MODEL-BASED VISION FOR AUTOMATIC ALARM INTERPRETATION

T.J. Ellis,
P.L. Rosin,
Centre for Information Engineering,
City University.
LONDON EC1V 0HB.

P. Golton,
Scientific Research and Development Branch,
Home Office,
Horseferry House,
Dean Ryle Street,
London SW1P 2AW.

Abstract. *A previous paper to the Carnahan Conference (Zurich, 1989)* [7] described the development of a knowledge-based vision recognition system for automating the interpretation of alarm events resulting from a perimeter intrusion detection system (PIDS). Measurements extracted over a sequence of digitised images are analysed to identify the cause of alarm. Models are maintained for both alarm causes and the scene and the measurements are matched with the models to derive an appropriate classification of the event.

In this paper we record progress on the further development of the system and present the results of applying it to a number of real alarms. The system is shown to behave robustly, correctly classifying genuine alarm events (i.e. human intruders) and providing statistics of false alarm events. It has also been exercised under a wide range of illumination conditions, for both long-term (e.g. day/night transitions, shadows cast by movement of the sun) and short-term (e.g. clouds, shadows caused by trees moving in the wind etc.) variations.

Introduction

The problem of detecting moving objects in sequences of images is of interest in many fields. Examples include target tracking for the military (e.g. [2]) and visual surveillance systems. Whilst the more general problem of motion includes both motion in the scene and ego motion of the image sensor, surveillance systems are more typically concerned with static camera scenes, locating and classifying object motions [5]. This constraint tends to simplify the task of motion detection and image differencing is a commonly used technique. However, this places considerable reliance on maintaining a reliable reference image, and much work has been done in this area [1].

In an earlier paper [7] we described the first phase in the development of a knowledge-based vision recognition system for automating the interpretation of an alarm event resulting from a perimeter intrusion detection system (PIDS). Sequences of digitised images, captured over the period of an alarm resulting from some physical event (typically, an animal), are analysed to identify the cause of alarm. (A complimentary system has been developed to detect weather-related alarms [4]). The system maintains models of possible alarm causes, as well as modelling the imaged scene. Images are processed to detect and extract measurements from motion-related events and these are matched with the models to derive an appropriate classification of the event.

In machine vision terms, the problem of tracking complex articulated objects in outdoor scenes of the real world using a monocular view is inherently under-constrained. Major problems arise from occlusion, shadows, changing viewpoint, and variations in lighting conditions caused by changes in the ambient light levels due, for example, to clouds. These give rise to segmentation errors and missing data, and introduces uncertainty and ambiguity to the model matching process.

A dominant paradigm in current machine vision systems uses explicit (usually rigid and geometric) models to describe objects. While appropriate for representing a wide class of man-made objects, geometric models do not adequately describe most natural objects such as trees, animals, and landscape because of their wide variety in form and shape. In addition, monocular images of such objects in motion (e.g. animals) introduces further ambiguities due to the changing viewpoint, complicating any recognition scheme based on projected shape profiles.

A further difficulty in the current work of using such models lies in the poor spatial resolution of target objects. As a consequence of the large range of depth (over 100 metres) and a minimum object size which would correspond to a small bird at the extreme of this range, the poor quality of structural information that can be extracted from the detected objects in the image precludes the use of structural models to represent the objects. In its absence, we have represented the objects by simple features (projected area, maximum velocities and accelerations, estimates of object height etc.) and observable behaviour patterns (e.g. birds tend not fly in heavy rain or wind). Without the image resolution to detect structural information, we must accept large error ranges for these parameters, since we will be unable to obtain precise information on viewpoint projection.

To help overcome some of these problems we take advantage of the constraint associated with the static camera position, together with the assumption that significant changes of intensity between images in the sequence correspond to moving objects. On this basis, a simple image segmentation algorithm (image subtraction followed by thresholding) is sufficient to detect movement. A further assumption is that objects are in contact with one of the modelled surfaces in the scene. Using a camera and scene calibration model we can then estimate the distance of the object from the camera and scale pixel measurements into real-world units.

