DURBAN
UNIVERSITY of
TECHNOLOGY

OBJECT MOTION DETECTION, EXTRACTION AND FILTERING USING ANN ENSEMBLES

KEVIN EMANUEL MOORGAS

2009

OBJECT MOTION DETECTION, EXTRACTION AND FILTERING USING ANN ENSEMBLES

by

Kevin Emanuel Moorgas

Student Number: 18651121

September 2009

Thesis submitted in compliance with the requirements for the

Master's Degree of Technology: Electrical Engineering – Light Current

DEPARTMENT OF ELECTRONIC ENGINEERING

FACULTY OF ENGINEERING AND THE BUILT ENVIRONMENT

Approved for Final Submission

_____

Supervisor: Dr. P. Govender

Department of Electronic Engineering

Durban University of Technology

_____

Date

DECLARATION

I declare that this thesis represents my own work and has not been previously submitted to any other university for any degree.

_____

Mr. K.E.Moorgas

_____

Date:

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, Dr. P.Govender for his dedication in assisting me throughout the duration of my research. His invaluable guidance, knowledge and encouragement have helped me achieve new heights and accomplish my dreams. I also thank Trevor Pillay for his continued support and advice.

I am also eternally grateful to my wife Devika for her unending support, love, encouragement and patience when I had to work late to complete this research. Also to my mother Maureen who was always there to support the family in my absence. To my children Cayley and Keanen, endless hugs, kisses and thanks for bringing happiness and joy at times when I needed stress relief.

Finally in all of this I thank God for my good health, strength, talents, knowledge and endurance because without God nothing is possible.

Kevin Emanuel Moorgas

ABSTRACT

This research is devoted to the development of an intelligent image motion detection system based on artificial neural networks (ANN's). Object motion detection, non-stationary image isolation and extraction, and image filtering is investigated, with the intention of developing a system that will overcome some of the shortcomings associated with the performance of conventional motion detection systems.

Motion detection and image extraction finds popular application in medical imagery and engineering based diagnostics systems. Conventional image processing systems utilise Digital Signal Processing (DSP) to perform the non-stationary image motion detection function. Aliasing and filtering are problematic processes in DSP based image processing systems. The proposed ANN motion detection system overcomes some of these shortcomings.

The study compares the performance of conventional DSP systems to that of the proposed ANN based system. The excellent noise immunity, ability to generalise and robustness of the ANN system is exploited in the design of the motion detection system. The ANN's are arranged as ensembles in order to improve the computation time of the proposed motion detection system. A *hybrid* system comprising DSP and ANN ensembles is also proposed in the study. The hybrid system exploits the positive characteristics of DSP and ANN's within a single system. The performance of the pure ANN system and the hybrid system is compared to that of DSP systems, using the image's signal-to-noise ratio and computation times as a basis for comparison.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
ADC	Analogue to Digital Converter
AI	Artificial Intelligence
AVI	Audio Video Interleave
CT	Computer Tomography
CED	Canny Edge Detection
CNN	Committees of Neural Networks
CVE	Congregation of Voting Experts
CWT	Continuous Wavelet Transform
DC	Direct Current
DCT	Discrete Cosine Transform
DVD	Digital Video Disc
DAC	Digital to Analogue Converter
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
ENN	Ensemble of Neural Networks
DSP	Digital Signal Processing
FT	Fourier Transform
HT	Hough Transform
HPF	High Pass Filter
JPEG	Joint Photographic Experts Group
LPF	Low Pass Filter
LMS	Least Mean Square
LoG	Laplacian of Guassian
ME	Mixture of Experts
MRI	Magnetic Resonance Imaging
MPEG	Moving Picture Expert Group
MLFF	Multilayer Feed Forward Neural Network
MSE	Mean Square Error
NN	Neural Network
PED	Prewitt Edge Detection

PET	Positron Emission Tomography
WT	Wavelet Transform
PSNR	Peak Signal to Noise Ratio
ROI	Region of Interest
IDFT	Inverse Discrete Fourier Transform
IWT	Inverse Wavelet Transform
HT	Hough Transfrom
CED	Canny Edge Detection
SED	Sobel Edge Detection
1D	One Dimension
2D	Two Dimensions
3D	Three Dimensions
1D-WT	One Dimension Wavelet Transform
2D-WT	Two Dimension Wavelet Transform
1D-IWT	One Dimension Inverse Wavelet Transform
2D-IWT	Two Dimension Inverse Wavelet Transform
1D-DWT	One Dimension Discrete Wavelet Transform
1D-IDWT	One Dimension Inverse Discrete Wavelet Transform
2D-IDWT	Two Dimension Inverse Discrete Wavelet Transform

Chapter 1

Introduction and Overview of the Study

1.1 Introduction

Motion is a powerful cue used by humans and animals to extract objects of interest from a background of irrelevant detail (Gonzalez and Woods, 2008). Motion arises from a relative displacement between a sensing system and the scene being viewed, such as in robotic vision systems, autonomous navigation systems and surveillance systems. Motion detection systems utilizing ANN's are relatively simple and use artificial models of biological neurons that are trained to perform the motion detection function. Artificial neurons display desirable characteristics such as robustness to noise and the ability to generalize in the face of small changes (Hagan *et al.*, 1996). This is not the case for traditional detection systems (Moorgas and Govender, 2008). Traditional image motion detection techniques compare a 2D reference image frame $f_1[x, y]$ to its subsequent image frame $f_2[x, y]$ taken at different times; x and y are variables of the function f and n denotes the frame number. This comparison is done pixel by pixel. The remaining pixels corresponding to non–zero elements indicate the non-stationery components of the image.

This study compares the performances of an AI based ANN system to that of a DSP motion detection system. The systems are first individually tested and are then combined into a single system with a view to enhancing processing speed, whilst simultaneously conserving image quality. Further testing of the ANN system is done using image compression techniques, with the objective of improving the end processing time of the ANN system.

1.2 Motivation for Using ANN Systems in Image Processing

ANN's have become a popular tool in AI research (Narendra and Pathasarathy, 1990; Polikar, 2006). Their applications extend to the fields of science, engineering, medicine, economics and the arts. Endeavors have been made to make technologically intelligent tools, and ANNs are one of those tools powerful enough for solving intelligent and complex tasks in image processing (Hagan *et al.*, 1996). Traditional imaging systems use DSP based models which require comprehensive programming in order to perform a given task (Dumontier *et al*, 1999; Hansen and Salamon, 1990).

ANN's are elementary processing units that are combined to form systems whose architectures may vary depending on the application of the network. Systems of this nature can be trained to process and learn data that is representative of the underlying task to be performed (Polikar, 2006). The self learning of ANNs is particularly useful when the comprehensive models that are required for conventional computing methods are either too large or too complex to represent accurately, or simply do not exist at all (Ham and Kostanic, 2001). The interconnectivity and distributed nature of ANN systems contributes towards a high degree of noise immunity, fault tolerance and generalization capability (Moorgas. and Govender, 2008). Although DSP techniques are used for most image processing functions and yield favorable results, this study will show that the processes involved are computationally intensive, are not immune to noise, need additional filtering and have a noticeable deterioration in image quality.

The study focuses on the development and implementation of an ANN ensemble which behaves as a committee of experts to perform a specific task within a motion detection system. The rationale for using a so-called committee of experts arose as a direct consequence of the huge computational burden that ANN's impose on a computer's processor. The ensemble contributes positively towards reducing the computational burden by processing smaller bits of data, where each bit contributes to the whole.

Motion detection, noise filtering and image extraction functions are carried out by the ANN ensemble system. This ensemble system is initially implemented using gray scaled

image frame sequences of the images under consideration. The performance of a conventional DSP motion detection system will be compared to that of the proposed ANN system. A hybrid system consisting of DSP combined with an ANN ensemble is also considered in the study, with a view to exploiting the positive aspects of DSP systems and ANN systems within a single model for enhancing system performance with regards to computation speed and image quality.

1.3 Overview of Ensemble Based Systems

Ensemble based systems (also known as ME, CNN, VPC, CVE and ENN) are decision making sub-systems that independently handle data assigned to them. The data is handled as packets that constituted a larger data base (Hansen and Salamon, 1979). Decision making by the sub-systems constitute less system noise because of the 'divide and conquer' phenomenon and contributes towards a clearer, accurate and faster decision making process (Polikar, 2006).

ANN systems can process large amounts of data accurately, quickly and contribute towards eradicating errors and noise. Hansen and Salamon (1979) showed that the generalization performance of a neural network system can be improved by using an ensemble of similarly configured ANNs. Since then research using ANN ensemble systems has expanded rapidly because of their superior performances over single system ANN models. The strategy applied to ENN systems involves the creation of many decision making neural network sub-systems. The outputs of these sub-systems are combined to improve system performance with regards to error reduction and faster processing speeds.

1.4 Summary

This chapter has provided a brief discussion on the techniques that will be used in this study to determine a best design for an image motion detection system. A more detailed

discussion on the relevant techniques and the processes involved will follow in the subsequent chapters. This report is presented as follows:

Chapter 2 discusses common problems encountered by using DSP methods and chapter 3 deals with image segmentation. Chapter 4 provides a comparative study between the DSP system and a proposed ANN system for object motion detection, image extraction and filtering. In chapter 5 the proposed ANN system, the DSP system and a hybrid system are tested and compared. The comparison is based on the quality of the image produced, and the processing time. Chapter 6 uses image compression techniques for preprocessing to improve the computation speed of the proposed ANN system. Chapter 7 concludes the study.

Chapter 2

Complexity and Noise Problems associated with DSP Systems

2.1 Introduction

Fundamental to DSP systems is the study of discrete–time systems that process digital signals to extract or modify information. When compared to analogue systems, DSP systems have fewer components, are not prone to the effects of component ageing, are much more flexible, and have higher noise immunity (Mitra, 1998). Conventional DSP systems for motion detection utilize DSP techniques such as image segmentation, image extraction and image filtering. These processes are computationally intensive, require more processing power and large memory capacities (Dumontier *et al*, 1999). Furthermore, aliasing which is a product of quantization and signal sampling, is problematic and requires additional filtering.

Soft computing based systems such as ANN's are more flexible and have a higher noise immunity than DSP systems (Govender and Moorgas, 2008). This will be demonstrated in this study. It will also be shown that ANN systems can handle changes in input data such as illumination and varying noise levels more robustly than DSP based image processing systems (Govender *et al*, 2006; Moorgas and Govender, 2008). DSP systems require continuous updating and reprogramming when input and output environmental conditions change. ANN systems on the other hand are adaptive and can produce optimum results for changes in the environment, with minimal human intervention (Hagan *et al*, 1996).

2.2 Noise in Image Processing Systems

Noise in images is a direct consequence of the sampling process, image reconstruction, morphological operations and image segmentation (Mitra, 1998; Heath *et al*, 1998). Noise often exists in the form of high frequency components that appear as high intensity

pixels in images. Electromagnetic drift also occurs in the transmission mediums that can modify digital data being transferred and can alter pixel values within images (Najarian and Splinter, 2006). Variations in scene illumination, dust, and reflections caused by poor lighting can also introduce stray pixels above set thresholds and image masks during image processing functions (Govender *et al*, 2006).

Noise in image processing systems can be either additive (2.1) or multiplicative (2.2).

$$f'(x,y) = \sum(f(x_1y_1),\ f(x_2y_2)) \tag{2.1}$$

$$f'(x,y) = f(x_1,y_1).\ f(x_2,y_2) \tag{2.2}$$

With regards to (2.1) and (2.2): $f'(x,y)$ denotes a noisy image, $f(x_1,y_1)$ represents the original uncontaminated image and $f(x_2,y_2)$ is the induced noise. Complex digital filters using DSP have to be introduced to imaging systems in order to remove noise (Cabrera *et al*, 1996). The reconstruction of images to reproduce the original image data after DSP processes are applied requires complex decoding algorithms, expensive decoding equipment and powerful processors to ensure accurate representation of the original images (Dumontier *et al*, 1999).

2.2.1 External Noise Sources and DSP Image Filtering

External sources of noise in imaging systems include environmental factors such as poor lighting levels and electromagnetic interference from ignition systems. Image noise is random and exists as high intensity isolated scattered pixels in the dark regions, and low intensity isolated scattered pixels in the bright regions. When filtering images corrupted with noise, the filtering operations and algorithms applied are most often frequency dependent DSP algorithms.

A model of an image processing system is given in Fig 2.1, showing additive and multiplicative induced noise. Noise in images can take the form of random unwanted fluctuations of pixel values outside and inside a ROI (Heath *et al*, 1998).

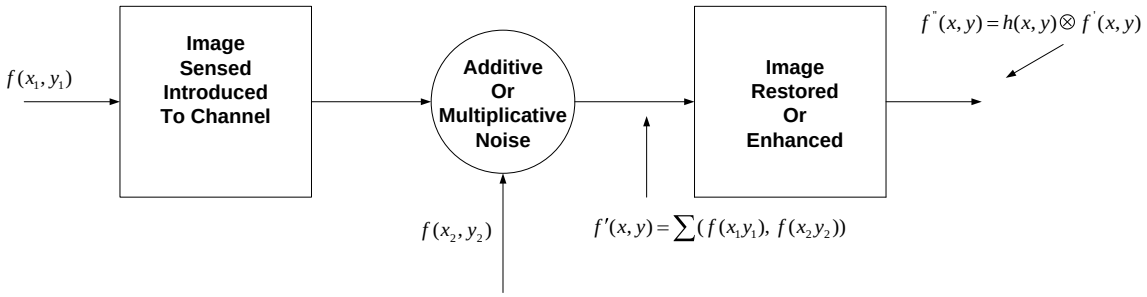


Figure 2.1 Basic imaging system with induced Noise

Applying equations (2.1) and (2.2) to the model shown in Fig 2.1 results in (2.3).

$$f''(x,y) = h(x,y) \otimes f'(x,y) \tag{2.3}$$

With regards to (2.3), *h(x,y)* is the image restoration or filtering function, *f''* represents the output of the model in Fig. 2.1 and *f'(x,y)* is the signal that results following noise insertion. Most image restoration and filtering algorithms use the DSP operation of convolution as can be seen in (2.3). Equation (2.4) represents the frequency response of the system in Fig 2.1.

$$F''(u,v) = H(u,v)F'(u,v) \tag{2.4}$$

With regards to (2.4), *H(u,v)* is the transfer function used to restore or enhance *f* ; *u* and *v* are the 2D frequency components.

2.2.2. Image Filtering Methods

Noise cannot be totally removed from an image without it suffering any loss of image data (Zhang *et al*, 2007). The loss of image data contributes to the loss of image resolution. Because of this, noise reduction filters have been developed using algorithms that can determine and preserve pixels within the ROI (Cabrera *et al*, 1996). Depending on the image processing applications, many different techniques can be used for image

filtering and image enhancement. Some examples of 2D filters used for image processing systems includes median filters, statistical filters, adaptive filters, band-reject filters, band-pass filters and notch filters. Once filtered, the quality of images can either be measured statistically, analytically or through observation. In this study the quality of the image frames were measured analytically using the PSNR ratio. This is a preferred method that is widely used in image processing systems to measure the quality of images during image reconstruction (Cabrera *et al*, 1996).

2.2.3 Problems with Aliasing and the Sampling Theorem

A mirror image of the original sampled data signal, referred to as aliasing, occurs during signal quantization and signal sampling. Aliasing is an undesirable phenomenon that occurs during DSP processing of data signals, and necessitates the use of anti-aliasing filters to remove any multiple mirror images (Mitra, 1998). Using these filters adds to the computational burden placed on a computer's processor and also increases its memory requirement (Dumontier *et al*, 1999; Moorgas and Govender, 2008).

Fig 2.2(a) and Fig 2.2(b) shows a 1D version of an ideal sampled time signal and its corresponding frequency transform, respectively. Fig. 2.3(a) also represents a 1D signal. The corresponding transform for Fig. 2.3(a) following over-sampling is given in Fig. 2.3(b). From Fig 2.3(b), any 'overlapping' of mirror images requires anti-aliasing filtering. A 2D signal is shown in Fig. 2.4(a) and the effects of under-sampling a 2D is given in Fig. 2.4(b)

There are two principal manifestations of aliasing in images, namely *spatial aliasing* and *temporal aliasing*. Spatial aliasing is due to sampling errors in a continuous signal and temporal aliasing is related to time intervals between images in a sequence of images. One of the most common examples of temporal aliasing is the "wagon wheel" effect, in which wheels with spokes in a sequence of images appear to rotate backwards. Aliasing in images consumes twice the processing power and requires approximately twice the memory capacity (Mitra, 1998).

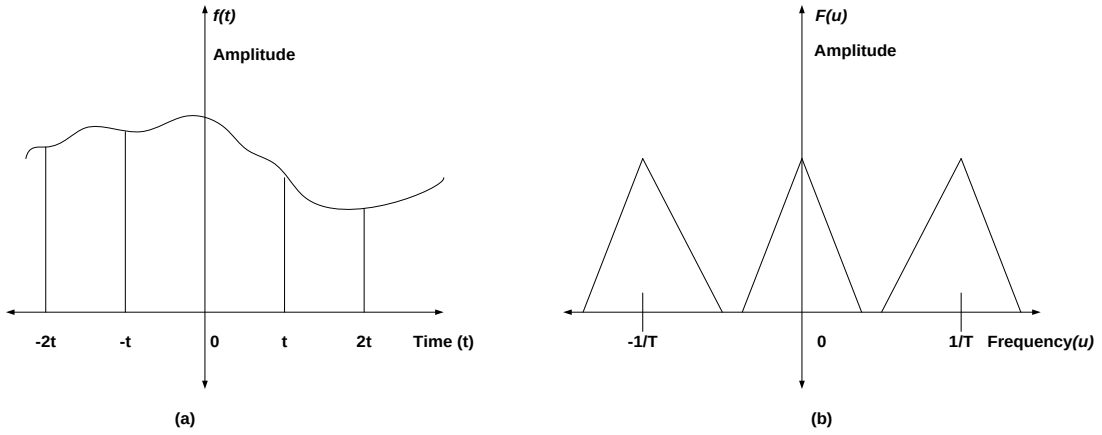


Fig 2.2: (a) 1D continuous time signal and (b) Transform of an over-sampled 1D continuous time signal (Gonzalez and Woods, 2008).

