Appearance Models for Occlusion Handling

Andrew Senior, Arun Hampapur, Ying-Li Tian, Lisa Brown, Sharath Pankanti and Ruud Bolle
{aws,arunh,yltian,lisabr,sharat,bolle}@us.ibm.com
IBM T. J. Watson Research Center,
PO Box 704,
Yorktown Heights, NY 10598

Abstract

Objects in the world exhibit complex interactions. When captured in a video sequence, some interactions manifest themselves as occlusions. A visual tracking system must be able to track objects which are partially or even fully occluded. In this paper we present a method of tracking objects through occlusions using appearance models. These models are used to localize objects during partial occlusions, detect complete occlusions and resolve depth ordering of objects during occlusions. This paper presents a tracking system which successfully deals with complex real world interactions, as demonstrated on the PETS 2001 dataset.

1. Introduction

Real world video sequences capture the complex interactions between objects (people, vehicles, building, trees, *etc.*). In video sequences these interactions result in several challenges to the tracking algorithm. Distinct objects cross paths and cause occlusions. A number of objects may exhibit similar motion, causing difficulties in segmentation. New objects may emerge from existing objects (a person getting out of a car) or existing objects may disappear (a person entering a car or exiting the scene). Maintaining appearance models of objects over time and using them to deal with complex interactions is key to a successful tracking system.

In this paper we present a tracking system which uses appearance models to successfully track objects through complex real world interactions. Section 2 presents a short review of related research. Section 3 presents the overall architecture of the system and its components: background subtraction, high-level tracking and appearance models are discussed in sections 4, 5 and 6 respectively. Section 6 discusses the appearance models. We have developed an interactive tool for generating ground truth using partial tracking result which is discussed in section 9. Section 10 discusses our method for comparing automatic tracking results to the ground truth. Section 11 presents results on the PETS test sequences. We summarize our paper and present future directions in section 12.

2. Related work

A video image will change over time due to object or camera motion, illumination variation, complex occlusion, and other variations. The analysis of appearance changes can be used to detect and track moving objects in video sequences. Many systems have been developed for video surveillance to detect and track people, vehicles and moving objects [5, 13, 15].

Occlusion is a significant problem in moving object detection and tracking. Some previous work does not deal with occlusion at all, or minimizes occlusions by placing the cameras at a high angle, looking down on the plane of motion of the objects [3, 4].

Methods to solve the occlusion problem have been previously presented [1, 2, 5, 8]. Chang *et al.* [1] and Dockstader *et al.* [2] use the fusion of multiple camera inputs to overcome occlusion in multi-object tracking. Khan and Shah [8] presented a system to track people in the presence of occlusion. First, they segmented a person into classes of similar colour using the Expectation Maximization algorithm. Then they used a maximum a posteriori probability approach to track these classes from frame to frame.

Lipton *et al.* [9] describe a simple method based on template matching and temporal consistency via object classification and motion detection. Their method can deal with partial occlusions. Template matching plays a similar role to appearance-based models but does not take into account the variable appearance of the object due to lighting changes, self-occlusions, and other complex 3-dimensional projection effects.

Several methods use Kalman filtering or probabilistic approaches to perform robust tracking which can deal with some instances of occlusion [7, 12, 14] These methods require estimation of prior distributions for modelling motion and appearance. Tao *et al.* [14] use a dynamic layer approach which relies on an appearance model. Their system can deal with partial occlusion of passing vehicles as seen

from above. Isard *et al.* [7] have built a system for tracking people walking by each other in a corridor. Each foreground object is statistically modelled using a generalized cylinder object model and a mixture of Gaussians model based on intensity. Rosales *et al.* [12] present an approach to detect and predict occlusion by using temporal analysis and trajectory prediction. In temporal analysis, a map of the previous segmented and processed frame is used as a possible approximation of the current connected elements. In trajectory prediction, an extended Kalman filter provides an estimate of each object's position and velocity. These methods make several assumptions about the types of objects in the scene and their shape and motion characteristics.

