A KNOWLEDGE-BASED APPROACH TO AUTOMATIC ALARM INTERPRETATION USING COMPUTER VISION, ON IMAGE SEQUENCES

Dr. T. J. Ellis,
Dr. P. Rosin,
Dr. P. Moukas,
Centre for Measurement and Instrumentation,
City University.
LONDON EC1V 0HB.

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

Abstract. This paper describes the development of a knowledge-based system which will be used to automate the interpretation of an alarm event resulting from a perimeter intrusion detection system. The knowledge-based system analyses a sequence of digital images captured before, during and after the alarm is generated. Additional data, pertaining to the alarm sensor, prevailing weather conditions and time-of-day are also available to assist the interpretation.

In order to cope with the diverse nature of the different data sources, a knowledge-based approach is used to perform the interpretation. Models are maintained for a variety of possible alarm causes (human, animal, environmental, false etc.) and each model characterises a number of properties associated with that particular alarm source. The event data is interrogated by the KBS following the selection of a particular model.

Introduction

This paper examines the development of a system to automate the interpretation of the cause of an alarm from a perimeter intrusion detection system (PIDS). The system is knowledge-based, classifying the cause of an alarm by combining information derived from a sequence of images from before, during and after the alarm event with non-visual data relating to environmental (weather, time-of-day, season) and alarm sensor characteristics. A more complete description of the PIDS can be found in an accompanying paper [1].

Alarms arise as a result of the triggering of some detection system and can occur due to a variety of causes. A genuine alarm results from the detection of an intruder (i.e. person). However, in most cases alarms are false, and may be attributed to other animals triggering the sensors, weather-related events, or unpredictable triggering of the alarm system due to noise. In order to verify the cause of an alarm, a image-based examination of the scene has been undertaken. (Note: it is not intended that the current work be used to determine alarms caused by weather-related events, as this has been the subject of a separate project on knowledge-based classification [2].)

In some cases, the image sequence alone contains insufficient detail or resolution to accurately identify the cause of the alarm, but can be used in conjunction with other available information to construct a valid and consistent interpretation.

Figure 1 shows an example of an image sequence spanning an alarm event. Figure 1a is a single image showing a pheasant activating a microwave beam alarm. Figure 1b is a composite picture assembled from a sequence of 8 images, spaced approximately 0.5 seconds apart. A second example, of a running dog, is shown in figure 2.

Interpreting visual data from real-world scenes, where a two-dimensional image is formed by projection of complex three-dimensional objects, is a difficult and demanding task. On the one hand, the digitised image simply consists of a sampled array of light intensity values. On the other hand, a complete interpretation requires the attachment of semantic labels to objects in the scene (e.g. a person walking, a house, a road etc.) as well as labels defining the relationships between objects (e.g. a person in a car).

Early stages in the image processing attempt to separate individual objects from the background, or to identify regions of interest. This image segmentation process is typically based on detecting regions of similarity (intensity, texture, colour etc.) or the boundaries (edges) between dissimilar regions. Following segmentation, a feature extraction process encodes the detected regions into primitive features. Where these primitives represent simple measurement features (size, length, shape etc.), statistical feature space analysis techniques can be used classify objects. However, in the majority of problems, classification is best achieved by matching the primitives against some appropriate model representations of the objects of interest.

Identifying motion-related information contained in a sequence of images, taken from natural, outdoor scenes can significantly complicate the interpretation process. The interpretation may now be called on to perform additional steps in order to interpret actions (e.g. a person climbing a fence), as well as the other recognition tasks. Whilst on the one hand, the inter-frame analysis can be used to reinforce interpretations, the additional processing overhead and changing views of objects in motion can introduce confusing and ambiguous information.