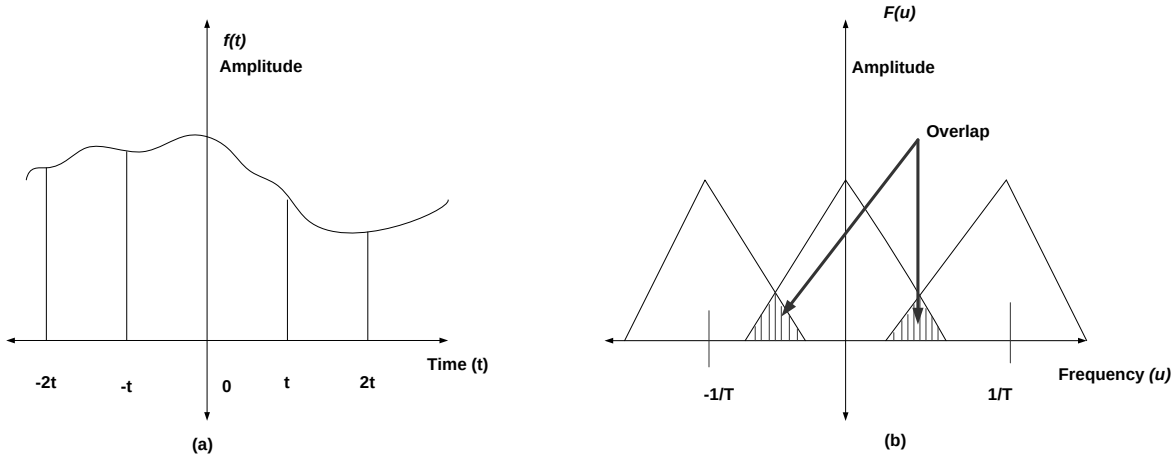


Fig 2.3: (a) 1D Continuous time signal and (b) Transform of an under-sampled continuous time signal (Gonzalez and Woods, 2008).

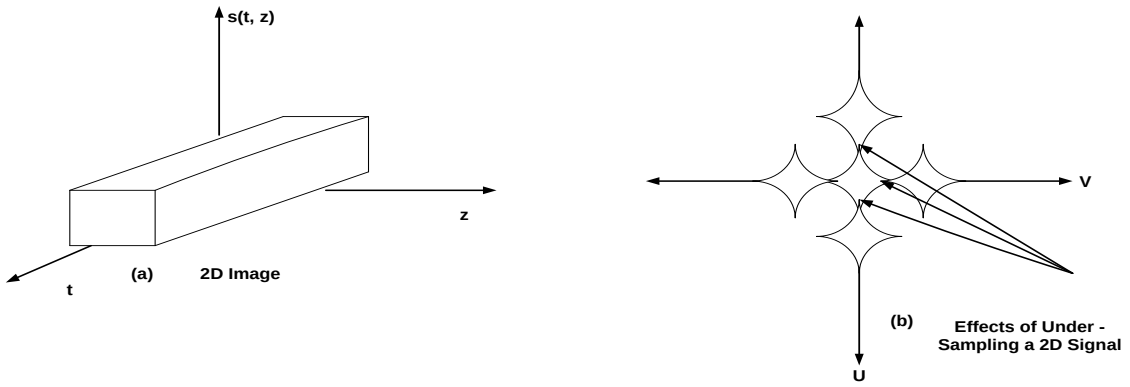


Fig 2.4: (a) 2D image, (b) 2D transform of the image showing aliasing of an under-sampled image (Gonzalez and Woods, 2008).

Signal Sampling

Shannon's sampling algorithm in (2.5a) is used extensively in 1D image signal sampling applications (Gonzalez and Woods, 2008). With regards to (2.5a), f_s denotes the sampling frequency and *f*max represents the signal's maximum frequency.

$$fs \geq 2f\max \tag{2.5a}$$

The equations for 2D image sampling are shown in (2.5b) and (2.5c).

$$fs_u \geq 2f\max_u \tag{2.5b}$$

$$fs_v \geq 2f\max_v \tag{2.5c}$$

With regards to (2.5b) and (2.5c), *u* and *v* denote the 2D frequency components of an image.

2.3 The Computational Burden of 2D DSP Systems

DSP systems are computationally intensive (Moorgas and Govender, 2008; Gonzalez and Woods, 2008). The following simple example will illustrate the computational intensiveness of DSP systems:

Consider a DFT which involves (n^2 x n^2) complex multiplications and *nm(n-1)(m-1)* complex additions during the processing of a 2D image having (*m* x *n*) rows and columns respectively. Hence, for a 2D image having *255* x *255* element points, its DFT will have 255^2 x 255^2 = 4228250625 complex multiplications and 255 x 255 (255 – 1) (255 – 1) = 4195152900 complex additions (Gonzalez *et al*, 2004). These 4228250625 complex multiplications and the 4195152900 complex additions represent an approximation of the computation points that the 2D DFT will have to perform when it processes a single image frame.

Discrete convolution is applied to processes involving image filtering and segmentation. Discrete convolution involves complex multiplication, addition and shifting operating during which an image mask is shifted pixel by pixel over the entire range of the original image. From (2.6), for an image of *m* x *n* dimensions with 65025 picture elements and an average sized motion mask of *h* x *l* dimensions with 9 mask elements, there will be approximately 5852225 complex multiplications, 5852225 complex additions and 65025 shifts.

$$f(x,y) \otimes h(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) h(x-m, y-n) \qquad (2.6)$$

2.4 Summary and Conclusion

This chapter has provided a brief discussion on the shortcomings of conventional DSP systems. These shortcomings are aliasing, computation burden and poor noise immunity. The following chapter will expand further on DSP systems and discusses image segmentation.

Chapter 3

DSP Image Segmentation Methods

3.1 Introduction

This chapter investigates DSP motion detection techniques, with a view to applying them to an ANN ensemble in order to improve end processing. Segmentation of objects within an image contributes towards the improvement of digital image filtering, and also improves the quality and processing speed of the image sequences by excluding unwanted background noise in image frame sequences during object motion detection and extraction (Heath *et al*, 1998). Motion detection system use conventional edge detection techniques such as CED, LoG, SED, PED and the HT as an edge detector for image segmentation.

3.2 Image Segmentation

Image segmentation is the separation of an image into a set of regions, or in some cases a single region. In this study these regions are referred to as the ROI. A ROI can be border pixels, line segments, groups of pixels, 2D objects and 3D objects. Segmentation can also be used to divide an image into foreground and background regions where the ROI could either be the foreground or the background. Segmentation is used to decompose the image into parts for low level and high level image analysis. Low level segmentation analysis is the extraction of areas of an image which need to be analyzed. Higher level segmentation analysis is performed for change of representation as in morphological operations, which are the reconstruction or modification of elements of objects of the ROI within an image either in the background or foreground. The segmentation of the ROI of an image must be uniform, continuous, and without holes. The adjacent regions of segmentation should have significantly different pixel values to ensure successful segmented areas (Gonzalez and Woods, 2008). The boundaries of each segment must be

smooth, and should be spatially accurate to obtain best results during analysis of an image (Gonzalez and Woods, 2008).

In this study the HT segmentation technique, edge detection techniques and image subtraction segmentation techniques will be investigated. The type of segmentation algorithms chosen always depends on the type of applications and analysis of the data within the images being analyzed (Vozikis and Jansa, 2008; Roushdy, 2006).

3.3 Hough Transform Segmentation

The HT algorithm for segmentation finds line segments in an image frame by using the straight line equation (3.1), where *f(x)* is a function of a straight line, *m* is the gradient, *x* is the dependant variable, and *c* is a constant. There are two variables for each segment, namely *m* and *c*. The line segments form 2D arrays identified by *m* and *c* values. For example, the two variables of an accumulator array will correspond to quantized values of *m* and *c*. The HT algorithm accumulates evidence for the existence of the lines. Using the HT algorithm, each pixel within that neighbourhood in the image is examined to match it to a family of line segments, thereby segmenting and isolating the ROI.

$$f(x) = mx + c \tag{3.1}$$

Equation (3.1) for straight lines does not work for vertical lines when applying the HT. (3.2) is used for vertical lines, where *d* is the perpendicular distance from the line to the origin and θ represents the angle the perpendicular makes with the *x*-axis. The HT can be used to find arcs and circles.

Equation (3.3) is the standard equation of a circle which has three variables. If a point (x_1, y_1) lies on the circle then the gradient at (x_1, y_1) points to the centre of the circle and if the radius *r* is selected, and the direction of the vector from (x_1, y_1) to the centre is computed, the coordinates of the centre can be found. The radius *r* is called the row coordinate of the center r_o and the column coordinate of the center is c_o . These variables are used to identify circles and curves using the HT algorithm. The row-column coordinates are represented by equations (3.4) and (3.5).

$$d = x\cos\theta + y\sin\theta \tag{3.2}$$

$$x^2 + y^2 = r^2 \tag{3.3}$$

$$r = r_o + d\sin\theta \tag{3.4}$$

$$c = c_o - d\cos\theta \tag{3.5}$$

During the study it was also found that the performance of the HT deteriorates during segmentation applications where images have multiple shadows and dark areas. This verifies the results of Vozikis and Jansa (2008). Although the HT is used in some applications for segmenting images, it was not used in this study because of its poor performance when segmenting selected areas of an image. This is shown in Fig 3.2(a) and Fig 3.2(b). Fig 3.1(a) and Fig 3.1 (b) are the captured image sequences taken during the HT experiments.

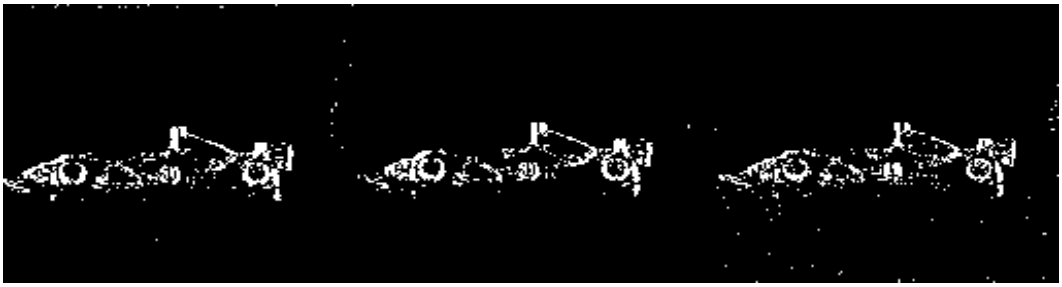
3.3.1 Experiments Using the Hough Transform

(a) Captured Image Sequence for HT Testing

(b) Captured Image Sequence for HT Testing

Fig. 3.1 (a) and (b) Captured Image Sequences for HT testing

(a) Result of HT Segmentation showing noise in ROI and background



(b) Result of HT Segmentation showing noise in ROI and background

Fig 3.2 (a) and (b) Results of Hough Transform Segmentation

3.4 Edge Detection Techniques used as Segmentation

Edge detection methods can be applied to images having shaded and dark areas (Vozikis and Jansa, 2008; Roushdy, 2006). For this reason it was preferred over other segmentation techniques. This is not the case for the HT method. Edge detection methods for segmentation rely on efficient and adaptable edge detection methods to ensure good

edges. Since gray level pixel values change significantly at the edges of objects, the ideal edge can be modeled as a step function. In practice, gray level image pixel values change sharply and can be modeled by a ramp function. Pixel differences in a 2D image $f[x, y]$ can occur in any direction. The maximum change occurs in the direction $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$ along the gradient on the image plane (Shapiro and Stockman, 2001). The estimated contrast at any arbitrary location in an image $i[x, y]$ along the x direction is obtained by computing $(i[x+1, y] - i[x-1, y)/2$, which is the intensity across the left and right neighbours of the pixel [x, y] divided by $\Delta x = 2$ pixel units. Since most pixel values contain noise, edges might actually cut across the pixel array at right angles. This helps to average the three different estimations at which a pixel could lie horizontally (x – direction), vertically (y – direction) and diagonally in the neighbourhood of [x, y]. Equation (3.6) estimates the contrast in the x direction by equally weighing the differences between row y and the row below it. The contrast in the y direction can be estimated similarly by equation (3.7)

$$\frac{\partial f}{\partial x} \equiv f_x \approx \frac{1}{3}\left[(i[x+1, y] - i[x-1, y])/2 + (i[x+1, y-1] - i[x-1, y-1])/2 + (i[x+1, y+1] - i[x+1, y-1])/2\right] \tag{3.6}$$

$$\frac{\partial f}{\partial y} \equiv f_y \approx \frac{1}{3}\left[(i[x, y+1] - i[x, y-1])/2 + (i[x-1, y+1] - i[x-1, y-1])/2 + (i[x+1, y+1] - i[x+1, y-1])/2\right] \tag{3.7}$$

Efficient edge detection methods consist of three stages, namely filtering, differentiation, and detection which use DSP techniques and methods (Shapiro and Stockman, 2001). In the filtering stage, the image is passed through a filter in order to remove noise introduced by DSP sampling and quantization. The differentiation stage highlights the locations in the image where the intensity changes are significant. In the detection stage those points where the intensity changes are significant are localized.

For this study we will test the following conventional edge detection methods, namely CED, LoG and CED.

3.5 The Sobel Edge Detector (SED)

The Prewitt, Sobel and Roberts edge detectors are similar edge detection masks which are scanned across an image while each pixel of the image is convolved with each weighted value of the mask. Each mask can be computed from (3.6) and (3.7). The masks of the Sobel detector are shown in (3.5). The masks which define the Prewitt detector are similar to that of the Sobel detector, except that it has different weighted values in the middle mask matrices. For this study the SED was investigated. The SED has a smoothing factor of 2 versus a smoothing factor of 1 for the Prewitt. The Roberts detector has a 2x2 weighted matrix and is used for images with less detail to save computational time. The advantage of these detectors is that they are computationally simple and can detect horizontal and vertical lines. In practice very few images have only vertical and horizontal lines. Fig. 3.3 shows the image sample used for testing image segmentation algorithms.

$$f_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad f_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$f = f^n[m,n] \otimes f_x\ [m,n] = m * n \tag{3.6}$$

$$f = f^n[m,n] \otimes f_y\ [m,n] = m * n \tag{3.7}$$

With regard (3.6) and (3.7), 'f' is the result of a 2D discrete space convolution which can be applied to any image frame with the detector masks f_x and f_y. The results of this edge detector can be seen in Fig 3.4, Fig 3.8(a) and Fig 3.8(b).

Equations (3.8) and (3.9) show the convolution masks of any image frame *f* at pixel level.

$$f[m,n] = f^{n}[m,n] \otimes f_{x}\ [m,n] = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} m[j,k]n[x-j, y-k] \tag{3.8}$$

$$f[m,n] = f^{n}[m,n] \otimes f_{y}\ [m,n] = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} m[j,k]n[x-j, y-k] \tag{3.9}$$

With regards to (3.8) and (3.9), $f[m,n]$ is the resulting frame after convolution, $f^{n}[m,n]$ is any frame from any image sequence n = 1, 2, 3, ..., n-1, $f_{x}[m,n]$ is the convolution in the x or horizontal direction, $f_{y}[m,n]$ is the convolution in the y or vertical direction and $m[j,k]$ is the convolution mask, with j and k being the elements of the of the convolution mask.

3.6 The LoG Edge Detection

The algorithm for the LoG edge detector is a combination of the Laplace operator and a smoothing filter. The Gaussian Function of one and two variables is shown in equations (3.11) and (3.12), respectively, where c is a scale factor and $r^2 = x^2 + y^2; \sigma$ is the standard deviation of the distribution of normalized pixel intensity. The second derivative which is with respect to r is the operation $\nabla^2 h(r)$ (3.13) which introduces smoothing.

$$\nabla^2 f = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \tag{3.10}$$

$$f(x) = -ce^{-\frac{x^2}{2\sigma^2}} \tag{3.11}$$

$$f(x,y) = ce^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{3.12}$$

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}} \tag{3.13}$$

The net effect of this operation yields a double edged image or *blurring*. Blurring is determined by factor σ and can be a disadvantage when applying this edge detector to moving objects within images as is evident from experimental results shown in Fig 3.5 and Fig 3.10.

3.7 The Canny Edge Detector

The CED is one of the powerful edge detectors and uses a combination of the Gaussian masks and convolution masks for edge detection, filtering and smoothing (Najarian and Splinter, 2006). The algorithm links weak pixels as edges and the strong pixels are called ridge pixels. For this study the canny filter provided very strong edge detection and linked weak noisy pixels as edges. This is evident in the experimental results shown in Fig 3.6, Fig 3.9 (a) and Fig 3.9 (b). This type of filter was too strong for segmentation as it highlighted background noise.

3.8 Edge Detection Experiments

The most common edge detection techniques used were the SED, CED and LoG (Najarian and Splinter, 2006; Gonzalez and Woods, 2008). Initial edge detection experiments were conducted in this study to find a suitable method of segmenting an object in motion. Fig 3.3 is the original captured image followed by the resulting images Fig 3.4, Fig 3.5, and Fig 3.6.

Fig 3.3 Captured Image of Ball

Fig 3.4 Sobel Edge Detection of Ball

Fig 3.5 Laplace of Gaussian Edge Detection of Ball

Fig 3.6 Canny Edge Detection of Ball

3.8.1 Image Segmentation Tests on Image Sequences

Tests to investigate end processing performance were conducted on motion detection in image sequences using the SED, LoG, and CED segmentation techniques. The images used for these tests are given in Fig. 3.7(a) and Fig. 3.7(b). Over-segmentation of an image must be guarded against. As a rule segmentation should stop when the ROI has been isolated. Over segmentation can introduce noise in line segments, borders and in unwanted regions. The ROI becomes larger because line segments and pixels are made thicker than necessary, and the ROI then becomes a false ROI. Under these conditions the ROI will extend into areas that the object does not occupy, as shown by the experimental results given in Fig 3.8, Fig 3.9 and Fig 3.10.

