Available online at www.sciencedirect.com
SCIENCE DIRECT
Image and Vision Computing 24 (2006) 1233–1243
image AND vision COMPUTING
www.elsevier.com/locate/imavis

Appearance models for occlusion handling

Andrew Senior*, Arun Hampapur, Ying-Li Tian, Lisa Brown, Sharath Pankanti, Ruud Bolle

IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA

Received 7 April 2003; received in revised form 6 June 2005; accepted 7 June 2005

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.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Probabilistic color appearance models; Visual vehicle and people tracking; Occlusion resolution; Moving object segmentation; Surveillance

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 is necessary for a visual tracking system to be able to model and understand such complex interactions.

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 4–6, respectively. We have developed an interactive tool for generating ground truth using partial tracking results, 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

Visually tracking multiple objects involves estimating 2D or 3D trajectories and maintaining identity through occlusion. One solution to the occlusion problem is to avoid it altogether by placing cameras overhead, looking down on the plane of motion of the objects [1–3] or to use the fusion of multiple cameras to determine depth [4,5].

In the case investigated here—of static monocular cameras viewing moving vehicles and people from a side view—occlusion is a significant problem. It involves correctly segmenting visually merged objects, i.e. during occlusion, and specifying depth layering. An useful categorization of the work in this field is based on the complexity of the model used to track objects over time. As model complexity increases, stronger assumptions about the object are invoked. In general, as the complexity of the models increases, more constraints are imposed, better

* Corresponding author.
E-mail addresses: aws@us.ibm.com (A. Senior), arunh@us.ibm.com (A. Hampapur), yltian@us.ibm.com (Y.-L. Tian), lisabr@us.ibm.com (L. Brown), sharat@us.ibm.com (S. Pankanti), bolle@us.ibm.com (R. Bolle).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.06.007

Table 1
Performance measures for Dataset 1, Camera 1

Multi-person tracking models	Constraints and limitations
Simple cues: color histogram, temporal consistency, shape features [6–8]	Cannot perform segmentation during occlusion, limited identity maintenance, good computational speed performance
Appearance based: evolving image information [9,10,20]	Better identity maintenance but not robust to 3D object viewpoint changes, similar textured, colored objects; simplistic occlusion rules
3D scene model and 2D human properties [11,12,14]	Requires camera calibration and input of scene information; improves reliability of constraints when correct but makes assumptions about normative human height stance, and behavior
2D human appearance model: active shape model, 2D body configurations, 2D clothing blobs [15–17]	Requires prior training, sometimes requires learning individual appearances, assumes people wearing clothing of different colors with uniform colors/textures
2d human temporal model: motion templates of people walking, higher order dynamics [14,18,19]	Assumes people are actively walking, moving in a periodic fashion or assumes second order dynamics

performance is achieved—when the appropriate conditions are met—but general robustness, applicability and real-time speed are more difficult to attain. Table 1 gives examples of multi-person tracking methods for progressively more complex tracking models. Several prototype projects are listed for each class of method. The right-hand column describes the constraints and limitations for each class. A few of the projects are listed in more than one category because they integrate model information from more than one class. In order to achieve the generality of simple methods and the accuracy of refined models, requires spanning a range of model complexity. In almost all of these approaches, background subtraction is the first step, and plays an important role including adapting to scene changes and removing shadows and reflections.

At the top of Table 1 are the simplest methods that rely on basic cues such as color histograms and recent temporal history of each object. These systems have achieved a high degree of robustness and real-time performance but are limited in the accuracy with which they can perform trajectory estimation and identity maintenance through occlusion. Each object is represented by a set of features. In [Roh 00] each object/person is represented by two or three colors temporally weighted depending on the size, duration and frequency of its appearance. In [6,7] each object/person is represented by a temporally updated color histogram. In [6], the histogram intersection method is used to determine the identity of an object and the histograms are used directly to compute the posterior probability of a pixel belonging to each object during an occlusion. In [7], people are tracked using the similarity of their color histograms from frame to frame using a mean shift iteration approach. To segment individuals from each other, temporal information is used.

