Computer Vision Winter Workshop 2006, Ondřej Chum, Vojtěch Franc (eds.)
Telč, Czech Republic, February 6–8
Czech Pattern Recognition Society

Fast pedestrian tracking based on spatial features and colour

Florian H. Seitner and Allan Hanbury

Pattern Recognition and Image Processing Group
Vienna University of Technology, Vienna, Austria
{seitnerf,hanbury}@prip.tuwien.ac.at

Abstract *A tracking with appearance modelling system for pedestrians is described. For pedestrian detection a cascade of boosted classifiers and Haar-like rectangular features are used. Statistical modelling in the HSV colour space is used for adaptive background modelling and subtraction, where the use of circular statistics for hue is proposed. A clipping algorithm based on this background model and extensions to the traditional boosted classifier for fast classification are introduced. By using the background model in combination with the detector, the system extracts a feature vector based on colour statistics and spatial information. Circular and linear statistics are applied on the extracted features to robustly track the pedestrians and other moving objects. An adaptive appearance model copes with partial or full occlusions and addresses the problem of missing or wrong detections in single frames.*

Keywords*: tracking, background segmentation, appearance model, HSV, circular statistic, pedestrian detection.*

1. Introduction

The proposed tracking system uses a background segmentation algorithm in combination with an object classifier to quickly find pedestrians in each video frame. After a possible pedestrian is detected, the moving object is subdivided into three zones (head, upper body, and lower body) and the colour and spatial properties of each part which form the basic appearance model in this system are extracted. The colour information is analyzed in the HSV (Hue, Saturation and Value) colour space. This colour model describes each colour by one angular (Hue) and two linear values (Saturation and Value). Although HSV has been applied to a wide range of applications like motion analysis, background modelling, and image retrieval, often its mixed topological nature of linear and circular domains is not appropriately taken into account. For example, it is clear that the mean of angles 359° and 1° is not 180° like the arithmetic mean would yield — it should be 0°. Therefore, important definitions of circular statistics are given in Section 2 and used in this work to accurately process directional hue data.

Section 3 describes how color distributions can be approximated by parametric descriptions and how the adaptive background model distinguishes between foreground and background. The detector (Section 4) uses a cascade of boosted classifiers and Haar-like features to describe pedestrians in a highly efficient way. Additionally, a method for fast pre-classification and a clipping algorithm based on the background model for reducing the number of classification windows are introduced in this section.

Section 5 describes how tracking features can be extracted by using the obtained pedestrian detections and the foreground data. The structure of the adaptive appearance model is described in Section 6 and results and conclusions of this work are given in Sections 7 and 8.

2. Circular statistics

The algebraic structure of the line and the circle are different and therefore methods of circular data analysis as discussed in [6] must be used when working with directional data. In contrast to the linear domain only one operation, the addition modulo 2π is available in the circular domain. Due to the fact that the circle is a closed curve, its natural periodicity must be taken into account. Accurate distributions for working with directional data are introduced in Section 3.

The arithmetic mean is unsuitable for directional data since the result is very dependent on the choice of origin in the circular domain. The circular mean for directional data can be computed as a vector sum $\hat{\mu} = \sum_{i=1}^{n} R_i$ of n unit vectors $R_{1..n}$ where each unit vector represents a sample direction. The circular variance $\hat{V}$ is then defined as $\hat{V} = \frac{|\hat{\mu}|}{n}$.

The circular variance $\hat{V}, \hat{V} \in [0, 1]$ cannot be compared directly with its linear equivalent σ^2 which lies in the domain $[0, \infty)$. However by using the relationship between the normal distribution on the circle (wrapped normal, [6]) and the normal distribution on the line a circular standard deviation $\hat{\sigma}$ in the range $[0, \infty)$ can be defined as

$$\hat{\sigma} = \sqrt{-2log_n(1 - \hat{V})}. \quad (1)$$

All statistical definitions in this work which are used in the context of hue values always refer to the above definitions from circular statistics.