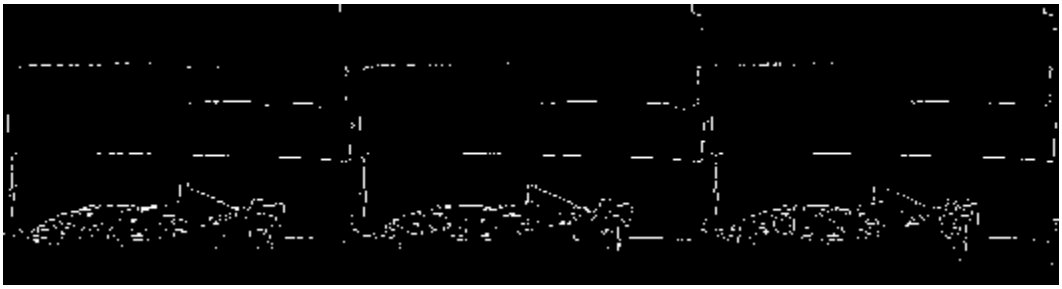
Due to the loss of geometric definition of the ROI, segmentation can cause noise within the ROI. This loss of geometric definition takes the form of discontinuities along the borders of the ROI, and includes regions that were initially excluded from the ROI. Edge information plays an important role in the selection of objects in motion; the detection of moving objects is done by identifying the shifts between edges and ROIs within edges. Based on the results obtained during the experimental stages, the conventional edge detection methods were not suitable for this study because:

- Over segmentation introduced noise in line segments and borders of the ROI by making them appear thicker than necessary,
- Over segmentation caused overshooting by adding or extending the ROI in areas that it does not occupy and
- Over segmentation included unnecessary background.

(a) Captured Image Sequence for testing SED, CED and LoG edge detection.

(b) Captured Image Sequence for testing SED, CED and LoG edge detection

Fig. 3.7 (a) and (b) Captured Image Sequences for testing SED, CED and LoG edge detection.



(a) SED experimental result showing background noise

(b)SED experimental result showing increased background noise

Fig 3.8 (a) and (b) SED experimental results

(a) CED Experimental results showing background noise and over-segmentation

(b) CED Experimental results showing background noise and over-segmentation

Fig 3.9 (a) and (b) CED Experimental Results

(a) LoG Edge Detection Experimental Results showing background noise and over-segmentation

(b)LoG Edge Detection Experimental Results showing background noise and over-segmentation

Fig 3.10 (a) and (b) LoG Edge Detection Experimental Results

3.9 Image Subtraction Techniques for Segmentation and Edge Detection

Research into motion detection and image segmentation shows that image subtraction for edge detection and segmentation of objects produces a good image quality (Shapiro and Stockman, 2001; Mat-Desa and Salih, 2004). A schematic of the image subtraction process for image sequences is shown in Fig 3.11

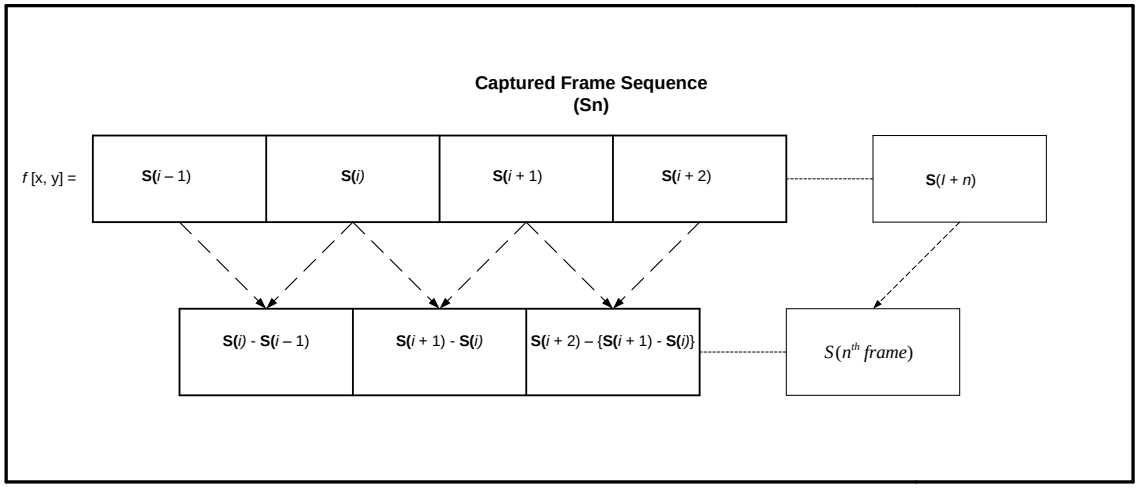


Fig. 3.11 Image subtraction processing for frame sequences

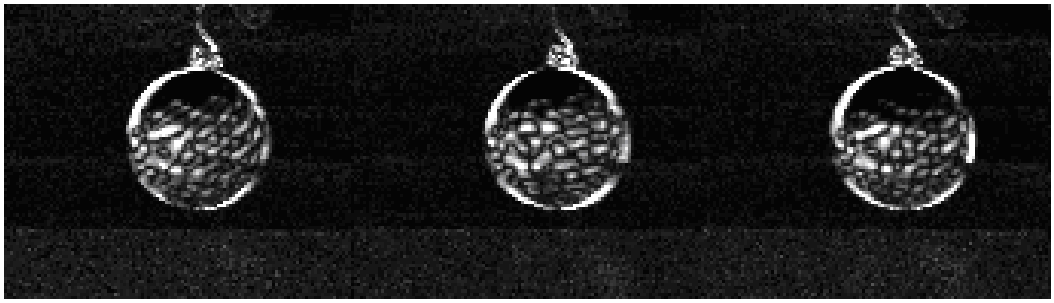
3.9.1 Image Subtraction Experiments used for Moving Object Segmentation

Initial experiments in this study were conducted by using the single frame image shown in Fig. 3.12. The result of motion detection using image subtraction is shown in a single frame image in Fig 3.13. Further experiments were carried out using image subtraction on image frame sequences. The captured image frame sequences and their results are shown in Fig. 3.14(a), Fig. 3.15(a), Fig. 3.16(a) and Fig. 3.17(a). From the results obtained in Fig 3.14 (b), Fig 3.15(b), Fig 3.16(b) and Fig 3.17(b) it is evident that image subtraction provided better results than the other segmentation methods discussed in this chapter.

Fig 3.12 Captured Image of Ball to test Image Subtraction

Fig 3.13 The Result of Image Subtraction between two frames of a Ball in Motion

(a) Captured Image Sequence of a Ball to test Image Subtraction of multiple frames



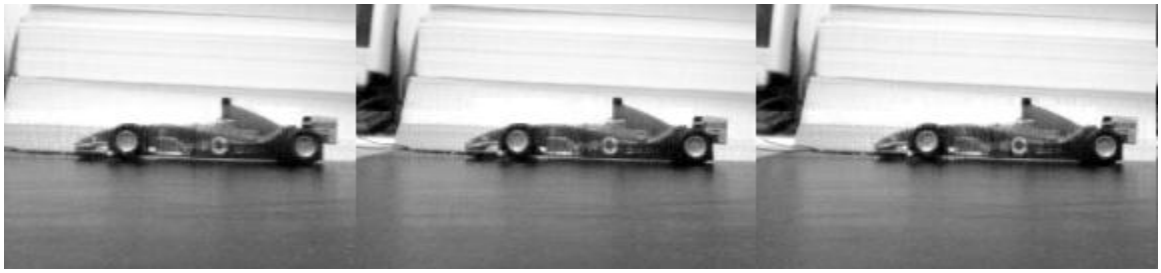
(b) Experimental Results of Image Subtraction of a Ball of Multiple frames

Fig 3.14 (a) and (b) Image Subtraction Tested on a Ball

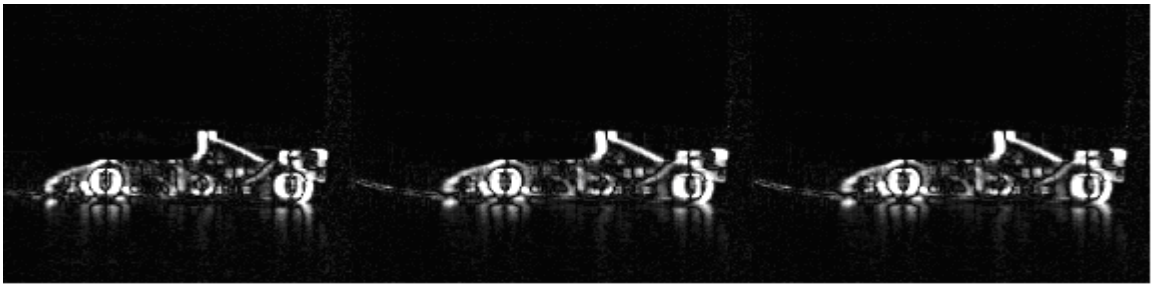
(a) Captured Image Sequence of a Ball to test Image Subtraction of Multiple Frames

(b) Experimental Results of Image Subtraction of a Ball of Multiple frames

Fig 3.15 (a) and (b) Image Subtraction Tested on a Ball in Motion



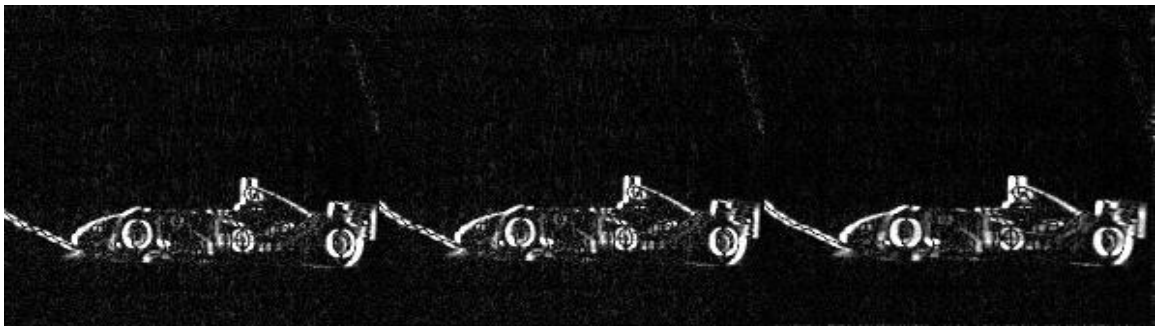
(a) Captured Image Sequence of Car to Test Image Subtraction of Multiple Frames



(b) Experimental Results of Image Subtraction of a Car of Multiple Frames

Fig 3.16 (a) and (b) Image Subtraction Tested on a Car in Motion

(a) Captured Image Sequence of Car to Test Image Subtraction of Multiple Frames



(b) Experimental Results of Image Subtraction of a Car of Multiple Frames

Fig 3.17 (a) and (b) Image Subtraction Tested on a Car in Motion

3.10 Conclusion

This chapter has discussed and tested image segmentation methods such as SED, CED, LoG and HT. From the experimental results obtained, these conventional segmentation methods were unsuitable because they resulted in over-segmentation of the ROI, and also segmented the noise in the images as part of the ROI. Further research also had to be performed on image subtraction segmentation for moving objects. This method provided suitable results and was used for segmentation of the moving objects. The next chapter will investigate DSP systems and will also propose an ANN ensemble system for motion detection, object extraction and filtering.

Chapter 4

DSP System *vs* ANN Ensembles for Motion Detection, Object Extraction and Filtering

4.1 Introduction

This chapter discusses a comparative study between DSP systems and ANN's for object motion detection and extraction. The ANN system is arranged as an ensemble of NN's to perform the motion detection, object isolation and filtering functions. The DSP motion detection system uses image subtraction for segmentation. Median filtering and image enhancement is used to produce improved image quality. The ANN ensemble motion detection system uses no preprocessing methods except gray scaling of the image frame sequences. All image frames are presented in parallel to each NN within the ensemble, and is mapped to the output pixel by pixel. Using this technique makes image preprocessing redundant. Image filtering, image extraction and isolation of the object in motion is accomplished using the proposed ANN motion detection system.

PART 1: MOTION DETECTION USING DSP SYSTEMS

4.2 DSP Image Processing

Digital Signal Processing techniques are widely used in image processing functions (Gonzalez and Woods, 2008). The 1D-FT for a continuous signal shown in (4.1) is modified for a digital image of *M x N* dimensions to produce a 2D continuous function $f(x, y)$ described in (4.2).

$$\Im\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}dx \tag{4.1}$$

$$\Im\{f(x,y)\} = F(u,v) = \int_{-\infty}^{\infty}\int f(x,y)e^{-j2\pi(ux+vy)}dxdy \tag{4.2}$$

With regards to (4.1), (4.2), (4.3) and (4.4), u and v denotes the frequency variables of $f[x,y]$ which is a square image frame of M x N dimensions of digital image matrix. Sampling of the 2D continuous signals is applied to produce discrete representations of the images to be processed. This produces the 2D-DFT and the 2D-IDFT as shown in (4.3) and (4.4) where $f_\alpha[x,y]$ is a square image frame and $M = N$.

$$F(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f_\alpha[x,y]e^{-j\frac{2\pi(ux+vy)}{N}}$$

$$u,v = 0,1,2,\ldots\ldots,N-1 \tag{4.3}$$

and the inverse 2D-IDFT is then defined as

$$f_\alpha[x,y] = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j\frac{2\pi(ux+vy)}{N}} \tag{4.4}$$

$$m,n = 0,1,2,\ldots\ldots,N-1$$

For digital image processing 2D continuous signals have to be digitized to produce discrete signals. For this study (4.3) and (4.4) represent frequency components of a grey scaled square image frame matrix $f_\alpha[x,y]$. Equation (4.4) is the IDFT where α represents the frame number of an image sequence. The DFT can then be modified for any given image $f[x,y]$ of a two dimensional array with m rows and n columns.

4.3 The DSP Motion Detection System

In higher level applications of image processing it is necessary to identify the boundary from the ROI of the object, and to separate the ROI from the surrounding irrelevant background. Detection and extraction of the ROI in an image are referred to as segmentation of the image into the required components of the image (Murta *et al*, 2006). In this study image subtraction was applied as the method of segmentation based on the results and experiments carried out in chapter 3.

After gray scaling of the captured image sequences and extracting the ROI by image subtraction segmentation, the boundaries of the ROI were represented by fast variations or rapid changes in the gray level of the neighbouring pixels. Such fast variations or intensity changes are high frequency components of the image that represent the edges and boundaries of the object of interest. The object of interest can for example be an object in motion as is the case shown in this study.

The high frequency components in an image were computed using the DFT to represent sharp variations, or *edges*, in pixel intensity levels that occur within an image or along its borders (Najarian and Splinter, 2006). This prepared the image and its isolated elements for further processing and is used for the DSP motion detection function. A schematic of the steps involved in DSP motion detection is given in Fig 4.1.

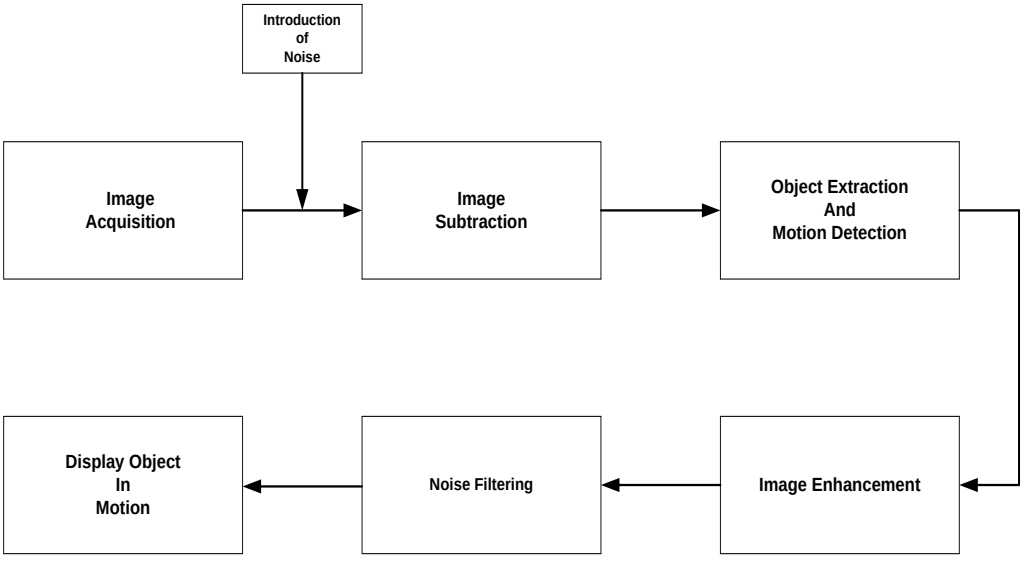


Fig. 4.1 Diagram of DSP Motion Detection System

4.3.1 Image Acquisition

Image sequences were acquired using a video camera and a frame grabber which buffered the captured images for processing. The image frames were gray-scaled and resized to produce a square 255 x 255 image frame.

4.3.2 Image Subtraction

In this study image subtraction is used for temporal differencing and also to remove irrelevant background and stationary elements from the frame sequences under consideration. An image subtraction function (4.5) is used to:

- Extract the static region of the image from its dynamic region. This dynamic region becomes the ROI and
- Segment the ROI and isolate the object in motion.

4.3.2.1 Background Subtraction

The background subtraction algorithm used in the study is given in (4.5).

$$f_{\alpha}[x,y] - \mathbf{B} = G \tag{4.5}$$

With regards to (4.5):

$G = \begin{cases} > 0 & motion \\ 0 & no\ motion \end{cases}$ and $f_{\alpha}[x,y]$ represents the composite background being considered; **B** denotes the static background matrix, G is the cluster of pixels in the ROI and α indicates the number of the captured frames. This motion can be shown as the pixel intensity of G in any captured frame $f_{\alpha}[x,y]$ (Desa and Salih, 2004).