The first part of this paper examines some of the

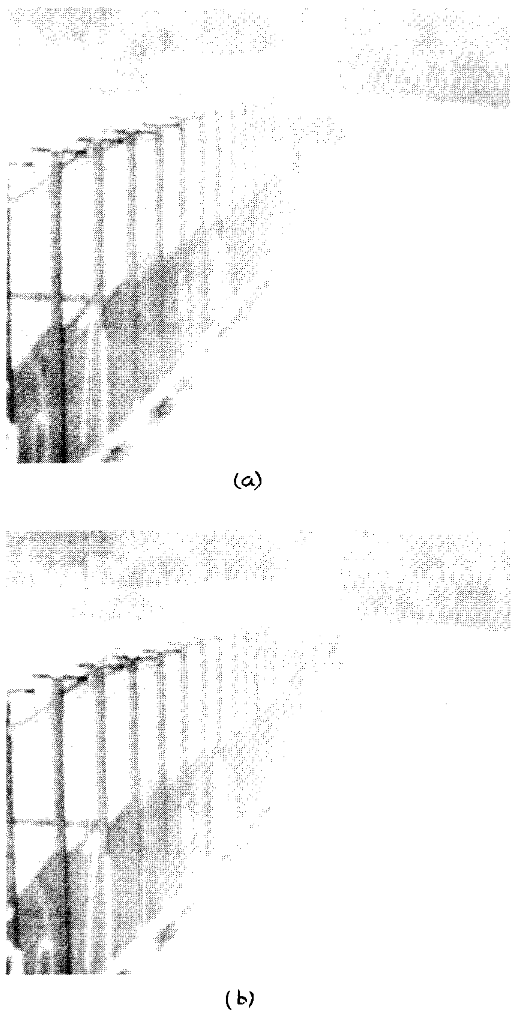


Figure 1. Example of an image sequence. 1a shows a single image from the beginning of the sequence. 1b shows a composite picture assembled from a sequence of 8 images.

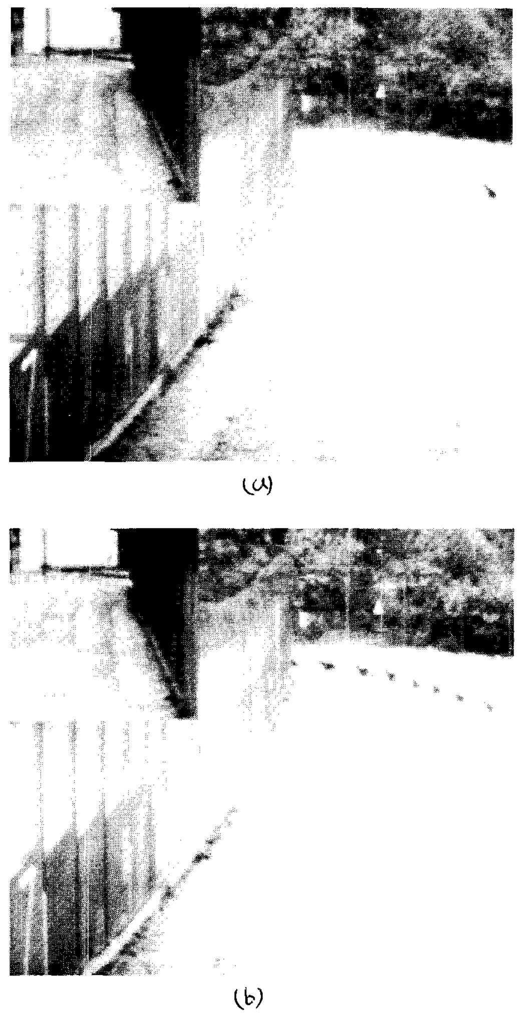


Figure 2. Second example of an image sequence, showing a dog running through the scene.

general problems associated with analysing images in order to interpret information pertaining to motion-related events. The second part details some of the advantages of using a knowledge-based approach for representing models which describe the cause of such events, and briefly describes the system which is being used for the current task. The third part details the operation of the interpretation system and shows some results from several examples.

Motion Analysis

Generally, the dynamic scene analysis problem can be tackled either by using differencing and/or correlation in order to detect motion and locate changing regions in the image. Subtracting two registered contiguous images of a scene will result in a difference image that will be zero in all areas that have not changed between the two images. Sections of images that undergo movement can be thus isolated and any further processing can thus be concentrated on those segments [3].

These techniques enable tracking of the moving object(s) in consecutive frames. They are effective in simple cases e.g. when the camera is fixed and the objects do not change their size and/or aspect from frame to frame. The fundamental problem with most cross-correlation and subtraction schemes, however, is the underlying assumption that the image moves as a whole from frame to frame, and that any objects depicted in the image present the same aspect. Images containing independently moving objects with occlusions, changes of aspect and articulated non-rigid objects pose difficult problems to these techniques [4]. Nevertheless, such methods enable the implementation of useful motion detectors and can form a basis for implementing higher-level inferencing mechanisms which can establish definitive relationships among motion parameters, object characteristics and the number of regions in difference images.

