Object Detection and Tracking in an Open and Dynamic World

Tim Ellis Ming Xu

Department of Electrical, Electronic and Information Engineering
City University, London, EC1V 0HB, UK
{t.j.ellis, m.xu}@city.ac.uk

Abstract

A motion detection and tracking algorithm is presented for monitoring the pedestrians in an outdoor scene from a fixed camera. A mixture of Gaussians is used to model each pixel of the background image and thus adaptive to the dynamic scene. Colour chromaticity is used as the image representation, which results in the illumination-invariant change detection in a daylit environment. To correctly interpret those objects that are occluded, merged, split or exit from the scene, a scene model is created and the motion of each object is predicted. A Bayesian network is constructed to reason about the uncertainty in the tracking. The results for detecting and tracking the moving objects in the PETS sequences are demonstrated.

1. Introduction

1.1 Previous Work

Tracking non-rigid objects in real world scenes contains several difficulties for computer vision. Problems include static and dynamic occlusion, variation of lighting condition, failure of foreground detection, etc. Therefore, many successful tracking systems work only in constrained environments, in which the targets are sparsely distributed and the background is less dynamic [12].

Frame differencing is a technique widely used for the change detection in dynamic images. It compares each incoming frame with a background image and classifies those pixels of significant variation into foreground. The background can be modeled with a single adaptive Gaussian [12] and learnt during an initialization period when the scene is empty. This method is efficient only in less dynamic scenes but has difficulties with vacillating backgrounds (e.g. swaying trees), background elements moving, and illumination changes. A more robust method is to model the background by a mixture of adaptive Gaussians [11]. However, it may fail in tracking a background pixel under fast illumination changes, e.g. flood of sunlight, shadows or artificial lights switching on/off. This causes spurious "foregrounds" and can lose targets in such cases. The reason is that most the existing applications use intensity-based image representations, e.g. (R, G, B) or I, which are the result of interaction between illumination from light sources and reflectance of object surfaces. To be able to identify and track the same object surface (e.g. a background pixel) under varying illumination, it is desirable to separate the variation of the illumination from that of the surface reflection.

In the existing algorithms that track occluded objects, Intille et al [5] compared the properties of detected foreground regions with those of the previously tracked objects. An object is identified as being occluded if two objects are found to match the same foreground region. Haritaoglu et al [3] identified the dynamic occlusion when the predicted bounding boxes of two objects overlap the same foreground regions. However, these are rule-based methods that are brittle and globally lack of consistency [10]. In addition, these algorithms did not consider static occlusions, e.g. buildings, trees, or road sign. Therefore, objects may suddenly disappear at some sites in a scene and many new objects appear at some other sites. Therefore, the trajectory of an object tends to be short and segmental.

1.2 Our Approach

In this paper a mixture of Gaussians is used to model each pixel of the background image and thus adaptive to the dynamic scene. The combination of colour chromaticity- and intensity-based image representations results in the illumination-invariant change detection in a daylit environment. To correctly interpret those objects that are occluded, merged, split or exit from the scene, a scene model is created and the motion of each object is predicted. A Bayesian network is constructed to reason about the uncertainty and noise in the observation.

2. Foreground detection

2.1. Background modeling and extraction

A mixture of up to *N* Gaussians has been used to model the probability of observing a value, $\mathbf{f}_t$, at each pixel:

$$P(\mathbf{f}_t) = \sum_{i=1}^{N} \omega_{i,t} G(\mathbf{f}_t, \boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}_{i,t}) \qquad (1)$$

where *G* is the Gaussian probability density function, $\boldsymbol{\mu}_{i,t}$ and $\boldsymbol{\Sigma}_{i,t}$ are the (temporal) mean vector and covariance matrix of the i-th distribution, respectively; the weight, $\omega_{i,t}$ reflects the likelihood that the *i*-th distribution accounts for the data and is limited between [0, 1]. To simplify the computation, the components of $\mathbf{f}_t$ are assumed to be independent and then $\boldsymbol{\Sigma}_{i,t}$ can be approximately represented using the sum of its diagonal elements, $\sigma^2_{i,t}$.