4.3.2.2 Temporal Differencing

Temporal differencing was applied to each successive frame following background subtraction as described in (4.6). The extracted ROI becomes the new image frame $f_r[x, y]$.

$$f_r\,[x,y] = f_\alpha\,[x,y] - \mathbf{B} \tag{4.6}$$

From (4.5) and (4.6) then G = $f_r[x, y]$. With regards to (4.6): $f_r\,[x,y]$ is the resulting new image frame following background subtraction, where *r* denotes the frame number, α indicates the number of captured frames. Equations (4.7) and (4.8) are used for temporal differencing. With regards to (4.7): $\mathbf{H}_1$, $\mathbf{H}_2$, …$\mathbf{H}_n$ represents the image matrix $f_r\,[x,y]$ $\mathbf{B}_1$, $\mathbf{B}_{2,\ \ldots}\ \mathbf{B}_n$ denotes the static background, $\mathbf{C}_1$, $\mathbf{C}_{2,\ \ldots}\mathbf{C}_n$ is the resulting matrices of the sequence S_1. Temporal differencing was then applied to each successive matrix frame of (4.8)

$$S_1=[\mathbf{H}_1 - \mathbf{B}_1],\ [\mathbf{H}_2 - \mathbf{B}_2],\ \ldots..,\ [\mathbf{H}_n^{\,th} - \mathbf{B}_n]\ =\ \mathbf{C}_1,\ \mathbf{C}_2\ ,\ldots..,\ \mathbf{C}_n \tag{4.7}$$

$$\mathbf{C}_{d1} = |\mathbf{C}_1\text{-}\mathbf{C}_2|,\ \ \mathbf{C}_{d2}= |\mathbf{C}_2\text{-}\mathbf{C}_3|,\ \ldots,\ \mathbf{C}_{dn} = |\mathbf{C}_{n-1}\text{-}\mathbf{C}_n| \tag{4.8}$$

With regards to (4.8): each ***C-matrix*** is subtracted from its preceding matrix. The difference between the two corresponding image matrix values were converted into an absolute value and stored in a difference matrix $\mathbf{C}_d$ to eliminate negative values. This process is continued for all images in the frame matrix. Motion mask ***U*** (4.9) was obtained from S_1 after background subtraction.

$$U = \begin{Bmatrix} 1 & C_d > 0 \\ 0 & no\ motion \end{Bmatrix} \tag{4.9}$$

(4.9) is applied to each successive difference matrix $\mathbf{C}_d$ and results in the image difference matrix sequence S_2 (4.10).

$$S_2 = \mathbf{C}_{d1}, \mathbf{C}_{d2}, \ldots, \mathbf{C}_{dn} \tag{4.10}$$

4.4 Image Enhancement and Filtering

The ROI of the resulting difference image matrix (4.10) might occupy a larger area of each image in S_2. Each image is enhanced by increasing the contrast of the images to clearly define the ROI in order to define the object in motion from the difference image sequence S_2. Image enhancement simultaneously enhances noise as indicated in Fig. 4.2(c), Fig. 4.3 (c) and Fig. 4.4(c). Median filters based on 2D convolution (4.11) and (4.12) were used to filter the system and any induced noise (see Fig 4.2 (d), Fig 4.3 (d) and Fig 4.4 (d)). These filters were chosen so as to avoid the side effect damage to pixels within the ROI (Liu *et al*, 2005).

$$f_{filter} = f_Z\ [x, y] \otimes f_{win}\ [x, y] \tag{4.11}$$

$$f_{filter} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_z[m, n] f_{win}[x - m, y - n] \tag{4.12}$$

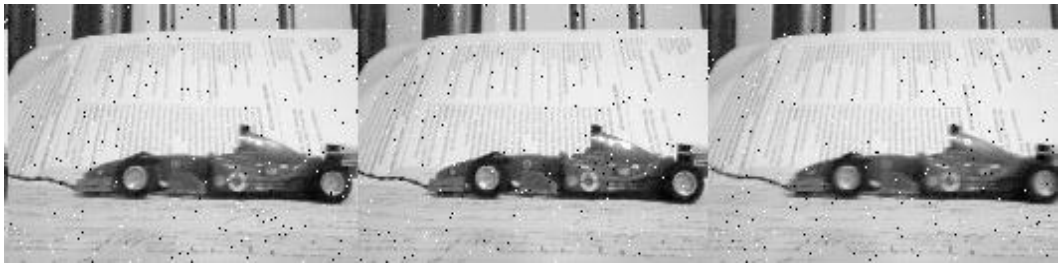
For x =0, 1, 2,……,M-1 and y = 0,1,2,….., N – 1

With regards to (4.11) and (4.12): f_{filter} represents the resulting filtered image, $f_{win}[x, y]$ is the filtering window having x-rows and y-columns and f_z [x,y] denotes the 2D frame sequence following the introduction of noise. In this study, the median filter minimizes noise whilst preserving edges and any high frequency components defined by (4.3). Equation (4.13) shows the introduction of noise into difference matrix (4.10) and the resulting image sequence **Z**, where ψ represents different levels of induced salt and pepper noise and **Z** denotes the image having noise as shown in Fig. 4.2(a), Fig. 4.3(a) and Fig. 4.4(a).

$$(\mathbf{C}_{d1}, \mathbf{C}_{d2}, \ldots, \mathbf{C}_{dn})\ \psi = \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n \tag{4.13}$$

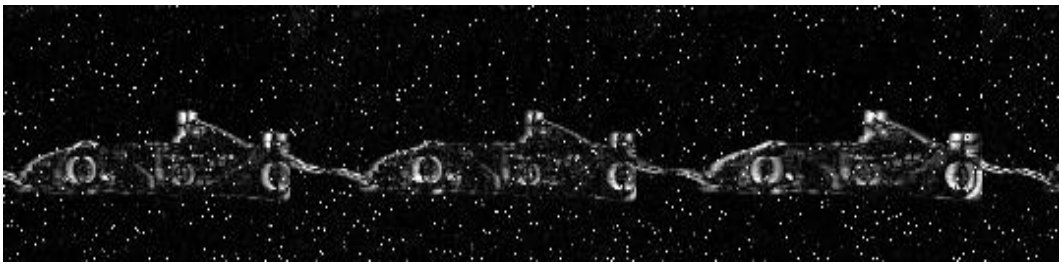
4.5 Testing of the DSP Motion Detection System

Fig 4.2 (b), Fig 4.3(b) and Fig 4.4 (b) are the results of DSP motion detection and extraction.



(a) Composite image Sequence

(b) Motion Detection and Extraction

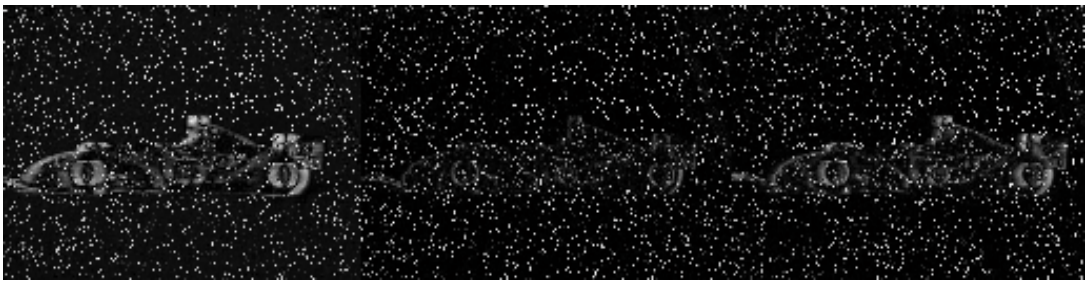


(c) Enhanced Images

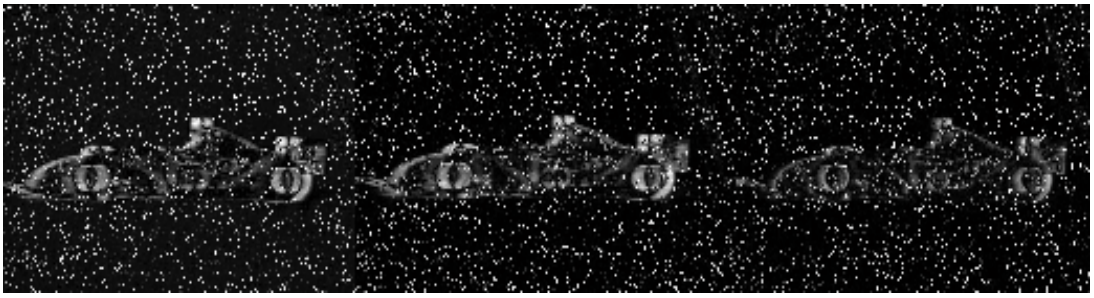
(d) Filtered Images

Figure 4.2: DSP Motion Detection with 0.01 Salt & Pepper Noise

(a) Composite Image Sequence



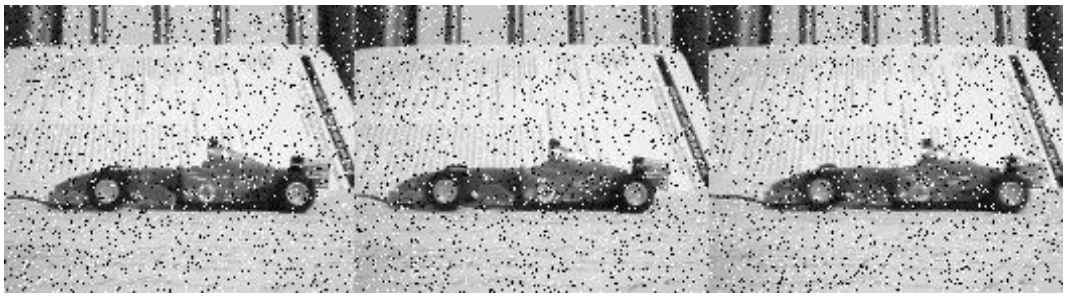
(b) Motion Detection and Extraction



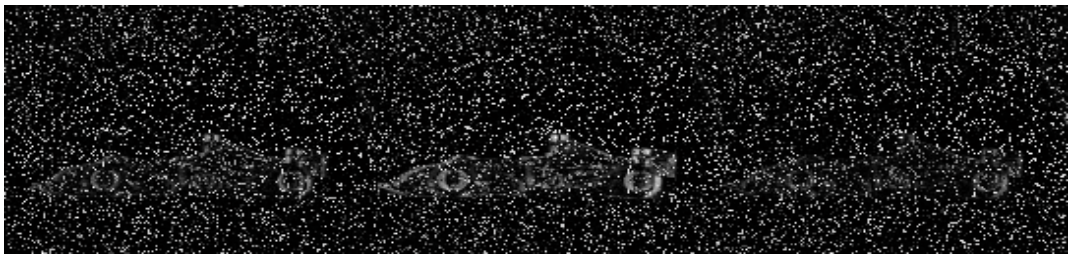
(c) Enhanced Images

(d) Filtered Images

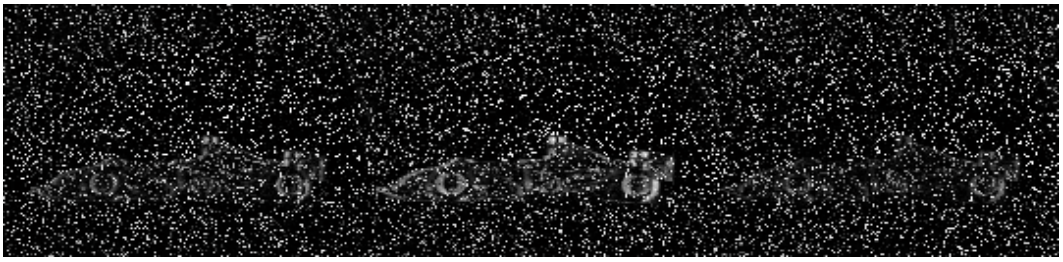
Figure 4.3: DSP Motion Detection with 0.05 Salt & Pepper Noise



(a) Composite Image Sequence



(b) Motion Detection and Extraction



(c) Enhanced Images

(d) Filtered Images

Figure 4.4: DSP Motion Detection with 0.09 Salt & Pepper Noise

PART 2: MOTION DETECTION USING ANN's

4.6 Introduction

ANNs consist of individual processing elements, termed neurons, which are modeled along the lines of the biological neuron. Each artificial neuron consists of an input, weight, summation, activation function and an output. ANN's are characterized as parallel computing networks having the ability to adapt or learn, to generalize and organize data. Unlike conventional non-intelligent systems, ANN's are first presented with data during training. A learning process within the artificial neural network occurs so that the ANN adaptively responds to inputs according to pre-programmed learning rules. After the neural network has learnt what it needs to know, the trained network can be used to perform tasks depending on the application. NN have the ability to learn from their environment and to adapt to it in an interactive manner similar to their human biological counterparts. This is an exciting prospect because of the vast possibilities that exist for performing certain functions with artificial neural networks that can emulate to a limited degree the comparable biological logical functions.

4.7 Artificial Neuron

An artificial neuron can also be referred to as a processing element, node, or threshold logic unit, however it is refer to as a "neuron". A neuron is an information processing unit that roughly resembles its biological counterpart as shown in Fig 4.5. Fig 4.5 shows the model of an artificial neuron.

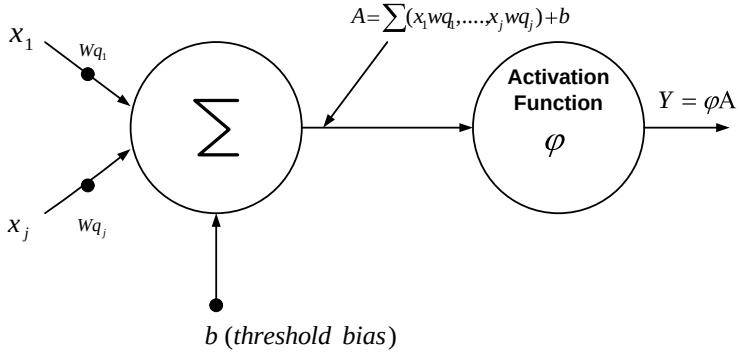


Fig. 4.5 Artificial neuron

4.7 Synaptic Weight and Inputs

Any input to the synapses can be represented by a vector signal $x = \Re(n \times 1)$, with the individual vector components given as x_j, for j = 1, 2,…,n, that is $\boldsymbol{x}$ = $[x_1, x_2,\ldots, x_n]^T$. Therefore, each vector component x_j is input to the jth synapse and connected to the neuron through a synaptic weight wq_j; that is, x_j is multiplied by the synaptic weight wq_j.

4.9 Algebraic Summer

The summer combines all the signals presented to it. All the operations up to and including the output of the adder *A* constitute a linear combiner which is a combination of the inputs to the synapses.

4.10 Activation Function (φ)

The activation function limits the amplitude of the neuron output *Y*. It can be continuous, binary, bipolar, linear or a non-linear function. When the activation function is nonlinear, its finite limits are typically normalized in the range of [0, 1] (Narendra and Pathasaraty, 1990). In highly interconnected massively parallel ANN architectures, the nonlinearities can serve to enhance the network's classification approximation and noise immunity capabilities (Ahmed *et al*, 2005).

4.11 Bias (b)

The bias is applied externally and lowers the cumulative input to the activation function. The threshold value is subtracted from the output of the linear combiner *Y* before the activation is applied. The input to the activation function could be raised by adding a term to *Y* which is the bias.

4.12 Output (*Y*)

Y is the output of the neuron and is a function of the applied inputs, weights, activation function and ANN bias after training.

4.13 The Multiple Layer Feed Forward Neural Networks

MLFF neural networks are made up multiple neural networks working as parallel units as in the human brain where biological neurons are connected as massive parallel units which process data simultaneously. ANN's are taught using either supervised training or *unsupervised* training.

Supervised Training: Supervised learning occurs when the NN is presented with a set of desired vectors which acts as a teacher. During training the network's output vectors are monitored and its weights are adjusted until the network has completing its learning phase under the guidance of a teacher or supervisor. Some popular supervised training methods include Hebbian learning and back-propagation training.

Unsupervised Training: In unsupervised training there is no knowledge of the output vectors and the network seeks to find patterns or regularity in the input data. To produce optimum results different training methods are used depending on the application. Unsupervised Learning methods include Kohonen self-organizing maps and adaptive. Learning networks.

For this study the supervised training has been used to train the MLFF-NN (see Fig. 4.6) with the backpropagation algorithm. Using this method, the weights are adjusted until the input and output pairs are representative of one another, and image patterns are generated until the input and output image patterns are representative of one another (Ham and Kostanic, 2001).

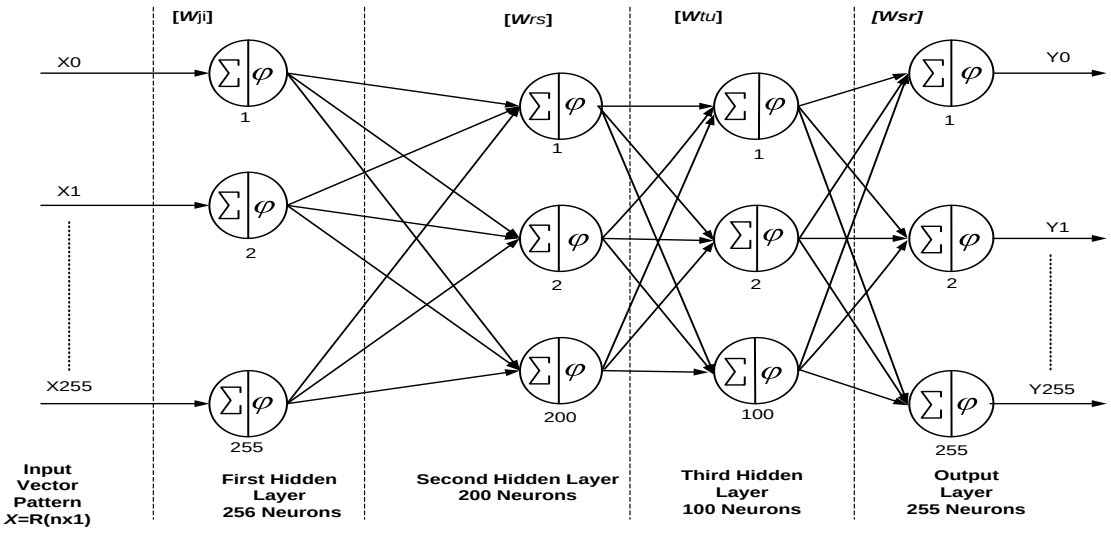


Fig 4.6 Architecture of each ANN System Comprising of ANN Ensembles
(Sigmoidal multilayer feedforward network (MLFF))

Figure 4.6 shows the architecture of a single MLFF neural network from the original NN ensemble designed for this study. Each layer has a synaptic weight matrix associated with all the connections made from the previous layer to the next layer. The input vector patterns of image frame $f_{\alpha}[x, y]$ where α = 1, 2, …., n indicates the number of frames presented to the each NN within the ensemble layers. The input pattern can be represented by $\boldsymbol{Xj}$ for j = 1, 2, …, n multiplied by the weight matrix $[\boldsymbol{W}_{ji}]$ with i = 1, 2, …, n. The second layer with 200 neurons has weight matrices given by $[\boldsymbol{W}_{rs}]$ with r = 1, 2, …., n and s = 1, 2, …., n. The third layer with 100 neurons has a weight matrix $[\boldsymbol{W}_{tu}]$ for t = 1,2,…,n; and u = 1,2,…,n. The weights for the 256 output neurons are $[\boldsymbol{W}vz]$ while the output patterns and the input patterns are mapped to the input matrix pattern by using 256 neurons. The output vector matrix is expressed using equation (4.14), where n = 1, 2, …., n

$$Y_n = [f^1[x_1, y_1],\ f^2[x_1, y_2],...,\ f^n[x_n, y_n]] \tag{4.14}$$

Equation (4.15) describes the non-linear mapping of the input vectors to the output vectors of a supervised backpropagation trained MLFF-NN..

$$Y_n = f^3[W^3 f^2[W^2 f^1[W^1 x]]] \tag{4.15}$$

The ANN training process involves presenting key patterns (vectors) to the network which are then transformed into weight vectors. The memory pattern can be stored in all weight layers of the MLFF after training. If the network is too small or too large, the network can then be pruned to reduce memory demands on the system, and to increase processing speed for faster convergence on the training parameter goal. With the memory matrix layers set, the overall desired response of the system can be achieved for input test pattern vectors which are corrupted with noise, have lost or added data, or have been altered in any way. The ANN system can present the original data vectors with an accuracy dependant on the training goal parameters set by the system user (Nelwamondo and Marwala, 2007). A schematic of the ANN motion detection system used in this study is shown in Fig 4.7.

