Fast Bayesian People Detection

Gwenn Englebienne [a] Ben J.A. Kröse [a]

[a] *Universiteit van Amsterdam, Science Park 107, 1098XG Amsterdam*

Abstract

Template-based methods have been shown to be effective at solving the problem of tracking specific objects, but their large number of free parameters can make them slow to apply and hard to optimise globally. In this work, we propose a template-based method for tracking people with fixed cameras, which automatically detects the number of people in a frame, is robust to occlusions, and can run at near-real-time frame rates. We demonstrate the effectiveness of the method by comparing it to a state-of-the-art background segmentation algorithm and show its important performance advantage.

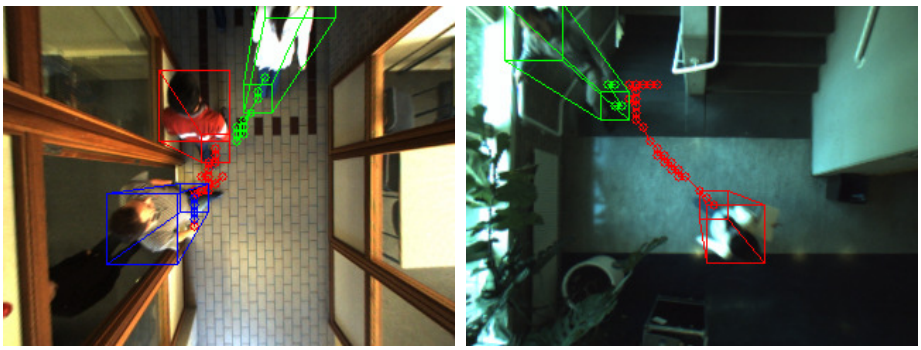
1 Introduction

Tracking the motion of people in video images is applied to a variety of situations, including athletic performance analysis, content-based video retrieval, surveillance applications, crowd flow analysis and people counting, but also as a preprocessing step to more advanced methods such as gait analysis, behaviour modelling, *etc.* In typical scenarios, accurate tracking can be extremely challenging. Multiple effects such as varying illumination, occlusions, shadows, specularities and non-static backgrounds all contribute to make the tracking process quite complex.

In this paper, we focus on the detection of humans in indoor scenes with fixed cameras. In many applications, most notably in surveillance applications, the computational cost of the detection is critical and one would want the detection process to happen at or above the video frame rate. We propose a simple but very effective probabilistic method, which allows the automatic evaluation of the number of people in the scene and the detection of those people's location. This method has the following advantages: (1) It can incorporate prior knowledge, including which areas in the scene can contain people and how probable it is for people to be in those locations; a probability distribution over the number of people in the scene; a probabilistic model of how close together people tend to walk; *etc.* (2) The complexity of the algorithm depends linearly on the number of people in the scene. When many people are present in the frame, detecting all of them requires more than 1/25th of a second with our current implementation of the algorithm, although it still requires far less than a second. Further optimisations could easily improve this performance. (3) The method is very robust to changes in illumination, shadows and occlusions, and it can easily be made to adapt to non-static background automatically. (4) Thanks to its generative probabilistic nature, the model can easily be incorporated into probabilistic models of motion across consecutive frames, such as Kalman filters or particle filters.

2 Related work

Foreground segmentation is typically done by background subtraction [5, 2], or using a probabilistic model of the background [17] after which, for each pixel, a hard decision is made whether to consider that pixel as foreground or as background. The obtained foreground regions are typically noisy, and an extra noise-cleaning step is performed to eliminate foreground regions that are too small, or too short-lived [2]. Connected components of foreground pixels can then be found, resulting in foreground "blobs". The main problem with this approach is that a single person will easily give rise to multiple blobs, and parts of multiple people will easily be combined into a single blob. Further processing, typically relying on temporal information, is then required to disambiguate the blobs [8].



(a) (b)

Figure 1: Example frames taken with different cameras in different locations.

The blobs are typically used for tracking over multiple frames, and consecutive frames are therefore informative of each other. As a result, most attention has been spent on techniques, including Kalman filters [3], particle filters [10] and graph-based methods [8], which use information from multiple frames to disambiguate the blobs and create accurate tracks from inaccurate observations.

