Background Subtraction in Videos using Bayesian Learning with Motion Information

Padmini Jaikumar, Abhishek Singh and Suman K Mitra
Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar, Gujarat, India - 382007
{padmini_jaikumar,abhishek_singh,suman_mitra}@daiict.ac.in

Abstract

This paper proposes an accurate and fast background subtraction technique for object tracking in still camera videos. Regions of motion in a frame are first estimated by comparing the current frame to a previous one. A sampling-resampling based Bayesian learning technique is then used on the estimated regions to perform background subtraction and accurately determine the exact pixels which correspond to moving objects. An obvious advantage in terms of processing time is gained as the Bayesian learning steps are performed only on the estimated motion regions, which typically constitute only a small fraction of the frame. The technique has been used on a variety of indoor and outdoor sequences, to track both slow and fast moving objects, under different lighting conditions and varying object-background contrast. Results demonstrate that the technique achieves high degrees of sensitivity with considerably lower time complexity as compared to existing techniques based on mixture modeling of the background.

1 Introduction

With increase in processing power, the use of computers for complex image processing applications has been on the rise. An important vision application is in the domain of object detection and tracking. Techniques based on mixture modeling of background remain most popular [11],[13],[9],[14],[3]. Many of the existing techniques compromise on the accuracy of the system, in favour of achieving fast processing speeds. Stauffer and Grimson [13] have used a fast online k-means based approximation to update the parameters of a Gaussian Mixture Model. While the method is very effective when the contrast between background and foreground is high, it yields poor results when the contrast is low [6], [11]. Block Matching based techniques have also been used for fast object tracking [7],[8]. However, these techniques yield at best a rough estimate of the moving body. They fail to yield an accurate contour of the moving body, required by applications such as object recognition, military surveillance etc.

On the other hand, relatively slower object tracking techniques such as [11] and [3] yield results with high sensitivity and appreciable performance in demanding conditions. Singh et al. [11] have used a combination of the EM algorithm and the online k-means approximation for updating parameters of the mixture model to obtain appreciable results

BMVC 2008 doi:10.5244/C.22.61

in low contrast conditions. However, the resulting technique is quite slow, in addition to having a high false alarm rate.

This paper presents a robust system that acheives both (1) high speed and (2) high degrees of sensitivity compared to existing techniques. To achieve these objectives a 2 step tracking system has been used.

Typically, in a still camera video sequence only a small portion of the each frame has motion relative to previous frames. Existing approaches to object tracking such as [13], [11], [9], [3] perform segmentation algorithms on all spatial pixel locations in the frame, leading to needless computational cost. To overcome this limitation, regions of the frame which have had motion relative to previous frames are first estimated using Sum of Absolute Differences (SAD). It is only on these regions that the segmentation algorithm is performed, leading to considerable saving in computational cost.

To achieve object tracking with high sensitivity and low false classifications, a Bayesian Learning based object-background classification technique is used. Bayesian learning techniques of determining the parameters of a model are generally more accurate as compared to classical probabilistic techniques, and most modern machine learning methods are based on Bayesian principles [1].

Pixel observations at a particular spatial pixel location are expected to form a certain number of clusters. The parameters of these clusters are thought to have probabilistic distributions of their own. These distributions are updated via a Bayesian 'Sampling-Resampling' learning technique (elaborated later) to obtain posterior distributions. These posterior distributions along with some criteria are used to classify pixel observations as background and foreground.

The results obtained using this method show a considerable improvement in the fraction of the actual foreground detected and reduction in incorrect classifications as compared to existing real-time techniques as proposed by Stauffer and Grimson [13] and offline techniques as proposed by Singh et al. [11]. At the same time the speed of the algorithm is shown to be comparable to real-time tracking techniques such as [13].

The next section describes a statistical 'sampling-resampling' technique given by Smith and Gelfand [12], which suggests easy implementation strategies and computational efficiency while implementing Bayesian learning. Section 3 describes our method used for object tracking. Sections 4 and 5 show results and conclusions respectively.

