Neural Networks for 3-D Motion Detection
From a Sequence of Image Frames

Chan Lai Wan Yip Pak Ching
Computer Science Department
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong.
email: lwchan@cusd.cuhk.hk yip06@cusd.cuhk.hk (bitnet)

ABSTRACT

The most prominent area where the potential of neural networks under study is that of engineering applications, such as image and speech recognition. In this paper, the design of neural networks for sequential domains is another interesting direction. This paper describes a 3-D motion detection system. This system consists of three stages; the Rough Motion Detection stage, the Moving Object Extraction stage, and the Object Identification and the 3-D Motion Detection stage. Five neural networks — the Correlation Network, the Rough Motion Detection network, the Edge Enhancement Network, the Background Remover, and the Normalization Network, are designed for the implementation of these three stages.

1. INTRODUCTION

A motion detection system has a lot of applications. It is of great military interest to be able to detect targets reliably within an acceptable processing time. In the proposed motion detection system, it is allowed to follow the moving object so that the system can be applied to image tracking. For example, it can be used in military to trace a military plan of the enemy having detected and controls a cannon to shoot at the estimated position (current position of the plan + the velocity). Discrimination of an object from its background is a central processing task of any visual system. Motion is one of the most powerful object features which can be exploited for this purpose.

A lot of researchers have selected the optical flow as their motion detectors [1][2][3]. For a real time motion detection system, in order not to loss the moving objects under our field of vision, only short time for processing is allowed. The optical flow models require a long time for computing the velocity vectors. Even if these vectors have been obtained, it is not easy to distinguish between an object and its background. Instead of the optical flow models, we proposed an alternative motion detection system and Figure 1 shows the system architecture of this system. The processing of the system is divided into:

The Rough Motion Detection Stage — The system receives a sequence of 2-D image frames which are projected from the real world scene. In the proposed motion detection system, the camera is allowed to move either willingly or unwillingly. It is said to be willing when the camera is moved to follow the moving objects controlled by either operators or the system itself. It is said to be unwilling when the camera is moved by extraneous factors. Therefore, the first task is to find out the shift between two image frames. We will obtain the shift by using a correlation network. The current projected image frame then will be shifted and will be passed to the rough motion detection network for the measurement of the varying degree of each small region. A recursive algorithm is used to find out the boundary of all involved regions for one detected moving object.

The Moving Object Extraction Stage — With the obtained boundary, two object's images (with background) can then be extracted from two image frames (both current image frame and previous image frame). Extracting the moving object from its background can be done by the edge enhancement network and the background remover.

The Object Identification and The 3-D Motion Detection Stage — Having output by the correlation network by passing the extracted images, the 2-D motion ($\Delta x$ and $\Delta y$ of the displacement) of detected moving object can be gained. The normalization process is done for two reasons — one for finding out the 3-D motion parameter $\Delta z$ and one for the object identification. By passing the normalized object's edge to the object identification network, the object name will be obtained at the last stage of motion detection processing.

Section 2 to 4 will describe the implementation of the each stages by using the specially designed neural networks.

2. ROUGH MOTION DETECTION
The first stage of motion detection is to have a rough measurement of image motion to determine if there is any object being moved in each small region. Since the camera is allowed to move in the proposed motion detection system, there should be a preprocessing before the rough motion detection. The purpose of such a preprocessing is for obtaining the shift between two projected image frames. The Correlation Network can be applied to find out the image motion.

2.1 The Correlation Network

In the correlation network shown in Figure 2, pixels from two consecutive images are passed to the input neurons, one to \( E' \) and one to \( E'^{-1} \). The output of each hidden neuron \( h_j \) is defined by

\[
h_j = \frac{1}{1 + e^{-(wE'_j + wE'^{-1}_j - \theta^h)}}
\]

where \( w \) is a small constant weight and \( \theta^h \) is the threshold of the hidden neuron \( h_j \). If the binary images are processed in the system, \( w \) can be chosen as 1. If the gray-level (0 to 255) images are used, \( w \) can be chosen as 0.1. In order to force \( h_j \) to approach to 0 when \( wE'_j + wE'^{-1}_j \) is 0, the value of threshold \( \theta^h \) is chosen as 5 in the proposed system. The output of the displacement neuron (the output neuron of the correlation network) \( d_i \) is given by

\[
d_i = \frac{1}{1 + e^{-(\sum_j h_j - \theta^d)}}
\]

where \( \theta^d \) is the threshold of the neuron \( d_i \). After all displacement neurons have obtained their outputs, the only one winner determines the shift (or the displacement). For example, the displacement should be -5 if the neuron \( d_{-5} \) is the winner.