Template-based tracking has been applied successfully to a variety of situations, including the tracking of rigid objects [9], segmenting and tracking humans in crowded scenes [16] and human body pose estimation [1]. However, in order to allow for sufficient flexibility, such methods either require adapting the templates over time [9] or extra parameters which need to be optimised to fit the template to the observation. In our approach, the template is adapted to a foreground object's position in the image, but not to the particular appearance of the foreground objects. This makes our approach fast and insensitive to the local optima that may arise when adjusting the template. It also makes the method more robust to noise in the image.

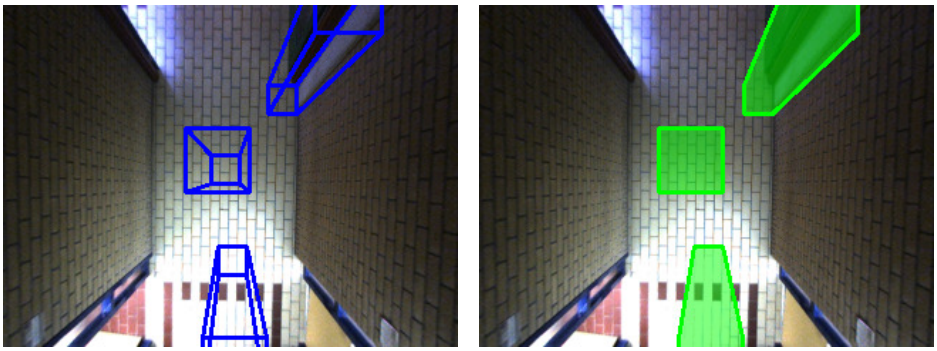
3 Segmentation

We assume fixed cameras looking straight down from the ceiling. Such a setup was proposed in [4], and has a number of advantages, including reduced numbers of occlusions and, if models are built of the tracked person's appearance, more holistic appearance models. Example images of our setup are depicted in Figure 1. Our purpose is to track the motion of people in a building without imposing any constraints on the subjects' behaviour (except for the constraints enforced by the architecture of the building), and in realistic conditions. In practice, this results in a variety of light conditions, shadows, differences in camera position (height of the camera), motion blur, as well as complex and dynamic backgrounds (doors being opened and closed, tables being moved around, *etc.*)

The sequences were captured by Point Grey Bumblebee stereo vision cameras, with one PC per camera. Each camera recorded data at an average frame rate of approximately 20Hz. Each frame consists of a left and a right image, which can be used to compute a disparity map and depth information. In practice, however, it turned out that the regular pattern of the floor in Figure 1 (a), and the bad illumination in Figure 1 (b) made it very hard to avoid artefacts in the depth of field information. The stereo information was not used for the results reported in this paper. Tracking within the cameras' field of view was performed as follows.

4 Model

The model we use is a generative model of the images. We consider an observation vector $\mathbf{x}$, containing the hue, saturation and value (HSV) components of the pixels of a video frame. For a 320 pixel image, this corresponds to a vector of $320 \times 240 \times 3 = 230400$ elements. The video frames are corrected for the optical distortion due to their lenses, using the camera calibration provided by the manufacturer. Each pixel $\mathbf{x}_i$, containing H,S and V components, is modelled as belonging either to foreground or to background. The probability of a foreground pixel is denoted as $p_f(\mathbf{x}_i)$, and we denote the vector of foreground pixel probabilities as $\mathbf{p}_f(\mathbf{x})$. Similarly, we use $p_b(\mathbf{x}_i)$ and $\mathbf{p}_b(\mathbf{x})$ to denote, respectively, the background probability of a pixel and the probability vector of the image.