3. Background model

An adaptive background model is used for background subtraction and motion-based foreground selection. A parametric model as used by Francois *et al.* [2] for real-time segmentation of video streams is used. The model operates on the HSV colour space since it clearly separates chromatic and intensity information which makes it suitable for both

intensity and colour measurements. Each colour channel of a background reference pixel is modelled as a single and separate distribution since we use a static camera sequences and assume that each pixel of the background can be represented as a single colour (single model).

Since intensity and saturation are linear variables, Gaussian distributions characterized by a mean μ and a variance σ are used for modeling those two channels of a pixel. Note that colours are not normally distributed [8] but as shown in numerous works can be well approximated by the normal distribution [2]. Better suited distributions like the *Beta distribution* for the saturation and the intensity would probably provide a more accurate description than the Gaussian for values near the extremes. The natural finite domain $S \in [0, 1]$ of the Beta function is a significant advantage.

Nevertheless, most cameras have only a limited sensitivity range and do not cope well near the extremes. Since we are mainly interested in an accurate modelling of pixels within the camera working range, the simpler Gaussian distribution is appropriate for saturation and intensity components. However, for the hue component the Gaussian distribution is inappropriate since circular data behaves quite differently from linear data. Here a von Mises distribution, the circular equivalent of the Gaussian distribution, is an adequate density function. Similar in shape to the von Mises distribution is the wrapped Gaussian [6]. This distribution of the form

$$P(H) = \frac{1}{\sqrt{2\pi}\sigma_h} \sum_{k=-\infty}^{\infty} e^{-\frac{(H+2\pi k)^2}{2\sigma_h^2}} \tag{2}$$

where H is $N(0, \sigma_h)$, wraps the ordinary normal distribution around a circle. It is shown in [5] that the wrapped normal distribution is a very accurate approximation to the von Mises distribution for moderate SNR (signal-to-noise ratio). For a color distribution described by a mean μ and a standard deviation σ, the SNR can be defined as $SNR = \mu/\sigma$.

For a unimodal, static background we can assume that background changes occur slowly and signal distortions produced by, say, camera sensors are in a moderate range. Therefore the wrapped normal distribution is an appropriate simpler alternative to the von Mises distribution for modelling the hue values and is used in this system.

Each pixel of the background is described by two Gaussians $N(\mu_s, \sigma_s)$ and $N(\mu_v, \sigma_v)$ for saturation and intensity and one wrapped Gaussian $\tilde{N}(\mu_h, \sigma_h)$. Initially, the means of all three colour channels of the reference distribution are set to the corresponding values of the pixels in the first frame. The variance for each background distribution is always set above a minimal variance value of $\sigma_{min} > 0$ to tolerate noise which is always present in an image. After the initialization the model continually performs two main tasks. First the background mask is generated by comparing the reference distributions and the current frame. Secondly the distributions of the background model are updated by using the current frame information.

3.1. Background generation

The model decides if a pixel with value $I = [H, S, V]'$ belongs to the background by thresholding the distance between the three colour channels and the means of the correspondent colour distributions $\mu = [\mu_h, \mu_s, \mu_v]'$ in the background model. The circular domain of the hue is taken into account when computing the hue difference δ_h. We make the simplified assumption that the colour channels are independent of each other to reduce computational complexity. It is observed in [7] that this assumption degrades the quality of the results only minimally. If for a pixel at position x the difference for one of the channels is larger than a foreground threshold $\lambda_{\{h,s,v\}}(x)$ the pixel is marked as foreground $F(x) = 1$, otherwise it is labelled as background $F(x) = 0$. The threshold $\lambda_{\{h,s,v\}}$ depends on the variance of the corresponding colour channel

$$\lambda_{\{h,s,v\}}(x) = 2\sigma_{\{h,s,v\}}(x). \tag{3}$$

The range of 2σ is equivalent to a 95.5% confidence interval for a standard normal distribution. Since colour information is not Gaussian distributed [8] we can still expect each colour value to lie in the interval $[\mu-2\sigma, \mu+2\sigma]$ with a confidence of at least 75% by applying Tchebychev's Inequality theorem.

