http://dx.doi.org/10.1109/TENCON.2000.888772

Fung, C.C. and Jerrat, N. (2000) A neural network based intelligent intruders detection and tracking system using CCTV images. In: TENCON 2000, 24-27 September 2000, Kuala Lumpur, Malaysia, pp II-409-II-414.

A Neural Network Based Intelligent Intruders Detection and Tracking System using CCTV Images

Chun Che Fung and Nicholas Jerrat
School of Electrical and Computer Eng.
Curtin University of Technology
Bentley, Western Australia 6102
email: tfungcc@curtin.edu.au

Abstract: This paper reports the development of a neural network based intelligent intruders detection and tracking system using Closed-Circuit Television (CCTV) images. It examines the techniques and algorithms used to identify a potential intruder and methods to eliminate other non threatening objects. Once the presence of an intruder is determined, the object will be monitored and tracked. The tracked information can be used to further identify any suspicious behaviour in the sparse and complex environments. The traditional approach to Intelligent Scene Monitoring (ISM) is examined and compared with the artificial neural network (ANN) approach. The ANN approach demonstrates how a system can learn how to distinguish suspicious movements from non-suspicious movements. The proposal has a potential to be used as an intelligent surveillance system.

I. INTRODUCTION

All over the world, security industry is growing at a rapid rate. In particular, more and more video surveillance systems have been installed for the monitoring of public places and private premises. While most of these systems require human operators to monitor the CCTV images at a centralised location, studies have shown that the operators suffer from a rapid loss of concentration once fatigue sets in. In addition, they have limited capability to monitor more than a few cameras at any given time. It is therefore desirable to have an automated system which does not suffer from these limitations.

An example is the Video Motion Detection (VMD) system which monitors live images from CCTV systems and uses detection and tracking algorithms to raise an alarm if an intruder is present. Ideally, it should be able to make an assessment of any detected motion as either a genuine intruder or a false alarm based on a sequence of tests or a set of criteria. If the detected motion fulfils the conditions in all the tests, the system will conclude that the motion is due to an intruder. In this sense, a VMD system can be considered as an intelligent system in that it performs the job of the operator by acting as an agent that perceives and acts based on the changes in the environment. Once a motion has been detected, the object should be tracked. A technique used to track intruders is presented in this paper as the Intelligent Scene Monitoring (ISM) algorithm. This algorithm eliminates any remaining false alarms and determines if the remaining units are exhibiting suspicious behaviour. A neural network based approach is introduced to learn what constitutes suspicious behaviour for the given environment.

II. VIDEO MOTION DETECTION

Basically, Video Motion Detection (VMD) is confronted by a number of serious technical difficulties. VMD systems must differentiate between an intruder and environmental conditions such as rain, wind, fog, birds, animals, and lightning effects. Table 1 shows the degree of difficulty in rejecting common false alarms [2].

Table 1 – Degree of difficulty of detecting false alarms

Grass movement	Easy
Wind	Not so easy
Falling rain	Not so easy
Wind blown debris	Not so easy
Driving rain	Not so easy
Hail	Not so easy
Small animals	Difficult
Birds	Difficult
Large animals	Difficult
Legitimate human activity	Very difficult

The VMD system must attempt to minimize the number of false alarms which can be defined by FAR, *False Alarm Rate* while keeping the sensitivity of the system as high as possible. The success of detection can also be defined by POD, *Probability Of Detection.* A successful VMD system will attempt to provide a significant reduction in FAR with a minimum reduction in POD. It attempts to make an assessment of any detected motion as an actual intruder or as a false alarm using a sequence of tests. If the detected motion passes each test the system concludes that the motion is an intruder.

A video motion detection system would normally rely on dedicated hardware to grab the image of each frame from a

0-7803-6355-8/00/ $10.00 © 2000 IEEE

CCTV. It then converts the images to a digital format and processes the data using a hardwired implementation of a detection algorithm. As the processing speed of the PC has increased and Charged couple device (CCD) cameras using high speed PC interfaces such as PCI or USB buses have become widely available, the PC has now become the new platform for VMD. However, a high-end PC or multi-processor system is essential for satisfactory performance, as the throughput required for the analysis of real time video images is extremely high. Some systems decrease the processing requirements by operating at a reduced frame rate, which can be quite adequate in most circumstances.