CH2892-8/90/0000-0062 $1.00 © 1990 IEEE

The alarm classification system has been developed within a knowledge-based system (KBS) for vision called FABIUS [8,9]. FABIUS is a frame-based system for image interpretation written in Prolog. It combines the object oriented taxonomic structure of frames, with the problem solving and general inferencing mechanism of a logic language like Prolog. It incorporates mechanisms to support property inheritance, which allows common properties to be inherited by links between frames and decompositional hierarchies which allow complex models to be described by decomposing them into sub-parts. Other features include defaults, value restrictions, and demons, as well as value and relational constraints. Probabilistic updating [6] is used to match image data to object models and determine the classification of the event and image processing algorithms, written in 'C' for efficiency, are called directly from within the system.

This paper records progress on the development of the system and presents the results of applying it to a number of real alarms. The system is shown to behave robustly on the current test data, correctly classifying genuine alarm events (i.e. human intruders) undertaking a range of activities (crossing alarm zones, crawling, climbing fences, etc.). It has also been exercised under a wide range of illumination conditions, for both long-term (e.g. day/night transitions, shadows cast by movement of the sun) and short-term (e.g. clouds, shadows caused by trees moving in the wind etc.) variations. The system is currently being modified to cope with other weather-related "noise", largely resulting from camera movement caused by the wind and is being tested on a wider range of alarm examples.

The next section provides an overview of the initial classification system, described more fully in [7]. Following this, we describe in detail modifications to the system. Finally, we present results of applying the system to a range of real alarm events, and discuss further improvements which are currently being investigated.

Earlier System

Since the cameras are fixed, scene models can be constructed for individual camera positions. Three models were made: an approximate range map, a map of the areas covered by the various alarm sensors, and a segmentation map of the scene in which distinct areas such as ground, fence, sky, etc. are labelled. All these models were stored as images for ease of access. Looking up a value is simply a matter of reading the image at the appropriate location.

Each object model was represented by a frame containing a list of properties to be matched against the properties of features extracted from the image. A pair of weighting values associated with each property are used to update the probabilities for and against the model. Measurements were converted to probability values by one of a set of pre-defined probability distribution functions before the weightings are applied. The size of the object and the range of expected locations were the sole properties used.

Successive frames in the image sequence were subtracted from a reference image which depicted the scene in its undisturbed state. The differenced images were thresholded using a fixed threshold value of 8. Isolated image blobs were detected in the image, and measurements made of their size, centroid, minimum bounding rectangle, location and sequence number, as well as several measures of shape. All objects with an area less than 10 pixels were discarded as likely to be noise.

A co-ordinate from the bottom edge of the bounding rectangle was used to look up the scene segmentation map and label the object's location within the image. The same co-ordinate also pointed to the range of the object in the scene range map, which determined the appropriate scaling factor for calculating the area of the object in square metres.

Object sequences were made up from blobs in consecutive frames with the Euclidian distance between blobs constrained to be less than a threshold. The sequences were matched against the set of models associated with the alarm causes, and classified as the best matching model.

Current System

Image Models and Calibration

In the original scheme, image-based models of the static scene were used to provide identification information about object location (segmentation map), detection sensor zone (alarm map) and a range map. Given the (x,y) coordinates of some particular point of interest in the image sequence (e.g. the location of a detected object) it is a simple matter to determine the object's location and range by looking up this information in the associated map. Similarly, we can determine if a track intersects with any specified alarm zone. Whilst this representation is convenient in image form, allowing a very quick and simple lookup operation to determine object location etc., it is very inefficient in storage terms, requiring 0.25 MByte of image memory storage. To reduce this requirement, at some small sacrifice to speed of access, we encode the boundaries of the prescribed regions using a set of polygonal approximations. The task of extracting object location or alarm zone then becomes a search through the set of closed polygons determining inside which polygon the point lies. For the segmentation map shown in the previous paper [7], only 40 coordinate pairs of points are required to represent the boundaries.

The second part of the scene modelling enables range (distances to the camera) measurements to be made on the objects. Consideration of the geometry of the scene and the focal length of the camera allows adequate estimates of object range. This, combined with a small number of camera measurements, the camera position and the orientation of surfaces in the scene, allows estimates of range to be calculated knowing the coordinates of an object that touches the ground plane or some other modelled surface. (Note: This assumption is invalid for birds in flight. In this case, we tend to underestimate the distance of the bird from the camera, producing an overestimate of the objects size. Similarly, we would tend to underestimate the speed of the object, though this is complicated if the bird is flying directly towards the camera).