An undesirable property of the HSV colour space is its achromatic range. In this range the pixel lies on the central line of gray values and its hue information is meaningless and not usable as a distance measure. We define a pixel as achromatic if its saturation lies below a saturation threshold $\lambda_{achr_s} = 0.2$ or if its intensity is below an intensity threshold $\lambda_{achr_v} = 0.2$. According to this we only use the reliable channels of the frame pixel and the reference distributions for comparison.

Instead of using a saturation threshold for deciding if a hue value is useful, another possibility would be to weight each hue value by its corresponding saturation [4].

3.2. Update background model

After the pixels in the current frame have been labelled as foreground or background, the colour distributions of all reference pixels are updated by

$$\mu(t) = [1 - \alpha]\mu(t-1) + \alpha I \tag{4}$$

$$\sigma^2(t) = [1 - \alpha]\sigma^2(t-1) + \alpha[\mu(t) - I]^2. \tag{5}$$

Here α is the learning rate which defines how quickly old frames are forgotten. As in the background generation step, only those channels of a pixel which contain useful information are updated. In the case where a reference pixel and a frame pixel are both in the achromatic range and no useful colour information is available, only its intensity distribution is updated.

4. Detector

A detector similar to the implementation of Viola et al. [10] searches for pedestrians in single video frames. Therefore each frame is broken up into multiple sub-images and a classifier decides if the window contains a pedestrian.

As basic features for classification, a set of static *Haar-like rectangle* features as shown in Figure 1 is used. This kind of feature builds the basic structure of each of our classifiers and can quickly be computed for a grayscale image by using its *integral image*. By using an adapted version of

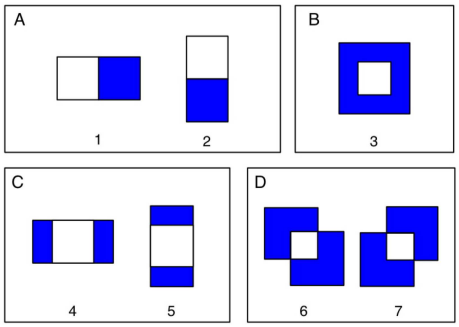

Figure 1: Features used in the detector.

the AdaBoost algorithm [3] we construct a *cascade of boosted classifiers* for quickly detecting pedestrians. Boosting is widely used in the field of pattern classification and is the idea of letting multiple and simple (=weak) classifiers decide a classification task by a majority vote.

The detector was trained with 1500 images of pedestrians from different angles. All training samples were manually extracted from multiple video sequences. The negative training samples were created by selecting random regions in images not containing any pedestrians (bootstrapping, [1]). The final detector cascade consists of 11 boosted classifiers. Each frame is divided into smaller sub images by using an uniform pyramid with 13 levels and a ratio of 1.2 between each level. The resulting 2684 sub images per frame are classified by the detector.

In our implementation an extension to the traditional boosted classifier is made. This extended boosted classifier is described in the next section.

4.1. Extended boosted classifier

A boosted classifier C normally uses a set of N_C weak classifiers $c_{1..N_C}$. The output of each weak classifier $c_i(x) \in \{0,1\}$ is weighted with a weak classifier weight α_i where $\sum_{i=1}^{N_C}\alpha_i = 1$ and $\sum_{i=1}^{N_C}\alpha_i c_i(x) \in [0,1]$. For deciding on the class $C(x)$ of a sample x the sum of all weighted outputs is used. If this sum is above the threshold λ_C of the boosted classifier C, the sample x is classified as positive, otherwise as negative.

$$C(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{N_C}\alpha_i c_i(x) \geq \lambda_C \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

In this implementation a constant threshold $\lambda_C = \frac{1}{2}$ was taken during the training, which means that if all classifier weights are equal, a sample has to be classified by at least 50% of the weak classifiers as positive to belong to class 1.