(a) (b)

Figure 2: Projection examples of bounding boxes on the ground plane, with fixed width and height (a), and a depiction of the corresponding mask vector (b)

4.1 Modelling people's locations

We can then introduce a foreground mask $\mathbf{m}$, which is a vector whose elements are one for the components of foreground pixels and zero otherwise, so that the probability of an image, given a mask, is given by

$$p(\mathbf{x}|\mathbf{m}) = \mathbf{m} \cdot \mathbf{p}_f(\mathbf{x}) + (\mathbf{1} - \mathbf{m}) \cdot \mathbf{p}_b(\mathbf{x}) \tag{1}$$

where $\cdot$ indicates the Hadamard (elementwise) product and $\mathbf{1}$ is a vector of suitable length containing all ones.

Since the camera position, orientation and lens aperture are fixed, we can compute what the projection in the camera's view would be of the bounding box (in 3D) of an object of a given width and height standing in a particular location on the ground plane. This is illustrated in Figure 2(a), and requires knowing the height and orientation of the camera. These are manually provided for the cameras and are fixed for the length of the sequence. From the bounding box's projection, we can compute which pixels would belong to that object and which belong to the surrounding background, resulting in a mask vector $\mathbf{m}$, as illustrated in Figure 2(b). In the case of person tracking, this mask would in general depend on the position in the ground plane, width, height and orientation of a person. However, we can approximate the mask by assuming that all persons have the same height, that they walk upright, and that their width and depth is the same. Although these approximations may seem very rough, they do turn out to be quite effective in practice.

Based on these approximations, the mask $\mathbf{m}$ depends only on the position, $\mathbf{l} = (x, y)$, of the person in the ground plane, which we denote as $\mathbf{m}_\mathbf{l}$. If more than one person is considered, the corresponding mask will contain all pixels that fall inside the bounding boxes of the set of locations $\mathcal{L} = \{\mathbf{l}_1, \ldots, \mathbf{l}_n\}$, which we denote as $\mathbf{m}_\mathcal{L}$. If the projections of two boxes overlap, the mask only contains the relevant pixels once.

The complete likelihood that we should see an image $\mathbf{x}$ and that a number of people should be visible in locations $\mathcal{L}$, is then given by

$$p(\mathbf{x}, \mathcal{L}) = p(\mathcal{L})\, p(\mathbf{x}|\mathcal{L}) \tag{2}$$

$$= p(\mathcal{L})\ [\mathbf{m}_\mathcal{L} \cdot \mathbf{p}_f(\mathbf{x}) + (\mathbf{1} - \mathbf{m}_\mathcal{L}) \cdot \mathbf{p}_b(\mathbf{x})] \tag{3}$$

Here, $p(\mathcal{L})$ indicates the prior probability — before we have seen the image — that a set of people would be located in the locations $\mathcal{L}$. In this work, we have factorised this probability as follows:

$$p(\mathcal{L}) = p(|\mathcal{L}|) \prod_{i=1}^{|\mathcal{L}|} p(\mathbf{l}_i)\, p(\mathbf{l}_i|\mathbf{l}_1 \ldots \mathbf{l}_{i-1}), \tag{4}$$

where we have defined:

$p(|\mathcal{L}|)$ the prior probability that $|\mathcal{L}|$ people are visible in the image. This prior is also used to limit the maximum number of people that the system looks for in an image, by setting it to zero from a certain number onwards.

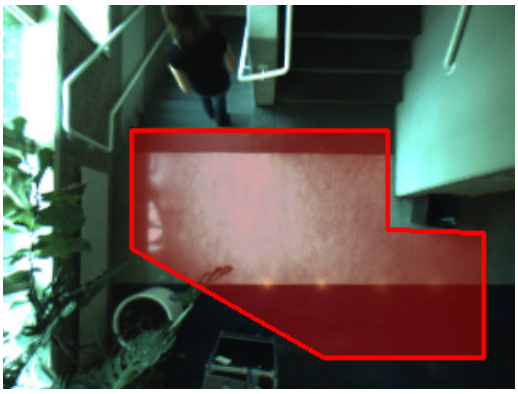


Figure 3: Example of the annotation of the ground plane, used to compute the prior probability that a person can be located at any particular position in the image. Similar regions were annotated for all three cameras.