Object Models

The frame-based representation supports relationships between models through hierarchical properties and also makes use of this hierarchical relationship in order to associate and match low-level image data with models. Figure 1 shows the hierarchy constructed for the current task. At the top-most level of the hierarchy, alarm causes are divided into two major classes - human and non-human. The non-human class breaks down into further sub-groups associated with other animal types and other possible types of alarm (i.e. genuinely false). Each object model has essentially two parts. The first describes the measurement parameters associated with single instances of the class, whilst the second reflects properties of the motion (maximum speed, acceleration).

Each object model is partitioned into two components: the first part is associated with characteristics of the invidual instances of the animals, and the second

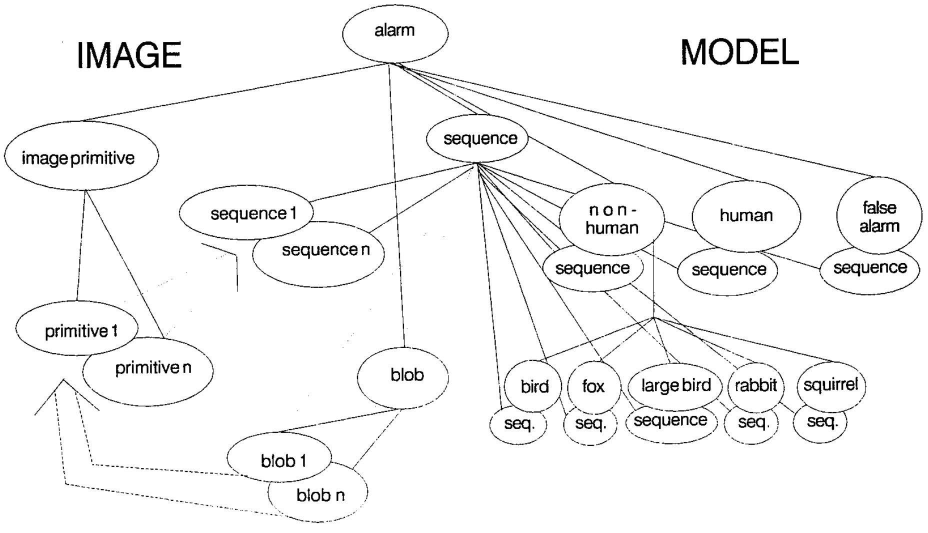

Figure 1. Hierarchy diagram for the interpretation system.

describes the dynamic behaviour of the animals (speed, acceleration). Figure 2 shows an example of a model used to describe a fox. This describes the object as a kind of animal, via which it inherits properties of the general class of animals. The model describes some common physical properties of the fox which must be matched against a set of primitive features extracted from the images. A pair of weighting values are associated with each property, which are used to update the probabilities for and against the current model. Evidence values (measurements) are first converted to probability values (pdf) by one of a set of pre-defined functions (e.g. downslope), before the weightings are applied. The large ranges associated with some of the measurements reflect the uncertainty due to natural variation, and image detection errors.

```
frame fox
            ako           value   alarm
            scaled_area   weight  [1,5]
            scaled_area   pdf     [band,0.06,0.1,0.30,0.35]
            location      weight  [3,10]
            location      one_of  [ground,trees]

frame fox_sequence
            ako           value   sequence
            sequence_of   value   fox
            speed         pdf     [band,0.0,0.0,8.0,15.0]
            speed         weight  [1,5]
            acceleration  pdf     [band,0.0,0.0,0.5,0.6]
            acceleration  weight  [1,5]
```

Figure 2. Model description frame for a fox.

Detection and Confusing Factors

In order to achieve acceptable levels of reliability, the system must be able to cope with a range of confusing factors which can give rise to alarm events, or which may occur during the period of an alarm. Although the object detection mechanism we employ (image differencing) is quick and efficient, it responds equally to both objects of interest, such as animals, and other contrast changes, arising for example from shadows.