The works most closely related to this paper are those of Haritaoğlu *et al.* [5] and Roh *et al.* [11] since they use appearance models to handle occlusion problem. The former combine the gray-scale texture appearance and shape information of a person together in a 2D dynamic template, but do not such appearance information in analyzing multi-people groups. Roh *et al.* use an appearance model based on temporal colour to track multiple people in the presence of occlusion. They use temporal colour features which combine colour values with associated weights. The weights are determined by the size, duration, frequency, and adjacency of a colour object.

3. Tracking system architecture

In this paper we describe a new visual tracking system designed to track independently moving objects, and using the output of a conventional video camera. Figure 1 shows the structure of the tracking system.

The input video sequence is used to estimate a background model, which is then used to perform background subtraction, as described in section 4. The resulting foreground regions form the raw material of a two-tiered tracking system.

The first tracking process associates foreground regions in consecutive frames to construct hypothesized tracks. The second tier of tracking uses appearance models to resolve ambiguities in these tracks that occur due to object interactions and result in tracks corresponding to independently moving objects.

A final operation filters the tracks to remove tracks which are invalid artefacts of the track construction process, and saves the track information (the centroids of the objects at each time frame) in the PETS XML file format.

In this paper we describe results using the PETS 2001 evaluation dataset 1, camera 1. For reasons of speed and storage economy, we have chosen to process the video at half resolution. The sytem operates on AVI video files (Cinepak compressed) generated from the distributed JPEG images. Naturally, higher accuracies and reliability are to be expected from processing the video at full size and without compression artefacts.

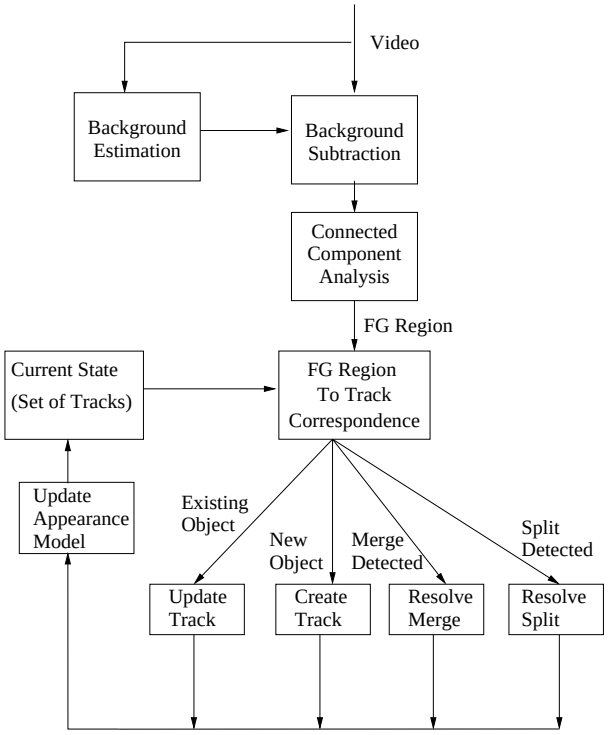

Figure 1: Block diagram of the tracking system

4. Background estimation and subtraction

The background subtraction approach presented here is similar to that taken by Horprasert *et al.* [6] and is an attempt to make the background subtraction robust to illumination changes. The background is modelled statistically at each pixel. The estimation process computes the brightness distortion and colour distortion in RGB colour space. Each pixel i is modelled by a 4-tuple (E_i, s_i, a_i, b_i), where E_i is a vector with the means of the pixel's red, green, and blue components computed over N background frames; s_i is a vector with the standard deviations of the colour values; a_i is the variation of the brightness distortion; and b_i is the variation of the chromaticity distortion. We have also developed an active background estimation method that can deal with moving objects in the frame. First, we calculate image difference over three frames to detect the moving objects. Then the statistical background model is constructed, excluding these moving object regions.