In a traditional boosted classifier normally all weak classifiers are evaluated on the sample image. If the weighted sum of all these outputs is above a certain threshold the image is classified as positive (or pedestrian) – otherwise as negative (or non-pedestrian). The fundamental idea behind extended boosting is to test already during the evaluation of the boosted classifier (e.g. after 50% of the weak classifiers) if it makes sense to evaluate the remaining weak classifiers. In particular in boosted classifiers with a large number of weak classifiers this would result in a performance increase.

We can easily pre-classify an image as pedestrian if after evaluating k weak classifiers the sum of their votes $\sum_{i=1}^{k}\alpha_i c_i(I)$ is above the pedestrian threshold λ_C of this boosted classifier C:

$$\underbrace{\sum_{i=1}^{k}\alpha_i c_i(I)}_{current} > \underbrace{\frac{1}{2}\sum_{m=1}^{N_C}\alpha_m}_{\lambda_C} \Rightarrow C(I) = 1 \quad (7)$$

If this is the case we can skip the remaining $N_c - k$ weak classifiers and the boosted classifier can classify the image immediately as pedestrian.

This *positive pre-classification* is implementable without changing the structure of the boosted classifiers. Another idea is if a *negative pre-classification* is possible where we can reject an image as non-pedestrian before we have evaluated all weak classifiers of a boosted classifier. The test at a weak classifier c_k determines if it makes sense to evaluate the remaining and unevaluated weak classifiers by

$$\underbrace{\sum_{i=1}^{k}\alpha_i c_i(I)}_{current} + \underbrace{\sum_{j=k+1}^{N_C}\alpha_j}_{possible} < \underbrace{\frac{1}{2}\sum_{m=1}^{N_C}\alpha_m}_{\lambda_C}. \quad (8)$$

Here the term $\sum_{i=1}^{k}\alpha_i c_i(I)$ represents the summed classification results of the already evaluated classifiers. If all remaining classifiers vote for the class pedestrian then the maximal possible increase could be $\sum_{j=k+1}^{N_C}\alpha_j$. An image I could be a pedestrian if the current sum $\sum_{i=1}^{k}\alpha_i c_i(I)$ and the maximal possible rest $\sum_{j=k+1}^{N_C}\alpha_j$ can reach the classification threshold $\frac{1}{2}\sum_{m=1}^{N_C}\alpha_m$. If they cannot reach it we can immediately classify the image as non-pedestrian and abort the classification. If we bring the term $\sum_{j=k+1}^{N_C}\alpha_j$ to the right side of the equation, we now have all constant terms on one side and can calculate the resulting sum β_k offline. The traditional boosted classifier structure is extended to an *extended boosted classifier* which contains, next to the weights $\alpha_{1..N_C}$, additional the sums $\beta_{1..N_C}$ for each of its features $f_{1..N_C}$. If the current sum at k is below β_k, the classification can be aborted immediately and the image labelled as non-pedestrian, i.e.

$$\sum_{i=1}^{k}\alpha_i c_i(I) < \beta_k \Rightarrow C(I) = 0 \quad (9)$$

For positive and negative pre-classification it makes sense to sort the weak classifiers c_i according to their weights α_i so that $\alpha_1 \geq \alpha_2 \geq ... \geq \alpha_{N_C}$. The weak classifiers are evaluated according to their importance since classifiers with a higher weight have a stronger influence on the final result.

For complex objects like pedestrians normally many weak classifiers must be evaluated until the image can be pre-classified by the tests of the Equations 7 and 9. Therefore we test only after 50%, 66%, 75% and 85% of the weak classifiers if the image could be a pedestrian[1].