2 Mathematical Preliminaries

2.1 Sampling-Resampling based Bayesian Learning

Given a model and some observation(s), the prior distribution of the parameters of the model is updated to a posterior distribution as,

$$p(\theta|x) = \frac{l(\theta;x)p(\theta)}{\int l(\theta;x)p(\theta)d\theta}, \tag{1}$$

which is a familiar form of Bayes' Theorem. Except in very simple cases, evaluation of a posterior distribution as above would require sophisticated numerical integration or other analytical approximation techniques, which can be totally off-putting for practical applications. Smith and Gelfand [12] address this problem by giving a new look to Bayes'

Theorem from a sampling-resampling perspective. In terms of densities, the essence of Bayes' Theorem is to relate the prior density to the posterior density via the likelihood function. Shifting to samples, this corresponds to obtaining a set of posterior samples from a set of prior samples (of the parameter distribution). A method described in [12] of doing so can be very briefly summarized in the following steps:

1. Given a prior distribution $p(\theta)$ of parameter θ, obtain n samples $\{\theta_1, \theta_2, ..., \theta_n\}$ from it.

2. Compute weight q_i for each sample θ_i, using the likelihood function as follows:

$$q_i = \frac{l(\theta_i; x)}{\sum_{j=1}^{n} l(\theta_j; x)} \tag{2}$$

3. Draw θ^* from the discrete distribution $\{\theta_1, \theta_2, ..., \theta_n\}$, placing mass q_i on θ_i. Then θ^* is approximately distributed according to the required posterior distribution $p(\theta|x)$, given the current observation x. The justification for this can be found in [12].

Note that this resampling technique is also a variant of the bootstrap resampling procedure as described in [2] and the SIR (sampling/importance resampling) procedure in [10].

3 Proposed Method

Segmenting moving objects in still camera video frames is done in three stages in the proposed method. Section 3.1 describes the first step of the tracking algorithm which involves estimating regions in the current frame which have motion. Section 3.2 describes the 'Sampling-Resampling' based Bayesian Learning technique which has been used for estimating parameters of the distribution formed by pixel observations at a particular spatial pixel position. Section 3.3 describes the criteria used for classifying pixel observations into background and foreground.

3.1 Isolation of Regions of Motion

The Block Matching Algorithm (BMA) is a standard way of encoding video frames [5]. A simplified variation of the BMA algorithm is used for determining regions of each frame which have had motion relative to a reference frame. Such regions have been called *regions of motion*. Each incoming frame is divided into non-overlapping blocks of equal size. Each block is compared to the corresponding block in the reference frame and the Sum of Absolute Difference (SAD) is determined for the block,

$$SAD = \sum_{i=1}^{X} \sum_{j=1}^{Y} (B_t(i,j) - B_{t-1}(i,j)), \tag{3}$$

In order to attenuate the effect of noise, a threshold value called *zero motion bias* has been used [5]. The threshold defines the minimum difference that the two corresponding blocks must have in order for the current block to be identified as a *region of motion*. If

the SAD value of the current block is below the *zero motion bias*, the block is said to be motionless and the segmentation algorithm is not performed on the block. This technique results in a significant reduction of false detections. The precise segmentation algorithm (section 3.2 and 3.3) is only performed on blocks which have their SAD values above the *zero motion bias*. The reference frame may be chosen to be a few frames before the current frame, to account for slow moving objects.

Figure 1: Result of Motion Region Estimation

Alternately, information from *Motion Vectors* (already available in MPEG videos) can be used to estimate the regions/blocks which have motion in them. *Motion Vectors* are used in the compression scheme employed in MPEG videos.

3.2 Bayesian Learning of cluster parameters

Pixel observations at a particular spatial pixel location (also called 'pixel process' in [13]) are expected to form a certain number (less than K) of clusters. The observations or data points would be a scalars in case of grayscale videos, and RGB vectors in case of color videos. The Mean value μ_i of each cluster is thought to have a probability distribution $p_i(\mu_i)$, where $i = 1, 2, ..., K$. Therefore, for each pixel position there exist K distributions of cluster Means. Whenever a pixel value is observed, the existing or prior distribution of one of these cluster Means is updated to a posterior distribution using a Bayesian learning technique. This learning process continues throughout the entire video sequence. The details of the Bayesian learning steps are described below.