We approximate the initial values of the temporal statistics using the spatial statistics over a local region (*n* pixels) of the start frame:

$$\begin{aligned} \boldsymbol{\mu}_{0,0}(\mathbf{x}) &= \frac{1}{n} \sum_{\Delta\mathbf{x}} \mathbf{f}_0(\mathbf{x} + \Delta\mathbf{x}) \\ \sigma^2_{0,0}(\mathbf{x}) &= \frac{1}{n-1} \sum_{\Delta\mathbf{x}} \left\| \mathbf{f}_0(\mathbf{x} + \Delta\mathbf{x}) - \boldsymbol{\mu}_{0,0}(\mathbf{x}) \right\|^2 \end{aligned} \qquad (2)$$

For the following frames, every new observation, $\mathbf{f}_t$, is checked against the *N* Gaussian distributions. A match is identified if $\left\| \mathbf{f}_t - \boldsymbol{\mu}_{i,t-1} \right\| < c\sigma_{i,t-1}$ ($c \approx 3$). The parameters of the matched distribution are updated as:

$$\begin{aligned} \boldsymbol{\mu}_{i,t}(\mathbf{x}) &= (1-\varphi)\boldsymbol{\mu}_{i,t-1}(\mathbf{x}) + \varphi \mathbf{f}_t(\mathbf{x}) \\ \sigma^2_{i,t}(\mathbf{x}) &= (1-\varphi)\sigma^2_{i,t-1}(\mathbf{x}) + \varphi \left\| \mathbf{f}_t - \boldsymbol{\mu}_{i,t}(\mathbf{x}) \right\|^2 \end{aligned} \qquad (3)$$

where φ controls the updating rate, and the weight $\omega_{i,t}(\mathbf{x})$ is increased. For the unmatched *j*-th distribution ($j \neq i$), $\boldsymbol{\mu}_{j,t}$ and $\sigma_{j,t}$ remain the same, but $\omega_{j,t}(\mathbf{x})$ is damped exponentially.

If none of the existing distributions matches the current pixel value, we have to either create a new distribution, given less than *N* existing distributions, or replace the least probable distribution with a new distribution. The distribution(s) with greatest weight is (are) considered as the background model(s).

2.2. Illumination-invariant detection

In an outdoor environment lit by sunlight, fast illumination changes tend to occur at the regions where shadows emerge or disappear. For the fast-moving cloud case, the shadows are larger-scale and illuminated by the reflection from the ambient clouds. These grey (or white) clouds are relatively balanced in all visible wavelengths. Therefore, the reflected light from shadow or lit regions has no difference in spectral distribution and only varies in magnitude [13]. The proportionality between (R, G, B) can be better represented using the colour chromaticity, (r, g, b), each component of which will keep constant for a given object surface under varying illumination and is more appropriate to model using an adaptive Gaussian [13].

To compensate the loss of the colour-based model in dim regions, we have combined the motion detection results using the intensity, I, with those using the (r, g, b) colour space to give a better detection. Suppose S_I and S_C are the binary sets of foreground detection using intensity- and colour-based models, respectively. A value of 1 represents the foreground and 0 for background. One combination scheme favouring the colour-based model is to add some points of S_I, which is spatially close to S_C, into S_C. The set of the fused foreground pixels, S, at pixel **x** can be computed as:

$$S(\mathbf{x}) = S_C(\mathbf{x}) + \overline{S_C(\mathbf{x})} \cdot S_I(\mathbf{x}) \cdot (S_C \oplus B)(\mathbf{x}) \qquad (4)$$

where $\oplus$ denotes the morphological dilation and *B* is the structuring element. .

The foreground pixels above are filtered by a closing (dilation plus erosion) morphological operation and then clustered into foreground “blobs” using a connected component analysis. A minimum number of foreground pixels is set for each blob to rule out small disturbances.

3. Object and scene modeling

3.1. Blob models

Each foreground blob detected at the current frame is ideally associated with an object or a group of interacting objects. It is characterized by:

- Positions: its bounding box and coordinate of its centroid.
- Colours: its colour template pyramid.
- Sizes: the number of foreground pixels.
- Status: allocated to an object (ALLOCATED) or not (UNALLOCATED).