[1]Note that both pre-classification methods neither change the classification result of a boosted classifier nor the detection or false positive rate but only influence the evaluation time.

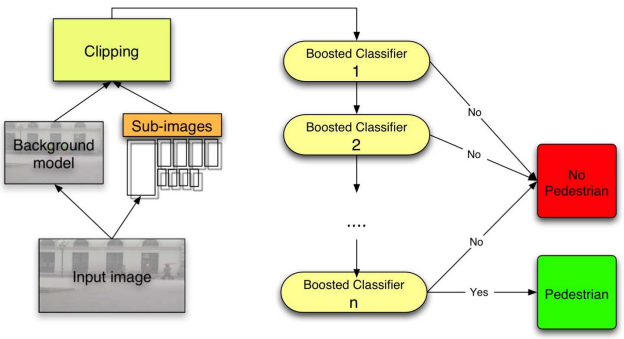

Figure 2: Adaptive Boosting with clipping.

4.2. Background clipping

A possibility to decrease the computations is to limit the number of images which are given to the classification cascade. The tracking algorithm in our system, described in Section 5, relies completely on the foreground information of a pedestrian which is provided by the background model. If no or only little foreground information is present in a sub-image it is impossible to find any correspondence between this detection and other objects. This implies that it does not make sense to give sub-windows without foreground information to the classifier cascade.

Therefore, after the background segmentation we compute the fraction of selected foreground pixels over the total number of pixels of a sub-image. If the fraction is above a threshold $\lambda_{clip} = 0.07$ then the sub-image is given to the classification cascade, otherwise rejected already before the classification. The new structure of the detector with clipping is shown in Figure 2 .

Long-term static pedestrians are adapted into the background and the windows which contain these pedestrians are clipped. This is legitimate since for these windows – even if they contain a pedestrian – no useful tracking information is provided. After we have clipped all windows with low foreground activity we classify the remaining windows with the boosted classifier cascade.

5. Tracking features

5.1. Feature vector

The effectiveness of the tracking process depends strongly on the choice of the tracking features. Our tracker uses the detector to get basic spatial information of a possible object and augments additional information by using the background model. The detector provides a set of K detection windows $d_k, k \in \{1..K\}$ in the current frame. Each window is defined by a size and a position. We firstly divide each detection window into three individual body zones (head, upper body, lower body) by using a fixed height ratio $r = (r_{head}, r_{ub}, r_{lb})' = [\frac{1}{4}, \frac{3}{8}, \frac{3}{8}]'$. Next each part is processed until it mainly contains a connected region of foreground pixels. After the processing the colour information is extracted. The feature vector for each body part contains the position P_k, the size S_k and the histograms $H_{k,\{h,s,v\}}$, means $\mu_{k,\{h,s,v\}}$ and variances $\sigma_{k,\{h,s,v\}}$ for all three colour channels. For deciding if a hue or saturation value represents useful information and should therefore be included in a histogram, the same rules as in the background segmentation are applied. For generating the hue histogram all hue values are additionally weighted by their corresponding saturation.

5.2. Distance measures

Since our feature vector contains spatial as well as colour information, different distance measures are used to calculate the distances between two of its elements. For measuring a spatial difference D_{spat} between two points $P_i, P_j \in \mathbb{R}^2$ the *Euclidean distance* is used.

The difference D_{hist} between two colour histograms H_i and H_j is computed by using the *Bhattacharyya distance*

$$D_{hist}(H_i, H_j) = 1 - \sum_{k=1}^{B} \sqrt{H_i(k) H_j(k)} \qquad (10)$$

where B is the number of histogram bins. In our work it was set to $B = 10$. The distance D_{dist} between two distribution $D(\mu_i, \sigma_i)$ and $D(\mu_j, \sigma_j)$ is computed by a modified *Mahalanobis distance*

$$D_{dist}(\mu_i, \sigma_i, \mu_j) = \frac{|\mu_i - \mu_j|}{2 \min(\sigma_i, \sigma_j)}. \qquad (11)$$

After the distances for all elements of the feature vectors are computed independently with the corresponding distance functions, all distance values are tested against border conditions like a maximal position difference $\Delta_{velocity}$ or a maximal scale difference Δ_{scale}. This rejects all objects which do not provide enough features to support robust tracking. After validation all distances are normalized, multiplied by a weight factor and summed to a total sum $D_{feature}$.