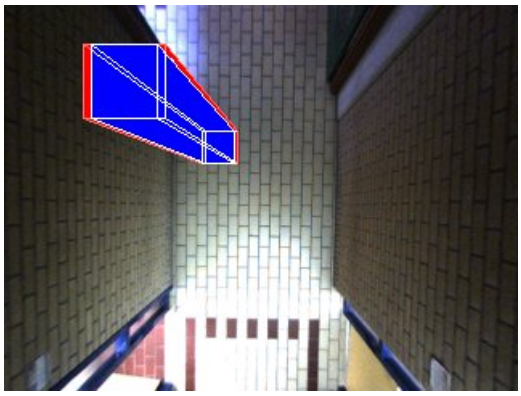


Figure 4: Mask difference due to shifting the latent location by the equivalent distance of one pixel in the ground plane. The red area is classified differently, the blue and white pixels remain foreground and the rest remains background.

$p(\mathbf{l}_i)$ the prior probability that a person is in position $\mathbf{l}_i$. For each camera, we have manually indicated the areas that show the ground plane, as depicted in Figure 3. The prior probability $p(\mathbf{l}_i)$ is set to zero outside this area, and is uniformly distributed over the pixels inside the area. In practice, we set the prior to include only the area where we're interested in tracking people, excluding the edges of the image (as very small mask areas tend to be more sensitive to noisy pixels), and excluding stairs (as our mask computation is not accurate for these).

$p(\mathbf{l}_i|\mathbf{l}_1 \dots \mathbf{l}_{i-1})$
the prior probability that a person should be in location $\mathbf{l}_i$, given that there are people in locations $\mathbf{l}_1 \dots \mathbf{l}_{i-1}$. This is used to model how close together people can be. In this work, this probability is set zero when the distance between the bounding boxes in the ground plane is smaller than half a person's width, and uniformly distributed otherwise. More informative priors could be used to capture the fact that people tend to walk in groups, and rather side by side than in front one of another, but this was not done in this work.

4.2 Background model

We built a separate "eigenbackground" model [15] for each of the cameras by performing Principal Component Analysis (PCA, [7]) using the technique described in [13], which allows us to do PCA without explicitly computing the covariance matrix of the data. A set of 35 training images containing no people were manually selected from each camera's set of recorded images. We kept the mean image and the $N = 3$ eigenvectors with largest eigenvalues as a model of the background. Reconstruction of the background image $\mathbf{b}$ was done by projecting the mean-subtracted new image $\mathbf{x}$ onto those eigenvectors $\mathbf{e}_1 \dots \mathbf{e}_N$, and projecting the result back up into image space:

$$\mathbf{b} = \boldsymbol{\mu} + \sum_{i=1}^{N} \left((\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{e}_i \right) \mathbf{e}_i, \tag{5}$$

where $\boldsymbol{\mu}$ denotes the mean of the background training images.

This is a very fast operation, and it allows the background model to capture most changes in illumination, including shadows from clouds and lights being switched on or off, as well as structural changes in the background that were also present in the training set. Most notably, open and closed doors are captured with this model.

The probability of a background pixel $\mathbf{x}_i$ is modelled, for each HSV component of each pixel independently, as a one-dimensional Gaussian distribution centred around the previously computed background image, and the same variance for all pixels, computed from the training data:

$$p_{b,i}(\mathbf{x}) = \mathcal{N}(\mathbf{x}_i; \mathbf{b}_i, \sigma^2) \tag{6}$$

Notice that this model could easily be extended to an online algorithm, which incorporates background information from new images into the model, by considering only those pixels which are not deemed part of the foreground masks.

4.3 Foreground model

For this work, we did not build person-specific models of appearance. The only information about a person's appearance that was included in the model is, as described above, the generic width and height of a person used to compute the masks. In our experiments, the height was set to 170 cm, and the width was fixed to the distance corresponding to 10 pixels in the ground plane directly below the camera. The probability of a pixel $\mathbf{x}_i$ given that it is foreground, is therefore set to a uniform distribution over the number of possible intensity values ($1/256$ for our 8-bit images). Intuitively, this captures the idea that a foreground object could look like anything, as long as it fills the bounding box reasonably well. This approach makes the model fast to train, since only a small number of training images are required for the background model and no training data is needed for the foreground model; it also makes the detection of people in new images very fast, since we can precompute the masks. Moreover, it makes the model extremely robust against changes in illumination and projected shadows, because the background model is quite good at capturing changes in the environment's illumination, and the foreground is insensitive to it. As long as a person looks sufficiently different from the background, it is detected correctly. Finally, thanks to its foreground-agnostic nature, this model handles artefacts such as motion blurring gracefully.