By comparing the difference between the background image and the current image, a given pixel is classified into one of four categories: original background, shaded background or shadow, highlighted background, and foreground

objects. thresholds are calculated automatically, details can be found in the original paper [6]. Finally, a morphology step is applied to remove small isolated spots and fill holes in the foreground image. The current algorithm works reasonably well indoors and outside without adapting the background after initial estimation, but we are currently adaptation to the system.

5. High-level tracking

The foreground regions of each frame are grouped into connected components. A size filter is used to remove small components. Each foreground component is described by a bounding box and an image mask, which indicates those pixels in the bounding box that belong to the foreground. For each successive frame, the correspondence process attempts to associate the foreground regions with one of the existing tracks. This is achieved by constructing a distance matrix showing the distance between each of the foreground regions and all the currently active tracks. We use a *bounding box distance measure*, as shown in figure 2. The distance between bounding boxes A and B (figure 2, left) is the lower of the distance from the centroid, C_a, of A to the closest point on B or from the centroid, C_b, of B to the closest point on A. In either centroid lies within the other bounding box (figure 2, right), the distance is zero. The motivation for using the bounding box distance as opposed to Euclidean distance between the centroids is the large jump in the Euclidean distance when two bounding boxes (objects) merge or split. A time distance between the observations is also added in to penalize tracks for which no evidence has been seen for some time.

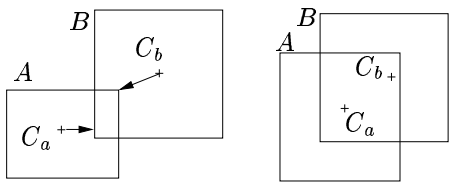

Figure 2: Bounding box distance measure

The distance matrix is then binarized, by thresholding, resulting in a correspondence matrix associating tracks with foreground regions. The analysis of the correspondence matrix produces four possible results as shown in figure 1: existing object, new object, merge detected and split detected.

For well-separated moving objects, the correspondence matrix (rows correspond to existing tracks and columns to foreground regions in the current segmentation) will have at most one non-zero element in each row or column — associating each track with one foreground region and each foreground region with one track, respectively. Columns with all zero elements represent new objects in the scene which are not associated with any track, and result in the creation of a new track. Rows with all zero elements represent tracks that are no longer visible (because they left the scene, or were generated because of artefacts of the background subtraction).

In the case of merging objects, two or more tracks will correspond to one foreground region, *i.e.* a column in the correspondence matrix will have more than one non-zero entry. When objects split, for example when people in a group walk away from each other, a single track will correspond to multiple foreground regions, resulting in more than one non-zero element in a row of the correspondence matrix. When a single track corresponds to more than one bounding box, all those bounding boxes are merged together, and processing proceeds. If two objects tracked as one do separate, the parts continue to be tracked as one until they separate sufficiently that both bounding boxes do not correspond to the track, and a new track is created.

Once a track is created, an appearance model of the object is initialized. This appearance model is adapted every time the same object is tracked into the next frame. On the detection of object merges and splits, the appearance model is used to resolve the ambiguity. A detailed discussion of the appearance model and its application to occlusion handling is presented in the following section.

6. Appearance-based tracking

To resolve more complex structures in the track lattice produced by the bounding box tracking, we use appearance-based modelling. Here, for each track we build an appearance model, showing how the object appears in the image. The appearance model is an RGB colour model with a probability mask similar to that used by Haritaoğlu *et al.* [5]. As the track is constructed, the foreground pixels associated with it are added into the appearance model. The new information is blended in with an update fraction (typically 0.05) so that new information is added slowly and old information is gradually forgotten. This allows the model to accommodate to gradual changes such as scale and orientation changes, but retain some information about the appearance of pixels that appear intermittently, as in the legs or arms of a moving person. The probability mask part is also updated to reflect the observation probability of a given pixel. Figure 3 shows the appearance model for a van from the PETS data at several different frames.