In the majority of the systems, monochrome images are used as they reduce the bandwidth required. In addition, colour images do not usually add to the reliability of the system. Eight-bit greyscale of values between 0 (black) and 255 (white) are often used (they are referred to as intensity from here on). If a colour system is in use, RGB images can be converted to greyscale on the fly by taking an average of the red, green and blue components for each pixel.

A. Motion Detection Algorithm

A typical detection algorithm comprises the following steps:

1. Grab the current video image.
2. Compare to the previous image and to determine whether there is significant motion.
3. Identify the units causing the motion if any.
4. Pass each unit through a series of tests to eliminate false alarms.
5. Trigger a response if a genuine intruder is detected.
6. Repeat the above steps.

Step 2 is accomplished by comparing the intensity (greyscale) values of the current image to that of the previous. In general, the current image is always compared to the previous, as small changes in the image such as global lighting effects (eg. the sun) can be integrated as time progresses without triggering an alarm condition.

Instead of comparing the pixels one by one, the current and previous images are divided up into blocks, usually of about 5 by 5 pixels. This is done to reduce the storage requirements and the impact of random pixel size changes. However, given the availability of high throughput machines, comparisons based on pixel-sized blocks could also be done. For each respective pixel in a block, the difference between the intensity of the current image and the previous is calculated. The average of the differences in the block is then determined. If the average of the differences exceeds a predetermined threshold value, the block is marked as *active*.

This procedure is performed for each block. The number of active blocks is summed and compared to a predetermined block threshold. If the number of blocks exceeds the block threshold, a simple VME algorithm would trigger an alarm. More complex algorithms continue as detailed below.

B. Intruder Identification

For each block marked active, the number of neighbouring active blocks is determined. Each group of blocks is referred to as a *unit*. A number of tests based on location, size and shape are now performed on this block to determine if it fits the profile of an intruder [3].

Each test uses a threshold value. Appropriate thresholds are determined by the environment being monitored by the VMD system, and are assigned during installation.

a. Location Test:

The location test is used to eliminate certain previously identified parts of the image from analysis. Some parts of images may be too oscillatory to monitor. It may also be highly unlikely that an intruder may be located in particular regions of the image. For example, it may be unlikely for an intruder to be detected high up in the sky. Units located in these areas can be ignored as false alarms. Care must be taken in using this technique, as an intruder with knowledge of these locations may be able to avoid detection. If the unit is partially located within an ignored region it is good practice to include the unit in further analysis. An expression of the Location Test is given below.

Is pixel (x,y) located in square region
$(a,b) \rightarrow (a+5,b+10)$?
(If $x \geq a$ AND $x \leq a+5$ AND $y \geq b$ AND $y \leq b+10$)
then ignore (x,y) (1)

b. Area Test:

The area test can be used to obtain the relative size of an object. This measure can give an indication of whether the object is sufficiently large to warrant further analysis. The area test essentially adds up the number of pixels in the unit and if this value is greater than a predetermined area threshold, the unit is elevated to the next test. Each pixel can also be weighted by a size factor. This technique must be used with caution as the size of an object decreases with distance from the camera, and intruders may change in size due to its movement such as crouching. Implementation of the test is as shown in the following expression:

If (Active Blocks in unit > Unit Threshold)
Then Continue. (2)

c. Intensity Test:

This test calculates an average of the differences between the intensities of each respective pixel in the current and previous image of the unit. If this value is greater than the unit threshold, the unit may be an intruder. This test provides a more accurate version of the block test performed before. This test eliminates the block averaging by analysing the change in every pixel of the unit. For this test not to be redundant the unit threshold must be set at a higher value.

d. Shape Test:
Shape tests are difficult as an intruder can form many complex shapes. A shape test developed by Freer in Reference 1 is presented below.

The shape factor, F_c, is defined as:

$$F_c = \frac{L(X)^2}{4\pi.A(X)} \qquad (3)$$

where A(X) is the area of the object X
L(X) is the perimeter of the object., which is defined by the number of pixels on the boundary.