The distances between feature vectors are used for assigning detections to objects in the appearance model but are also used by the detector to find and reduce multiple detections of the same object in a frame to a single detection.

6. Appearance model

The *adaptive appearance model* (AAM) used in this work has to address multiple tasks. A major task is the stable handling of occlusions. In the case where an object is partly or completely occluded the AAM should be able to predict its current position and size. This is done by using the information about the velocity, direction, size and position of the occluded object, which was collected during the previous frames. Since a missing object detection due to the detector can be regarded as a complete occlusion, this special case can also be addressed by the AAM. Additionally non-pedestrian objects which are incorrectly classified as persons by the detector can be filtered by the AAM. They normally occur only briefly or stay stationary for long periods of time and therefore can be distinguished from "real" pedestrians.

The AAM can be divided into multiple steps. At the beginning the distances $D(o_j, d_k)$ between all existing objects $o_j, j \in \{1..N\}$ of the AAM and the detections $d_k, k \in \{1..K\}$ in the current frame are calculated. Here N and K represent the numbers of objects in the appearance model and the number of detections in the current frame. An existing object o_j is assigned to a detection if the distance $D(o_j, d_k)$ is smaller than a similarity threshold ξ_{sim}.

All objects in the AAM which could be assigned to a detection are updated with the new information. For objects without a corresponding detection in the current frame,

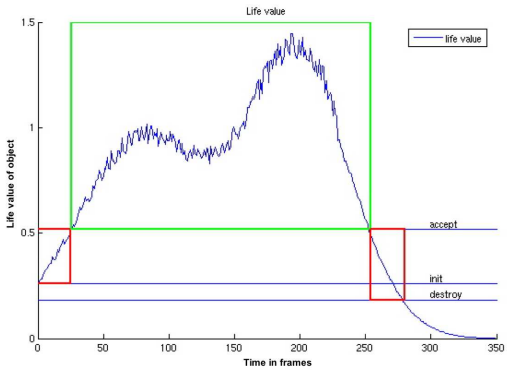

Figure 3: Life span of an object.

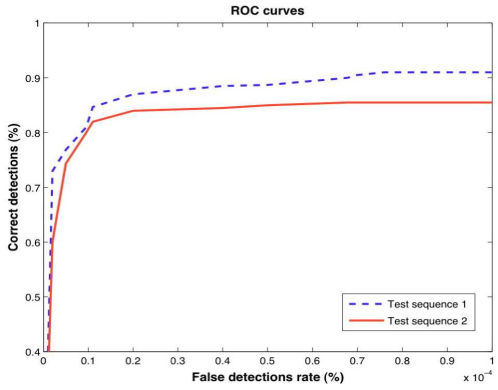

Figure 4: ROC curves of our detector.

the position and size are updated according to the model assumptions. Additionally the AAM creates new objects for all detections which could not be assigned to an existing object. When an object is created it gets an initial life value α_0 and is regarded as a possible pedestrian candidate. If a detection d_k is assigned to the object o_j, the life value α_j of the object is increased by a life bonus α_{bonus}.

If the life value α_j of an object rises above a threshold ξ_{acc} the object is accepted as a pedestrian and the object can be post-labelled as a pedestrian in all the previous frames. If the life span decreases below a threshold $\xi_{destroy}$ it is removed from the appearance model. All thresholds for the appearance model were manually adjusted to the scene properties.