For each foreground pixel of the blob, the colour chromaticity (r, g, b) is transformed into the two opponent colours (rg, by):

$$rg = r - g$$
$$by = \frac{2b - r - g}{2} \qquad (5)$$

which represent colours along the red to green, and blue to yellow axes [1]. The 3-D colour information has been mapped to 2-D, and for each blob a 2-D *m*×*m* colour histogram is generated as a template (*m*=16). This 2-D template needs less storage than a 3-D one, and so is faster to handle. The colour template for a blob needs to be compared with that for each object that has been tracked in the previous frames. Intuitively some tracked objects will have radically different colours from others, and an efficient search is desirable. This is achieved by having a pyramid of colour templates at four resolutions, and by trying to match initially at the coarsest resolution, and only going to higher resolution for objects considered similar [1].

3.2. Object models

Each tracked object is recorded in an object database and ideally corresponds with a blob in the new frame. An object is described by:

(1) The characteristics of its matched blob

- Positions: its bounding box and coordinate of its centroid.
- Colours: its colour template pyramid.
- Sizes: the number of foreground pixels.

(2) The tracking information

- Status: NEW (objects entering the scene), TERMINATED (objects leaving the scene), UPDATED (an object optimally matched to a blob), MERGED (colliding objects), SPLIT (objects colliding and then separate), OCCLUDED (objects hidden by static occlusions), MISSING (cannot be interpreted).
- Dynamic model: direction, velocity, acceleration.
- History: last position, frame of being first seen, frame of being last seen.
- Prediction: predicted position (estimated by a first-order motion model), predicted status, predicted bounding box.
- Interacting object: index of another object that interacts or is assumed to interact the underlying object.

3.3. Scene models

Because the camera is fixed, a scene model can be constructed for a specific camera position. Whilst this is currently done manually, we are investigating automatic methods for learning the scene model [6]. This helps reasoning about the termination and occlusion of objects by scene elements. Three types of static occlusions in a scene are identified (Fig. 1):

- Border occlusions (BO) due to the limits of the camera field-of-view (FOV).
- Long-term occlusions (LO). These are locations where objects may leave the scene earlier than expected, corresponding to the termination of a record in the object database. These occlusions often have one side touching the border of the image and make objects leave the scene from the other side, at a distance away from the border of the image, e.g. buildings and vegetation. The long-term occlusion may also exist in the middle of an image, e.g. at the doors of a building. Without some prior knowledge of these long-term occlusions, an object disappearing at a LO may later be mismatched with other objects that are present nearby.
- Short-term occlusions (SO). These are the locations where an object may be temporarily occluded by a static occlusion, e.g. a tree or a road sign. Prior knowledge of these occlusions helps avoid missing existing objects and creating “new” objects.

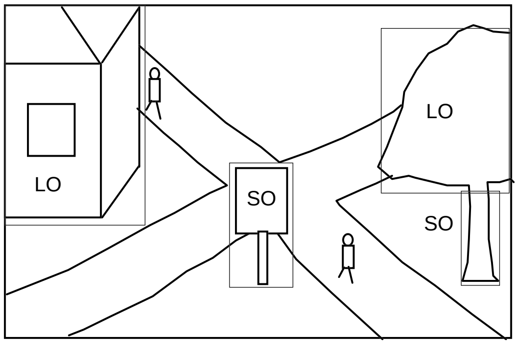


Figure 1. Long-term occlusions (LO) and short-term occlusions (SO) in a scene.

All the occlusions are stored in an occlusion database. Each occlusion is characterized by its:

- Type (BO, LO or SO).
- Bounding box, representing its location and dimension.

The overlap of these static occlusions with the predicted bounding box of an object can be used to predict object termination and occlusion. Currently a rectangular bounding box is being used for each static occlusion to minimize the computational cost. A more accurate representation of these occlusions, e.g. using polygons, is straightforward but not so important here, because these occlusion bounding boxes are used only for the prediction of some events (termination and occlusion). The determination of such events also depends on the result of tracking (e.g. an object fails to find a corresponding blob), because objects may pass in front of an occlusion.

3.4. Model-based prediction

After the status of an object is determined at each frame, the object is subject to a process of status prediction which is based on a first-order motion model and scene model.

- An object is labelled as PREDICTIVE TERMINATED, if its predicted bounding box overlaps a long-term occlusion (LO) or the outer of the border occlusion (BO).
- An object is labelled as PREDICTIVE OCCLUDED, if its predicted bounding box overlaps a short-term occlusion (SO).
- An object is labelled as PREDICTIVE MERGED, if its predicted bounding box overlaps that for another object. Then the index of the second object is recorded as the "Interacting Object" of the first object.
- An object is labelled as PREDICTIVE SPLIT, if its status is MERGED and its predicted bounding box does not overlap that of its "Interacting Object".