In more complicated cases, however, e.g. when a complete interpretation of the scene is required, this low-level approach is not sufficient. Higher-level schemes of optical flow or feature correspondence come into play. These schemes establish velocity vectors representing the velocity of change of the image intensity structure (e.g. gradient) or image features (points, edge segments, etc.), respectively. From the visual motion vector field thus estimated, under certain circumstances represented as constraints in an optimisation problem, the 3-dimensional structure and motion in the scene can be estimated.

The correspondence process [5,6] establishes a match of features from frame to frame. Originally, the features used were the elements of the primal sketch [7] and the correct correspondence was that which maximised the overall similarity between frames. The similarity was defined as the sum of cost functions that give a similarity value to each pairing in the two frames. In general, matching is achieved by successive approximation under a relaxation scheme. Various correspondence algorithms differ only in the complexity and dimensionality of the matching primitives chosen, and the type of similarity or affinity measure used [8].

Correspondence and optical flow can be used for recovering 3-dimensional structure of the scene depicted in a sequence of images from motion. But this has so far proved to be a very difficult task indeed, and has only been tackled under very constrained circumstances. To ensure that the 2-dimensional velocity field that is computed from the changing image corresponds to the true projected velocity field of the actual surfaces moving in space, requires additional constraints based upon assumptions about the physical world which generally hold true. A very general intuitive assumption is that the physical world consists of predominantly solid objects with smooth surfaces which usually generate a smoothly varying velocity field.

The simplest additional assumption is that the velocity field is constant over an area of the image. This type of field is usually the result of pure translation along a straight line parallel to the image plane at constant speed. Most schemes are based on assumptions of increasingly complexity about the motions that take place in the scene i.e. translations and/or rotations and the way these scenes are projected on the image plane i.e. orthographic or perspective. In addition, assumptions about the shape of the moving object(s) are also made i.e. curved or polyhedral rigid bodies. It is clear that some form of models of the environment are assumed.

Optical flow and correspondence approaches presently seem to be at experimental stages and have not yet been implemented in a working system. Their role is, at the moment, to provide an understanding and a computational model of the visual processes of motion. The solution of the optical flow and correspondence problem is heavily dependent on iterative methods of either solving differential equations or relaxation techniques and are therefore very time-consuming. Nevertheless, these approaches seem to be the ones that promise a general solution to the dynamic scene analysis problem. Very recently, however, the idea of correspondence as an independent preliminary process for the further analysis of dynamic images has come under severe criticism [8]. Algorithms based on simple token matching have not been successful in most applications and this has led to the definition of increasingly elaborate and sophisticated token models and matching criteria. Additional psychophysical evidence indicates that Ullman's model [5] is in need of review.

On the other hand, the lower level pixel-based techniques seem to be suitable for practical applications due to their speed. But the range of problems that can be solved is very small and the solutions have to be tailored to the application by exploiting all a priori knowledge about the environment and the expected events in the image sequence. Several ad hoc schemes based on an assortment of pixel-based techniques combined with intelligent heuristics have been reported with various degrees of success.

Knowledge-based Classification

Classification based on traditional pattern recognition systems are generally application dependent, and restricted to highly constrained scenes. More sophisticated techniques are required for a flexible vision system that can deal with less structured and more varied environments. Due to the variability of scenes and viewing conditions such scenes tend to provide only uncertain, incomplete, and ambiguous data. Intelligent knowledge-based systems (IKBS) use large amounts of domain knowledge to aid interpretation of such data. This knowledge can be complex, encompassing hierarchical, causal, uncertain, ambiguous, and incomplete knowledge. Since knowledge is seen as the key to high performance systems, the choice of knowledge representation is important.

Choice of knowledge representation is determined by what knowledge is to be represented; in this case the models are of causes of alarm triggers. With the available images, visual appearance of alarm triggers is often insufficient to distinguish the different causes. Therefore the models include in addition to their visual properties non-visual properties such as ranges of possible speed of travel, and environmental factors such as the effect adverse weather conditions have on the daily movements of animals. Since likely candidates for the cause of alarm triggers are found by differencing several frames from a video loop, moving objects show as a sequence of related objects through the frames. Object sequences must be modelled, including knowledge about constraints on the change of size and location of the objects throughout the sequence.