Figure 3 shows different phases in the lifetime of an object. The red rectangle on the left side represents the phase of the object creation until it is accepted as a pedestrian. The green rectangle shows the phase where the object is regarded as a pedestrian. Note that the objects in all frames of the first rectangle are post-labelled as pedestrians. The red rectangle on the right side represents the time when the tracker loses the object because it has left the field of view or is occluded for too long a time. Finally the object is removed from the object pool.

During partial or complete occlusions the colour histograms of the colliding objects are mixed and no clear extraction of the colour information of a single object is possible. Therefore no update of the colour tracking features is done and no new objects are added to the appearance model during occlusions.

7. Results

We did the performance evaluation of our detector with two test sequence with 250 and 300 frames and with available ground truth information. To build the ROC curve we slowly lowered or raised the constant threshold λ_C for the boosted classifiers. The decreasing or increasing of the threshold λ_C leads to a rising or descent of the detection and the false positive rate of the detector which is used for creating the ROC curve. This global lowering of the threshold avoids the adjusting of each boosted classifier and always regards the performance of the complete classifier cascade. During the generation of the ROC curve the clipping algorithm was deactivated to avoid a performance deterioration of the detector.

We can see in Figure 4 that for both test sequences the detection rate is around 80% at a false positive rate of 10^{-5}. At a higher false positive rates of $F \geq 2 \times 10^{-5}$ the detector reaches a detection rate of 88% in the first and 84% in the second test sequence. In the second sequence this was the highest detection rate which could be achieved. In test sequence 1 a detection rate $D \approx 0.91$ at a false positive rate of $F \geq 7 \times 10^{-5}$ was achievable. These rates have a similar order of magnitude to those obtained by Viola et al. [10].

The positive pre-classification with the extended boosted classifier was tested with a random set of negatives which was created with bootstrapping. We compared the percentage of windows rejected at one of the defined pre-classification points at 50%, 66%, 75% and 85% of the total feature number. Additionally the window percentage which required more than 85% of the weak classifiers was counted.

Only around 4% of the windows can be rejected after evaluating 50% of the weak classifiers. After 66% already 30% of the windows can pre-classified as a pedestrian and we can classify more than the half of the windows with only 75% of the weak classifiers. The other half of the images is harder to classify and requires at least 85%.

For the negative pre-classification only a relatively small amount of 5% of the windows can already be rejected after the evaluation of only one half of the weak classifiers. After 66% already around 15% of the negative samples can be rejected. After evaluating 75% of the weak classifiers already 40% of the negative samples can be classified as negative. A large ratio of more than two thirds of all negatives is rejected after 85% of the weak classifiers and only a rest of 26% requires the complete evaluation of the boosted classifier.

The theoretic velocity gain of the two pre-classification steps seems quite high but in practice the effective speed up is much smaller. The negative pre-rejection only saves computations in the boosted classifier which rejects the negative sample. The highest speed upbuilds the basic structure of each of our classifiers for negative pre-classification would be possible for complex boosted classifiers which contain many weak classifiers and are located at the end of the cascade. Due to the cascade structure only few negatives normally reach these complex classifiers and therefore the efficiency of negative pre-classification is limited. However, for negatives which are hard to classify and which reach higher stages in the cascade a speed up is achieved since they are

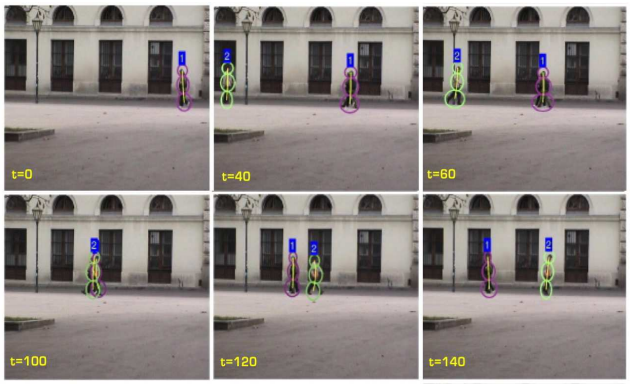

Figure 5: Tracking with occlusion.

classified as positives in all previous boosted classifiers before the rejecting one. The positive pre-classification saves in contrast to the negative pre-classification the evaluation of weak classifiers in each boosted classifier and is more efficient.