4. Object tracking

A staged and ordered matching process has been used to make a correspondence between the blobs detected at the current frame and the objects tracked at the previous frame.

4.1. Blob to object matching

The visible blobs are first compared with the tracked objects. To determine blob-to-object correspondence, a match score for every blob and object combination is computed as the weighted sum of several distance measures, as in [5]. Each distance measure reflects the difference of some characteristic between the blob and object. It is limited by an allowable tolerance for possible matching and normalised by the tolerance value. Because the objects are assumed to have no drastic change in some selected characteristics between two consecutive frames, the distance measure is low for a possible match. The characteristics that have been considered include:

- Predicted position. It is given the greatest weight. For an object identified as NEW at the previous frame, this characteristic is replaced by the object position.
- Colour. The distance measure between the colour template, T, of a blob and that, T', of an object is calculated as:

  $$d = \sqrt{\sum_{rg=0}^{m-1}\sum_{by=0}^{m-1}\left(\frac{T_{rg,by}}{S_T} - \frac{T'_{rg,by}}{S_{T'}}\right)^2} \qquad (6)$$

  where $S_T = \sum_{rg=0}^{m-1}\sum_{by=0}^{m-1} T_{rg,by}$ such that $0 \le d \le 1$. The scaling down by the sum of values in the template ensures that the distance measure is invariant to scale.
- Direction
- Size

The match scores between all pairs of the detected blobs and tracked objects constitute a match score matrix. This matrix is sparse in that many pairing relations are inhibited by using the tolerance value for each distance measure. For each object, its match scores with all the blobs are compared, and the best-matched blob is selected. For each blob, its match scores with all the objects are also compared and the best-matched object is identified.

For a pair of object O_1 and blob B_1:

- If O_1 is the only object that has B_1 as its best-matched blob and B_1 is also the only blob that has O_1 as its best-matched object, then blob B_1 is considered to correspond to object O_1 and O_1 is set to the status UPDATED (Fig. 2(a)).
- If both the objects O_1 and O_2 have the same best-matched blob B_1, then the match scores of (B_1, O_1) and (B_1, O_2) are compared (Fig. 2(b)). Suppose that (B_1, O_1) has a better match, then object O_1 is assigned to blob B_1 and set to the status UPDATED; object O_2 checks its next best-matched blob B_2. If O_2 is the best-matched object to B_2, then object O_2 is assigned to blob B_2 and set to the status UPDATED; otherwise, O_2 is labeled with POTENTIALLY MERGED and left to the next stage for checking unmatched objects.
- If object O_1 is the best-matched object to both blobs B_1 and B_2, then the match scores of (B_1, O_1) and (B_2, O_1) are compared (Fig. 2(c)). Suppose that (B_1, O_1) has a better match, then object O_1 is assigned to blob B_1 and set to the status UPDATED; blob B_2 checks its next best-matched object O_2. If B_2 is the best-matched blob to O_2,

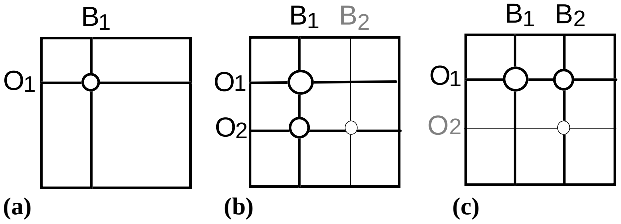


Figure 2. (a) an UPDATED object, (b) a POTENTIALLY MERGED object, and (c) a POTENTIALLY SPLIT blob. The size of each circle indicates the amount of match score.

then object O_2 is assigned to blob B_2 and set to the status UPDATED; otherwise, B_2 is labeled with POTENTIALLY SPLIT and left to the final stage for checking unmatched blobs.

Therefore, when two objects are potentially merged, the object, whose properties are less influenced by the object merging (with a better match score), is favoured and considered as a normally tracked object, e.g. the car in a car-and-pedestrian merging group or the pedestrian who partially occludes another one. On the other hand, when two objects are potentially split from a single object, the object whose properties are more consistent with that of the original object is favoured and considered as a normally tracked object, e.g. the pedestrian who drops a baggage on the ground.