The Bayesian learning process described above is only performed for pixel observations in the regions of motion identified for the frame. The prior distributions of the pixel observation which are not in the regions of motion (of the current frame) are left unchanged.

The first few frames (the first few *learning observations* of a pixel process) are required to build a stable distribution of the cluster Means. No classification is done for these frames.Henceforth, for each observation, a classification step is also performed wherein the observation is adjudged (based on certain criteria) as foreground or background. The details of the classification steps are described in Section 3.3.

3.2.1 Steps for Bayesian Learning

The following steps are performed for each observation made at a particular spatial pixel location:

1. Draw N samples each from all the prior distributions of the K Means. Let us call the obtained samples as $\{\mu_{11}, \mu_{12}, ..., \mu_{1N}\}$, $\{\mu_{21}, \mu_{22}, ..., \mu_{2N}\}$, ..., $\{\mu_{K1}, \mu_{K2}, ..., \mu_{KN}\}$.

2. When a pixel value $\mathbf{x}$ is observed at the particular location, compute the sum of likelihoods for each Mean distribution, given that observation:

$$L_r = \sum_{i=1}^{N} l(\mu_{ri}; \mathbf{x}), \qquad r = 1, 2, ..., K. \tag{4}$$

The likelihood of each Mean sample μ_{ri} is calculated as the probability of observing $\mathbf{x}$ in a Gaussian distribution centered at μ_{ri}, with covariance matrix Σ_M.

$$l(\mu_{ri}; \mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_M|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mu_{ri})' \Sigma_M^{-1} (\mathbf{x}-\mu_{ri})} \tag{5}$$

The *model variance*, Σ_M can be thought of as a parameter to control the sensitivity of the system. Its effect on clustering is described later.

3. The next step is to determine which cluster the pixel observation belongs to. The observation would belong to the cluster having the highest sum of likelihoods value L_r. The prior distribution of the Mean of this cluster is updated to obtain a posterior distribution using step 4. The distributions of the Means of the other clusters are left unchanged.

4. The steps to update a prior distribution to a posterior one are:

   (a) If the r^{th} distribution is to be updated, compute weights q_i for each sample μ_{ri} of the prior distribtion as follows:

   $$q_i = \frac{l(\mu_{ri}; \mathbf{x})}{L_r}, \quad i = 1, 2, ..., N \tag{6}$$

   (b) $\{\mu_{r1}, \mu_{r2}, ..., \mu_{rN}\}$ are then resampled using the weighted bootstrap method with weights $\{q_1, q_2, ..., q_N\}$ to obtain samples from the posterior distribution of μ_r, which are $\{\mu^*_{r1}, \mu^*_{r2}, ..., \mu^*_{rN}\}$

5. When the next pixel observation is made, the posterior samples $\{\mu^*_{r1}, \mu^*_{r2}, ..., \mu^*_{rN}\}$ become the *prior* samples of the r^{th} cluster Mean.

Steps 1 through 5 are repeated for every observation of the pixel process.

It is important to note that this entire process is done just for one pixel process. Therefore, if *regions of motion* were not used, in a 80x120 pixel video sequence for example, the entire process would need to be done for all the 9600 (80*120) pixel positions, independently. However, implementing the learning process on *regions of motion*(which is typically only a small fraction of the frame) significantly improves speed without compromising the tracking ability of the system, as shown in the results section.

3.2.2 Effect of changing *Model Variance*

Having a narrower Gaussian (small values in the matrix Σ_M) for computing likelihoods (equation 5) would mean that the bootstrap weights would decrease more rapidly as the distance of the sample, μ_{ri}, from the current pixel observation, $\mathbf{x}$, increases. Only those

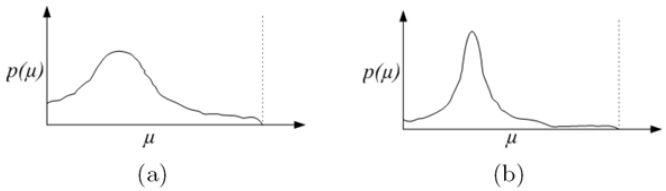

Figure 2: Schematic showing posterior distributions of a univariate Mean. (a) Posterior distribution obtained when a high *model variance* is used. (b) Posterior distribution obtained when a low *model variance* is used

samples which are very close to the pixel observation would be assigned high weights. As a result, posterior samples would form a narrow distribution, as shown qualitatively in Fig. 2.