These appearance models are used to solve a number of problems, including improved localization during tracking, track correspondence and occlusion resolution.

Given a one-to-one track-to-foreground-region correspondence, we use the appearance model to provide improved localization of the tracked object. The background subtraction is necessarily noisy, and the additional layers of morphology increase the noise in the localization of the ob-

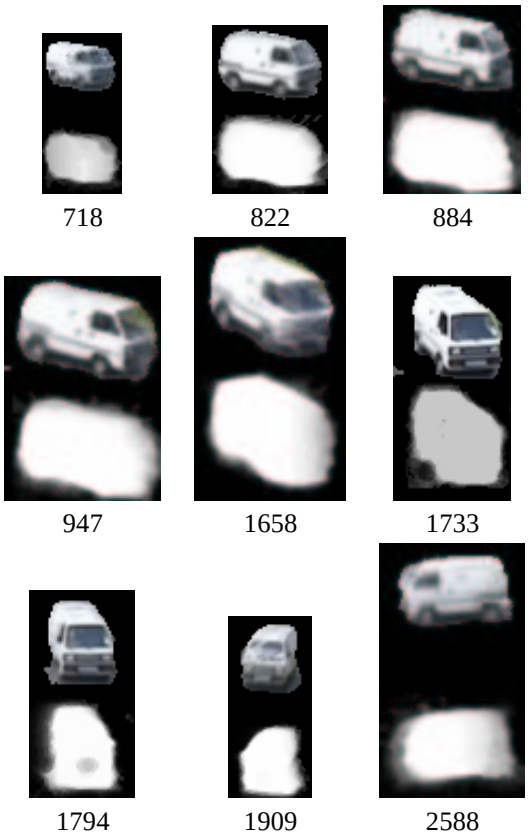

Figure 3: The evolution of an appearance model. In each figure, the upper image shows the appearance for pixels where observation probability is greater than 0.5. The lower shows the probability mask as grey levels, with white being 1.) The frame numbers at which these images represent the models are given, showing the progressive accommodation of the model to slow changes in scale and orientation.

jects, by adding some background pixels to a foreground region, and removing extremities. The appearance model however, has an accumulation of information about the appearance of the pixels of an object and can be correlated with the image to give a more accurate estimate of the centroid of the object. The accumulated Euclidean RGB distance is minimized over a small search region and the point with the lowest distance taken as the object's location. The process could be carried out to sub-pixel accuracy, but the pixel level is sufficient for our tracking.

When two tracks merge into a single foreground region, we use the appearance models for the tracks to estimate the separate objects' locations and their depth ordering.

This is done by the following operations, illustrated in figures 4&5:

1. Using a first-order model, the centroid locations of the objects are predicted.
2. For a new merge, with no estimate of the depth-ordering, each object is correlated with the image in the predicted position, to find the location of best-fit.
3. Given this best-fit location, the 'disputed' pixels — those which have non-zero probabilities in more than one of the appearance model probability masks — are classified using a maximum likelihood classifier with a simple spherical Gaussian RGB model, determining which model was most likely to have produced them. Figures 4c & 5c show the results of such classifications.
4. Objects are ordered so that those which are assigned fewer disputed pixels are given greater depth. Those with few visible pixels are marked as occluded.
5. All pixels are reclassified, with disputed pixels being assigned to the foremost object which overlapped them.