Foreground segmentation methods decide what pixels are part of the objects before attempting to track the object. This results in misclassified pixels: background pixels which are considered as foreground and are therefore wrongly taken into consideration when tracking, and foreground pixels which are classified as background, and are therefore not contributing any information to the tracking process. In contrast, our method does not create a hard segmentation of the image before tracking the people's position. All pixels contribute to the decision of the person's location, whether they are foreground our background, and whether they are starkly contrasted or not. The result is a much more holistic, more robust, faster method.

5 Inferring the position of multiple people

Based on the likelihood given in Equation (3), we can now infer the probability of a set of person locations $\mathcal{L}$ using Bayes' rule:

$$p(\mathcal{L}|\mathbf{x}) = \frac{p(\mathbf{x}, \mathcal{L})}{\sum_{\mathcal{L}} p(\mathbf{x}, \mathcal{L})}. \tag{7}$$

Similarly, we can compute the probability that the image contains a number n of people, by marginalising out the exact locations and computing the posterior probability of that number of people.

Unfortunately, this computation is intractable. In a low-resolution 320×240 pixel image, there are 76800 locations that can meaningfully be discriminated. If we have a number m of people in the image, we need to consider 76800^m combinations of locations, which is not manageable for even small numbers of people. However, we can reduce the search space of the algorithm as follows:

1. Since the foreground model is the same for all objects, the swapping of any two locations in $\mathcal{L}$ will result in the same likelihood. This allows us to restrict the search space, by not considering such equivalent pairs of locations. In particular, it means that if no location can be found which, when added to a set of locations $\mathcal{L}$, results in a higher likelihood than the original set $\mathcal{L}$, adding even more people to the model is guaranteed to lower the likelihood even further. This allows us to stop searching as soon as the likelihood goes down.

2. When analysing the posterior probability distributions of the locations, these turn out to be very sharply peaked at a few — and more often, a single – location. This is because moving the mask by the equivalent of a single pixel (in the ground plane) results in tens, hundreds or even thousands of pixels being classified differently, as is illustrated in Figure 4. As a consequence, the posterior probability is changed by the product of the difference in probability of this many pixels and is exponentially larger than the difference in probability of a single pixel.

Dataset	Number of frames	Annotation Sq. Dist.	Baseline Number	Baseline Sq. Dist.	Proposed Number	Proposed Sq. Dist.
Set 1	50	1.95	1.18	11.65	0.50	13.43
Set 2	51	1.74	1.19	10.11	0.21	11.63
Set 3	50	2.11	1.54	64.16	0.12	16.30
Set 4	171	2.01	0.92	46.97	0.28	8.49

Table 1: Comparison of detection and position accuracy. Distances are in pixels

3. If two masks do not overlap, their contributions to the likelihood of an observation are independent. They can therefore be calculated once in the first pass, the resulting contribution can be stored and they do not need to be recomputed in later passes.

4. If a foreground mask, in isolation, does not improve the likelihood of the observation, then if considering part of that mask as foreground *would* improve the likelihood, considering the rest of the mask as foreground is guaranteed to decrease the likelihood even more. As a consequence, if a mask does not increase the likelihood in isolation, it is guaranteed not to increase it either when some people have already been found: either it does not overlap with other masks that have already been found to increase the likelihood, and in that case its contribution is not affected, or it does overlap, and in that case its contribution is guaranteed to decrease the likelihood more than it would in isolation.