This would result in a finer clustering of observations. Closely separated observations would be clustered into different classes. Therefore a low value of *model variance* results in high sensitivity and better results in cases where foreground and background clusters are close (low contrast conditions).

3.3 Classification of Pixel Observations into Background and Foreground

For every pixel observation, classification involves determining if it belongs to the background or the foreground. The first few initial frames in the video sequence (called *learning frames*) are used to build stable distributions of the cluster means, using the process detailed in Section 3.2. No classification is done for these *learning frames*. Classification is done for subsequent frames using the process given below.

Typically, in a video sequence involving moving objects, at a particular spatial pixel position a majority of the pixel observations would correspond to the background. Therefore, background clusters would typically account for much more observations than the foreground clusters. This means that the prior weight (ω) of any background cluster would be higher than that of a foreground cluster. The clusters are ordered based on their prior weight. Based on a certain threshold Th, the first B clusters are chosen as background clusters, where

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > Th \right) \tag{7}$$

Th is a measure of the minimum portion of the data that should be accounted for by the background. A relatively lower value of Th (≈ 0.5) can be used when the background is unimodal. A higher value of Th (> 0.7) allows more than one Gaussian to be a part of the background, enabling the mixture model to adapt to lighting changes, repetitive motion etc.

The sum of likelihoods (L_r) is used to determine the cluster to which the observed pixel belongs. If this cluster is *not* one of the first B clusters as described above, the pixel would be a foreground pixel.

The classification process is only performed on the pixel locations inside the *regions of motion* in the current frame. All pixel locations in the current frame outside the *regions of motion* are classified as background.

4 Experimental Results

The proposed technique has been tested on a variety of indoor and outdoor video sequences. It has been used to track both fast and slow moving objects under different lighting conditions, varying object-background contrast and situations in which the object is camouflaged by the background. Fig. 3 and 4 show the two steps of the tracking process.

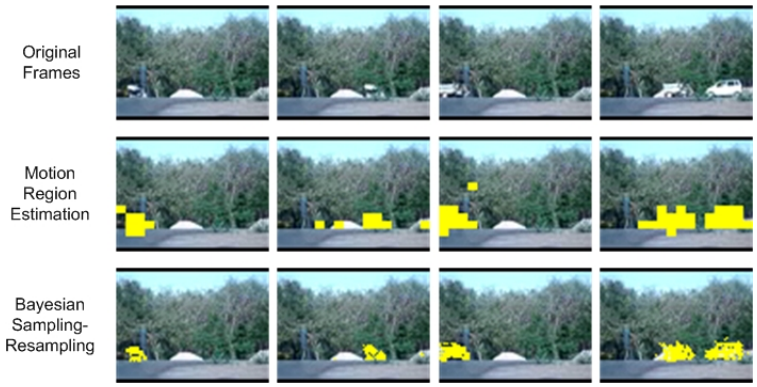

Figure 3: The first row shows original frames from a video sequence. The second row shows the results of motion region estimation. The third row shows the final Bayesian Sampling-Resampling results. Note that fast moving objects which seem to be camouflaged by the background are also accurately detected.

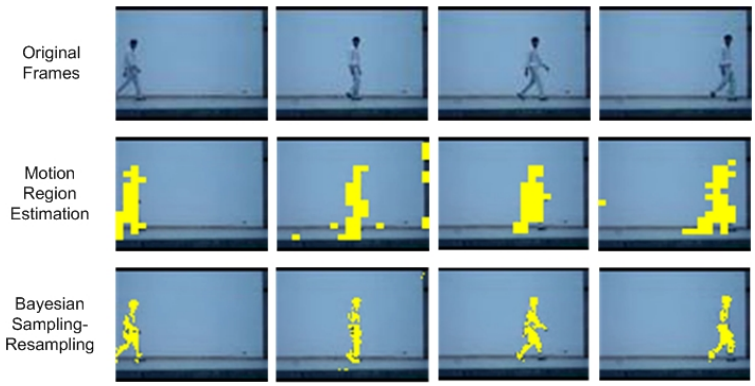

Figure 4: The first row shows original frames from a video sequence. The second row shows the results of motion region estimation. The third row shows the final Bayesian Sampling-Resampling results.