In addition, the splitting of two originally merged objects is treated differently in our algorithm according to their relative moving directions. If one object heads toward and then passes by another, both objects are considered as UPDATED. If a pair of objects move along non-interfering directions, one is UPDATED and another is SPLIT.

4.2. Unmatched objects

After sorting out the UPDATED objects, there remain some objects that do not have a correspondence with any detected blob. This may arise from the objects leaving the scene, the occlusion of objects by scene elements, the merging of multiple objects, or the failure of foreground detection. The ambiguity here can be partly relieved by using domain knowledge. For example, if it is known that an unmatched object was very close to a long-term occlusion in the last frame, it is quite possible that this object left the scene in the current frame. However, there exist uncertainties in such domain knowledge:

- Not all of the objects close to a long-term occlusion will leave the scene (they may walk in front of it).
- An object may merge with another one near the border of a long-term occlusion.
- The foreground detection may fail (i.e. the corresponding blobs are missing) at any position in a scene.

Given the uncertain and incomplete information, the object tracking can be inferred through a process of deduction. A Bayesian network [8] is a framework for representing and using domain knowledge to perform probabilistic inference. It is a directed acyclic graph in which nodes represent random variables and arcs represent causal connections among the variables. Associated with each node is a probability table that provides conditional probabilities of the node's possible states given each possible state of its parents. In the case that a node has no parents, conditional probabilities degenerate to priors. When values are observed for a subset of the nodes, posterior probability distributions can be computed for any of the remaining nodes. Bayesian networks have been used in object tracking and behaviour identification [2][4][9][10].

The Bayesian network used for reasoning about unmatched objects is shown in Fig. 3. Except the query nodes “terminated at t”, “occluded at t”, “merged at t” and “missing at t”, the nodes correspond to image-measurable quantities, i.e. evidence nodes. All quantities in the network are binary variables. The conditional probability distributions attributed to each variable in the network are specified using domain knowledge. Although the prior probability of an object being merged is very low (0.05), the conditional probability runs up to 0.75 given that the object was merged at the previous frame (time t-1). This is also true for the “occluded at t-

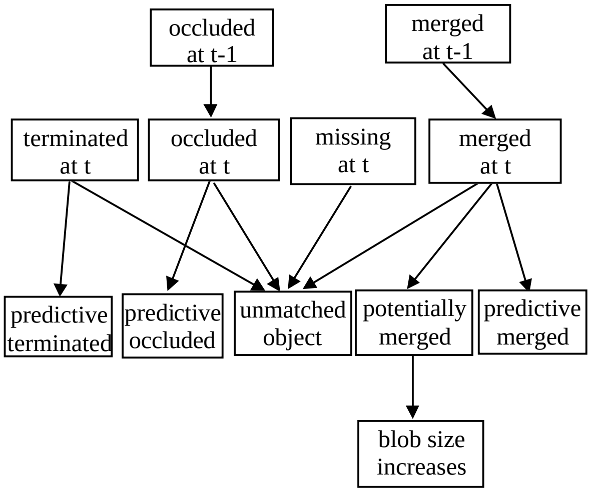


Figure 3. The Bayesian network for reasoning about unmatched objects.

1" and "occluded at t" nodes. For a merged object, it is very likely to be labeled as POTENTIALLY MERGED in the matching stage and the corresponding blob is often significantly larger than that object.

Given the observed values for the evidence nodes, the probability of any unmatched object being terminated, occluded, merged or missing can be computed and the most probable explanation can be given. It is noted that four causes, "terminated", "occluded", "merged" and "missing" compete to explain the evidence "unmatched object". Hence they become conditionally dependent given that their common child is observed. For example, suppose the underlying object is unmatched, but that we know that this object was merged at the previous frame. Then the posterior probability that the object is terminated, occluded or missing goes down, which is called "explaining away".

To make the computation more efficient, the posterior probabilities of the four query nodes were precomputed using the Bayes Net Toolbox in [7], given all the possible values of the evidence nodes. The result is saved in a look-up-table.