Knowledge representations can be split into declarative and procedural approaches, i.e. the distinction is between "knowing what", and "knowing how". Declarative representations describe the static aspects of knowledge: facts about objects, events, and their relations. These facts are manipulated by a control procedure, kept separate from the declarative knowledge. Procedural representations embed knowledge within programs, which can be used to find relevant facts, and make inferences; the knowledge is implicit in contrast to the explicit knowledge in declarative representations.

Commonly used knowledge representations include production systems, semantic networks and frames. Each scheme has its advantages and disadvantages; many strengths of one are the weaknesses of the other. Declarative representations offer readability, flexibility, and ease of modification. Procedural representations have the advantages that meta-knowledge and heuristics can be easily represented, enabling directness of line of inference; ease of coding; and understandability of the reasoning process.

The representation used for the current task is based on a declarative, structured object representation, which combines related items of knowledge into chunks. These chunks are known as frames. In frames, all assertions about a particular entity are held together. A frame is the basic unit of knowledge, and consists of some slots that describe attributes of that frame. A frame can describe either classes or individual instances of stereotypical objects, acts or events. By supplying a place for knowledge the slot mechanism facilitates expectation driven processing.

Although slots will initially be empty, they may have a range of expected values, which can act as a frame person

ako	value	mammal
date_of_birth	value	[29,10,1953]
age	if_needed	calculate_age
height	if_needed	query_user
height	default	average_height

Figure 3. Example of a simple frame. Two demons are attached to slots which do not have values. The value slots will be filled-in after the successful completion of the demon. If, for some reason, the "query_user" does not return a value, the default value (defined as a persons average height) may be used.

validity check, and default values, which can substitute for known values if they are unavailable. If the frames are structured as a hierarchy, slots may inherit values from similar slots in ancestor frames.

In addition to its declarative aspect, procedures can be attached to slots within a frame to drive the problem solving behaviour of the system. "Attached procedures" or demons are triggered when slot values are accessed. Take for example a frame describing a person, including amongst its slots one to store the date of birth. That slot could have a demon triggered when the slot is filled, which will execute a procedure to calculate the person's age and store it in the age slot. In addition, when the slot is read, if there is no value, another demon could be triggered that obtains the required value, either by calculating it, or by asking the user.

FABIUS [9,10] 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 in an **ako** (a-kind-of) specialisation hierarchy. In addition, decompositional hierarchies are facilitated through **ipo** (is-part-of) links, allowing 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.

Models of objects are represented as frames, to be matched against sensor (image) data as each model is evaluated. One of the principal advantages of using a knowledge-based system for interpretation is that it allows the knowledge (including the extracted data) to be made explicit, and kept independent from the control mechanism which is performing the interpretation. Hence it is a much simpler task to develop and test alternative evaluation strategies. In addition, since demons can be attached arbitrarily to frame slots and activated on access, much of the algorithmic computation involved in analysing the data can also be separated from the control mechanism, to be triggered not by explicit function calls, but only when a particular value or data is required.

This matching operation must attempt to cope with a variety of confusing and ambiguous processes which distort the image data. Moving objects in the scene can become occluded or obscured as they pass behind or in front of other objects. Objects are inevitably self-occluding, hiding details according to the view or aspect currently projected onto the image frame. In addition, over the sequence of images, an object will inevitably present differing viewpoints, adding to the confusion.

A number of methods may be used to determine how well a particular model matches the data. FABIUS uses a variation of Baye's theory [11] about conditional probabilities called Subjective Bayesian Updating [12] as used in Prospector, a well-known KBS for analysing mineral deposits. Probabilities propagate up from the leaves of a frame tree until they reach the root (top level model) hypothesis. In other words, the calculation of the hypothesis of each frame in the tree takes as its items of evidence the values of its children's hypotheses.

As well as representing individual objects, frames can also be used to describe groups or combinations of objects. For instance, a sequence frame can be decomposed into a number of individual sequence element frames. Associated with the frame is a set of constraints over its sub-elements, constraining the elements to belong to sequential image frames, and defining their variability of size and proximity.