The tracking accuracy and the time complexity of the proposed technique has been compared using a low contrast benchmark video (obtained from the Advanced Computer Vision Gmbh - ACV[4]) to two existing techniques proposed by Stauffer and Grimson[13] and Singh et al.[11] in Fig 5. Plots of Sensitivity (the fraction of the actual foreground detected) and False Alarm Rate (fraction of pixels incorrectly classified as foreground) are also shown to better understand the results quantitatively. A table comparing the CPU time taken by the 3 techniques has also been shown. The values were obtained by implementing the techniques on 128x96 pixel videos, in Matlab 7.2 using a 1.7 Ghz processor. Note that these are time taken for running computer simulations of the techniques, meant for comparative purposes only. Actual speeds on optimized real time systems may vary.

As can be seen from the results, the Stauffer and Grimson approach[13] has the least time complexity, however, the showing low Sensitivity. The approach proposed by Singh et al.[11] achieves the highest sensitivity, but at the cost of higher processing time.

The proposed technique achieves Sensitivity comparable to [11] while maintaining a time complexity comparable to [13]. Also, the False Alarm Rate of the proposed technique is lower as compared to [11].

Table 1: Comparison of Sensitivity, False Alarm Rate and CPU Time for results shown in Fig. 5

Approach used	Avg. Sensitivity (%)	Avg. False Alarm Rate (%)	CPU Time for 100 frames (min:sec)
GMM with online approx.[13]	36	0.05	1:25
GMM with online approx. and EM[11]	87	1.8	28:30
Bayesian Learning with Motion Region Estimation	78	0.46	1:55

5 Conclusion

This paper has presented an accurate and fast background subtraction approach in still camera videos. Unlike existing real-time techniques that compromise on quality of segmentation, the proposed method achieves high processing speed with no compromise accuracy. The high sensitivity is achieved using an accurate Bayesian learning approach. The accurate contours of segmented objects allow for their use in higher level vision applications as well, such as object extraction, recognition etc. The proposed segmentation technique using Bayesian learning also retains the advantages of using mixture models to model the background, such as adaptation to lighting changes and multimodal backgrounds. Results have indicated, both quantitatively and qualitatively, the superiority of the proposed technique.