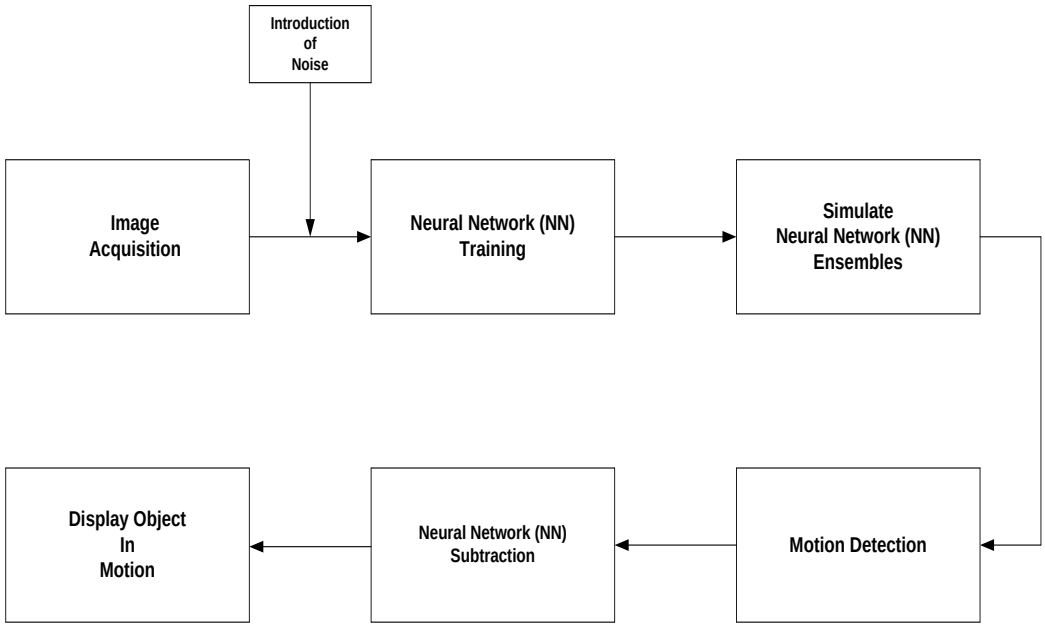


Fig. 4.7 ANN Motion Detection System

4.14 The ANN Motion Detection System

The internal topology of the ANN's shown in Fig. 4.7 is given Fig 4.8. It uses MLFF backpropagation networks and a log-sigmoid nonlinear activation function to detect motion and remove any stationary background. The MLFF architecture is designed with two hidden layers and an output layer, having 100, 200 and 256 neurons respectively (see

Fig. (4.6). The output layer maps corresponding pixel elements from an input image matrix to the output image matrix. The ANN's are arranged to operate as an ensemble to relieve the computational burden, reduce noise and promote faster processing times when handling the large quantities of image data (Moorgas and Govender, 2008; Zuo and De With, 2008). The image sequences were split into single frames and applied to the ANN system for motion detection.

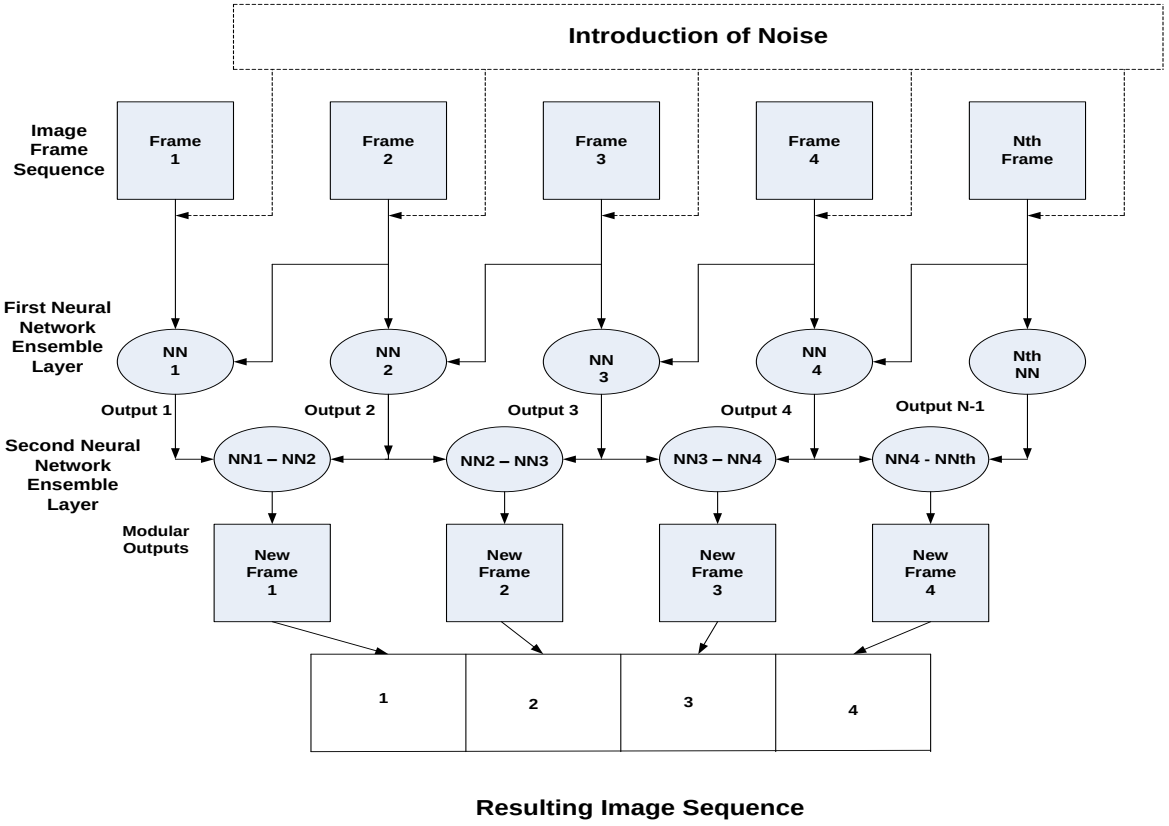


Fig. 4.8 ANN Ensemble Motion Detection System

(Source: Moorgas and Govender, 2008)

4.15 Operation of the ANN Motion Detection System

Fig 4.9 is a test sample of the image sequence used in the NN and is similar to that used for the DSP detection system described in chapter 3. Each gray scaled 255 x 255 image frame matrix is first normalized to the [0, 1] range for faster convergence during training, before being applied into the network (Ahmed *et al.*, 2005). Each subsequent frame forms

the target for the preceding NN for motion detection. Captured image frame sequence S_3 (4.16) which are the gray scaled normalized image sequence is presented to the ANN system.

Fig. 4.9 Sample of ANN training image sequence

$$S_3 = (\mathbf{H'}_1, \mathbf{H'}_2, \ldots\ldots, \mathbf{H'}_n) \quad (4.16)$$

With regards to (4.16), **H'** represented the respective image frame matrix applied to its corresponding network in the ensemble. With regards to Fig 4.9, the first layer of ANN's behaved as a motion detecting ensemble to indicate the new position of the object and filtered the image sequence. The output of the first ensemble layer is represented by (4.17).

$$S_4 = (\mathbf{C'}_1, \mathbf{C'}_2, \ldots\ldots\ldots, \mathbf{C'}_n) \quad (4.17)$$

The second ensemble layer performs the image subtraction function to remove any stationary background. The output of the second ensemble layer is the output of the system and indicates the non-stationary object sequence (4.18).

$$S_5 = (\mathbf{C'}_{d1}, \mathbf{C'}_{d2}, \ldots\ldots, \mathbf{C'}_{dn}) \quad (4.18)$$

4.16 ANN Noise Filtering and Image Enhancement

ANN's are robust processing elements and have the ability to remove unwanted noise signals from the images under consideration (Moorgas and Govender, 2008; Bitten and

Osorio, 2000). The noise immunity of the ANN ensemble system in Fig 4.8 is tested by adding different levels of 'salt and pepper noise' to the image sequence in Fig 4.10. Fig 4.11 is the result of motion detection without induced noise. Fig 4.12(a), Fig 4.13(a), and Fig 4.14(a) shows Fig 4.10 after different levels of salt and pepper noise is added to it. The output of the ANN ensemble system is shown in Fig. 4.12(b), Fig. 4.13(b) and Fig. 4.14(b). Image subtraction is performed within the NN system and removes any stationary background from the sequences represented by Fig. 4.12(c), Fig. 4.13(c) and Fig. 4.14(c). To ensure the validity of the comparison between the ANN and DSP systems, the image sequence is enhanced and filtered under the same conditions as the DSP in order to standardize the end processing (see Fig 4.12(d), Fig 4.13(d) and Fig 4.14(d)).

Fig: 4.10: Captured Image Sequence for testing ANN Motion Detection System

Fig. 4.11Experimental Results of ANN Motion Detection without Induced Noise

4.17 Testing the ANN motion Detection System

(a) Composite Image Sequence

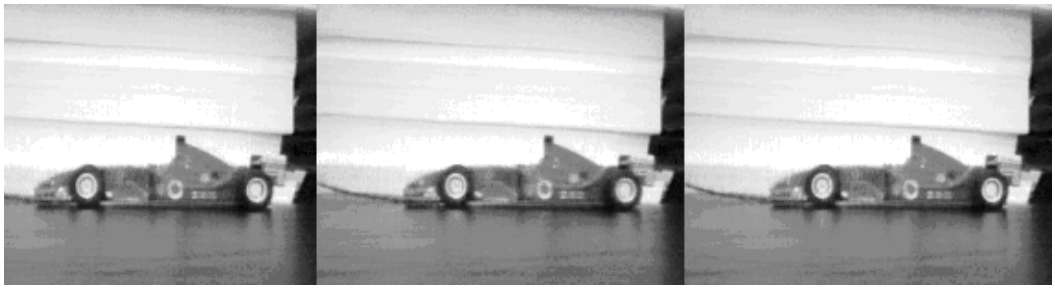
(b) Motion Detection and Filtering

(c)ANN Image Subtraction

(d)Image Enhancement

Fig. 4.12 ANN Motion Detection with 0.01 Noise

(a) Composite Image Sequence



(b) Motion Detection and Filtering

(c) ANN Image Subtraction

(d) Image Enhancement

Fig. 4.13 ANN Motion Detection with 0.05 Salt & Pepper Noise

(a) Composite Image Sequence

(b) Motion Detection and Filtering

(c) ANN Image Subtraction

(d) Image Enhancement

Fig. 4.14 ANN Motion Detection with 0.09 Salt & Pepper Noise

PART 3:

4.18 Comparison between the Performance of DSP Systems *vs* ANN Ensembles

Equation (4.19) represents the PSNR ratio for measuring the quality of the images produced by the DSP and ANN systems during end processing. With regards to equation (4.19), $g_{\max}$ and $g_{\min}$ indicate the maximum and minimum pixel values for an image frame; *MSE* is the mean squared error for two *m x n* monochrome images defined by equation (4.20)

$$PSNR = 10 \log_{10} \frac{[g_{\max} - g_{\min}]^2}{MSE} \tag{4.19}$$

where

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \| I[x, y] - K[x, y] \|^2 \tag{4.20}$$

and $I[x, y]$ and $K[x, y]$ represent the noise contaminated image and the filtered image respectively.

4.18.1 Noise Analysis of DSP System *vs* ANN Ensembles

Graph 4.1 shows PSNR plots comparing the tests conducted on the captured image sequences during DSP and ANN motion detection. These plots correspond to the frames shown in Fig. 4.2, Fig. 4.3 and Fig. 4.4 for the DSP system and Fig 4.12, Fig 4.13, and Fig 4.14 for the ANN Ensemble system.

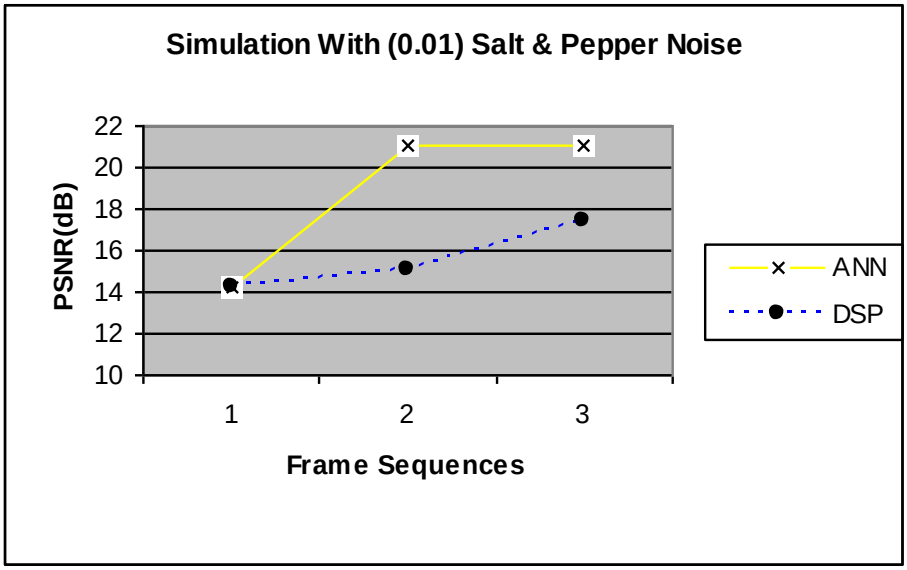
With regards to Graph 4.1 (a), Graph 4.1 (b) and Graph 4.1(c):

\

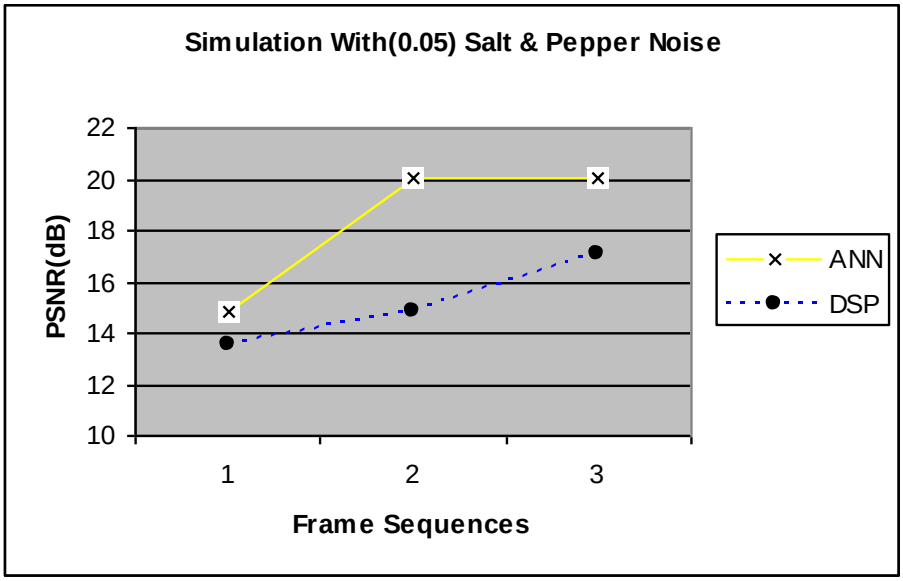
Noise induced in 1st Frame Sequence: The ANN system response in Fig 4.12(a), Fig 4.13(a) and Fig 4.14 (a) has a higher level of PSNR over the DSP Fig. 4.2(a), Fig. 4.3(a), Fig. 4.4(a).

Motion is detected in 2nd Frame Sequence: The ANN system response in Fig 4.12(b), Fig 4.13(b) and Fig 4.14(b) inhibits any negative effects of noise on the image quality Fig. 4.2(b), Fig. 4.3(b), Fig. 4.4(b).

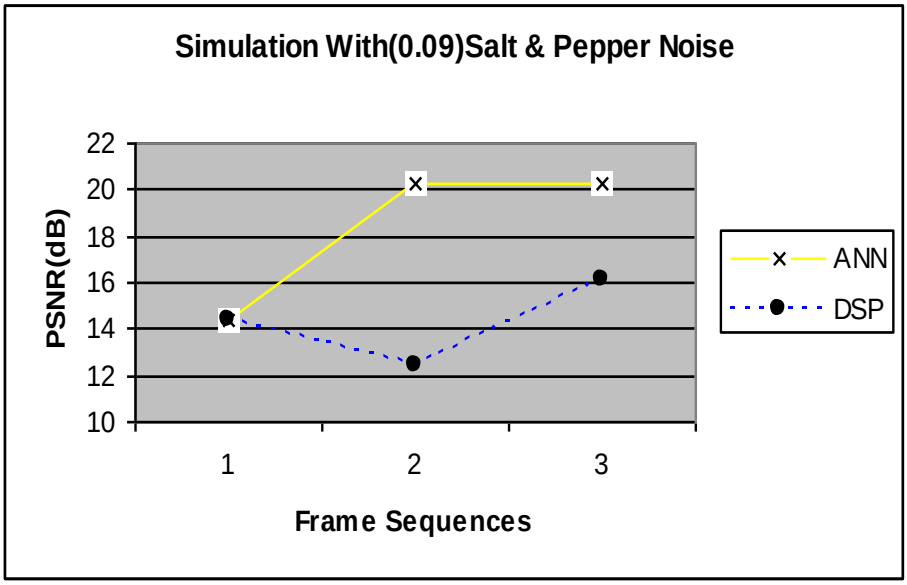
DSP median filtering in 3rd Frame Sequence: The performance of the ANN system as a filter is given in Fig 4.12(d), Fig 4.13(d) and Fig 4.14(d) yields a higher image quality over that of the DSP system given in Fig 4.2(d), Fig 4.3(d) Fig 4.4(d).