Models

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 four images before and four after the actual event. Each image is a 512x512 resolution image, digitised to 8 bits intensity resolution. Associated with each image sequence is a data file (see figure 4) which describes the conditions under which the sequence was acquired: alarm sensor type, camera location, weather conditions, inter-frame time interval etc.

```
          DESCRIPTION:   A, SRDB018
          DATE/TIME  :   29/04/88  05:26:43
          CAMERA NO  :   20
          ALARM NO   :   18
          IMAGE SIZE :   512
          NO. IMAGES :   8
          FRAME DELAY:   24
          LIGHTS     :   ON
          WIND DIR   :   E
          WIND SPEED :   2
          RAINING    :   NO
COMMENTS : Pheasant in foreground ,birds in distance.
           One or both break microwave beam.
```

Figure 4. Example of event log file for image shown in figure 1.

Various kinds of model are used to perform the interpretation. Since the camera is viewing an essentially static scene (typically, only strong winds affect this, occasionally causing small movement of the camera), image models can be constructed for individual camera positions which allow estimates of the range of an object (based on it actually touching one of the modelled surfaces, e.g. the ground or the fence), the area covered by various alarm sensors, and a segmentation map of the scene, which can be used to determine the objects location in the image. All these models are stored as images, so that looking up a value becomes simply a matter of accessing an appropriate location in the image to determine, for instance, whether the object is located on the ground, on the fence, in the trees etc. An example of an image segmentation model is shown in figure 5.

Figure 6 shows an example of a simple object model (a dog) using the frame description language of FABIUS. This describes the object as a kind of mammal, via which it can inherit properties of the general class of mammals (not defined in this case). The model describes some common physical properties of the dog which must be matched against a set of primitive features extracted from the images. A pair of

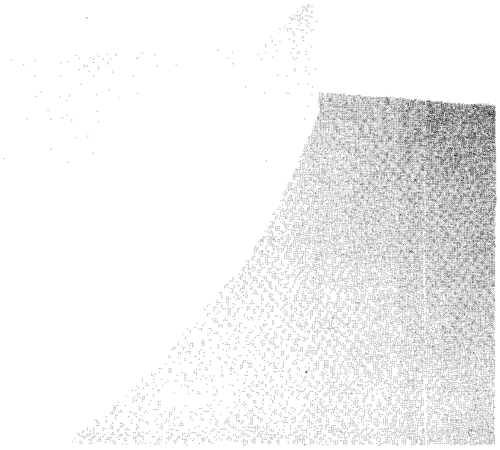


Figure 5. Example of one of the image models which are constructed manually. This shows the segmentation map which is used to determine the location of an object in the scene. For this camera position, only three areas are defined: ground, trees and fence.

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.

```
frame dog

  ako        value      mammal
  size       if_needed  get_size
  size       weight     [1,3]
  size       pdf        [band,0.2,0.4,0.8,1.0]
  size       maxmin     [1,0]
  location   one-of     [ground]
  location   weight     [1,5]
  location   pdf        [downslope,0,1]
  location   maxmin     [1,0]
  max_speed  weight     [1,2]
  max_speed  pdf        [downslope,1.0,2.0]
  max_speed  maxmin     [1,0]
```

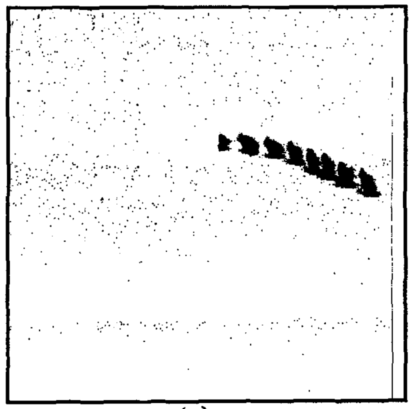
Figure 6. Example of a crude model for a dog.

Image Interpretation

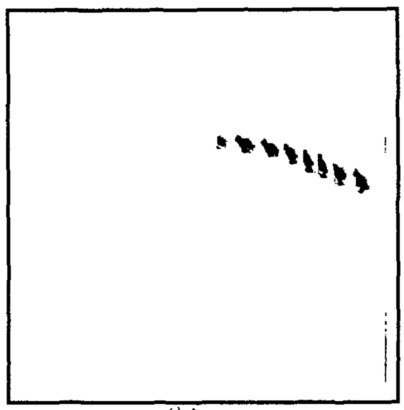
In its present state, the control algorithm initially adopts a bottom-up processing strategy, which first generates a set of measurements from the image for each of the detected objects (image primitives). These objects are first detected by subtracting successive frames in the image sequence from a reference image of the scene. This reference image depicts the scene in its undisturbed state. The differenced images are then thresholded using a threshold value of 8, which adequately suppresses the pixel noise in the image whilst acting as a sensitive detector of genuine of motion-related events. Measurements are then made of isolated objects detected within the image, determining size, location and sequence number, as well as several crude measures of shape (compactness, aspect ratio etc.). A simple threshold is used to discard objects which are likely to be noise, based on a very low area (currently, a value of 10 pixels is used).