As a consequence of the above, we can apply the greedy search strategy suggested by Titsias *et al.* [14]. This search strategy was developed for modelling both foreground and background, which is computationally expensive. Adapted to the present model, however, this strategy becomes easily manageable:

1. Compute the likelihood if $\mathcal{L} = \emptyset$, *i.e.*, no person is present in the image.

2. Compute and store the contribution to the likelihood of each possible person location.

3. Find the most likely position of the first person.

4. If this most likely position improves the likelihood, add it to $\mathcal{L}$, otherwise exit the algorithm.

5. Find the next most likely position, only considering positions that did improve the likelihood in isolation. Go to 4.

Since the likelihood is so sharply peaked, we can also reasonably approximate the probability distribution with its modes: *i.e.*, approximate the likelihood of the sets of locations which do not result in the maximal likelihood (except for the permutations mentioned above) with zero. When we do so, the most likely set of locations is close to the expectation of the set of locations, and the most likely number of people is a good approximation for its expectation. Note that in practice, the likelihood is a vanishingly small number due to the product over the individual pixels, and would lead to numerical underflows if implemented directly. Instead, our implementation uses the logarithm of the likelihood, hence avoiding numerical problems.

6 Experiments and Discussion

To evaluate the proposed method, we selected 322 images, organised in four sets, containing a widely varying number of visible people, in different poses and locations.[1] We manually annotated these images with the position in the ground plane of the people present in the image. This annotation was done three times for each image and by different annotators, in order to obtain a measure of the annotation error. We used the mean locations of the annotations as ground truth measurements in the experiments detailed below.

We compared our method to an advanced blob tracking system. This baseline system works as follows: background segmentation is performed using [17], an extension of the adaptive background mixture model proposed in [12], which uses a mixture model of up to 4 Gaussian distributions to model the background. The shadow and highlight detection algorithm proposed in [6] was then applied to remove those artefacts. The resulting foreground pixels were then grouped into blobs, using the connected-component labelling

[1] The images were selected to ensure that a reasonable number of images do indeed contain people.

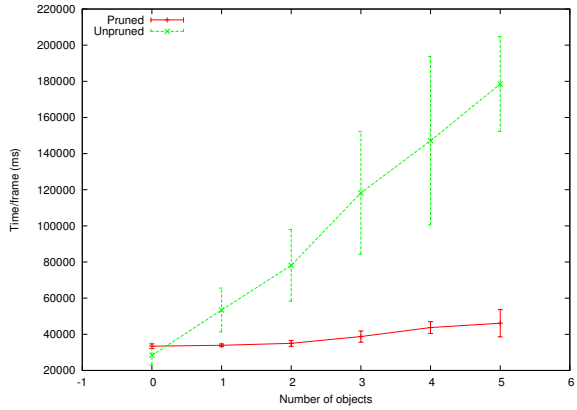


Figure 5: Evolution of the processing speed in function of the number of detected people

algorithm described in [11]. The various parameters of the different modules of this system were independently hand-tuned to obtain optimal results. The position of the object was estimated by computing the centroid of the blob and using the stereoscopic information and camera's calibration to compute its projection on the ground plane. These distances are then converted back to pixels, so as to compensate for the distance from the camera in different areas of the image.

Since neither the annotation nor the blob detection returns any information about the identity of the person, we compared each detected position with the closest annotated position. When more persons were detected than were annotated, the remaining erroneous detections were not included in the computation of the distance error, since in those cases there is nothing to compute the distance with. Instead, we list the average error in the number of detected people separately. The results are listed in Table (1). The first column in this table lists the average squared Euclidean distance (in image pixels) between the mean annotated locations and the individual annotations, *i.e.*, the variance in the annotations. This provides a reference with which to compare the algorithmic results. The average number of misdetections per frame is shown in the column labelled *number* in Table (1), while the squared Euclidean distance between the detected locations and the ground truth is listed in column *Sq. Dist.*.

The first two sets consists of images with up to 7 people (3.4 on average), containing many occlusions, taken from the area shown in Figure 1(a). These images were selected from a short time window, so that the illumination was all but constant. In such circumstances, the major difficulty is in correctly segmenting the people, and detecting the correct number of people. Our method substantially outperforms the baseline with respect to the number of misdetections. It does result in marginally less precise predictions for the locations, but this difference is not statistically significant. Also note that, when the algorithm erroneously detects a person, the average distance to the true position tends to decrease. This is because only the best detections are used in computing this distance, and an erroneous detection has a non-zero probability of being closer to the annotation than any correct detections. The higher the number of wrongly detected people, the more conservative the estimate of the location error therefore becomes.