(a) First Set of Image Sequences



(b) Second Set of Image Sequences



(c) Third Set of Image Sequences

Graph 4.1: (a),(b) and (c) PSNR Response of DSP versus the ANN system

4.18.2 Average Processing Time of DSP vs ANNs

Table 4.1 shows the average processing time performance for the DSP motion detection system versus the ANN ensemble motion detection system. The DSP system is faster than the ANN system even when the training time is not considered. Although the DSP is faster, its algorithms are mathematically intensive and complex (Dumontier *et al*, 1999). DSP systems become expensive at high bandwidths and the cost of highspeed analogue to digital and digital to analogue converters at speeds higher than 100 MHz makes them impractical for many applications. The high power consumption and size of DSP systems can make them unsuitable for simple very low–power or small size applications. If computational time is not critical then the proposed ANN system is much more efficient. DSP processing time is much faster than the ANN's but at the cost of image quality when compared to the ANN.

Computer System Pentium 4, 3.0Ghz, 2.0GB RAM	**DSP Motion Detection System**	**ANN Motion Detection System**
(without noise)	4.5s	7min
(with induced noise)	5s	7min

TABLE 4.1Average System processing time

4.19 Conclusion

This chapter described the design and implementation of the proposed ANN motion detection system and compares its performance with a conventional DSP motion detection system. The results in terms of PSNR show that the ANN system out-performs the DSP system. The DSP system has superior speed but has produced poor quality images after motion detection. In order to improve the quality of the images a median filter has to be introduced. However this added filter did not match the quality of the ANN system. In order to improve the processing speed of the proposed ANN motion detection system, the DSP system was combined with the ANN system to produce a hybrid system (Hybrid DSP + ANN Ensemble System) – this is discussed in the following chapter.

Chapter 5

Object Motion Detection Methods and Applications

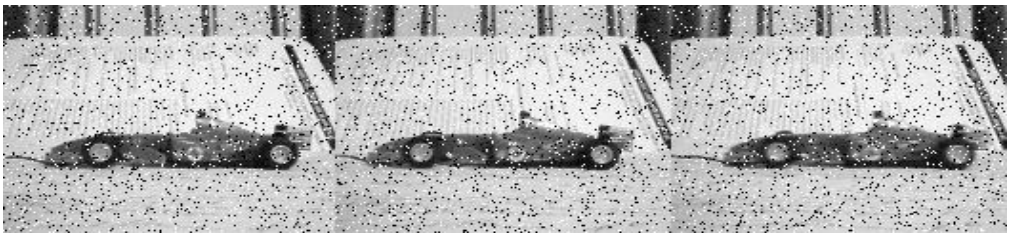
5.1 Introduction

The performance of the DSP motion detection system and the proposed ANN ensemble motion detection system was compared in chapter 4. The results showed that the ANN system displayed a superior performance in all aspects save computation time. This chapter focuses on combining selected aspects of DSP to the ANN ensemble based motion detection system, with a view to improving processing time without sacrificing image quality.

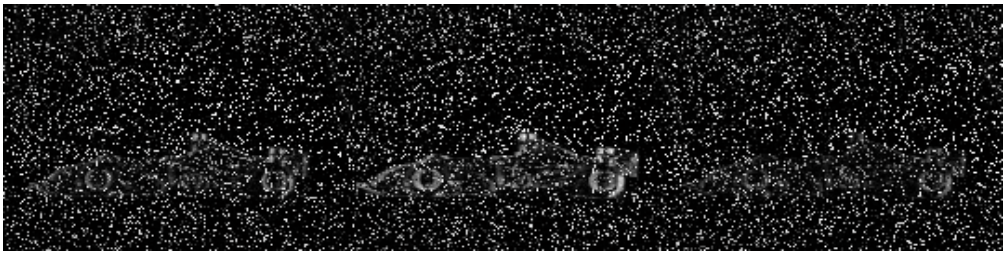
5.2 Further Testing of the DSP Motion Detection System

Chapter 3 described motion detection using DSP for a single object. This section will discuss how DSP motion detection can be performed on a variety of different objects. The results of the tests are shown in Fig 5.1, Fig 5.2 and Fig 5.3. The conditions under which these tests were conducted are the same as was mentioned in chapter 3.

(a) Captured Images Sequences



(b) Induced Noise Salt & Pepper at 0.09

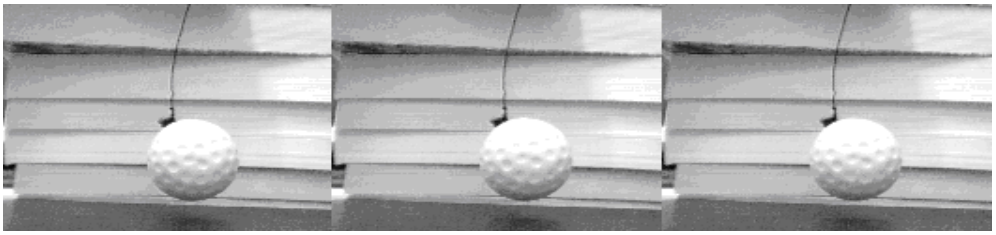


(c) DSP Motion Detection

(d) Median Filtered Images

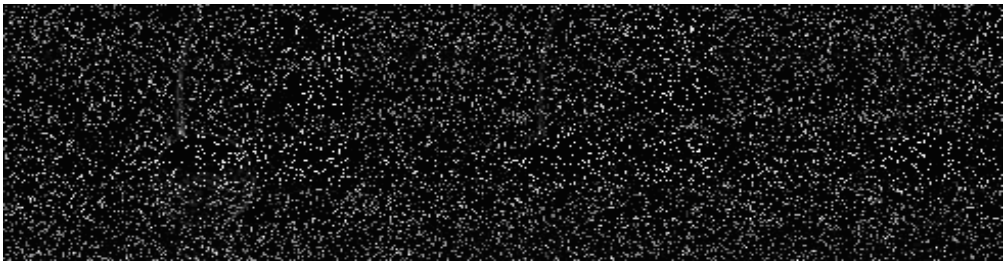
(e) Enhanced Images

Fig 5.1 Car Motion Detection using DSP System

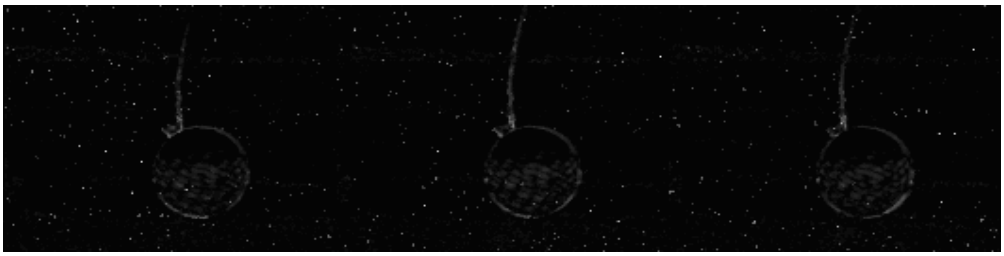


(a) Captured Image Sequences

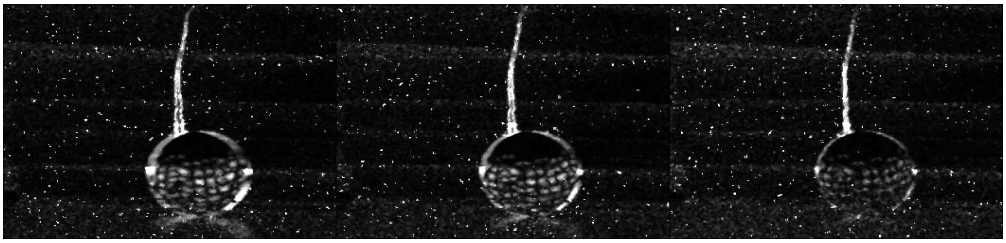
(b) Induced Salt and Pepper Noise at 0.09



(c) DSP Motion detection



(d) Median Filtered Images

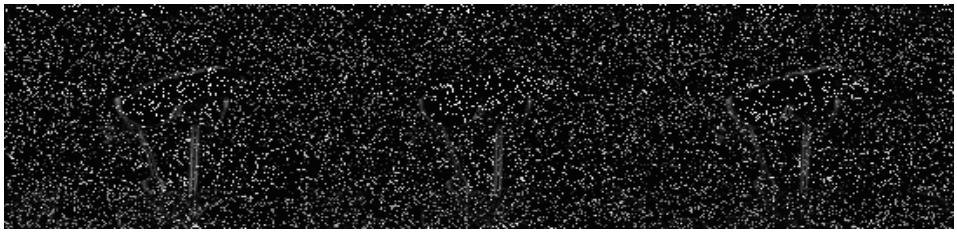


(e) Enhanced Images

Fig 5.2: Ball Motion detection using DSP System

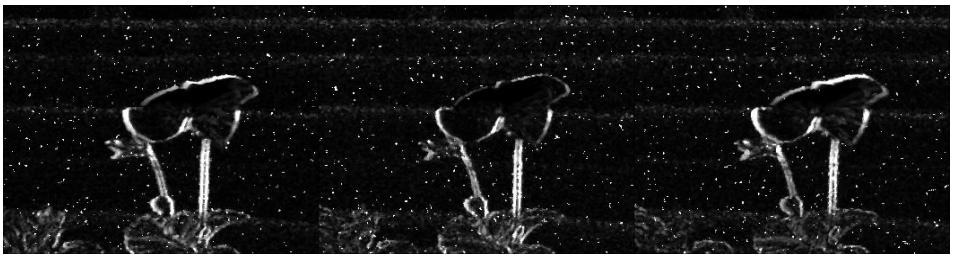
(a) Captured Image Sequences,

(b) Induced Salt and Pepper Noise at 0.09



(c) DSP Motion Detection

(d) Median Filtered Images



(e) Enhanced Images

Fig 5.3: Plant Motion detection using DSP System

5.3 Further Tests conducted using the ANN Ensemble Motion Detection System

Further tests to asses the performance of the ANN ensemble based motion detection system were conducted using the system as an ensemble filtering system because of its inherent immunity to noise and generalization. The results are shown in Fig 5.4, Fig 5.5 and Fig 5.6.

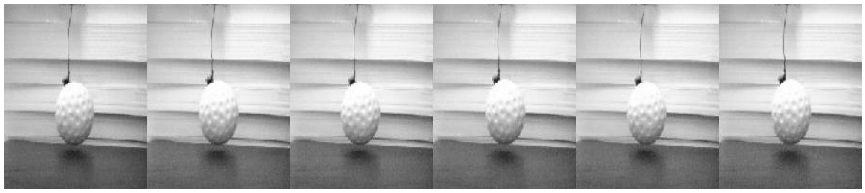
(a) Captured Images Sequences

(b) Induced Noise Salt & Pepper at 0.09

(c) ANN Motion Detection + Filtering

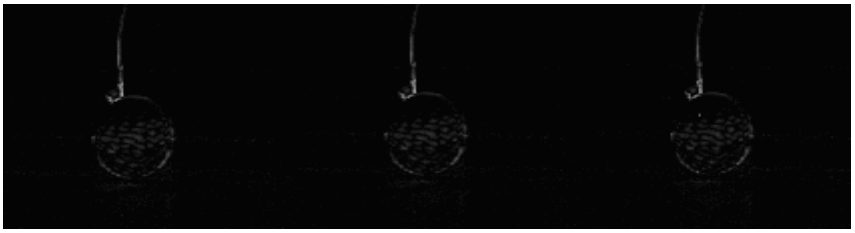
(d) Enhanced Images

Fig 5.4 Car Motion Detection using ANN Ensemble System

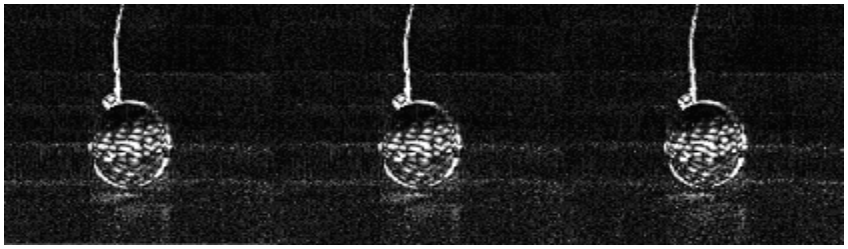


(a) Captured Image Sequences

(b) Induced Salt and Pepper Noise at 0.09

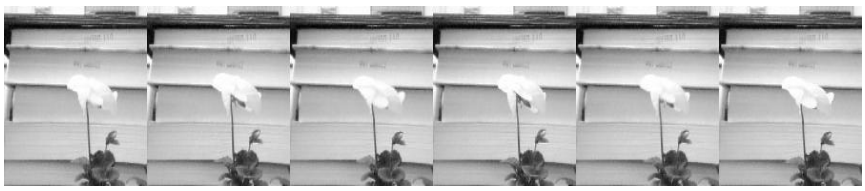


(c) ANN Motion Detection + Filtering



(d) Enhanced Images.

Fig 5.5: Ball Motion Detection using ANN Ensemble System



(a) Captured Image Sequences

(b) Induced Noise Salt and Pepper at 0.09

(c) ANN Motion Detection + Filtering

(d) Enhanced Images.

Fig 5.6: Plant Motion Detection using ANN Ensemble System

5.3.1 ANN+DSP Hybrid Motion Detection System

Following from the previous discussions, we have observed that DSP systems display a superior speed and ANN systems have excellent generalization and noise immunity characteristics. Given these two excellent characteristics of these two systems, it was decided that the *positive characteristics of each* should be *combined into a single system* that yields *excellent image quality* at reasonably *fast processing speeds*. A schematic of the hybrid DSP + ANN system is given in Fig 5.7.

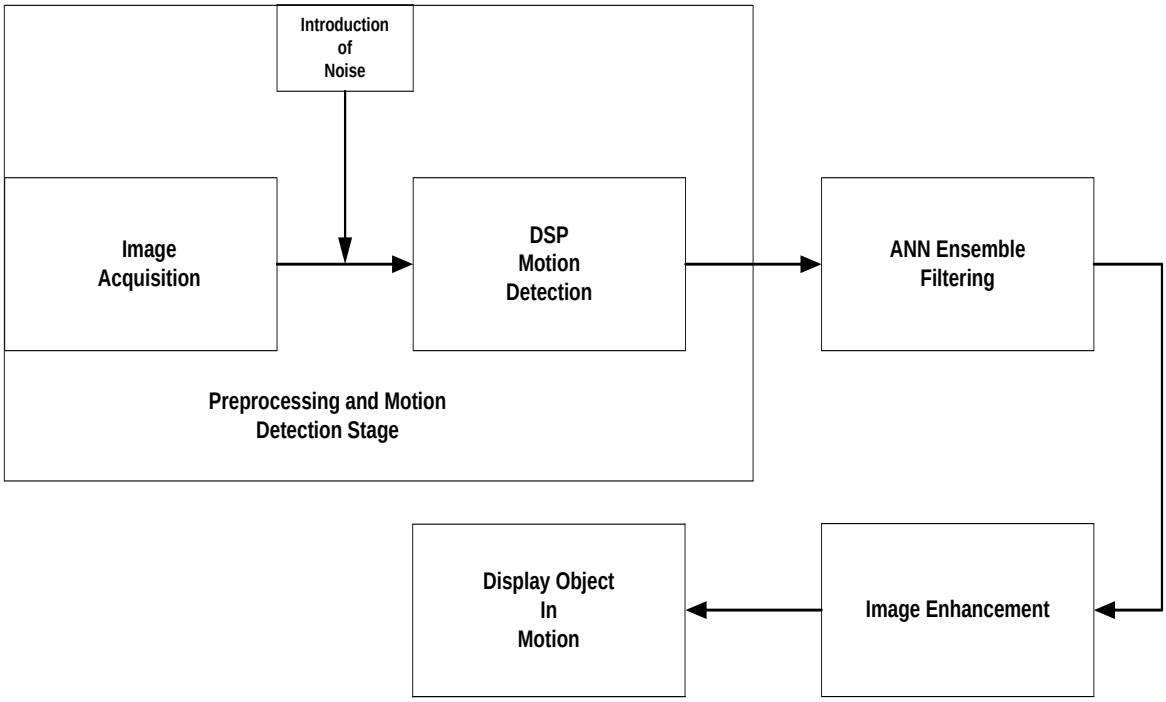
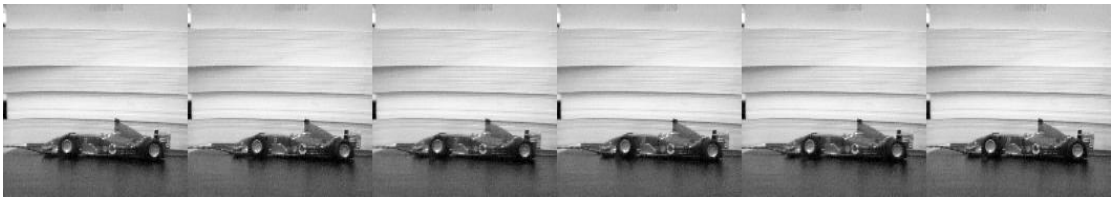


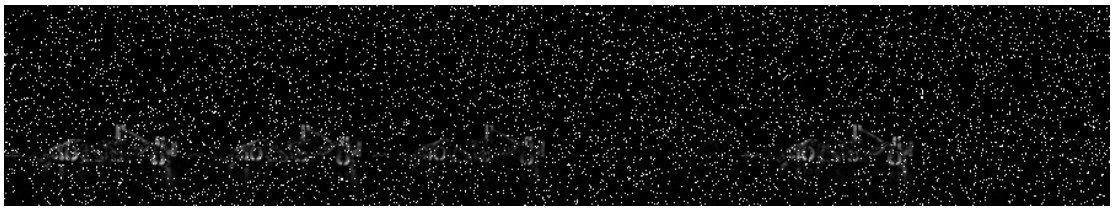
Fig. 5.7 Hybrid DSP + ANN ensemble motion detection system

The results of the tests of the hybrid system are shown in Fig 5.7, Fig 5.8 and Fig 5.9.