In general, in an outdoor scene, the illumination levels change continuously. Long term variations (as opposed to moving objects) arise from changes in the ambient light levels and the gradual movement of object shadows resulting from the changing position of the sun. By maintaining a reference image which is regularly resampled (in time), we can adapt to such variations. Sudden but long term changes (e.g. floodlights turned on) are detected independently of the vision system, and such events can inform the system to sample a new reference image.

Sudden short-term changes generally result from the effects of the wind - fast moving clouds give rise to large scale intensity changes over the image; trees and the shadows they cast can result in objects in the detected image which are characteristic of (usually small) animals.

The differencing technique we have used for extracting areas of movement relies on the availability of good reference images. Reference images that do not accurately depict the same scene background as the image sequence will result in poor difference images, generating spurious blobs. The recording system currently provides reference images updated every fifteen minutes. While this is usually adequate, certain conditions can lead to problems. Early or late in the day, when the sun is low in the sky, objects (such as trees) cast long shadows. Even a delay of fifteen minutes can produce a significant shift of the shadow. Another problem is that since the reference image is automatically generated there is no guarantee that there are no objects such as animals or birds in the scene. When differenced, these will generate spurious stationary animals or birds.

Alternative methods exist which generate a reference image directly from the image sequence [1]. We have developed a temporal median filter for this purpose. Each pixel in the reference image is generated by reading the corresponding pixels at the same location in each of the eight sequence images, and choosing the median of the eight pixel values. When the objects move substantially this produces good results. However, if some pixels are set to the foreground for the majority of the sequence, then the reference image will contain some foreground as well as background.

Currently blobs are extracted from the differenced image after applying a dynamic global threshold. The histogram of the differenced image is calculated, and the potential threshold is stepped down until the

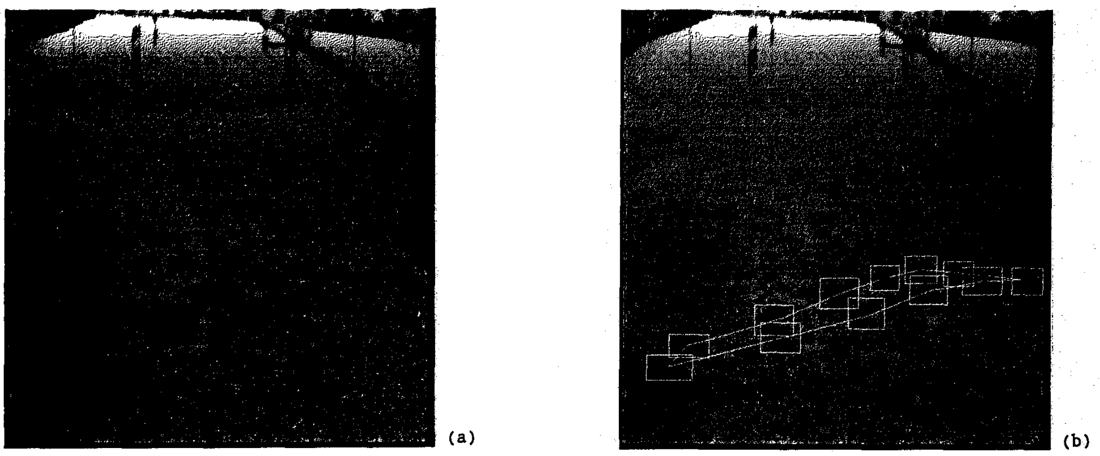

Figure 3. (a) Composite image (over 8 frames) showing two rabbits running through the scene. (b) Results of sequence detection, showing two tracks detected.

thresholded area shows a very large increase. This signifies that the noise level has been reached, and the threshold is stepped back to before the increase. An improvement would apply local thresholds rather than a global one, since local contrast levels can vary considerably across image.