Appearance-based systems have improved the performance of tracking and identity maintenance through occlusion but are still limited to simple interactions and scene conditions. Appearance-based systems, by our definition, maintain information about each pixel in an evolving image-based model for each moving object/person. In the simplest case, this is just an image template. For example in [8], people and vehicles are classified using shape characteristics then tracked by template matching combined with temporal consistency information such as recent proximity and similar classification. Their method can deal with partial occlusion but does not take into account the variable appearance of objects due to lighting changes, self-occlusions, and other complex 3-dimensional projection effects. The work most closely related to this paper is that of [9]. Their system combines gray-scale texture appearance and shape information of a person together in a 2D dynamic template, but does not use appearance information in analyzing multi-people groups. In our system, we use a color appearance model (thereby incorporating texture, shape, temporal and occlusion history) and are able to successfully perform identity maintenance in a wide range of circumstances but our system is still vulnerable to misclassification for similarly colored/textured objects particularly for complex non-rigid 3D objects when they interact, such as people, since their appearance depends on viewpoint and is often complicated by shadows and scene geometry.

As the complexity of tracking models, the methods become progressively more sophisticated at performing identity maintenance but at the expense of greater constraints. For example, if it is possible to calibrate the camera and manually input specific information regarding the scene, then systems can be designed which take advantage of knowledge of where people will enter/exit the scene, where the ground plane is, where occluding objects occur, and the typical height in the image a person will occupy. This process is referred to as ‘closed-world tracking’ in [10] and was applied to tracking children in an interactive narrative play space. The contextual information is exploited to adaptively select and weight image features used for correspondence. In [11] a scene model is created, spatially specifying short term occluding objects such as a street sign in the foreground, long term occluding objects such as a building, and bordering occlusion due to the limits of the camera field of view. In [12] static foreground occluding objects are actively inferred while the system is running without prior scene information. In [13] not only is

the expected height of a person used based on the ground plane location, but also the time of day is utilized to calculate the angle of the sun and estimate the size and location of potential shadows.

Several systems make assumptions regarding the composition of a typical person in a scene. For example, it is frequently assumed that each person is composed of a small number of similarly colored/textured blobs, such as their shirt and pants. One of the earliest works to exploit this type of assumption was [14]. In their work, a single person is tracked in real-time. The human is modeled as a connected set of blobs. Each blob has a spatial and color distribution and a support map indicating which pixels are members of the blob. Body parts (heads and hands) are tracked through occlusions using simple Newtonian dynamics and maximum likelihood estimations. Several researchers have extended this work to tracking multiple people during occlusion. In [15] each person is segmented into classes of similar color using the Expectation Maximization algorithm. Then a maximum a posteriori probability approach is used to track these classes from frame to frame. In [16] the appearance of each person is modeled as a combination of regions; each region has a color distribution represented non-parametrically. Each region also has a vertical height distribution and a horizontal location distribution. Assuming the probability of each color is independent of location for each blob (i.e. each region can have multiple colors but their relative positions are independent) and the vertical location of the blob is independent of the horizontal position, the system computes the product of the probabilities to estimate the best arrangement of blobs (i.e., maximum likelihood estimation) and therefore the relative location of the people in the scene. These assumptions can be powerful constraints that reduce the search space for multi-person tracking but inevitably they also reduce the range of applicability. Although these constraints improve performance, there are always natural exceptions and environmental conditions which will cause these systems to fail. Examples include peculiar postures and transactions involving objects.

The performance of tracking can also be improved by the use of temporal or motion constraints. All systems assume proximity, most methods employ simplistic first order dynamics, but a more recent generation of systems is assuming higher order dynamics and specific periodicities which can be attributed to people. [17] presents 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. [18] have built a system for tracking people walking by each other in a corridor. Each foreground object is statistically modeled using a generalized cylinder object model and a mixture of Gaussians model based on intensity. A simple particle filter (often called condensation) is used to perform joint inference on both the number of objects present and their configurations. Assumptions about motion include modeling translational dynamics as damped constant velocity plus Gaussian noise and a turn parameter that encodes how much the person (represented by a generalized cylinder) has turned away from the camera. [13] track multiple humans using a Kalman filter with explicit handling of occlusion. Proper segmentation is verified by 'walking recognition'. This is implemented via motion templates (based on prior data of walking phases) and temporal integration. The basis for walking recognition relies on the observation that the functional walking gaits of different people do not exhibit significant dynamical time warping, i.e. the phases should correspond linearly to the phases of the average walker. However, in practice, people start and stop, are not always walking, and it is difficult to find phase correspondence if the viewed in the same direction as the walking motion.