The perimeter designates the length of the object boundary. In the discrete case this can be estimated as the number of points which lie on the object boundary [1].

The shape factor measure is invariant to rotation, reflection and scaling. It measures the elongation of an object. The human figure is often elongated. An expression of the test is given here.

If Shape_Min < Fc < Shape_Max then Continue. (4)

If the unit passes each of the above tests, it is considered to be an intruder. An alarm will then be triggered, or the activities of the unit monitored through tracking, the subject of the next section.

III. TRACKING

When a new unit is detected after it has exceeded the thresholds in the tests above, a template is created for the unit recording its current location, size and shape. The location is determined by a single pixel position. A technique reported in Reference 1 calculates the barycentre of a unit. The barycentre can loosely be compared to the centre of gravity of an object and is defined by

$$M_{1x} = \frac{1}{A(X)} \sum_X x_i \qquad (5)$$

and

$$M_{1y} = \frac{1}{A(X)} \sum_X y_j \qquad (6)$$

Where M_{1x} is the first moment of inertia in the x plane
M_{1y} is the first moment of inertia in the y plane
A(X) is the area of the object X
(x_i,y_j) is a point in the object

Based on the location of the barycentre in the image, a unit can travel a certain distance in the image between each frame. This distance is scene dependent and will have different values depending on the distance from the camera. The barycentre is the origin of the subregion determined by the distance. The subregion must be calculated for each unit and added to its template.

Once the possible intruder units for a given frame have been determined each unit is examined to see if a close match exists with any of the templates of the previous frame. If a current unit and a template have similar size and shape, and the unit is located within the template's sub-region, the unit and template are considered to be the same. If the match is high the template is updated with the characteristics of the current unit, otherwise the template remains unchanged. Each unit is checked, new units are added to the template list and existing templates that have no matches are discarded [4].

The overall performance can be improved by using position prediction from several previous positions [4]. The predicted vector of motion defines the starting point for the template matching routine. For example, a unit in the predicted vector of motion would be given greater leeway in the shape and size matching criteria. The predicted vectors of motion would be indicated at installation.

Tracking alone is of little use without any analysis of the motion detected. A high level algorithm is given the location of each unit every frame in order to perform this analysis, commonly known as Intelligent Scene Monitoring (ISM). A description of ISM is given in the following section.

IV. INTELLIGENT SCENE MONITORING

Once the movement of a unit has been tracked, this motion is tested against a number of rules to determine if the unit is a genuine intruder. Two cases exist here – the monitoring of a relatively sparse scene where any human activity constitutes an intrusion, and the monitoring of a complex scene where an alarm should be triggered by a series of suspicious events associated with an individual. A complex scene is difficult to analyse, as the detection of a human does not necessarily mean an intruder is present.

A. The Traditional Approach

The traditional approach is to use a number of rules to eliminate false alarms and to detect any human activity. This is appropriate for a sparse scene where any human intrusion requires an alarm response. A number of rules can be used to determine if the unit is an intruder. Several examples of typical rules are listed below.

(a) The unit moves back and forth in an oscillating pattern over the same area – Swaying vegetation – *Negative.*

(b) Unit moves across image at an improbable speed for a human – Fast moving shadow or car headlights – *Negative.*

(c) Unit moves very fast and is small – Bird or insect. Should have been eliminated already by the size criteria but if not are eliminated here due to high speed.

(d) Small oscillating movements in a subregion – Grass movement – *Negative.*

(e) Global oscillation of several units in the same pattern – Wind – *Negative.*

(f) Movement in a specific direction across the image, such as left to right or up over a fence – ***Intruder***!

The rules above classify the unit as either a false alarm or a genuine intruder. This procedure is successful for a sparse scene where any human activity constitutes an intrusion. However, in a complex scene this approach is inadequate because the system must differentiate between legitimate human activity and genuine intruders.

B. A Neural Network Approach

In a busy scene a neural network approach can be useful. The network can be trained to recognize suspicious activity in units that are being tracked. In the set-up phase of the system, a scene is monitored and input data mapped to the appropriate response of false or genuine alarm.