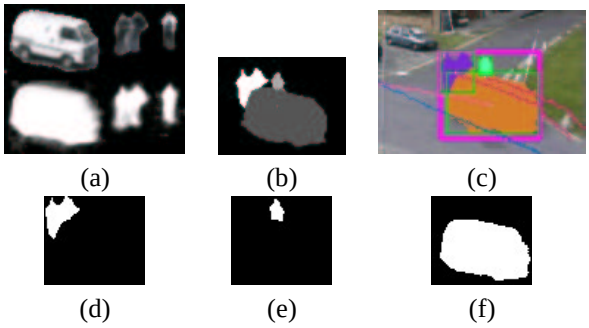

Figure 4: An occlusion resolution (Frame 921 of dataset 1, camera 1). (a) shows three appearance models for tracks converging in a single region. (b) shows the pixels of a single foreground region, classified independently as to which of the models they belong to. (d,e,f) show the pixels finally allocated to each track, and (c) shows the regions overlaid on the original frame, with the original foreground region bounding box (thick box), the new bounding boxes (thin boxes) and the tracks of the object centroids.

On subsequent frames, the localization step is carried out in depth order, with the foremost objects being fitted first, and pixels which match their appearance model being ignored in the localization of 'deeper' objects, as they are considered occluded. After the localization and occlusion resolution, the appearance model for each track is updated using only those pixels assigned to that track.

Because of failures in the background subtraction, particularly in the presence of lighting variation, some spurious foreground regions are generated, which result in tracks. However mosto of these are filtered out with rules detecting their short life or the fact that the appearance model created in one frame fails to explain the 'foreground' pixels in

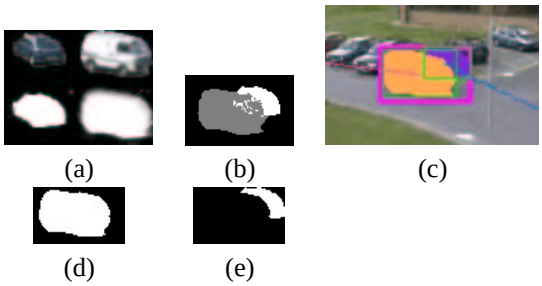

Figure 5: An occlusion resolution (Frame 825 of dataset 1, camera 1). (a) appearance models. (b) independently classified foreground region pixels as to which of the models they belong to. (d,e) the pixels allocated to each track, and (c) the regions overlaid on the original frame.

subsequent frames. An additional rule is used to prune out tracks which do not move. These are considered to be static objects whose appearance varies, such as moving trees and reflections of sky.

7. Multi-object segmentation

The appearance models can also be used to split complex objects. While the background subtractions yields complex, noisy foreground regions, the blending process of the model update allows finer structure in objects to be observed. The principal way in which this structure is used in the current system is to look for objects which are actually groups of people. These can be detected in the representation if the people are walking sufficiently far apart that background pixels are visible between them. These are evidenced in the probability mask, and can be detected by observing the vertical projection of the probability mask. We look for minima in this projection which are sufficiently low and divide sufficiently high maxima. When such a minimum is detected, the track can be divided into the two component objects, though here we choose to track the multi-person object and flag its identity.

Figure 6: The appearance model for a group of people.

8. Object classification

For the understanding of video it is important to label the objects in the scene. For the limited variety of objects in the test data processed here, we have written a simple rules-based classifier. Objects are initially classified by size and shape. We classify objects as: Single Person, Multiple People, Vehicle, and Other. For each object we find the area, the length of the contour, and the length and orientation of the principal axes. We compute the 'dispersedness', which is the ratio of the perimeter squared to the area. Dispersedness has been shown to be a useful cue to distinguish 2D image objects of one or more people from those of individual vehicles [9]. For each 2D image object, we also determine which principal axis is most nearly vertical and compute the ratio of the more-nearly horizontal axis length to the more-nearly vertical axis length. This ratio, r, is used to distinguish a foreground region of a single person from one representing multiple people since a single person's image is typically significantly taller than it is wide while a multi-person blob grows in width with the number of visible people. From these principles, we have designed the ad-hoc, rule-based classification shown in figure 7. In addition, we use temporal consistency to improve robustness so a cleanly tracked object, which is occasionally misclassified, can use its classification history to improve the results.