2.2 The Rough Motion Detection Network

Two projected image frames are shown in Figure 3. All pixels of the frame \( f' \) will pass to the input neurons \( X' \) of the neural network shown in Figure 4. There are two types of input neurons in this network architecture. They are \( X'^{-1} \) and \( X' \). Here, \( X'^{-1} \) are used to memorize the brightness of \( E(x, y, t-1) \) and \( X' \) are used to receive the brightness of \( E(x+Ax, y+Ay, t) \). The on-off gate is used for the connection between the input features \( E(x, y, t) \) and input neurons \( X'^{-1} \) (see Figure 4). When the gate is 'off', no stimulation flows. When the gate is 'on', full stimulation flows. The gate functions as a control on assignment of input flow for the discrete purpose. Having output the motion, the gate will be 'on' within a short time. The new features of \( f' \) will replace the old value \( E(x,y,t-1) \) of all input neuron \( X'^{-1} \) by \( E(x,y,t) \).

The output of the motion neuron \( m_i \) can be given by

\[
m_i = \frac{1}{1 + e^{-(\sum_j w_j - \theta_m)}}
\]

where \( w \) is a small constant weight (in Figure 4, \( w=0.1 \)) and the output of hidden neuron \( h_j \) is defined by

\[
h_j = |X'_j - X'^{-1}_j|
\]

Each motion neuron \( m_i \) is used to output the degree of variation in the region \( R_i \). The outputs of motion neurons for all regions at time \( t \) by passing image frame \( f' \) (all image features of image frame \( f'^{-1} \) are held in input neurons \( X'^{-1} \) are given in Figure 5.

From Figure 5, each number indicates the output of one motion neuron. If a motion neuron outputs a '0', it means that no object motion has been detected. Otherwise, there should be something moved in the region responded by the neuron. With this approach, the rough motion detection stage can be done in an acceptable processing time. On a 80386 with 387, this stage takes about 8 seconds for the processing of image frame of 320*256 pixels.

3. MOVING OBJECT EXTRACTION

Having obtained the motion information of all regions, all positions of moving objects can be gained easily as a result. From Figure 5, two moving objects have been detected and they are located in Region \( R_{1,9} \) and \( R_{1,11} \). The boundary of all involved regions for each moving object can be obtained from the
outputs of motion detection network. In this stage, main steps for extracting one moving object are the edge enhancement and the background elimination.

3.1 Edge Enhancement

Extracting the edge of a detected object is a crucial point to discriminate from its background. Using the normal operators for edge detection, such as Roberts gradient and Sobel smoothed gradient, all information of the image will be lost except the edges. In order to get the edge of detected object without losing some important information, a high-pass-filter-like network architecture shown in Figure 6 is designed to enhance the edges. All images (a) and (b) of the examples shown in Figure 7 are the outputs of the edge enhancement network. This figure confirms that not only are the edges obtained but also the bright information are kept. Two enhanced images will be obtained from frame $t$ and frame $t-1$ for one detected object. For example, the object's image $R_a$ is obtained from the frame $f^t$ and the object's image $R_b$ is obtained from the frame $f^{t-1}$.

3.2 Background Remover

Since the noises from the background are retained in two enhanced images obtained from the outputs of the edge enhancement network, a special network called "the background remover" is designed to extract the detected object from its background. The output of the background remover $B_{zy}$ is defined by

$$B_{zy} = \begin{cases} 0, & \forall h \in N_{zy} \land h \neq e, \\ H_{zy}, & \text{otherwise} \end{cases}$$

(5)

where $N_{zy}$ is a set of the hidden neuron $H_{zy}$'s neighbors. $N_{zy}$ can be defined by

$$N_{zy} = \{H_{ij} | (x - b < i < x + b) \land (y - d < j < y + d)\}$$

(6)

where both $b$ and $d$ are the constant numbers of neurons for indicating the spread degree of one's neighbors. In both Equation 5 and Equation 6, the output of the hidden unit $H_{zy}$ is defined by

$$H_{zy} = \begin{cases} 0, & I_{zy} = I_{zy}^{t-1}, \\ 1, & (I_{zy} \neq I_{zy}^{t-1}) \land (I_{zy} = e), \\ 0.9, & \text{otherwise} \end{cases}$$

(7)

The first option of Equation 7 when two inputs are same is the most important factor for removing background. Setting such position to '0' (nothing) can remove almost all noises from the background. The constant $e$ is a selected value for the detected edge. In a grey-level-pixel-input system (e.g. '0' to '255'), '255' can be selected for the representation of the detected edge. In a normal-neuron-input system (i.e., '0' to '1'), '1' can be selected for the representation of the detected edge. By the second option of Equation 7, the edges of detected object can be extracted. In order to be gotten the correct displacement by the correlation network, the brightness of the detected object can be enhanced by the last option of Equation 7.

All objects' images (c) and (d) of the examples shown in Figure 7 are the outputs of the background remover. It should be mentioned that not all noises of background have been removed by the background remover. There may be a few number of noises pixels remained. The images $C_c$, $C_d$ and $R_c$ shown in Figure 7 are worse examples. These remained noises are ascribable to the hidden part of the moving object.