4.3. Unmatched blobs

After checking the object database, the detected blobs that have not been interpreted are most likely split or new objects. Another Bayesian network has been used to infer the posterior probabilities of the query nodes, "new at t" and "split at t", given the observed values of the evidence nodes (Fig. 4). In order to have efficient computation, the "distance to BO or LO" is approximated with a set of discrete values: touching, close and far. It is noted that most of the split objects was previously merged unless the objects entered the scene in a group. This is reflected in the high conditional probability of "split at t" given "merged at t-1". For a split object, it is most likely labeled as POTENTIALLY SPLIT in the blob-to-object matching stage and tends to be significant smaller than the merging group.

Once the status of all the objects is determined, the records in the object database need to be updated. The record of an UPDATED, SPLIT or NEW object is replaced by the characteristics of the corresponding blob. For a MERGED or OCCLUDED object, its position is updated according to its visible history and the first-order motion model; its colour and size remain unchanged. The record of a MISSING object is kept unchanged until this object is re-tracked or automatically terminated after "missing" for three consecutive frames.

5. Results

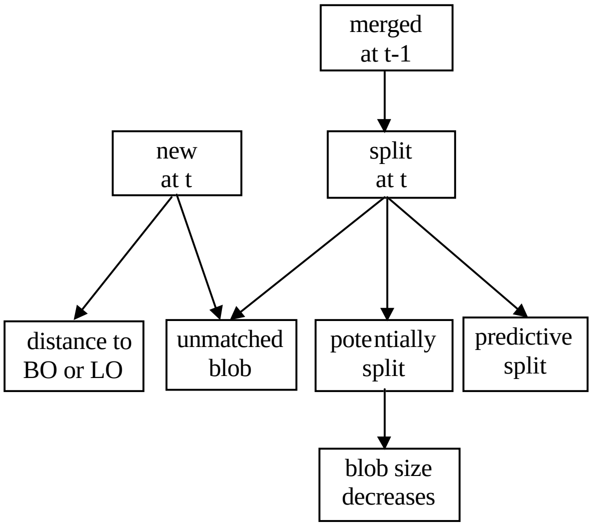
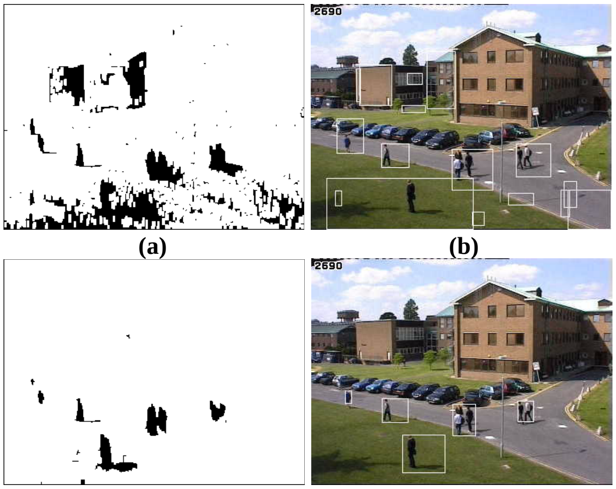


Figure 4. The Bayesian network for reasoning about unmatched blobs.

To assess the significance of the detection and tracking algorithm, we have applied it to the PETS2001 sequences which include significant lighting variation, occlusion and scene activity. The sequences were spatially sub-sampled to half-PAL (384×288 pixels) and temporal sub-sampling has been investigated in our experiments. The results presented below use 2.5 fps for foreground detection and 5 fps for object tracking. These rates provided a reasonable trade-off between computational efficiency and robust detection and tracking.

5.1 Foreground detection

Fig. 5 shows the results of the motion detection at frame 2690 of Dataset 3 (camera 1, testing) at 2.5 fps. The corresponding result sequences ("31_1.avi" and "31_2.avi" using intensity-based model, and "31_3.avi" and "31_4.avi" using the combination of colour- and intensity-based models) start at frame 1500 and end at frame 3000. The foreground pixels in the colour-based results are those that go beyond [μ-3.5σ, μ+3.5σ] of the most probable Gaussians. The foreground pixels in the intensity-based results arise from a global threshold on the difference between the observation and the mean of the most probable Gaussian. The thresholding level is selected as 10% of the maximum intensity so as to produce "blobs" of similar sizes to those in the corresponding colour-based results. In order to rule out isolated "foreground" pixels and fill gaps and holes in "foreground" regions, a 1×3 closing (dilation-erosion) opera-



(a) (b)

(c) (d)

Figure 5. Motion detection, at frame 2690 of Dataset 3 (camera 1, testing), with FPS=2.5: the detected blobs (a) and bounding boxes overlaid on the frame (right) using the intensity-based model (top) and the combination of colour- and intensity-based models (bottom).

tion has been applied to the binary image of detected “foreground” pixels.