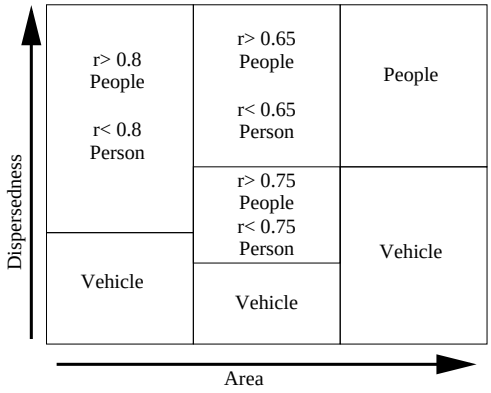

Figure 7: The classification rules for a foreground region. r is the horizontal-to-vertical principal axis length ratio.

9. Ground truth generation

The tracking results were evaluated by comparing them with ground truth. This section overviews the ground truth generation process. A semi-automatic interactive tool was developed to aid the user in generating ground truth. The ground truth marking (GTM) tool has the following four major components: (i) iterative frame acquisition and advancement mechanism; (ii) automatic object detection; (iii) automatic object tracking; (iv) visualization; (v) refinement. After each frame of video is acquired, the object detection component automatically determines the foreground objects. The foreground objects detected in frame n are related to those frame $n-1$ by the object tracking component. At any frame n, all the existing tracks up to frame n and the bounding boxes detected in frame n are displayed by

the visualization component. The editing component allows the user to either (a) accept the results of the object detection/tracking components, (b) modify (insert/delete/update) the detected components, (c) partially/totally modify (create, associate, and dissociate) track relationships among the objects detected in frame $n-1$ and those in frame n. Once the user is satisfied with the object detection/tracking results at frame n, she can proceed to the next frame.

Generating object position and track ground truth for video sequences is a very labour intensive process. In order to alleviate the tedium of the ground truth determination, GTM allows for *sparse* ground truth marking mode. In this mode, the user need not mark all the frames of the video but only a subset thereof. The intermediate object detection and tracking results are interpolated for the skipped frames using linear interpolation. The rate, t, of frame subsampling is user-adaptable and can be changed dynamically from frame to frame.

The basic premise in visual determination of the ground truth is that the humans are perfect vision machines. Although we refer to the visually determined object position and tracks as "*the* ground truth", it should be emphasized that there is a significant *subjective* component of human judgment involved in the process. The objects to be tracked in many instances were very small (*e.g.* few pixels) and exhibited poor contrast against the surrounding background. When several objects came very close to each other, determination of the exact boundary of each object was not easy. Further, since the judgments about of the object location were based on visual observation of a single (current) frame, the motion information (which is a significant clue for determining the object boundary) was not available for marking the ground truth information. Finally, limited human ability to exert sustained attention to mark minute details frame after frame tends to introduce errors in the ground truth data. Because of the monotonous nature of the ground truth determination, there may be an inclination to acceptance of the ground truth proposed by the (automatic) component of the GTM interface. Consequently, the resultant ground truth results may be biased towards the algorithms used in the automatic component of the GTM recipe. Perhaps some of the subjectiveness of the ground truth data can be assessed by juxtaposing independently visually marked tracks obtained from different individuals and from different GTM interfaces. For the purpose of this study, we assume that the visually marked ground truth data is predominantly error free.

10. Performance metrics

Given a ground truth labelling of a sequence, this section presents the method used for comparison of the ground truth with tracking results to evaluate the performance. The approach presented here is similar to the approach presented by Pingali and Segen [10]. Given two sets of tracks, a correspondence between the two sets needs to be established before the individual tracks can be compared to each other. Let N_g be the number of tracks in the ground truth and N_r be the number of tracks in the results. Correspondence is established by minimizing the distance between individual tracks. The following distance measure is used, evaluated for frames when both tracks exist:

$$D_T(T1,T2) = \frac{1}{N_{12}^2} \sum_{i:\exists T1(t_i)\ \&\ \exists T2(t_i)} \sqrt{d_{\mathbf{x}}^2(i) + d_{\mathbf{v}}^2(i)} \quad (1)$$