4. OBJECT IDENTIFICATION AND MOTION DETECTION

4.1 2-D Motion Detection

Having obtained two object's image extracting from its background, the 2-D motion ($\Delta x$ and $\Delta y$) of the moving object can be gained by the output of the correlation network described in section 2.1. In this stage, all information of two extracted object's images will be passed to the correlation network. The 2-D displacement, $\Delta x$ and $\Delta y$, is obtained (see Figure 1).

4.2 Object Normalization & 3-D Motion Detection

Since the extracted object image obtained from the output of the background remover may contain a few noises, it is dangerous to use such an imprecise object image for the object identification. Therefore, it is necessary to remove all remained background noises. A supplementary network has been specially designed for this purpose.
In Figure 7, each image (e) is combined by two extracted images (c) and (d). From these examples, we can find that all remained background noises are on the outside of the mixed object. So now those noises can be removed easily by the supplementary network. The output neuron $o_{xy}$ is defined by

$$o_{xy} = \begin{cases} e, & (I_{xy} = e) \land (\exists p \in M_{xy} \land p = e) \\ 0, & \text{otherwise} \end{cases}$$

where each input neuron $I_{xy}$ receives one pixel of the object's image extracted from frame $t$ and $M_{xy}$ is a set of $I_{xy}$'s neighbors. $M_{xy}$ can be defined similarly as Equation 6 by

$$M_{xy} = \{I_{xy}^{-1}[(x + \Delta x - b < i < x + \Delta x + b) \land (y + \Delta y - d < j < y + \Delta y + d)] \}$$

where both $b$ and $d$ are the constant numbers of neurons for indicating the spread degree of one's neighbors. All images (f) of Figure 7 are the examples of the outputs of this supplementary network. Now, only object's edges are remained.

The next stage is the normalization process. The normalization process is done for two reasons — one for finding out the 3-D motion parameter $\Delta z$ and one for the object identification. A special designed normalization network is used to convert any sized image to a fixed-sized image. There are two types of output neurons in the network. They are the size-changed neuron $c$ and normalized image neurons $n_{xy}$. The size-changed neuron $c$ is given by $S/m$, where $m$ is a constant for indicating the size of the normalized object image. The parameter $S$ should be passed as a special input feature to the normalization network. This special input feature are different from the normal input features $o_{xy}$. It should be calculated independently before passing to the network. $S$ denotes the computed size of the original object image.

The output neuron $n_{xy}$ of the normalization network is defined by

$$n_{xy} = \begin{cases} 1, & \exists g \in B_{xy} \land g = c \\ 0, & \text{otherwise} \end{cases}$$

where $B_{xy}$ is a set of $o_{ij}$ and is defined by

$$B_{xy} = \{O_{ij}[(c + h_x \leq i < c + h_x) \land (c + h_y \leq j < c + h_y)] \}$$

where $h_x$ and $h_y$ should be passed as special input features to the normalization network. These two parameters $h_x$ and $h_y$ indicate the first position where there is an edge information.

The 3-D motion parameter $\Delta z$ can be obtained by the difference between two outputs of the size-changed neurons $c'$ and $c''$ of the normalization network by passing two object’s images extracted from frame $t$ and frame $t-1$.

### 4.3 Object Identification

The object identification network is a multi-layer perceptron[4]. The training process is done by the back propagation algorithm[5]. The normalized object image now is passed to the object identification network. The correct class of the detected object can be given as the output of this network. Having obtained the object class, the proposed motion detection system outputs all motion information of the object — $\Delta x$, $\Delta y$, $\Delta z$ and the object name immediately.

### 5. CONCLUSION

This paper describes a 3-D motion detection system. Five special designed neural networks are introduced. They are the Correlation Network, the Rough Motion Detection network, the Edge Enhancement Network, the Background Remover, and the Normalization Network. The processing time of the system depends on the number of moving objects detected. Normally, it will take about 9 seconds on a sequential machine, such as 80386. With the proposed motion detection system, two moving objects (a man and a ball) and their motions were detected correctly at time $t$ (see Figure 3 and Figure 7). Since there were only two objects detected, it took only 8 seconds for the processing of all three stages.

### REFERENCE

sequence of 2-D image frames
boundary of detected object 1
boundary of detected object n

Figure 1. System Architecture.
One trapezoid box with many circles represents one neural network. Each rectangle represents one image.

Figure 3. Two projected image frames.
Left, the projected image frame at time t-1. Right, the projected image frame at time t.
Figure 2. Correlation Network. All pixels of \( f \) will pass to the input neurons \( E^f \). All pixels of the previous frame will pass to the input neurons \( E^{f-1} \).

Figure 3. Rough motion detection network. \( d \) is displacement (Ax and Ay or the shift) obtained by the preprocessing of the correlation network.

Figure 4. Edge enhancement network.

Figure 5. The outputs of motion neurons for all regions.

Figure 6. Some examples for object extraction. Left, \( d \) is perfect. Center, it is a worse example from Figure 3. Right, the object is moving forward. All images (a) and (b) are the outputs of edge enhancement network. The background remover produces all images (c) and (d). Each image (e) is combined by two extracted images (c) (with a shift) and (d). All images (f) are the results of moving object extraction.