(a) Captured Image Sequences

(b) Induced Salt and Pepper Noise at 0.09

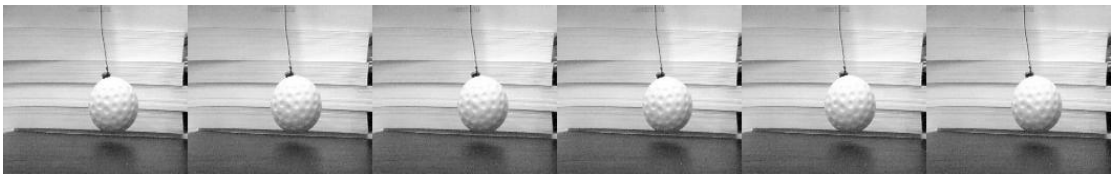


(c) DSP Motion Detection

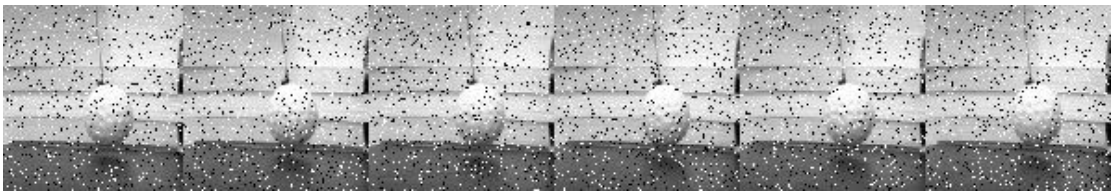
(d) ANN Filtering

(e) Enhanced Images

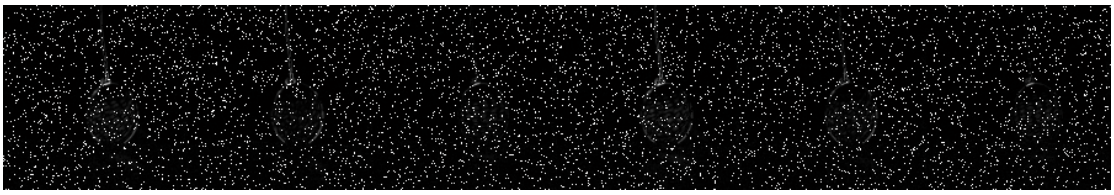
Fig. 5.8 Car motion detection using DSP + ANN system



(a) Captured Image Sequences

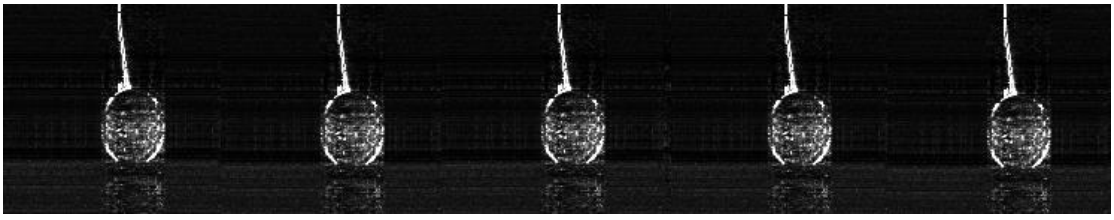


(b) Induced Salt and Pepper Noise at 0.09



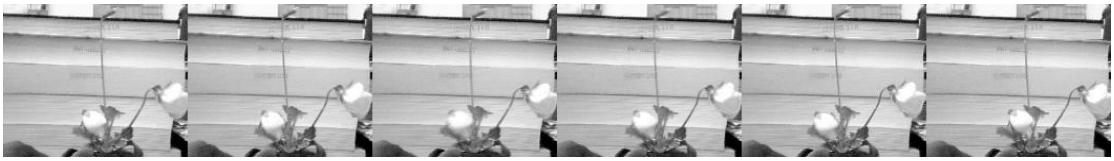
(c) DSP Motion Detection

(d) ANN Filtering



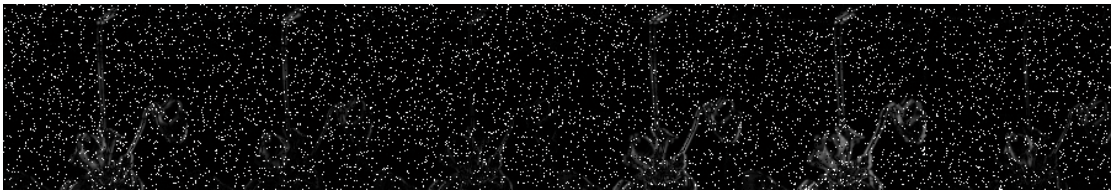
(e) Enhanced Images

Fig. 5.9 Ball motion detection using hybrid DSP + ANN system

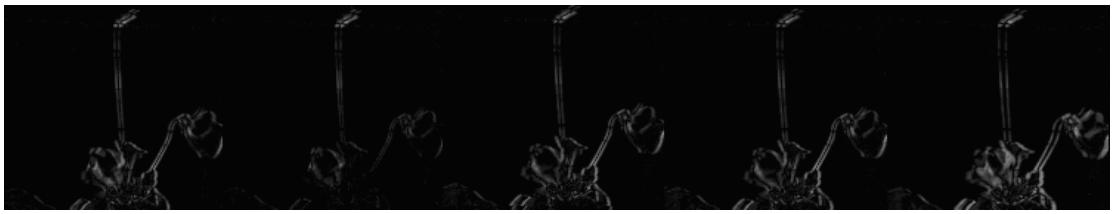


(a) Captured Image Sequences

(b) Induced Salt and Pepper Noise at 0.09



(c) DSP Motion Detection



(d) ANN Filtering

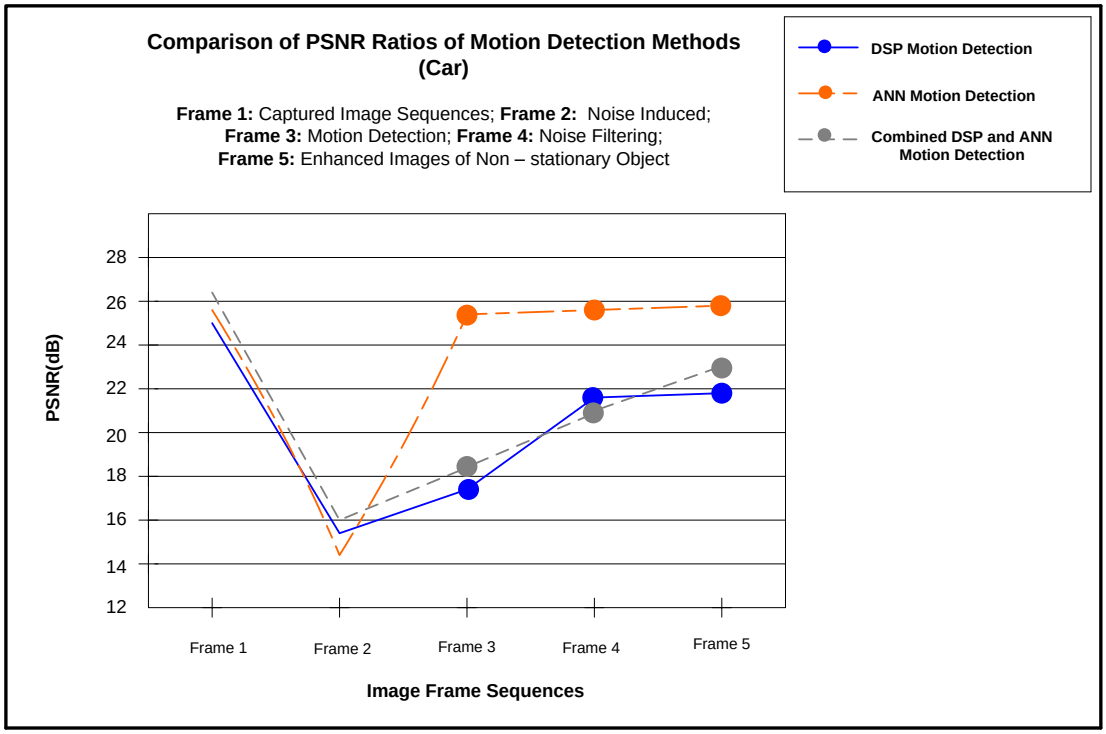
(e) Enhanced Images

Fig. 5.10 Plant motion detection using hybrid DSP + ANN system

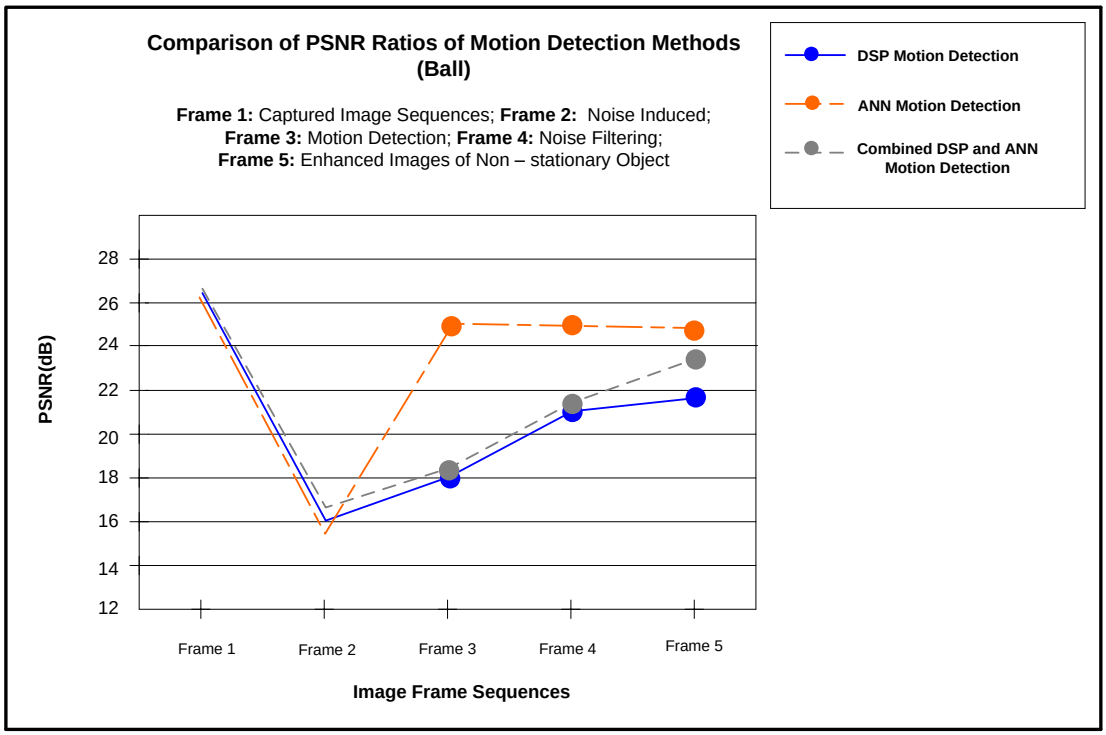
5.4 PSNR and Computational Time Performance of the DSP, ANN and Hybrid ANN + DSP Motion Detection Systems

5.4.1 Noise Responses for DSP Only, ANN Ensembles Only and DSP + ANN Ensemble Motion Detection Systems.

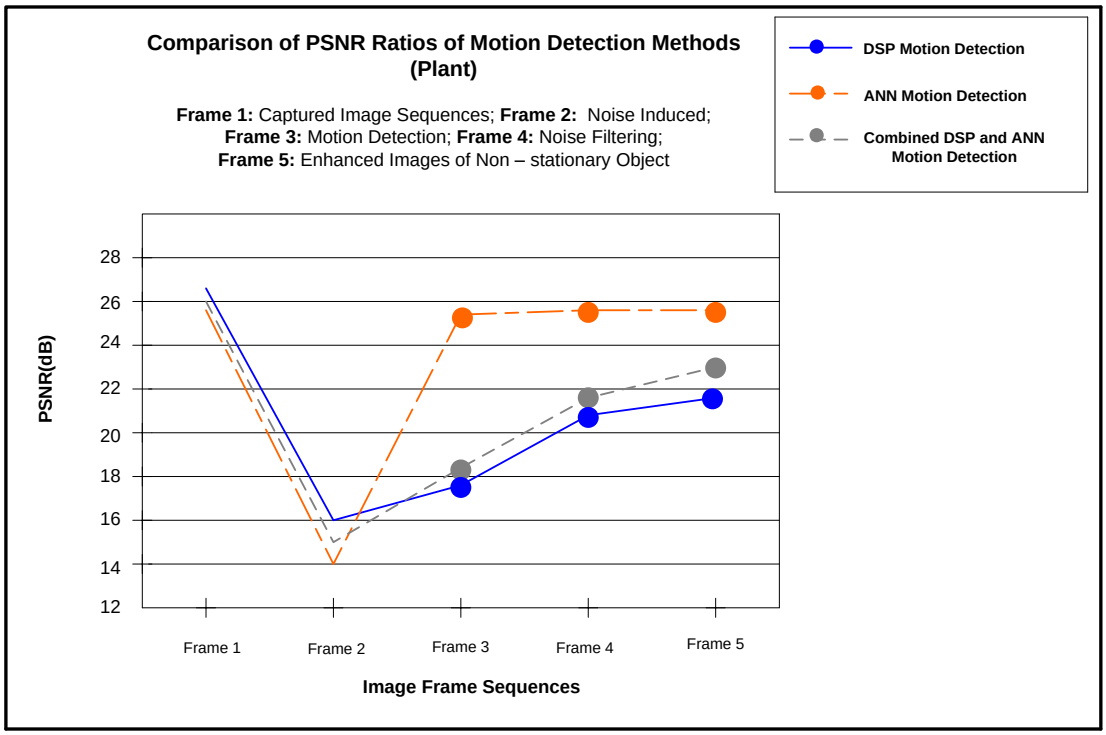
Graph 5.1 is the PSNR plot comparing the results of the tests conducted of the captured image sequences given in Fig. 5.8 – Fig 5.10. The PSNR shows that the ANN motion detection system has a superior noise advantage over the other motion detection systems. The picture quality of combined DSP and ANN Ensemble system is an improvement over the pure DSP system.



(a) PSNR of Hybrid Motion Detection System (Car)



(b) PSNR of Hybrid Motion Detection System (Ball)



(c) PSNR of hybrid motion detection system (Plant)

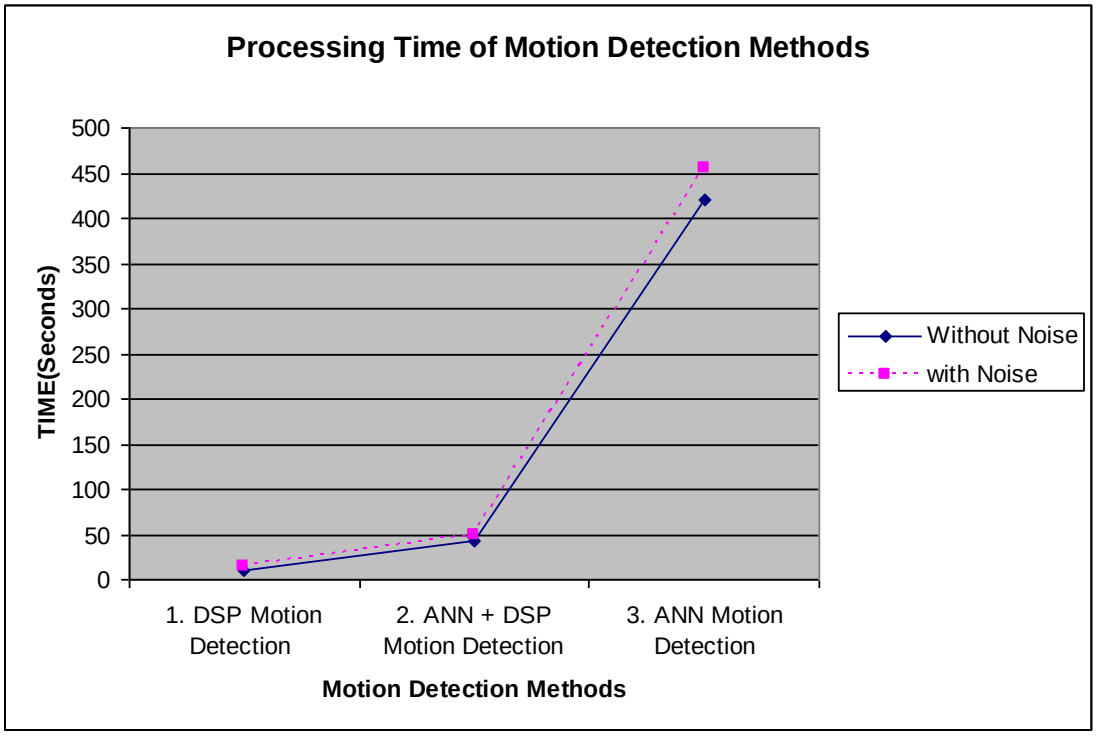
Graph 5.1 PSNR of hybrid motion detection system

5.4.2 Average Processing Time of the Hybrid Motion Detection System

The processing times of the motion detection systems are shown in Table 5.1 and Graph 5.2. A Pentium 4, 3.0 GHz, with 1.0 GB of RAM memory was used to conduct the experiments.

Computer System Pentium 4, 3.0Ghz, 2.0GB RAM	**DSP Motion Detection System**	**ANN Motion Detection System**	**ANN + DSP Motion Detection System**
(without noise)	11.75s	7 min	32.115secs
(with induced noise)	17.33s	7 min	34.795secs

TABLE 5.1: Average System processing time of Motion Detection Systems



Graph 5.2Processing Times of Hybrid System

5.5 Analysis of the Performance of the DSP, ANN and Hybrid ANN + DSP Motion Detection Systems

With regards to Table 5.1, Graph 5.1 and Graph 5.2:

5.5.1 System Noise Response

The end processing of the combined DSP + ANN Ensemble Systems displays a poor noise response than the ANN ensemble system, but an improvement when compared to the pure DSP system. This as a result of the inherent noise problems associated with DSP systems such as aliasing, additive noise and multiplicative noise from DSP systems. DSP systems also require complex digital filters to overcome these shortcomings (Moorgas and Govender, 2008). The improvement of the noise response can be attributed to the inherent noise immunity characteristics of the ANN ensemble system.