A simplified example is now examined.

The scene is divided into three regions that are assigned different levels of risk. If a unit is detected in a high risk area this may indicate malicious activity.

The time that the unit has been in the image can also indicate suspicious behaviour. Examples of undesirable activities may include loitering or attempting to gain entry to a structure under surveillance. Time is sliced into short, medium and long. Scene dependent thresholds quantify these times as defined below.

0 < short < T_Short
T_Short < medium > T_Medium
T_Medium < long

For example, if T is in short then Short=1, medium=0 and Long=0 as inputs to the network.

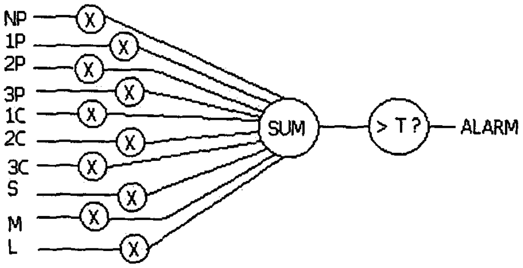
A straight-through perceptron neural network has been chosen as a simple example as perceptrons are easy networks to construct and train. A straight through perceptron neural net is characterised by a single neuron, binary inputs, and logic boxes with just one input, where the output is always the same as the input. A straight through perceptron can be viewed as a perceptron without logic boxes [5].

The output of a perceptron is either 0 or 1 depending on whether the weighted sum of the logic-box outputs is greater than the threshold [5].

Fig. 1 shows an untrained straight through perceptron. The inputs are given in Table 2.

Table 2: Input definitions of Fig. 1
NP – No Zone previously occupied
1P – Zone 1 previously occupied
2P – Zone 2 previously occupied
3P – Zone 3 previously occupied
1C – Zone 1 currently occupied
2C – Zone 2 currently occupied
3C – Zone 3 currently occupied
S – Small time in image
M – Medium time in image
L – Long time in image

Fig. 1 Straight Through Perceptron Neural Network



The intention is to train the network to respond with an alarm when the combinations of inputs of Table 3 are present. In a real life situation when the network is being trained the system monitors each of the inputs and receives a simple alarm/no alarm indication from the trainer. The advantage of the neural network is that the input conditions that constitute an alarm do not have to be quantified by the trainer as we have done in this example. The trainer would work with the system allowing it to generate a large number of input-output conditions. The network then learns from these conditions over and over until the weights are set correctly.

Table 3: Training Patterns for System in Fig 1.

Zone 2 previous + Zone 3 current + medium
Zone 2 previous + Zone 3 current + long
Zone 1 previous + zone 3 current + long
Zone 3 previous + Zone 2 current + long
Zone 3 previous + Zone 1 current + long

Table 3 was manufactured by the following situation:

- Zone 3 is an entrance to a building under surveillance.
- Units in Zone 1 or Zone 2 that never enter Zone 3 are of no interest.
- Risk factor increases from zone 1 to zone 3.
- The scene cannot be exited from zone 3.

A real network would determine this situation and Table 3 on it's own using the input-output combinations generated during it's training sessions.

Training of a straight through perceptron is given in the following procedure:

```
Until the perceptron yields the correct result for
each training sample, for each sample,

•   If the perceptron yields the wrong answer,
    •   If the perceptron says no when it should
        say yes, add to each weight it's
        respective input.
    •   Otherwise subtract each respective input
        from it's respective weight.
•   Otherwise do nothing.
```

This procedure for training the network will always discover a successful set of weights given that a successful set of weights exists. All possible input samples for the network are shown in Table 4.

The network processes these inputs over and over until the correct alarm condition is given for each set of inputs.

Table 5 shows some possible first steps of training the network. Note that the threshold value (T) is considered to be an extra input, whose value is always assumed to be 1. With this addition the perceptron can be viewed as having a threshold of 0. This enables the threshold to be trained to the appropriate value.

The training exercise takes several hundred iterations and the trained network is shown in Fig. 2.

In this network alarms are only triggered by the conditions of Table 3.

The Neural network approach provides an easy and effective way of training a system to monitor a new environment, without the use of advanced statistics.