All of the systems mentioned here are prototypes. Each project usually evaluates its own performance based on a small number of video sequences. In most cases, these sequences are produced by the investigators and typical results are primarily a small number of visual examples. Hence it is very difficult to compare the capabilities of these methods or quantify their accuracy or robustness. However, in general these methods are brittle and will fail if their large number of stated and unstated assumptions is not met. The IEEE Workshop on Performance Evaluation of Tracking and Surveillance is an attempt to address some of these issues by making a public dataset widely available and inviting researchers to compare the performance of their algorithms. However, there is still a need to provide ground truth evaluation of predefined performance criteria.

3. Tracking system architecture

In this paper, we describe a new visual tracking system designed to track independently moving objects using the output of a conventional video camera. Fig. 1 shows the a schematic of the principal components 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,

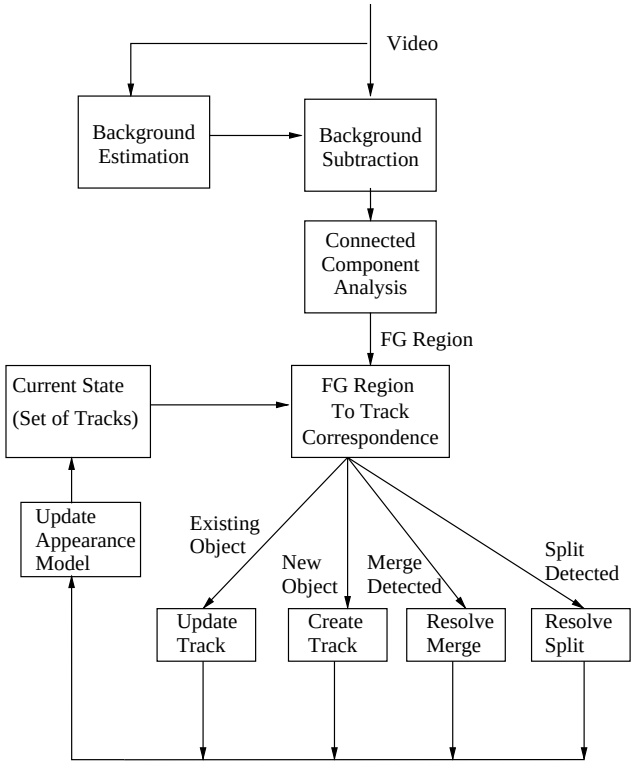

Fig. 1. Block diagram of the tracking system.

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.

4. Background estimation and subtraction

The background subtraction approach presented here is based on that taken by Horprasert et al. [19] and is an attempt to make the background subtraction robust to illumination changes. The background is modeled statistically at each pixel. The estimation process computes the brightness distortion and color distortion in RGB color space. Each pixel i is modeled 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 color values; a_i is the variation of the brightness distortion; and b_i is the variation of the chromaticity distortion.

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. The pixels in the foreground objects, are passed to the next stage (Section 5, and the remaining categories are grouped together as background pixels. The categorization thresholds are calculated automatically; details can be found in the original paper [19]. Finally, isolated pixels are removed and then a morphological closing operator is applied to join nearby foreground pixels.

We have also developed an active background estimation method that can deal with objects moving in the training images (except the first). The first frame is stored as a prototype background image, and differenced with subsequent training frames—areas of significant difference being the moving objects. When the statistical background model is constructed, these moving object regions are excluded from the calculations. To handle variations in illumination that have not been seen in the training set, we have also added a further two modifications to the background subtraction algorithm that operate when running on test sequences. The first is an overall gain control that applies a global scaling factor to the pixel intensities before comparing them to the stored means. The scale factor is calculated on the non-foreground regions of the previous image, under the assumption that lighting changes between adjacent frames are small. Further, background adaptation is employed by blending in the pixel values of current non-foreground regions, thus slowly learning local changes in appearance not attributable to moving objects. These processes reduce the sensitivity of the algorithm to the lighting changes seen at the end of sequence 1, and throughout sequence 2.