The clipping algorithm is also tested on the two test sequences. Due to a clear background segmentation it is possible to reduce the number of detection windows constantly to around 12% in both video sequences. This relatively small percentage is then classified by the classifier cascade. The activated clipping does not deteriorate the performance of the detector in either test sequence. This is probably due to the constant movement of the pedestrians in both test scenes. If a pedestrian would remain still for a longer period, he/she would be adapted into the background and therefore clipped in the next detection step. This is intended because the pedestrian then does not provide any tracking information. For a video sequence where the background cannot be separated clearly enough from the foreground, this clipping algorithm would neither result in a performance boost nor deteriorate the detection rate.

To test how well the AAM copes with occlusions we raised the classification threshold. Therefore, fewer pedestrian detections were registered and the AAM had to cope with multiple frames where object detections are not provided by the detector. The AAM provides good approximations if an object is occluded for not longer than 15 frames. Since the color features of the pedestrians remain nearly constant, objects could be robustly found and reassigned after an occlusion.

The background model performs well. It separates foreground and background well enough to extract the tracking features of moving objects. Since all test sequences are created with static cameras, one distribution per colour channel is sufficient for modeling the colour distribution.

8. Conclusion

In this paper we introduced a tracking system based on a combination of an adaptive background model and an object detector to quickly locate pedestrians in video frames and to extract their colour and spatial information. We demonstrated in two test sequences how the appearance model can use this information to robustly track the objects through the scene. Here a direct comparison of how well the use of appropriate circular statistics improves the processing of directional colour data provides interesting possibilities for future investigations.

The tests showed that the pre-classification leads to a speed up since it saves unnecessary feature evaluations in nearly all boosted classifiers. However, the use of the clipping algorithm already reduces the number of windows which have to be classified significantly before the classification and leads to a increase in the detection speed. Full details on the system are available in [9].

Future work includes a separation of the training samples into different view angles and using multiple and view specialized detectors. This will probably improve the detection rate of the detector.

To additionally improve the background model, another colour space or shadow removal techniques can be used. An improved foreground segmentation would increase the quality of the tracking features and make the prediction of the object location and size more reliable.

Acknowledgements

This work was supported by the European Union Network of Excellence MUSCLE (FP6-507752), and the Austrian Science Foundation (FWF) under grant SESAME (P17189-N04).

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2nd edition, 2000.

[2] A. R. Francois and G. G. Medioni. Adaptive color background modeling for real-time segmentation of video streams. In *Proc. of the Int. Conf. on Imaging Science, Systems, and Technology*, pages 227–232, 1999.

[3] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. In *The Annals of Statistics*, volume 28, pages 337–374, 2001.

[4] A. Hanbury and J. Serra. Colour image analysis in 3d-polar coordinates. In *DAGM 2003*. Springer-Verlag, 2003.

[5] B. C. Lovell. *Techniques for Non-Stationary Spectral Analysis*. PhD thesis, University of Queensland, 1991. Brisbane.

[6] K. V. Mardia. *Statistics of directional data*. Academic Press, London, 1972.

[7] F. Porikli and O. Tuzel. Human body tracking by adaptive background models and mean-shift analysis. In *IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, March 2003.

[8] N. Sebe and M. S. Lew. A maximum likelihood investigation into color indexing. In *Proc. Visual Interface 2000*, pages 101–106, 2000.

[9] Florian H. Seitner. Robust detection and tracking of objects. Technical Report TR-PRIP-100, TU Vienna, 2005.

[10] P. A. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. of ICCV03*, volume 2, pages 734–741, 2003.