The static camera model provides reasonable detection in the majority of cases. However, in relatively heavy winds, camera shake can severely disrupt the picture subtraction algorithm, generating considerable noise in the differenced image. We are currently using a cross-correlation technique to detect and correct this camera shake by applying shifts to the image. The correlation is performed in the horizontal and vertical directions, using a pair of arbitrary lines through the centre of the image, and are correlated with two equivalent lines in the reference image. In the current implementation, we use a zeroth-order interpolation scheme for correcting the shake, shifting the image with respect to the reference by a whole number of pixels, rounded up from the correlation value. Whilst this does not entirely eliminate the noise from the differenced image, it can considerably reduce it. This correction procedure need only be invoked in windy conditions, a situation which can be anticipated by examining the associated data file for the sequence.

Because the shake is rather fast (due to the stiffness and length of the mounting pole), pairs of image fields may also exhibit considerable differences as a result of using an interlaced image, complicating both the detection and correction processes. To overcome this, we halve the resolution of the image by discarding one of the interlaced fields, and perform the correlation on the remaining field. Of course, this has the effect of reducing the spatial resolution of the image and will reduce our ability to detect very small objects, but under such circumstances we can at least satisfy our main goal of detecting very large (human-size) events.

Sequence Detection

Object recognition is applied to a consistent temporal sequence of detected image blobs. In general, we are not interested per se in tracking objects through the scene. However, identifying blobs that form a consistent sequence is a valuable tool for identifying significant blobs and eliminating "noise" blobs that only occur sporadically. A search is made over the set of image primitives measured from each image in the sequence, and consistent blob sequences are determined with the following criteria:

1) The 3D distance between blobs is less than a threshold which represents the maximum distance an object would be expected to move over the interframe sampling period. Currently, a threshold of 10 metres per second is used, approximating a bird in (slow) flight. As this is only an upper limit, it is a weak constraint, and groups of slow moving blobs will generate many sequences.

2) There must be some consistency in the area blobs within a sequence. However, due to changing viewpoint and the possible articulation of subparts, an object's size may vary significantly. In addition, allowance must be made for measurement errors. Thus, the area of blobs are loosely constrained to lie within a factor of three.

3) The sequence must exist over a contiguous, minimum number of images in the sequence. At present, this is set to 5.

4) The blobs must form a relatively smooth track. This is determined by averaging the magnitude of the acceleration of each blob over a sequence of three, and accumulating this value over the entire sequence.

Candidate sequences are generated over the full set of image primitives, using criteria 1-3, for all possible combinations. Valid sequences are then selected from these sequences by identifying those with the smoothest tracks.

Model Matching

Finally, the valid sequences are evaluated by the probabilistic model matching function within FABIUS. Probabilities are derived from image feature measurements and are weighted to reflect their significance to the object model. Feature probabilities within an object model are combined and propagated up to the sequence model. The sequence is finally classified as the model with the highest probability rating over all te object models, over all the object models.

Results

The image sequences used in the interpretation are typically 8 frames in length, with frames spaced at approximately 0.5-1.0 seconds. The sequence is taken during the alarm event, with typically four images before and four after the actual event. Each image is a 512x512 resolution image, digitised to 8 bits intensity resolution [3].

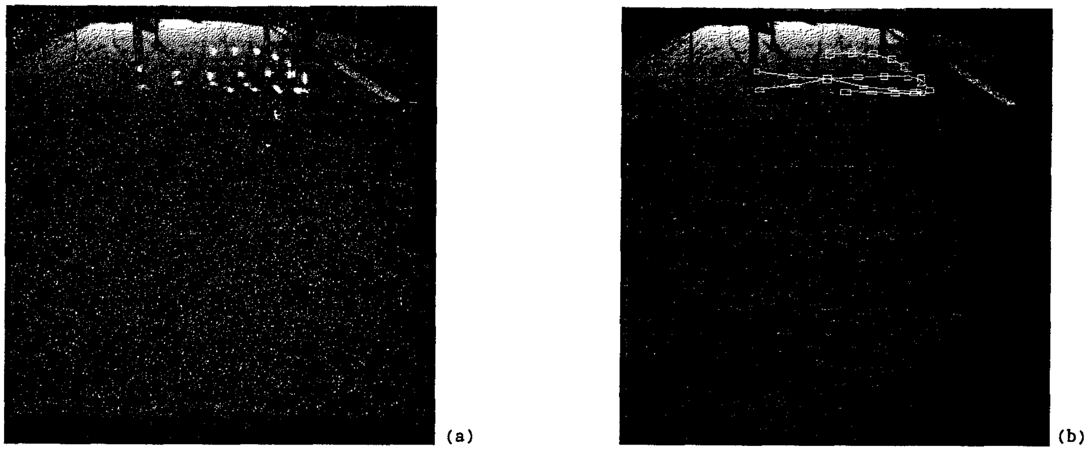

Figure 4. (a) Binary detected objects overlaid onto original image from the sequence, showing two birds (and shadows). (b) Results of sequence detection for four tracks.