Figure 7(a-h) shows a set of thresholded binary images, subsequent to image differencing, of the image sequence depicted in figure 1. Figure 7i shows the result of combining the binary information from all eight of the detected images in the sequence. As can be seen, a large number of small noise points are generated by the grey-level thresholding of the differenced images. A total of 181 objects are found in this combined image, of which only 29 are retained, following the area thresholding operation.

Figure 8a shows the results of similar processing applied to the sequence in figure 2. One drawback of using a fixed value for the image thresholding is that strong shadows (as seen in this image) are detected equally with the dog. Figure 8b shows a similar combined sequence, obtained using a threshold level of 28.



(a)



(b)

Figure 8a. This shows a composite detected image for the image sequence shown in figure 2. Figure 8b shows the results of processing the same sequence but with a threshold level of 28 because of the strong shadow cast by the dog.

Measurements on individual objects are asserted into the PROLOG data base as a frame. A typical example of an image primitive frame is shown in figure 9. Some of the measurements are those directly derived from the image. The centroid value represents the first order moments of the binary object and can be used to crudely locate the object relative to other primitives. The rectangle value is a list which contains the minimum and maximum X and Y co-ordinates contained in the object boundary. One of these values

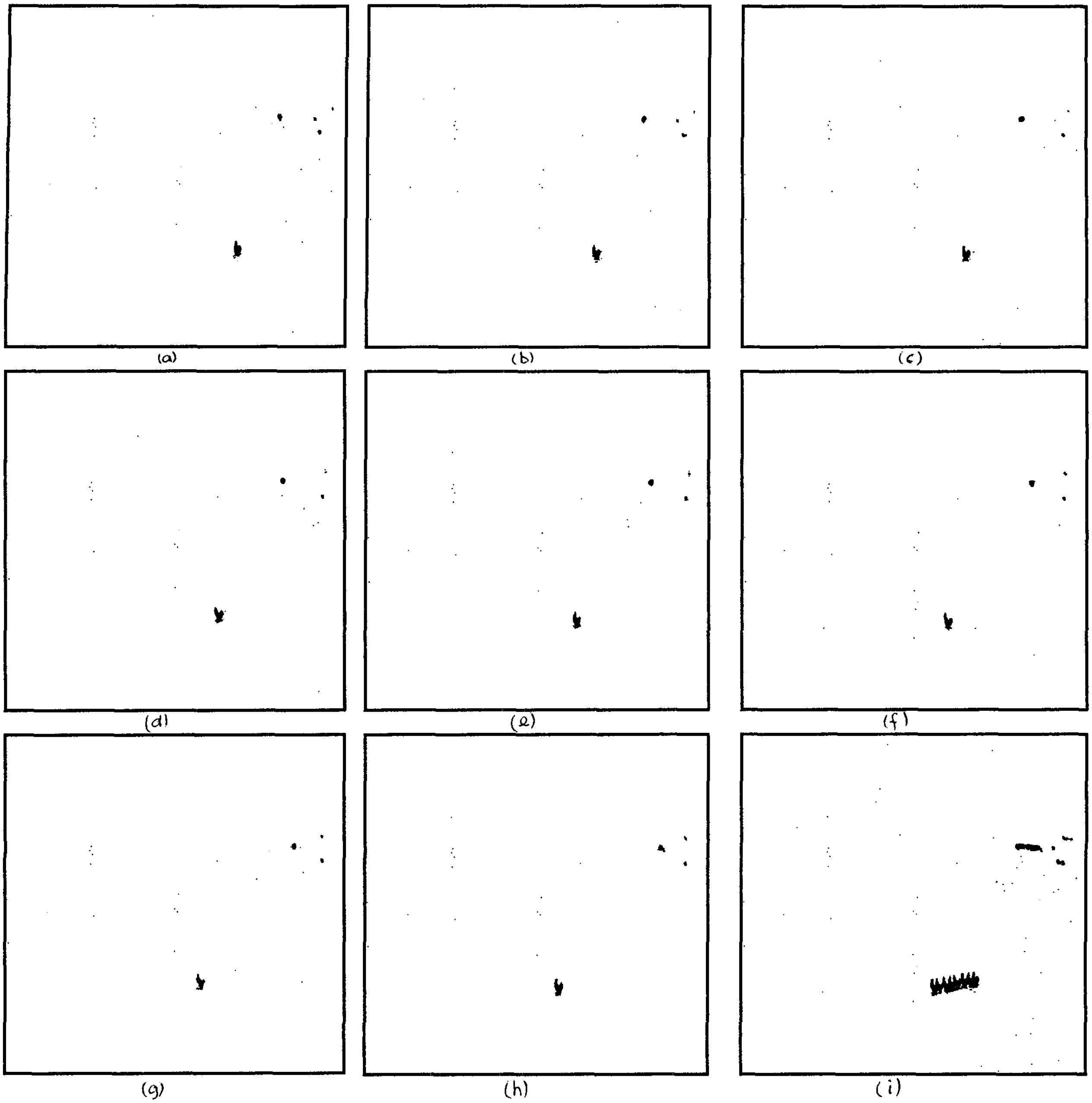


Figure 7. 7a-h shows a binary image, obtained by thresholding the differenced images, using a threshold level of 8. The objects detected include the pheasant in the foreground and a group of birds (on the ground) in the top right corner. 7i is a composite image, made by combining images 7a-h.

frame primitive1

ako	value	image_blob
alarm	value	[11,18]
area	value	43
centroid	value	[410,140]
first_coord	value	[408,136]
location	value	ground
perimeter	value	38
range	value	77.0000
rectangle	value	[407,136,413,144]
scale	value	0.0017
scaled_area	value	0.0741
seq_no	value	1

Figure 9. Example of a frame of data from an image primitive.

(Ymax) is used in conjunction with the segmentation map to determine the actual location of the object within the image (e.g. on the ground, on the fence etc.). This co-ordinate is also used to lookup the range of the object from the associated range map, and hence to determine the appropriate scaling factor for calculating the area of the primitive.