$$d_{\mathbf{x}}(i) = \mid \mathbf{x}_1(i) - \mathbf{x}_2(i) \mid \quad (2)$$

$$d_{\mathbf{v}}(i) = \mid \mathbf{v}_1(i) - \mathbf{v}_2(i) \mid \quad (3)$$

Where N_{12} is the number of points in both tracks $T1$ and $T2$, $\mathbf{x}_k(i)$ is the centroid and $\mathbf{v}_k(i)$ is the velocity of object k at time t_i. Thus the distance between two tracks increases with the distance between the centroids and the difference in velocities. The distance is inversely proportional to the length for which both tracks exist — so tracks which have many frames in common will have low distances. An $N_g \times N_r$ distance matrix is constructed using the track distance measure D_T. Track correspondence is established by thresholding this matrix. Each track in the ground truth can be assigned one or more tracks from the results. This accommodates fragmented tracks. Once the correspondence between the ground truth and the result tracks are established, the following error measures are computed between the corresponding tracks.

- Object centroid position error: Objects in the ground truth are represented as bounding boxes. The object centroid position error is approximated by the distance between the centroids of the bounding boxes of ground truth and the results. This error measure is useful in determining how close the automatic tracking is to the actual position of the object.

- Object area error: Here again, the object area is approximated by the area of the bounding box. The bounding box area will be very different from the actual object area. However, given the impracticality of manually identifying the boundary of the object in thousands of frames, the bounding box area error is a reasonable measure of the quality of the segmentation.

- Object detection lag: This is the difference in time between when the ground truth identified a new object versus the tracking algorithm.

- Track incompleteness factor: This measures how well the automatic track covers the ground truth:

$$\frac{F_{nf} + F_{pf}}{T_i}$$

where, F_{nf} is the false negative frame count, *i.e.* the number of frames that are missing from the result track. F_{pf} is the false positive frame count, *i.e.* the number of frames that are reported in the result which are not present in the ground truth and T_i is the number frames present in both the results and the ground truth.

- Track error rates: These include the false positive rate f_p and the false negative rate f_n as ratios of numbers of tracks:

$$f_p = \frac{\text{Results without corresponding ground truth}}{\text{Total number of ground truth tracks}} \quad (4)$$

$$f_n = \frac{\text{Ground truth without corresponding result}}{\text{Total number of ground truth tracks}} \quad (5)$$

- Object type error: This counts the number of tracks for which our classification (person/car) was incorrect.

11. Experimental results

The goal of our effort was to develop a tracking system for handling occlusion. Given this focus, we report results only on PETS test dataset 1, camera 1. The current version of our system does not support continuous background estimation and hence we do not report results on the remaining sequences which have significant lighting variations. Given the labour intensive nature of the ground truth generation, we have only generated ground truth up to frame 841. Table 1 shows the various performance metrics for these frames.

Of the seven correct tracks, four are correctly detected, and the remaining three (three people walking together) are merged into a single track, though we do detect that it is several people. This accounts for the majority of the position error, since this result track is compared to each of the three ground truth tracks. No incorrect tracks are detected, though in the complete sequence, five spuious tracks are generated by failures in the background subtraction which are accumulated into tracks. The bounding box area measure is as yet largely meaningless since the bounding boxes in the results are only crude approximations of the object bounding boxes, subject to the vagaries of the background subtraction and morphology. The detection lag is small, showing that the system detects objects nearly as quickly as the human ground truther.

12. Summary and conclusions

We have written a computer system capable of tracking moving objects in video, suitable for understanding moderately complex interactions of people and vehicles, as seen in the PETS 2001 data sets. We believe that for the sequence on which we have concentrated our efforts, the tracks produced are accurate. The two tier approach proposed in the paper successfully tracks through all the occlusions in the dataset. The high level bounding box association is sufficient to handle isolated object tracking. At object interac-

	Dataset 1, Camera 1
Track error f_p	8/7
Track error f_n	2/7
Average position error	5.51
Average area error	-346
Average detection lag	1.71
Average track incompleteness	0.12
Object type error	0

Table 1: Performance Measures for Dataset 1, Camera 1

Figure 8: A comparison of estimated tracks (black) with ground truth positions (white), for two tracks superimposed on a mid-sequence frame showing the two objects.