There is a major illumination change around frame 2690. In the intensity-based result, Figs. 5(a) and (b), a large area of the background is detected as a huge foreground object, in which the ground-truth targets (pedestrians) are submerged and lost. On the other hand, in the result of the combination of colour- and intensity-based models, Figs. 5(c) and (d), fast illumination changes give no additional “foreground” blob and the ground-truth targets are clearly visible.

Table 1 shows the number of the detection errors in the same image sequence, from frame 1600 (skipping the learning period) to frame 3000. Multiple objects are considered as a single ground-truth object if they are grouped. The colour-based model is much more successful in dealing with illumination changes.

Models	Intensity	Colour
Ground-truth objects	509	
Undetected positives	54	22
False positives	151	8

Table 1. The detection errors in an image sequence with fast illumination changes.

5.2 Object Tracking

Figs. 6-8 show part of the tracking results using Dataset 2 (camera 1, testing) at 5 fps. The corresponding result sequence ("31_5.avi") starts at frame 1 and ends at frame 701. The results of the first five frames are noisy and not included, because the Gaussian mixture model needs to learn the initial parameters for each distribution.

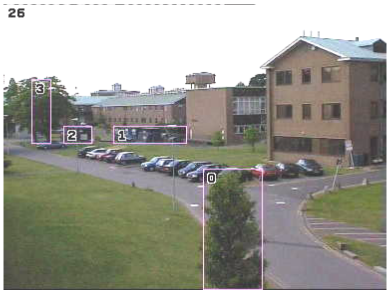
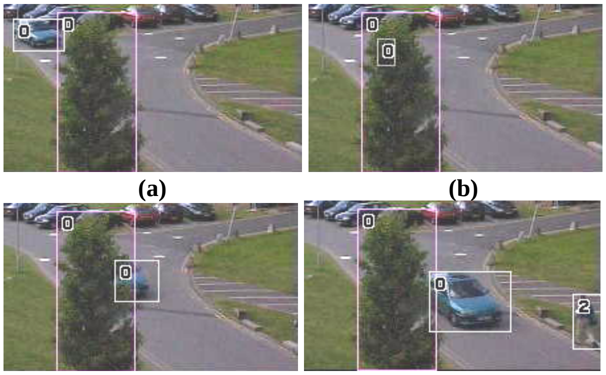


Figure 6. The occlusion models for the Dataset 2 (camera 1, testing).

Fig. 6 shows the manually selected occlusions in the scene, in which No. 0, 2 and 3 are short-term occlusions and No. 1 is a long-term occlusion. The building in the right is a potential long-term occlusion but not used here.

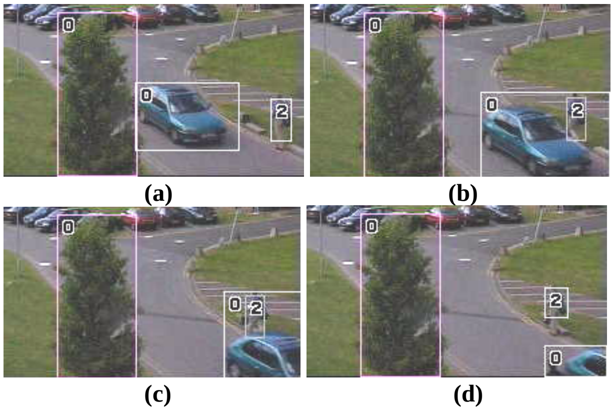
Fig. 7 shows an example when an object (No. 0, white bounding box) is passing by a short-term occlusion (No. 0, in pink bounding box). At frame 351 (Fig. 7(a)), the predicted bounding box (invisible here) of object 0 overlaps occlusion 0, the predicted status is set to PREDICTIVE OCCLUDED. At frame 386 (Fig. 7(b)) when no corresponding blob is found, object 0 is determined as OCCLUDED by the Bayesian network. Its position is updated according to the first-order motion model based on its visible history. Therefore, the object bounding box (grey) is not observed but estimated. At frame 396 (Fig. 7(c)) when a blob is detected at the other side of the occlusion and matches object 0, object 0 is re-tracked and its record is then updated using the



(a) (b)

(c) (d)

Figure 7. A tracking example in Dataset 2 when object 0 (white) passes by a short-term occlusion 0 (grey).