The next step in the evaluation is the assembly a likely set of image primitives which are consistent with an object moving through the scene. This is a two stage process, with image primitives being instantiated to "blob" frames in order to ensure that the matched primitives are consistent: in this case, that the first blob is instantiated with a primitive from the first frame in the sequence, the second blob from the second etc. Frame Blob is defined (temporarily) as an "ako dog", so that image primitives matched to blob1-8 are compared with the dog model. In addition, a further constraint is used to ensure that the euclidean distance between consecutive blobs is within some threshold. The second stage defines the sequence frame (shown in figure 10) which is used to assemble such a set of image primitives. The inventory list describes the set of primitives (via an intermediate frame, the blob) which might form such a set, and associated with the complete set are several relational constraints which apply a more global proximity constraint over the entire set of blobs.

Figure 11 shows the results of this sequence extraction process. Each of the binary objects measured from all of the detected images in the image sequence are candidates for this process. One of the consistent sets which have been selected are shown bordered.

Given a consistent set of image primitives that form a sequence, the individual blobs in the sequence are matched against the set of models associated with the alarm causes. This allows the interpretation system to classify the sequence as a dog sequence, or a bird sequence or a man sequence according to the degree of match with each of the models over the sequence (i.e. the model with the highest probability).

Discussion.

This paper has described the initial stages in the development of a knowledge-based system for identifying the cause of alarms in a PIDS. Analysing a sequence of images from static cameras sufficiently constrains and simplifies the motion analysis, in order to allow an interpretation. In the first instance, the image can be easily modelled prior to analysis. This provides a simple mechanism for determining the location and range of an object detected in the image. Secondly, objects can be reliably detected by image subtraction, and fall into

frame sequence

ako	value	thing
inventory	value	[blob1,blob2,blob3, blob4,blob5,blob6, blob7,blob8]
constraints	value	[all_near(blob1,blob2, blob3,blob4,blob5, blob6,blob7,blob8)]
blob1	weight	[1,1]
blob2	weight	[1,1]
blob3	weight	[1,1]
blob4	weight	[1,1]
blob5	weight	[1,1]
blob6	weight	[1,1]
blob7	weight	[1,1]
blob8	weight	[1,1]

frame blob

ako	value	dog
seq_no	if_needed	get_seq_no
seq_no	weight	[1,5]
seq_no	pdf	[band,1,1,8,8]
seq_no	maxmin	[1,0]

frame blob1

ako	value	blob
constraints	value	[check_seq_no(blob1,1)]
primitive	value	image_blob

frame blob2

ako	value	blob
constraints	value	[check_seq_no(blob2,2) near(blob1,blob2,20)]
primitive	value	image_blob

Figure 10. This describes the hierarchy of frames that extract a valid sequence of primitives and attach them to candidate "blob" frames and then assign them to the frame sequence. Frames for blobs 3 to 8 are ommitted, but are identical to blob2, except for the appropriate tags in the constraint slot.