Table 4 – All possible input combinations to Perceptron Network, Fig. 1

X=Don't Care										
NP	**1P**	**2P**	**3P**	**1C**	**2C**	**3C**	**S**	**M**	**L**	**ALARM**
1	0	0	0	1	0	0	X	X	X	0
1	0	0	0	0	1	0	X	X	X	0
1	0	0	0	0	0	1	X	X	X	0
0	1	0	0	0	1	0	X	X	X	0
0	1	0	0	0	0	1				1(L=1) ; 0(M=1) ; 0(S=1)
0	0	1	0	1	0	0	X	X	X	0
0	0	1	0	0	0	1				1(L=1) ; 1(M=1) ; 0(S=1)
0	0	0	1	1	0	0				1(L=1) ; 1(M=1) ; 0(S=1)
0	0	0	1	0	1	0	X	X	X	1

Table 5 – Possible Initial training iterations of Perceptron Network, Fig. 1

W=Weight I=Input	**NP**	**1P**	**2P**	**3P**	**1C**	**2C**	**3C**	**LMS**	**T**	**OUTPUT**	**DESIRED OUTPUT**
W1	0	0	0	0	0	0	0	0	0		
I1	1	0	0	0	1	0	0	L=1	1	0	0
W2	0	0	0	0	0	0	0	0	0		
I2	0	0	1	0	0	0	1	L=1	1	0	1
W3	0	0	1	0	0	0	1	L=1	1		
I3	0	1	0	0	0	1	0	L=1	1	1	0
W4	0	-1	1	0	0	-1	1	0	0		

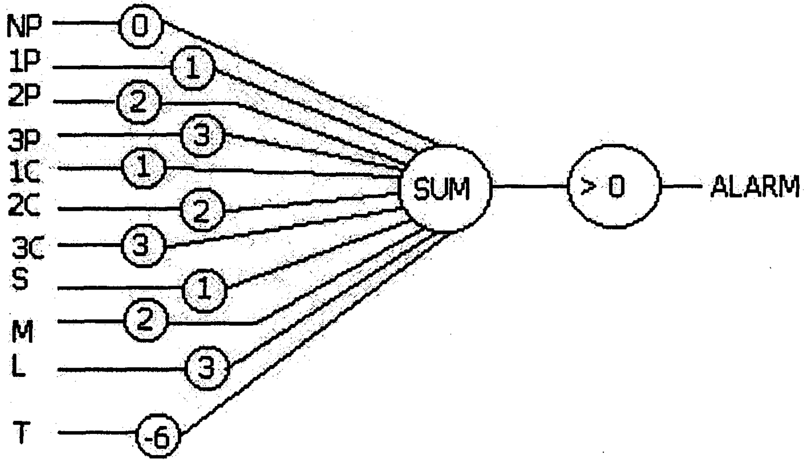


Fig. 2 Trained Perceptron Neural Network

V CONCLUSIONS

This paper detailed techniques which may be used in intruder detection, tracking and intelligent scene monitoring from CCTV images. A neural network approach based on perceptron has been proposed for the implementation of an ISM System. When properly established, the proposed system could become an effective tool for discriminating genuine threats from false alarms in a practical situation.

VI REFERENCES

[1] Freer, J.A. (1995), "Moving Object Surveillance and Analysis for Camera Based Security Systems", Proceedings of the 1995 IEEE International Carahan Conference on Security, pp 67-71.

[2] Horner, M. (1997), "AMETHYST: Automatic Alarm Assessment: Becoming a Reality", Proceedings of the 1997 IEEE International Carahan Conference on Security Technology, pp 88-92.

[3] Takano, T. (1994), "Intruder Detection System by Image Processing", Proceedings of the 1994 IEEE International Carahan Conference on Security Technology, pp 31-33.

[4] Klima, M. (1994), "Motion Detection and Target Tracking in a TV Image for Security Purposes" Proceedings of the 1994 IEEE International Carahan Conference on Security Technology, pp 43-48.

[5] Winston, P.H. (1992), *Artifical Intelligence*, Addison-Wesley Publishing Co., Sydney, pp 471-488.