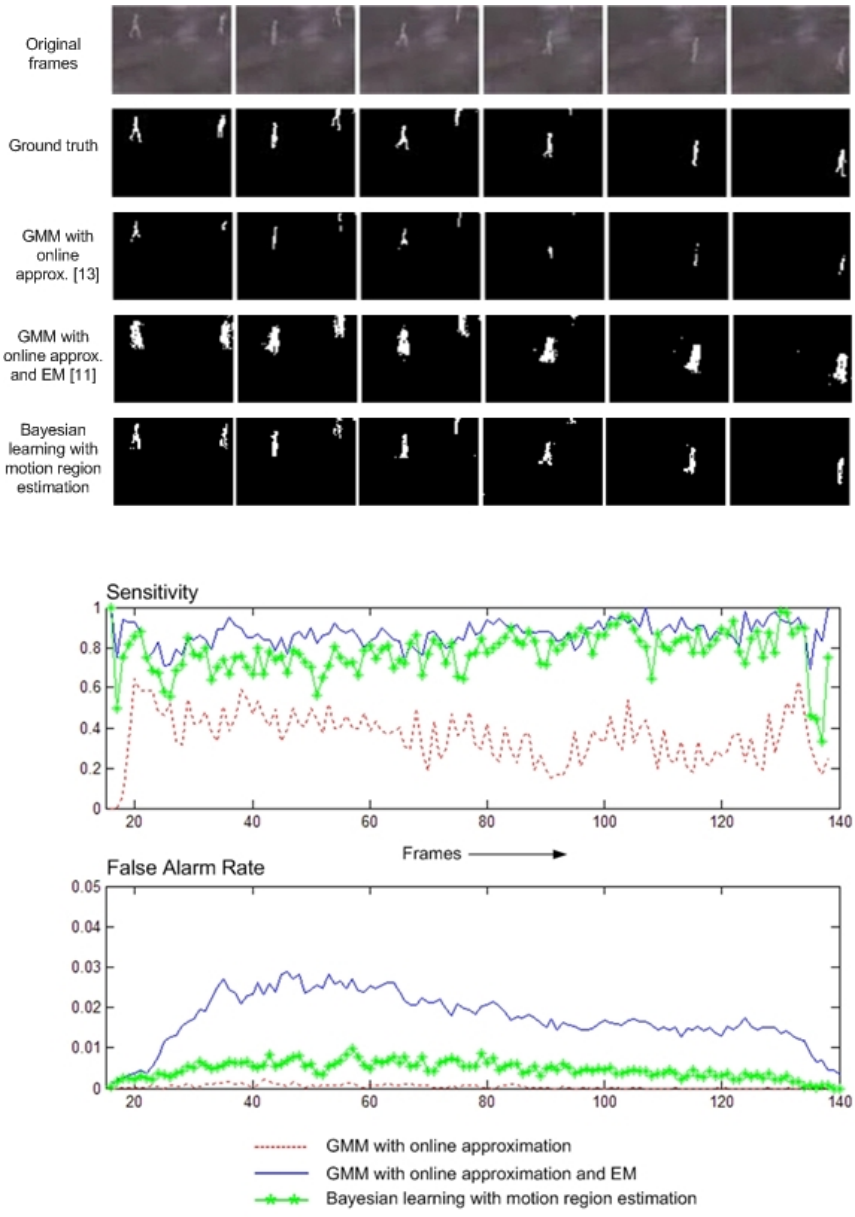

Figure 5: The first row shows original frames from a low contrast benchmark video sequence obtained from [4]. The second row shows the ground truth frames of the same video. The third row shows the tracking results using the Stauffer and Grimson approach [13], where only some high contrast regions are tracked well. The fourth row shows the tracking results using the Singh et al. approach [11], where object detection is not sharp and accurate. The fifth row shows results obtained using the proposed technique

References

[1] C.M Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

[2] B Efron. *The Bootstrap, Jackknife and Other Resampling Plans*. Society of Industrial and Applied Mathematics, Philadelphia, 1982.

[3] N Friedman and S Russel. Image segmentation in video sequences. In *Thirteenth Annual Conference on Uncertainty in Artificial Intelligence, San Francisco, USA*, pages 175–181, 1997.

[4] Advanced Computer Vision GmbH. Motion detection video sequences. http://muscle.prip.tuwien.ac.at/data_here.php.

[5] A Gyaourova, C Kamath, and S.-C Cheung. Block matching for object tracking. Technical report, Lawrence Livermore National Laboratory, Livermore, Calif, USA, 2003.

[6] Dar-Shyang Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, May 2005.

[7] J Lu and M.L Liou. A simple and efficient search algorithm for block-matching motion estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(2):429–433, April 1997.

[8] S Mattoccia, F Tombari, L.D Stefano, and M Pignoloni. Efficient and optimal block matching for motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, Colorado, USA*, pages 599–608, June 1999.

[9] C Ridder, O Munkelt, and H Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. In *Proceedings of International Conference on recent Advances in Mechatronics,ICRAM95, UNESCO Chair on Mechatronics*, pages 193–199, 1995.

[10] D.B Rubin. Using sir algorithm to simulate posterior distributions. *Bayesian Statistics 3. Oxford University Press*, pages 395–402, 1988.

[11] A Singh, P Jaikumar, S.K Mitra, M.V Joshi, and A Banerjee. Detection and tracking of objects in low contrast conditions. In *IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Gandhinagar, India*, pages 98–103, January 2008.

[12] A.F.M Smith and A.E Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46(2):84–88, May 1992.

[13] C Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition, Colorado, USA*, pages 599–608, June 1999.

[14] C.R Wren, A Azarbayejani, T Darrell, and A Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, July 1997.