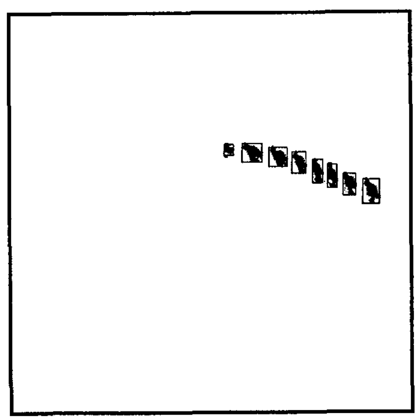


Figure 11. This shows the results of applying the model evaluation to the binary image in figure 8b. The objects selected for the sequence from the set of measured primitives extracted from the image are shown bordered.

only a small number of classified categories. This is ultimately only two (i.e. human and false), but this initial investigation is using a wider range of classification groups in order to more precisely categorise the false classes.

The system has thus far been developed on only a small number of examplar images. Work is in hand to test it on a larger number of alarms. It is anticipated that the current models will require considerable refinement and that alternative control strategies will be developed.

There are several shortcomings with the current implementation which have yet to be broached: camera movement during the sequence (caused by strong winds) can produce significant false alarm features in the detected image. However, since a measure of the windspeed is available from the event log, such conditions can be anticipated, and it is intended that models will be developed to interpret such data. Short term lighting variations (i.e. within the sequence) can result in more significant errors in the detected image and are more difficult to predict from the event log. Again, models of such events will be developed to cope with this problem. Other problems are caused by flocking birds, which are too large in number to be tracked in the same manner as for small numbers of slower moving objects. However, the very nature of this type of event makes it fairly unique, and hence easy to detect and assign an appropriate interpretation.

The flexible framework provided by the KBS provides a powerful and easily updateable environment for developing models and control strategies, and represents an excellent tool for such tasks.

Acknowledgements.

Drs. Rosin amd Moukas acknowledge the support of the Home Office (SRDB) in this project. Dr. Ellis acknowledges the support of the UK Science and Engineering Research Council.

References.

[1] 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.", presented at 1989 International Carnahan Conference, Zurich, 3 October, 1989.

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

[3] R. Jain, "Dynamic Scene Analysis Using Pixel-Based Processes", Computer, 14(8), pp 12-19, 1981.

[4] S. Ullman, S, "Analysis of Visual Motion by Biological and Computer Systems", Computer, 14(8), pp 57-69, 1981.

[5] S. Ullman, "The Interpretation of Visual Motion", Cambridge, MA: MIT Press, 1979.

[6] J.K. Aggarwal, L. S. Davis, & W.N. Martin, "Correspondence Processes in Dynamic Scene Analysis", Proceedings of the IEEE, 69(5), p562-572, 1981.

[7] Marr, D. "Vision", Freeman Press, 1982.

[8] M. Jenkin, D. Kolers, "Some Problems with Correspondence", in Motion Understanding: Robot and Human Vision, In Martin, W.N. & Aggarwal, J.K. (Eds.), Boston: Kluwer Academic Publishers, pp.269-296, 1988.

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

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

[11] R.O. Duda, P.E. Hart, "Pattern Classification and Scene Analysis", Wiley, New York, 1973.

[12] 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.

Data on Influencing Factors

In drawing up a picture of the general situation in which users of PIDS find themselves it is appropriate to look at the performance of systems from Home Office test sites and operational environments. Table 1 indicates the methodology used to classify the cause of an alarm. It is important that this table be studied as all alarms which are not due to human intervention are classified as "false". This terminology is different from that adopted by some organisations, but reflects the concern felt by users of systems, namely that any non-human trigger of a PIDS causes an action to be carried out in a control room. The need for the control room to act to a false trigger creates an unnecessary financial burden on the organisation. Table 2 shows the breakdown of alarm