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 and an image mask, which indicates those pixels in the bounding box that belong to the foreground. The set of foreground pixels is called $\mathcal{F}$. For each successive frame, the correspondence process attempts to associate each foreground region 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 Fig. 2. The distance between bounding boxes A and B (Fig. 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. If either centroid lies within the other bounding box (Fig. 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

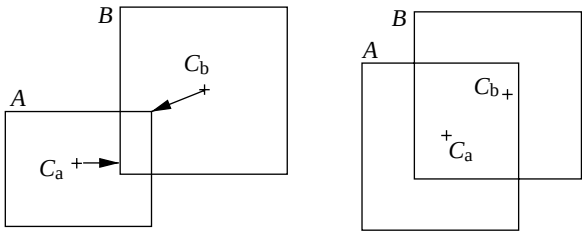

Fig. 2. Bounding box distance measure.

(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.

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 Fig. 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 hitherto tracked as one should 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, 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 Section 6.

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, most 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 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.

6. Appearance-based tracking

To resolve more complex structures in the track lattice produced by the bounding box tracking, we use appearance-based modeling. Here, for each track we build an appearance model, showing how the object appears in the image. The appearance model is an RGB color model with an associated probability mask. The color model, $M_{\mathrm{RGB}}(\mathbf{x})$, shows the appearance of each pixel of an object, and the probability mask, $P_c(\mathbf{x})$, records the likelihood of the object being observed at that pixel. For simplicity of notation, the coordinates $\mathbf{x}$ are assumed to be in image coordinates, but in practice the appearance models model local regions of the image only, normalized to the current centroid, which translate with respect to the image coordinates. However, at any time an alignment is known, allowing us to calculate P_c and M_{RGB} for any point $\mathbf{x}$ in the image, $P_c(\mathbf{x})$ being zero outside the modeled region.

When a new track is created, a rectangular appearance model is created with the same size as the bounding box of the foreground region. The model is initialized by copying the pixels of the track's foreground component into the color model. The corresponding probabilities are initialized to 0.4, and pixels which did not correspond to this track are given zero initial probability.

On subsequent frames, the appearance model is updated by blending in the current foreground region. The color model is updated by blending the current image pixel with the color model for all foreground pixels, and all the probability mask values are also updated with the following formulae ($\alpha=\lambda=0.95$):

$$M_{\mathrm{RGB}}(\mathbf{x},t) = M_{\mathrm{RGB}}(\mathbf{x},t-1)\alpha + (1-\alpha)I(\mathbf{x}) \text{ if } \mathbf{x}\in\mathcal{F} \quad (1)$$

$$P_{\mathrm{c}}(\mathbf{x},t) = P_{\mathrm{c}}(\mathbf{x},t-1)\lambda \text{ if } \mathbf{x}\notin\mathcal{F} \quad (2)$$

$$P_{\mathrm{c}}(\mathbf{x},t) = P_{\mathrm{c}}(\mathbf{x},t-1)\lambda + (1-\lambda) \text{ if } \mathbf{x}\in\mathcal{F} \quad (3)$$

In this way, we maintain a continuously updated model of the appearance of the pixels in a foreground region, together with their observation probabilities. The latter can be thresholded and treated as a mask to find the boundary of the object, but also gives information about non-rigid variations in the object, for instance retaining observation information about the whole region swept out by a pedestrian's legs.

Fig. 3 shows the appearance model for a van from the PETS data at several different frames. The appearance models are used to solve a number of problems, including improved localization during tracking, track correspondence and occlusion resolution.