This section presents some preliminary results of applying the interpretation system to approximately 30 image sequences. Even for very low contrast images it performed well, reliably detecting movement and extracting the image blobs. Forming sequences has proved effective as a temporal filter, removing noisy image blobs while robustly detecting valid object tracks. In all image sequences reliable human/non-human classification is achieved. In three non-human sequences the subclassification is incorrect, labelling rabbits as large birds, for instance.

Figures 3-5 show examples of the blob and sequence detection. They show the results of processing three example image sequences. Figure 3a is a composite image (formed by combining a sequence of 8 images) of a pair of rabbits running though the foreground. Figure 3b shows the two sets of tracks generated by the sequence detection. Since the rabbits are not present in all eight of the frames, the sequence detector selects a common object for the last frame, apparently merging the two sets of tracks.

Figure 4a shows the binary detection image of two birds (plus shadows) taking off, overlaid onto one of the images from the sequence. Figure 4b shows the four sets of tracks that are detected the motion (i.e. both birds and shadows).

Finally, figure 5a shows the detection image of a person running across the scene. Due to only slight contrast changes between parts of the persons clothes and the tree shadow, the detection image is "broken up", as shown by the set of detected image blobs shown in figure 5b. The sequence detector selects the most consistent set of blobs to form a track, as shown in figure 5c.

We are currently performing more extensive testing on a larger set of several hundred example sequences.

Discussion

Several stages in the image analysis are problematic, and their improvement would increase the robustness and generality of the system. Some of these difficulties are outlined, and possible solutions are given.

Object classification is principally based on size. However, an object's size may vary significantly due to several causes: changing viewpoint (e.g. animal turning), articulation of subparts (bird wings opening and closing), and shadows becoming attached to the object during thresholding. Such variability of size necessitates loose bounds on object model sizes, which in turn causes considerable overlap of object models. Thus, fine distinctions between objects cannot be reliably made. This is the reason several rabbits were mis-classified as large birds.

Camera shake, experienced in strong winds, remains a problem. The correlation method described earlier works adequately if the camera motion is purely horizontal or vertical, but if the camera suffers any rotational movement, then it performs poorly. To correct for this, a more general affine transformation of the image would be more appropriate, but computationally more expensive.

The affects of wind are also apparent for trees, or tree shadows, which are imaged in the scene. These tend to generate a number of both large and small objects in the differenced image, and tend to overload the sequence detection (see below). We are currently trying to generate appropriate models of such tree motion in order to minimise this problem.

Forming sequences is a combinatorial operation since all combinations of blobs through the image sequence that satisfy the loose constraint on an object's upper speed limit are calculated. Only then can the best paths (in terms of smoothness, etc.) be selected. When many blobs are present this technique becomes impracticably slow. Two solutions are currently being considered. Sequences with many small blobs usually arise from flocks of birds. In this case, tracking all the birds is unnecessary. Instead, it is more efficient to directly classify the sequence as a flock of birds if many small blobs exist, but no larger blobs are present. Alternatively, a sequence finder could be implemented that does not consider all possible tracks, but applies heuristics to prune the search space. This would have the advantage of speed, but would result in sub-optimal sequences.

Acknowledgements.

Dr. Rosin acknowledges the support and help of staff of the Home Office (Scientific Research and Development Branch, SRDB) in this project. Dr. Ellis acknowledges the support of the UK Science and Engineering Research Council.