Figure 9: An image showing all the tracks detected by the system for dataset 1, camera 1, overlaid on a background image.

tions, the appearance model is very effective in segmenting and localizing the individual objects and successfully handles the interactions.

To evaluate the system, we have designed and built a ground truthing tool and carried out preliminary evaluation of our results in comparison to the ground truth. The attempt to ground truth the data and use it for performance evaluation lead to the following insights. The most important aspect of the ground truth is at object interactions. Thus ground truth can be generated at varying resolutions through a sequence, coarse resolutions for isolated object paths and high resolution at object interations. The tool we designed allows for this variation.

13. Future work

The implementation of the appearance models holds much scope for future investigation. A more complex model, for instance storing colour covariances or even multimodal distributions for each pixel would allow more robust modelling, but the models as described seem to be adequate for the current task. The background subtraction algorithm is currently not adaptive, and so begins to fail for long sequences with varying lighting conditions. Continuous updating of background regions will improve its robustness to such situations. The system must also operate in real-time to be applicable to real-world tracking problems. Currently the background subtraction works at about 9 fps and the subsequent processing takes a similar amount of time. Without further optimization, the system should run on live data by dropping frames, but we have not tested the system in this mode.

Acknowledgments

We would like to thank Ismail Haritaoğlu of IBM Almaden Research for providing the background estimation and subtraction code.

References

[1] Ting-Hsun Chang, Shaogang Gong, and Eng-Jon Ong. Tracking multiple people under occlusion using multiple cameras. In *Proc. 11th British Machine Vision Conference*, 2000.

[2] S.L. Dockstader and A.M. Tekalp. Multiple camera fusion for multi-object tracking. In *Proc. IEEE Workshop on Multi-Object Tracking*, pages 95–102, 2001.

[3] A. F. Bobick *et al.* The KidsRoom: A perceptually-based interactive and immersive story enviroment. In *Teleoperators and Virtual Environment*, volume 8, pages 367–391, 1999.

[4] W. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Conference on Computer Vision and Pattern Recognition*, pages 22–29, 1998.

[5] I. Haritaoğlu, D. Harwood, and L. S. Davis. W^4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[6] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV'99 Frame-Rate Workshop*, 1999.

[7] M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *International Conf. on Computer Vision*, volume 1, page 111, 2001.

[8] S. Khan and M. Shah. Tracking people in presence of occlusion. In *Asian Conference on Computer Vision*, 2000.

[9] A. Lipton, H. Fuyiyoshi, and R. Patil. Moving target classification and tracking from real-time video. In *Proc. Fourth IEEE Workshop on Applications of Computer Vision*, 1998.

[10] G. Pingali and J. Segen. Performance evaluation of people tracking systems. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 33–38, 1996.

[11] Hyungki Roh, Seonghoon Kang, and Seong-Whan Lee. Multiple people tracking using an appearance model based on temporal color. In *Proc. International Conference on Pattern Recognition*, volume 4, pages 643–646, 2000.

[12] Romer Rosales and Stan Sclaroff. Improved tracking of multiple humans with trajectory prediction and occlusion modelling. In *IEEE CVPR Workshop on the Interpretation of Visual Motion*, 1998.

[13] J. Segen and G. Pingali. A camera-based system for tracking people in real time. In *Proc. International Conference on Pattern Recognition*, pages 63–67, 1996.

[14] H. Tao, H. Sawhney, and R. Kumar. Dynamic layer representation with applications to tracking. In *Proc. International Conference on Pattern Recognition*, 2000.

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