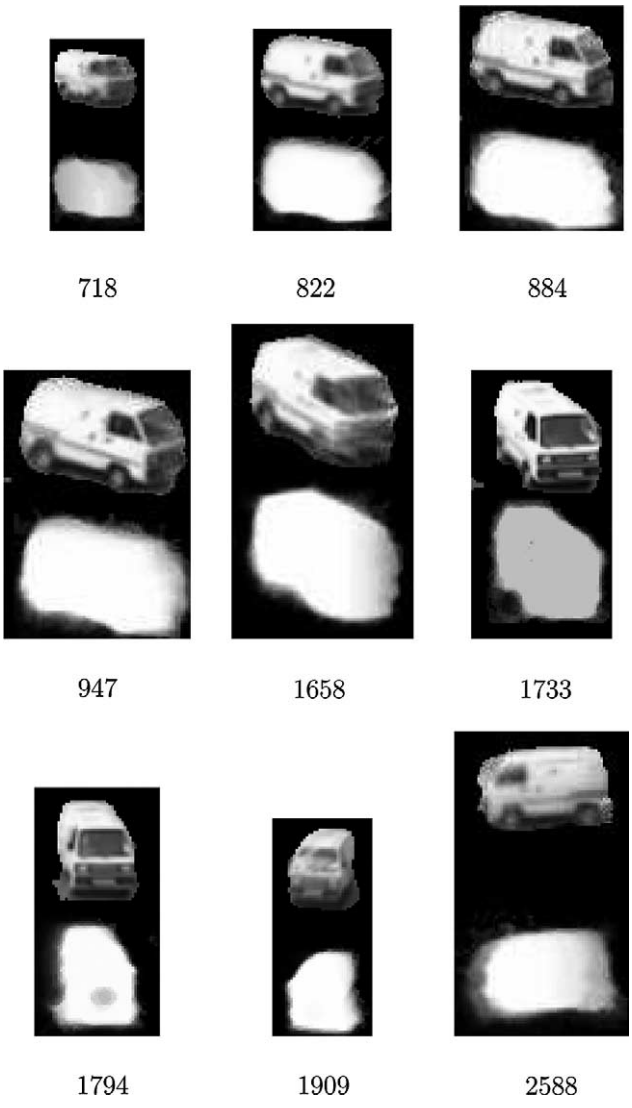

Fig. 3. The evolution of an appearance model. In each figure, the upper image shows the appearance for pixels where observation probability is greater than 0.5. The lower shows the probability mask as grey levels, with white being 1. The frame numbers at which these images represent the models are given,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

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 unavoidably noisy, and the additional layers of morphology increase the noise in the localization of the objects, 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, $p(I, \mathbf{x}, M)$, is minimized over a small search region and the point with the lower 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.

$$p(I, \mathbf{x}, M) = \prod_{\mathbf{y}} p_{\mathrm{RGB}}(\mathbf{x} + \mathbf{y}) P_{\mathrm{c}}(\mathbf{x} + \mathbf{y}) \tag{4}$$

$$p_{\mathrm{RGB}}(x) = (2\pi\sigma^2)^{-(3/2)} \mathrm{e}^{\frac{-||I(x)-M(x)||^2}{2\sigma^2}} \tag{5}$$

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 Figs. 4–6:

(1) Using a first-order model, the centroid locations of the objects i 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 observation 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.

$$p_i(\mathbf{x}) = p_{\mathrm{RGB}_i}(\mathbf{x}) P_{\mathrm{c}_i}(\mathbf{x}) \tag{6}$$

Figs. 4c and 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 disputed pixels are reclassified, with disputed pixels being assigned to the foremost object which overlapped them.

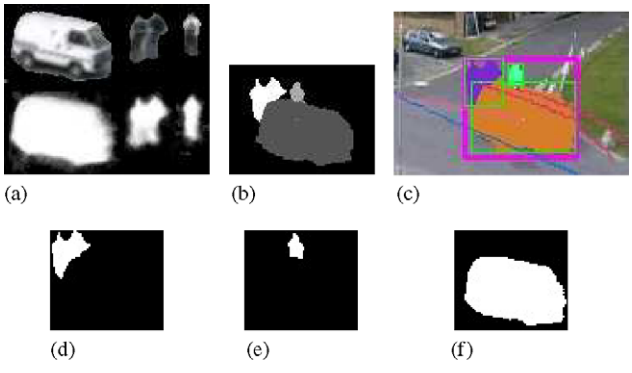

Fig. 4. An occlusion resolution (Frame 921 of dataset 1, camera 1). (a) Shows three appearance models for tracks converging in a single region. (b) Shows the pixels of a single foreground region, classified independently as to which of the models they belong to. (d–f) show the pixels finally allocated 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.

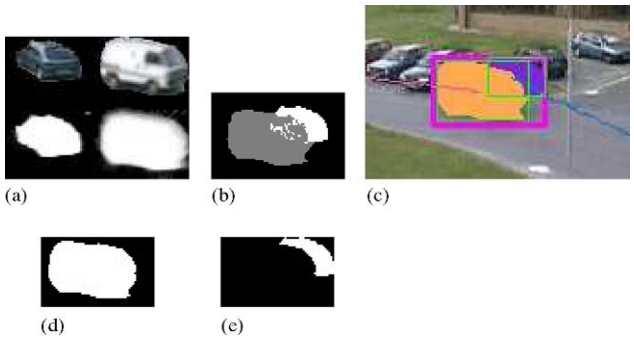

Fig. 5. An occlusion resolution (Frame 825 of dataset 1, camera 1). (a) appearance models. (b) Independently classified foreground region pixels as to which of the models they belong to. (d, e) the pixels allocated to each track after enforcing the depth-ordering, and (c) the regions overlaid on the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

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.