Sets three and four were taken in much more challenging light conditions, but contain fewer people (up to four, and 2.1 on average). In these circumstances, the major problem is incorrect foreground segmentation. We can see that the baseline method struggles on this data: it makes far more misdetections, and the average location accuracy is much worse. In contrast, our proposed method deals very gracefully with the bad light conditions, because it does not perform foreground segmentation per pixel. The resulting tracking is very robust to varying light conditions.

6.1 Computational performance

We explained in section 5 that we could prune those locations that, when considered in isolation, do not increase the likelihood. To illustrate the importance of this pruning, we performed a test on a set of 646 images, taken at random from our dataset. In this test, two runs were performed: one with pruning and one without. The results are shown in Figure 5. If no pruning is done, the algorithm needs to search the whole state space anew for each new person. In both cases, the complexity is linear in the number of people, thanks to the use of the greedy algorithm. In practice, however, pruning can drastically improve the computation

time of the algorithm.

7 Conclusion

We have proposed a novel and efficient probabilistic model for locating people in video frames. The model relies on an approximate camera calibration (height and angle of the camera), and on prior knowledge of the average size of people. It easily integrates any known prior knowledge about the environment (such as which areas can be walked on), and about typical walking behaviour (such as keeping distances, or walking in groups.) Thanks to its probabilistic nature, this method can also be trivially integrated in advanced probabilistic tracking systems.

It was tested on frames from different cameras in different locations, in challenging light conditions, and was shown to significantly outperform the highly tuned baseline, both in accuracy and in computational performance. In future work, it will be interesting to evaluate how well this method performs for tracking other objects, such as vehicles or animals, and how informative models of motion such as the Kalman filter or particle filters, improve on the results reported here.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.

[2] S. Bahadori, L. Iocchi, G. Leone, D. Nardi, and L. Scozzafava. Real-time people localization and tracking through fixed stereo vision. *Applied Intelligence*, 26(2):83–97, April 2007.

[3] D. Beymer and K. Konolige. Real-time tracking of multiple people using continuous detection. In *IEEE Frame Rate Workshop*, 1999.

[4] Gwenn Englebienne, Tim Van Oosterhout, and Ben J. A. Kröse. Tracking in sparse multi-camera setups using stereo vision. In *Proceedings of the 3rd International Conference on Distributed Smart Cameras*, Como, Italy, September 2009.

[5] I. Haritaoglu, D. Harwood, and L. S. Davis. W 4: real-time surveillance of people and theiractivities. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):809–830, 2000.

[6] Thanarat Horprasert, David Harwood, and Larry S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of the International Conference on Computer Vision*, Greece, September 1999.

[7] I. T. Jolliffe. *Principal component analysis*. Springer, New York, 1986.

[8] Yunqian Ma, Qian Yu, and Isaac Cohen. Target tracking with incomplete detection. *Computer Vision and Image Understanding*, 113(4):580–587, 2009.

[9] Hieu T. Nguyen, Marcel Worring, and Rein van den Boomgaard. Occlusion robust adaptive template tracking. In *ICCV 2001*, volume 1, page 678, 2001.

[10] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE international conference on robotics and automation, 2001. Proceedings 2001 ICRA*, volume 2, 2001.

[11] Linda G. Shapiro and George C. Stockman. *Computer vision*. Prentice Hall, 2001.

[12] Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2, page 252 Vol. 2, 1999.

[13] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 591, page 586–591, 1991.

[14] Christopher K.I. Williams and Michalis K. Titsias. Greedy learning of multiple objects in images using robust statistics and factorial learning. *Neural Computation*, 16(5):1039–1062, May 2004.

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

[16] Tao Zhao, Ram Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1198–1211, 2008.

[17] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the international conference on Pattern Recognition*, volume 2, pages 28–31 Vol.2, 2004.