ratio between the probability for new foreground object and background depends on the position in the image.

2.4. Computational complexity

All objects are assumed to have a limited speed of movement and deformation, so each object is relevant only in a part of the image. For each pixel location $\vec{s}$ we know which known objects are relevant.

The prior probability that a relevant object is observed in pixel $\vec{s}$, given that there are no other objects relevant is $P(F_i|R_i)$ were R_i denotes that only object i is relevant for pixel location $\vec{s}$. This probability is not necessary equal to one. Depending on the distance $d(F_i, \vec{s})$ to the core of the object, this probability has a value given by

$$P(F_i|R_i, \vec{s}) = \begin{cases} 0 & \text{for } d > d_{max} \\ 0 < p(d) < 1 & \text{for } 0 < d < d_{max} \\ 1 & \text{for } d = 0 \ , \end{cases} \quad (6)$$

with d_{max} a parameter specifying the area in which the object is relevant. This parameter depends on the speed of the object and the amount it can deform between two frames. $p(d)$ is a function with homogeneous decay and existing derivative. In our experiments we used $p(d) = 1 - \frac{d}{d_{max}}$.

As more objects become relevant, equations 4 becomes more expensive to compute. Its complexity increases with $N_R!$. For real-time implementations this is prohibitive.

Fortunately there exist two situations where the complexity is lower. The first situation occurs when the ordering is known. Then equation 4 only has to be computed for one ordering.

The other situation for which equation 4 simplifies occurs when no information is used about the ordering of objects. For this situation all orderings are equally probable. Equation 5 then simplifies to:

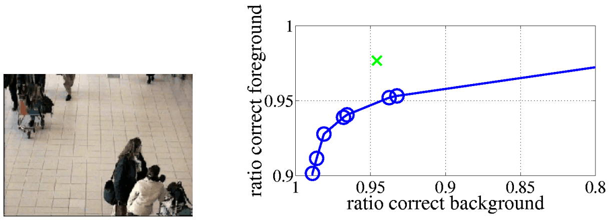
$$P(B|\vec{s}) + P(F_{new}|\vec{s}) = \prod_{i=1}^{N_R} (1 - P(F_i|R_i)) \quad (7)$$

and using $p_i = P(F_i|R_i)$, for one relevant object equation 4 simplifies to $P(F_1|\vec{s}) = p_1$, for two objects to $P(F_1|\vec{s}) = p_1(1 - \frac{1}{2}p_2)$ and for three objects to $P(F_1|\vec{s}) = p_1(1 - \frac{1}{2}(p_2 + p_3) + \frac{1}{3}(p_2 p_3))$. Simplified equations for more objects exist, but we choose to discard higher-order terms as they quickly become negligible.

2.5. Object detection and removal

New objects are detected by considering blobs of connected pixels which are all labelled as depicting a possible new foreground object. Before a blob is accepted it must be sufficiently probable that it is indeed an object. Therefore two demands need to be satisfied. First the number of pixels in the blob needs to exceed threshold N_{detect}.

Second, the color of the blob needs to be sufficiently different from the background. This prevents shadow areas from being detected as object. We define q as the quotient of the values of all pixels belonging to the blob of the background image I_{BG} and the current image I_t: $q = \frac{I_t}{I_{BG}}$. New objects are only allowed when their average of q over all pixels: $\langle q \rangle$ significantly differs from unity, i.e. $|\langle q \rangle - 1| > S_1$, and there is also sufficient variation of q over the object: $\langle |q - \langle q \rangle| \rangle > S_2$.



(a) Frame 500 (b) Results

Figure 1. (a) Frame 500 of the used sequence, (b) ROC of the results (circles denote the reference method, x the proposed method)

An objects is removed when its number of object pixels does not exceed threshold N_{keep} or when the histogram matching error is larger than e_{max}. Also, objects are removed when they exist for more than T_{max} frames.

3. Experiments

We evaluate the performance of the proposed algorithm by comparing its foreground/background classification result to that a reference method. For this reference method we use a Gaussian mixture model which is updated every frame for each pixel, regardless of whether the pixel depicts foreground or background. Classification is done as described in [5].

To prevent differences in update equations having an effect on the results, the models of both the reference algorithm and the proposed algorithm are updated using standard online Expectation Maximization [3]. Therefore, differences in the results will be caused only by the different classification algorithm.

For our experiments we used a color image sequence of 1800 frame. The data contains changes in illumination, moving people and people standing still for some time. Of this dataset six frames were manually labelled between foreground, background and don't care. See figure 1(a) for a frame from the sequence. The percentage of correctly labelled foreground and background for different parameter settings for both methods are shown as a ROC in figure 1(b).

The ROC for the reference method is the convex hull of over 1700 different combinations of parameters, among which are update speed and threshold.

For the proposed method we used the following parameter setting:

- Object detection: $S_1 = 0.6$, $S_2 = 0.2$, $N_{\text{detect}} = 100$.
- Object removal: $T_{\text{max}} = 250$, $e_{\text{max}} = 0.7$, $N_{\text{keep}} = 40$
- Update speed objects: $\gamma_F = 0.1$, background: $\gamma_B = 0.1$;
- Size of shell: $d_{\text{erode}} = 4$, $d_{max} = 10$.
- Object histogram size: $8 \times 8 \times 8$ bins.

To evaluate the proposed method we used all pixels in detected objects and those pixels labelled as possible new object as foreground and all other pixels as background.

Figure 1(b) shows the ROC of the results. For a typical cost of misclassification: $C(B, F) = 2C(F, B)$ the error of the proposed method is 3% and the error of the reference method is 5%, so a significant reduction of almost a factor 2. The parameters of the proposed method were not fully optimized, so even better performance can be expected.

The proposed method was evaluated as method for classification between foreground and background, but it also tracks deformable objects in image sequences. Evaluation of this part of the algorithm remains as future work.

4. Conclusions

We proposed a optimal 2-class classification method to classify between foreground and background. Preliminary experiments show that classification improves a factor two using the proposed method.

An additional advantage of the proposed classification method is that many of the problems presented in [6] now are solved at the object level instead of the pixel level. This is an advantage because it is easier to implement knowledge about the behavior of objects at the object level.

References

[1] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM-algorithm. *Journal of the Royal Statistical Society*, B 39(1):1–38, 1977.

[2] D. H. I. Haritaoglu and L. Davis. W^4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analisys and Machine Intelligence*, 22(8):809–830, August 2000.

[3] C. Priebe. Adaptive mixtures. *Journal of the American Statistical Association*, 89(427):796–806, 1994.

[4] A. Senior. Tracking people with probabilistic appearance models. In *Proceedings of the IEEE International Workshops on Performance Evaluation of Tracking and Surveillance (PETS)*, pages 48–55, 2002.

[5] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757, Aug. 2000.

[6] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Seventh International Conference on Computer Vision, September 1999, Kerkyra, Greece*, pages 255–261. IEEE Computer Society Press, 1999.

[7] P. J. Withagen, K. Schutte, and F. C. Groen. Likelihood-based object tracking using color histograms and EM. In A. Tekalp, editor, *Proceedings of the IEEE International Conference on Image Processing (ICIP 2002)*, pages 589–592, Rochester, NY, Sept. 2002. IEEE.