If a tracked object separates into two bounding boxes, then a new track is created for each part, with the appearance models being initialized from the corresponding areas of the previous appearance model.

Fig. 6. The appearance model for a group of people.

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.

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 [8]. 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 Fig. 7–10. 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.

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

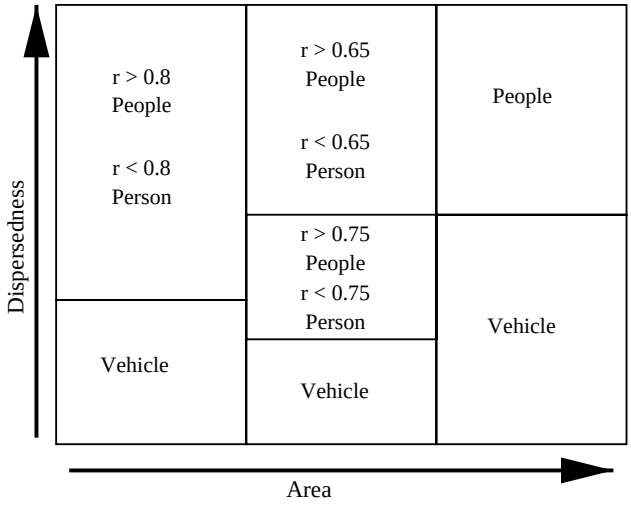

Fig. 7. The classification rules for a foreground region. r is the horizontal-to-vertical principal axis length ratio.

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 in 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.

Fig. 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.

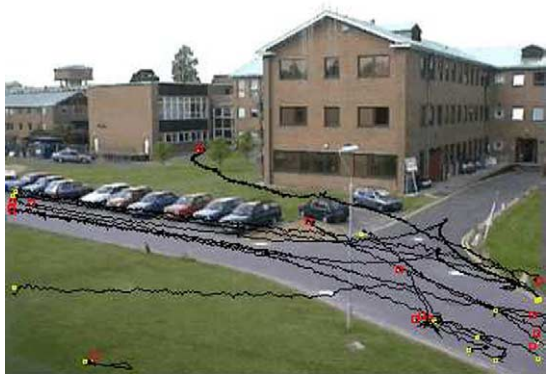

Fig. 9. An image showing all the tracks detected by the system for dataset 1, camera 1, overlaid on a background image.

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, τ, 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

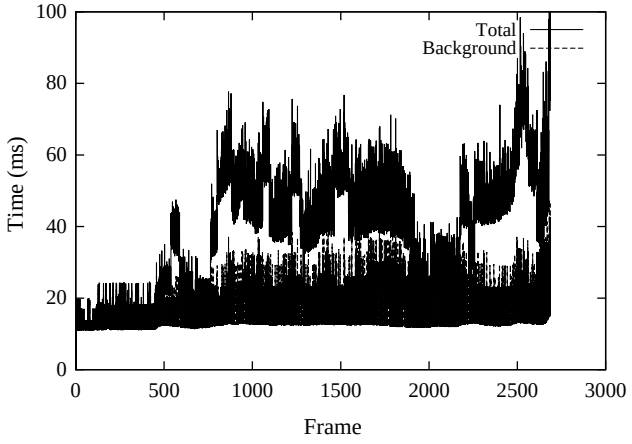

Fig. 10. Per frame time requirements for the tracking and background subtraction. The lower line shows the time required for the background subtraction, and the upper line shows the total time required to process each frame. Times vary from about 10 ms when there are no objects to be tracked to a peak of about 130 ms when there are several overlapping objects being tracked.

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 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 [20]. 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:

$$D_T(T1,T2) = \frac{1}{N_{12}^2} \sum_{i:\exists T1(t_i)\&\exists T2(t_i)} \sqrt{d_{\mathbf{x}}^2(i) + d_{\mathbf{v}}^2(i)} \tag{7}$$

$$d_{\mathbf{x}}(i) = |\mathbf{x}_1(i) - \mathbf{x}_2(i)| \tag{8}$$

$$d_{\mathbf{v}}(i) = |\mathbf{v}_1(i) - \mathbf{v}_2(i)| \tag{9}$$

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 gives some idea of the quality of the segmentation.
- Object detection lag: This is the difference in time between when a new object appears in the ground truth and when the tracking algorithm detects it.
- Track incompleteness factor: This measures how well the automatic track covers the ground truth:

$$\text{Trackincompleteness} = \frac{F_{nf} + F_{pf}}{T_i} \tag{10}$$

  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:

$$f_p = \frac{\text{Results without corresponding ground truth}}{\text{Total number of ground truth tracks}} \tag{11}$$

$$f_n = \frac{\text{Ground truth without corresponding result}}{\text{Total number of ground truth tracks}} \tag{12}$$