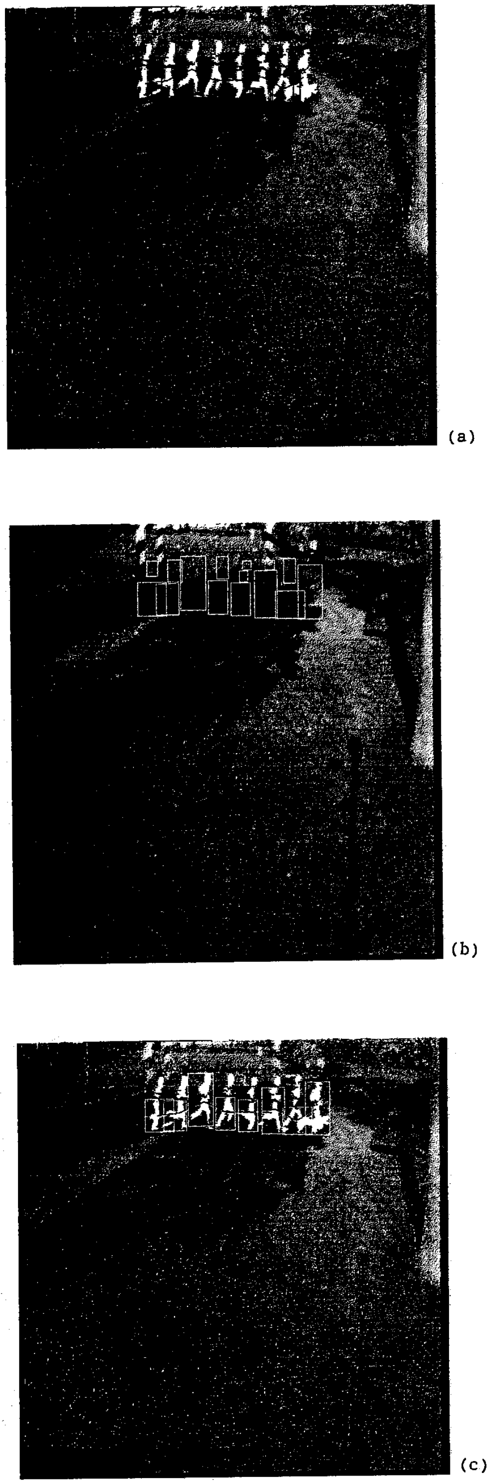

Figure 5. (a) Detected object for a person running across the scene. (b) set of image 'blobs' detected. (c) extracted sequence.

References.

[1] S. Brofferio, L. Carnimeo, D. Comunale, G. Mastronardi, "A Background Updating Algorithm for Moving Object Scenes", in Time-Varying Image Processing and Moving Object Recognition 2 (Proc. 3rd. Workshop, Italy), ed. V. Capppellini, Elsevier, Amsterdam, 1990.

[2] J.F. Bronskill, J.S.A. Hepburn, W.K. Au, "A Knowledge-Based Approach to the Detection, Tracking and Classification of Target Formations in Infrared Image Sequences", Proc. Computer Vision and Pattern Recognition, pp 153-158, 1989.

[3] N. Custance, P. Golton, T.J. Ellis, P. Rosin, P. Moukas, "The design, development and implementation of an imaging system for the automatic alarm interpretation using IKBS techniques", 1989 International Carnahan Conference on Security Technology.

[4] N. Custance, "Perimeter Detection Systems: Correlation of false alarm cause with environmental factors.", 1988 International Carnahan Conference on Security Technology.

[5] I. Dinstein, "A New Method for Visual Motion Alarm", Pattern Recognition Letters, vol 8, pp 347-354, 1988.

[6] R.O Duda, P.E. Hart, N.J. Nilsson, "Subjective Bayesian methods for rule-based inference systems", Proc. 1976 Nat. Comp. Conf. (AFIPS Conf. Proc.), vol. 45, pp. 1075-1082, 1976.

[7] T.J. Ellis, P. Rosin, P. Moukas, P. Golton, "A Knowledge-Based Approach to Automatic Alarm Interpretation using Computer Vision", 1989 International Carnahan Conference on Security Technology.

[8] P. Rosin, "Model Driven Image Understanding: A Frame-Based Approach", Ph.D. dissertation, City University, 1988.

[9] P.L. Rosin, T.J. Ellis, "A Frame-based System for Image Interpretation", to be published.