(a) (b)

(c) (d)

Figure 8. A tracking example in Dataset 2 when objects 0 and 2 are merged and then split.

new observation.

Fig. 8 shows an example when one object is merged with another. At frame 416 (Fig. 8(a)) when the predicted bounding boxes (invisible) of objects 0 and 2 overlap, the predicted status is set to PREDICTIVE MERGED for both the objects. At frame 426 (Fig. 8(b)) when only one blob is detected, object 0 is matched to that blob because its properties are less influenced by the merging. Object 0 is determined as a normal UPDATED object. On the other hand, object 2 is determined as MERGED by the Bayesian network and its interacting object is set to No. 0. Its position is updated and predicted according to the first-order motion model based on the visible history (note the grey bounding boxes in Figs. 8(b)(c)). At frame 441 (Fig. 8(d)) object 2 are matched to a newly detected blob and thus re-tracked as an UPDATED object.

Table 2 shows the tracking errors, when the objects are interacting with each other or the scene elements, for the first four testing sequences (Datasets 1-2, cameras 1-2). The result indicates that the current algorithm is proper to track two interacting objects or object groups. Its performance degrades when multiple objects are clustered in a local region, such as that in Dataset 3.

Events	Merged	Split	Occluded
Examples	12	8	13
Errors	0	1	3

Table 2. The tracking examples and errors for interacting objects in the Datasets 1 and 2.

6. Conclusions

The combination of the colour- and intensity-based Gaussian mixture models can better adapt to fast illumination changes when detecting foregrounds. The scene model and motion prediction provide relatively reliable evidence in inferring and tracking objects through Bayesian networks, especially when ambiguity in observation arises.

Future work includes considering multiple objects that interact within a group, using dynamic Bayesian networks and even continuous variables to infer object status, using multi-view co-operation to interpret the incomplete observation from each single view.

Acknowledgements

This work is supported by the EPSRC under grant number GR/M58030. We would like to thank James Black for his help on transforming our tracking result files into the XML format.

References

[1] S. A. Brock-Gunn and T. Ellis, "Using colour templates for target identification and tracking", *Proc. BMVC'92*, 207-216, 1992.

[2] H. Buxton and S. Gong, "Visual surveillance in a dynamic and uncertain world", *Artificial Intelligence*, 78:431-459, 1995.

[3] I. Haritaoglu, D. Harwood and L. S. Davis, "W4: real-time surveillance of people and their activities", *IEEE Trans. on PAMI*, 22(8): 809-830, 2000.

[4] T. Huang, D. Koller, J. Malik, *et al*., "Automatic symbolic traffic scene analysis using belief networks", *Proc. AAAI'94*, 966-972, 1994.

[5] S. S. Intille, J. W. Davis and A. F. Bobick, "Real-time closed-world tracking", *Proc. CVPR'97*, 1997.

[6] D. Makris and T. Ellis, "Finding paths in video sequences", *Proc. BMVC'2001*, pp. 263-272, 2001.

[7] K. Murphy, "Bayes net toolbox for Matlab", http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html.

[8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference, Morgan Kaufmann*, San Mateo, CA, 1988.

[9] P. Remagnino, T. Tan and K. Baker, "Agent orientated annotation in model based visual surveillance", *Proc. ICCV'98*, 857-862, 1998.

[10] J. Sherrah and S. Gong, "Tracking discontinuous motion using Bayesian inference", *Proc. ECCV'2000*, 150-166, 2000.

[11] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking", *Proc. IEEE CVPR Conf.*, 1999.

[12] C. Wren, A. Azarbayejani, T. Darrell and A. Pentland. "Pfinder: real-time tracking of the human body", *IEEE Trans. PAMI*, 19(7):780-785, 1997.

[13] M. Xu and T. Ellis, "Illumination-invariant motion detection using colour mixture models", *Proc. BMVC' 2001*, pp. 163-172, 2001.