- 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

Table 2
Performance measures for Dataset 1, Camera 1

	Dataset 1, Camera 1
Track error f_p	5/7
Track error f_n	2/7
Average position error	5.51 pixels
Average area error	−346 pixels
Average detection lag	1.71 frames
Average track incompleteness	0.12
Object type error	0 tracks

truth generation, we have only generated ground truth up to frame 841. Table 2 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 spurious tracks are generated by failures in the background subtraction in the final frames, 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 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 interactions, 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 interactions. 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 color covariances or even multi-modal distributions for each pixel would allow more robust modeling, 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.

Acknowledgements

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

References

[1] A.F. Bobick, et al., The KidsRoom: a perceptually-based interactive and immersive story environment, Teleoperators and Virtual Environment 8 (1999) 367–391.
[2] W. Grimson, C. Stauffer, R. Romano, L. Lee, Using adaptive tracking to classify and monitor activities in a site, Conference on Computer Vision and Pattern Recognition (1998) 22–29.
[3] H. Tao, H. Sawhney, R. Kumar, Dynamic layer representation with applications to tracking, Proceedings of International Conference on Pattern Recognition (2000).
[4] T.-H. Chang, S. Gong, E.-J. Ong, Tracking multiple people under occlusion using multiple cameras, Proceedings of 11th British Machine Vision Conference (2000).
[5] S. Dockstader, A. Tekalp, Multiple camera fusion for multi-object tracking, Proceedings of IEEE Workshop on Multi-Object Tracking (2001) 95–102.
[6] S. McKenna, J. Jabri, Z. Duran, H. Wechsler, Tracking interacting people, International Workshop on Face and Gesture Recognition (2000) 348–353.
[7] I. Haritaoğlu, M. Flickner, Detection and tracking of shopping groups in stores, CVPR (2001).
[8] A. Lipton, H. Fuyiyoshi, R. Patil, Moving target classification and tracking from real-time video, Proceedings of Fourth IEEE Workshop on Applications of Computer Vision (1998).
[9] I. Haritaoğlu, D. Harwood, L.S. Davis, W^4: Real-time surveillance of people and their activities, IEEE Transaction on Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–830.
[10] S. Intille, J. Davis, A. Bobick, Real-time closed-world tracking, Conference on Computer Vision and Pattern Recognition (1997) 697–703.
[11] T. Ellis, M. Xu, Object detection and tracking in an open and dynamic world, International Workshop on Performance Evaluation of Tracking and Surveillance (2001).
[12] A. Senior, Tracking with probabilistic appearance models, Third International workshop on Performance Evaluation of Tracking and Surveillance Systems (2002).

[13] T. Zhao, R. Nevatia, F. Lv, Segmentation and tracking of multiple humans in complex situations, Conference on Computer Vision and Pattern (2001).
[14] C.R. Wren, A. Azarbayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.
[15] S. Khan, M. Shah, Tracking people in presence of occlusion, Asian Conference on Computer (2000).
[16] A. Elgammal, L. Davis, Probabilistic framework for segmenting people under occlusion, Eighth International Conference on Computer Vision, II, IEEE (2001) 145–152.
[17] R. Rosales, S. Sclaroff, Improved tracking of multiple humans with trajectory prediction and occlusion modelling, IEEE CVPR Workshop on the Interpretation of Visual Motion (1998).
[18] M. Isard, J. MacCormick, BraMBLe: a Bayesian multiple-blob tracker, International Conference on Computer Vision 2 (2001) 34–41.
[19] T. Horprasert, D. Harwood, L.S. Davis, A statistical approach for real-time robust background subtraction and shadow detection, ICCV'99 Frame-Rate Workshop (1999).
[20] G. Pingali, J. Segen, Performance evaluation of people tracking systems, Proceedings of IEEE Workshop on Applications of Computer Vision (1996) 33–38.