5.5.2 Processing Time

The combined system shows significant improvement in processing time. These results can be attributed to using the DSP system for preprocessing and motion detection. The DSP system for motion detection used image subtraction as segmentation and isolation of the moving object to eliminate all stationary data from image frames as was discussed in part 1 of chapter 3. The ANN ensemble is presented with image frames having a dark background in order to reduce the processing speed of the ANN ensemble.

5.6 Conclusion

Following from the previous discussions and observations of the test results, we conclude that there is a significant improvement in the PSNR ratio for the hybrid motion detection system, plus a significant reduction in processing time. The improved PSNR ratio can be attributed to the excellent noise immunity characteristics of the ANN system, and the faster processing speed is a direct result of the fast computations by the DSP system. The next chapter describes the testing of the proposed ANN system using image compression as image compression to improve processing speed.

Chapter 6

Image Compression as a Preprocessing Technique for Improving ANN system Processing Time

6.1. Introduction

The previous chapters have illustrated and discussed the performance of ANN motion detection systems, DSP motion systems and hybrid ANN + DSP motion detection systems. From the performance of these systems we observe the following:

- *ANN systems* display excellent PSNR ratios but are computationally burdensome and require long processor times.
- *DSP systems* are much faster than ANN. This speed comes at the expense of image quality.
- *Hybrid DSP + ANN systems* harnesses the positive aspects of DSP and ANN systems. The DSP system contributes towards reducing computation time whilst the noise immunity and generalization characteristics of the ANN enhance image quality by improving the PSNR ratio.

In order to maintain high quality images large quantities of image data had to be processed by each NN within the ensemble. The ANN ensemble consisted of two layers of NNs with each NN layer having 100 neurons and 200 neurons, respectively. In order to ensure a high resolution the input layer of each NN had to map the output layer to produce an image of 255 x 255 pixels points. The large NN architectures each contributed to the slow processing speed. Furthermore there was no preprocessing of the images presented to the system except for gray-scaling. The main focus of this chapter is to consider the application of image compression in our motion detection, with a view to improving the processing speed of the ANN system. A brief description of popular image compression techniques is also discussed.

6.2. Purpose of Image Compression

Image compression is used to reduce the dimensionality of image data by removing redundancies from image frames. One technique of removing data redundancies in an image involves the removal of high frequency data invisible to the human eye. Image compression methods can be either lossy or lossless. Lossless compression methods are applied to images where there is a demand for high resolution and high quality images such as in medical imagery scans, crime scene investigative images and scans for archival purposes. Lossy compression methods are applied especially to natural photographs, transmission of image advertisements and video movies.

6.3. Image Compression Techniques

Typical examples of image compression formats include JPEG and JPEG-2000 for still images, MPEG for moving images and AVI for both images and sound. Some of the methods of compression that these formats use are Huffman Coding, Symbol Based Coding, Predictive Coding (used primarily in moving images), WT and DCT. For this study DCT and wavelet image compression techniques were tested. DCT and wavelet image compression has been a popular compression technique used over the older compression techniques mentioned (Nadarajah and Kotz, 2007).

6.4. 2D-DCT for Image Compression

The 2D-DCT can be applied to both still image compression and moving image compression (Dai *et al.,* 2005). In transform coding the compression technique applied divides an image into small non-overlapping blocks of equal size N x N dimensions. The blocks are then processed independently using the 2D-DCT transform. The 2D-DCT of an image is a sum of sinusoids of varying magnitudes and frequencies. After computation of the 2D-DCT, the significant information of the image is represented by coefficients in the frequency domain. This process forms the heart of the JPEG lossy image compression algorithm (S. Nadarajah and S. Kotz, 2007). The 2D-IDCT then converts this information

back to the time domain. The 2D-DCT of an M x N image matrix $f[x,y]$ used in this study is defined in equation 6.1 (Gonzalez and Woods, 2008).

$$B(p,q) = \alpha_p \alpha_q \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x,y] \cos\frac{\pi(2x+1)p}{2M} \cos\frac{\pi(2y+1)q}{2N} \tag{6.1}$$

$$0 \le p \le M-1, \quad 0 \le q \le N-1$$

$B(p,q)$ are the 2D DCT coefficients of $f[x,y]$. The inverse 2D-IDCT for 6.1 is defined as:

$$f[x,y] = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q B(p,q) \cos\frac{\pi(2x+1)p}{2M} \cos\frac{\pi(2y+1)q}{2N} \tag{6.2}$$

$$0 \le x \le M-1, \quad 0 \le y \le N-1$$

where

$$\alpha_p = \begin{pmatrix} 1/\sqrt{M}, & p = 0 \\ \sqrt{2/M}, & 1 \le p \le M-1 \end{pmatrix} \qquad \alpha_q = \begin{pmatrix} 1/\sqrt{N}, & q = 0 \\ \sqrt{2/N}, & 1 \le q \le N-1 \end{pmatrix}$$

The inverse 2D DCT equation can be interpreted as meaning that any M x N matrix $f[x,y]$ can be written as a sum of MN functions of the form (6.3).

$$f[x,y] = \alpha_p \alpha_q \cos\frac{\pi(2x+1)p}{2M} \cos\frac{\pi(2y+1)q}{2N}, \tag{6.3}$$

$$0 \le p \le M-1, \quad 0 \le q \le N-1$$

These functions are called the basis functions of the 2D-DCT. The 2D-DCT coefficients B(p,q), can be regarded as the weights applied to each basis function.

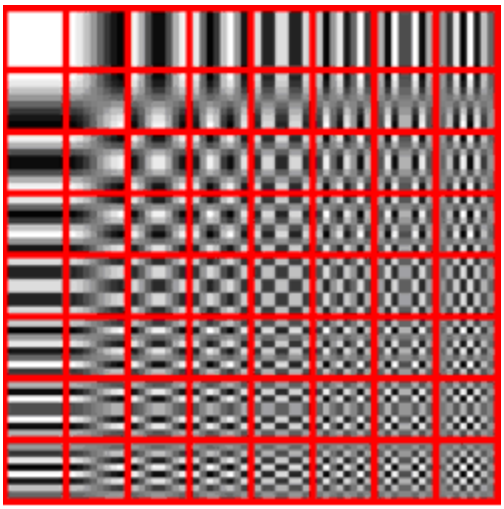


Fig. 6.1 2D-DCT Transform Coefficients in Image Scaling (8 x 8)

The image compression algorithm can be divided into (8 x 8) coefficient blocks as in Fig. 6.1 or into (16 x 16) or (64 x 64) coefficient blocks and the 2D-DCT is then computed for each block. The 2D-DCT coefficients are then quantized, coded, and applied. The decoder then applies the 2D-IDCT of each block, and puts the blocks together as a single frame to form an image. The coefficients close to zero are DC and are discarded without seriously affecting the quality of the reconstructed image frame.

6.5 Wavelet Transforms – Preliminary Discussion

From FT theory analysis a signal can be expressed as the sum of a series of sines and cosines. This sum is known as the Fourier expansion. The disadvantage of the FT is that it only caters for frequency resolution and not time resolution (Najarian and Splinter, 2006). This means that only the frequencies within a signal can be determined but not the time at which these frequencies were introduced to the signal. In order to represent a signal in the frequency domain and display the time associated with it, the signal of interest has to be analyzed in discrete parts. By analyzing a signal in this way more information about when and where different frequency components were introduced can be determined. The WT is used to overcome the shortcomings of the FT (Najarian and Splinter, 2006).

Wavelet analysis uses a fully scalable window to achieve signal cutting. The window is then shifted along the signal and for every position the frequency spectrum is evaluated. This process is continued for each cycle. The end result is a number of time–frequency representations of the signal with different resolutions shown in Fig 6.2. The WT is used extensively in image processing applications (for segmentation, morphological operations and image compression) and exhibits a superior performance over other methods (Buckheit and Donoho, 1995). For this study the WT is used as an image preprocessor prior to application into the ANN ensemble system with the objective of reducing the detection time of the ANN system. A comparison between DCT and WT will also be discussed in the following discussion.

6.5.1 Basic Theory of Wavelet Transforms

Fig. 6.2 shows a signal broken into its various frequency components using the FT whereas the WT shown in Fig. 6.3 scales and shifts the signal into packages of the original or mother wavelet.

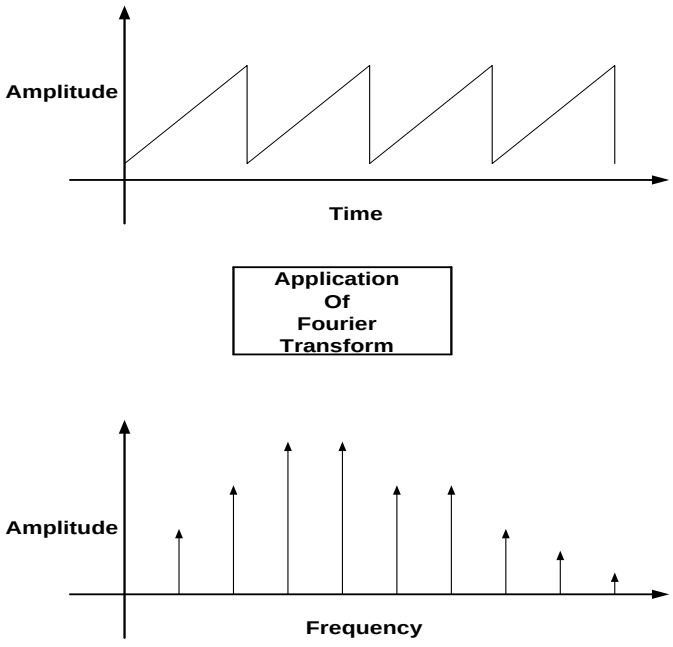


Fig. 6.2 Application of the Fourier Transform to a Continuous Time Signal

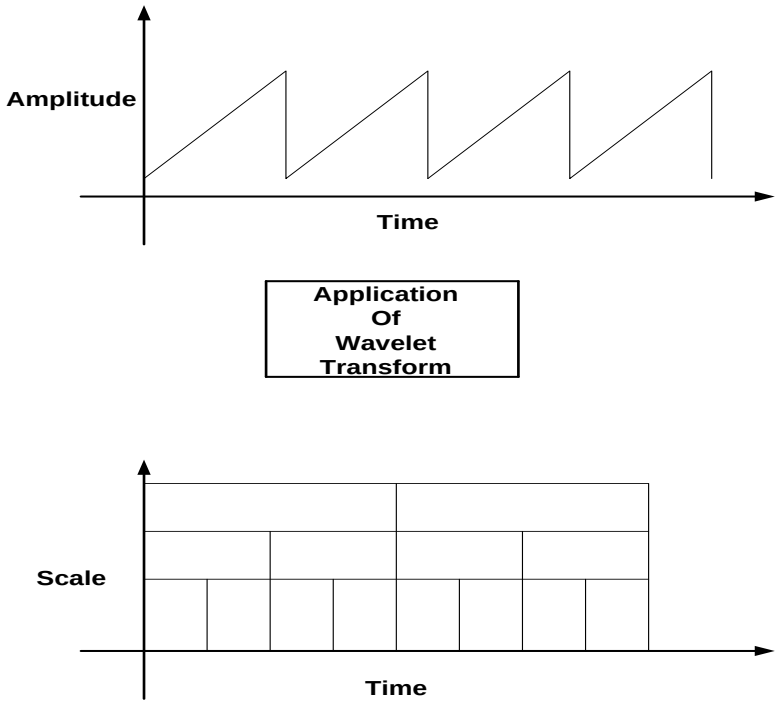


Fig 6.3 Application of the Wavelet Transform Scaling to a Continuous Time Signal

6.6 Advantages of Wavelets

Both wavelet compression and DCT compression can be applied to still JPEG images and moving MPEG images. Wavelets have an added advantage in that they can also be used for de-noising images, image decomposition and image reconstruction (Aboufadel and Schlicker, 1999). De-noising images is the removal of induced noise and resultant noise from pre-processing and compression functions (Zhang *et al*, 2007). This is an added advantage of wavelets over other image compression techniques and makes wavelets a popular choice for image compression applications. Another major advantage of wavelets is its ability to perform local analysis, that is to analyze a localized area of a larger signal and detect any discontinuities, trends and breakdown points in that signal which otherwise cannot be detected by other compression and reconstruction methods (Wang *et al.*, 2007). These advantages make using wavelets an attractive image pre-processing tool.

6.7 Wavelet Definitions

The continuous FT is defined by (6.4)

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \tag{6.4}$$

(6.4) produces the complex and the real components of signal $f(t)$. The results of (6.4) yield Fourier coefficients which when multiplied by sine frequencies produce the components of the original signal. Similarly, the CWT is defined as the sum over time of the signal multiplied by scaled, shifted versions of the wavelet function ψ: Equation (6.5) shows the CWT with the resultant coefficients C.

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\psi(scale, position, t)dt \tag{6.5}$$

By substituting variables for scale, position and time in (6.5), equation (6.6) is derived.

$$\gamma(s,\tau) = \int f(t)\psi^{*}{}_{s,\tau}(t)dt \tag{6.6}$$

Where $\psi*$ represents a complex conjugate and $f(t)$ is decomposed to a basis function. The variables 's'and 'τ' represent scaling and translation of $f(t)$. Wavelets are generated from a single basic wavelet called a "*mother wavelet*" by scaling and translation as shown in (6.7)

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{6.7}$$

In digital image processing, transforms have to be applied as discrete algorithms because of the digital nature of the image data. Like the FT and DCT, wavelets can be applied as

discrete transforms. Like the FT, WT maps a function of a continuous variable into a sequence of coefficients. The function expansion is a discrete number of coefficients of the WT, giving rise to the DWT which is expressed as (6.8) and (6.9) for function *f(x):*

$$W_{\varphi}(j,k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\varphi_{j},k(n) \tag{6.8}$$

$$W_{\psi}(j,k) = \frac{1}{\sqrt{M}} \sum_{x} f(x)\psi_{j},k(n) \tag{6.9}$$

Consider (6.8) and (6.9): $\varphi_{j},k(n)$ and $\psi_{j},k(n)$ are sampled versions of the basic function $\varphi_{j},k(x)$ and $\psi_{j},k(x)$ of *f(x).* Also n = 0, 1, 2, …,M – 1, where M is equally spaced samples of the basis function. The complementary 1D-WT is given by equation (6.10) and is used in reconstruction of images after decomposition and image compression functions.

$$f(x) = \frac{1}{\sqrt{M}} \sum_{k} W_{\varphi}(j,k)\varphi_{j},k(n) + \frac{1}{\sqrt{M}} \sum_{j}^{\infty} \sum_{k} W_{\psi}(j,k)\psi_{j},k(n) \tag{6.10}$$

For the 1D-WT n = 0, 1, 2, …, M – 1, j = 0, 1, 2, …, J – 1, and k = 0, 1, 2, …, $2^{j} - 1$. Images are 2D data fields as described in the previous sections. For this study 2D-DWT were applied for image decomposition and compression. The 2D-DWT is defined by (6.11) and (6.12) for any image frame $f[x, y]$ of size M x N.

$$W_{\varphi}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x,y]\varphi_{j},m,n(x,y) \tag{6.11}$$

$$W_{\psi}^{i}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f[x,y]\psi_{j}^{i},m,n(x,y) \tag{6.12}$$

The 2D-IDWT is given by (6.13)

$$f[x,y]=\frac{1}{\sqrt{MN}}\sum_{m}\sum_{n}W_{\varphi}(j,m,n)\varphi_{j},m,n(x,y)+\frac{1}{\sqrt{MN}}\sum_{i}\sum_{j}^{\infty}\sum_{m}\sum_{n}W_{\psi}^{i}(j,m,n)\psi_{j}^{i},m,n(x,y) \tag{6.13}$$

With regards to (6.11), (6.12) and (6.13): $W_{\varphi}(j,m,n)$ are coefficients defined as approximations of $f[x,y]$ at scale j. The $W_{\psi}^{i}(j,m,n)$ coefficients add horizontal, vertical and diagonal details for scales j. $N=M=2^{j}$ so that j = 0, 1, 2, …, j – 1 and m = n = 0, 1, 2, …, 2^{j} – 1.

6.8 Common Examples of Wavelet Functions

Fig 6.4 illustrates the common base functions of the WT namely Haar, Morlet, mexican hat, and Meyer wavelets. There are many WT families which incorporate slight variations in functions for different applications. These include the Haar, Daubechies, biorthogonal, coifets, symlets, complex Gaussian, and reverse biorthogonal (Goswami and Chan, 1999).

6.8.1 The Haar Wavelet

The Haar wavelet shown in Fig 6.4 is one of the simplest of wavelets and resembles a step function that can be applied to most applications (Elfouly *et al*, 2008; Talukder and Harada, 2007). A 1D dimensional image can be treated as a sequence of coefficients. Thus from the 1D-DWT, the 2D-DWT equations in (6.11) and (6.12) can be obtained. A vector space V^{j} or image matrix $f^{n}[x,y]$ will contain 2^{j} pixels in V^{j} with j = 0, 1, 2 ….*n-1*. The basis functions in the vector space are called scaling functions and are denoted by the symbol ϕ. A basis function for V^{j} is given by the set of scaled functions in (6.14).

$$\phi_{i}^{j}(x):=\phi(2^{j}x-i) \quad and\ i=0,1,2.....2^{j}-1 \tag{6.14}$$

where

$$\phi(x) = \begin{Bmatrix} 1 & for\ 0 \le x < 1 \\ 0 & otherwise \end{Bmatrix}$$

The corresponding Haar wavelets for the basis function is given by (6.15).

$$\varphi_i^j(x) : \varphi(2^j x - i) \quad and \quad i = 0, 1, 2, ... 2^j - 1 \tag{6.15}$$

where

$$\varphi(x) = \begin{Bmatrix} 1 & for\ 0 \le x < 0.5 \\ -1 & for\ 0.5 \le x < 1 \\ 0 & otherwise \end{Bmatrix}$$

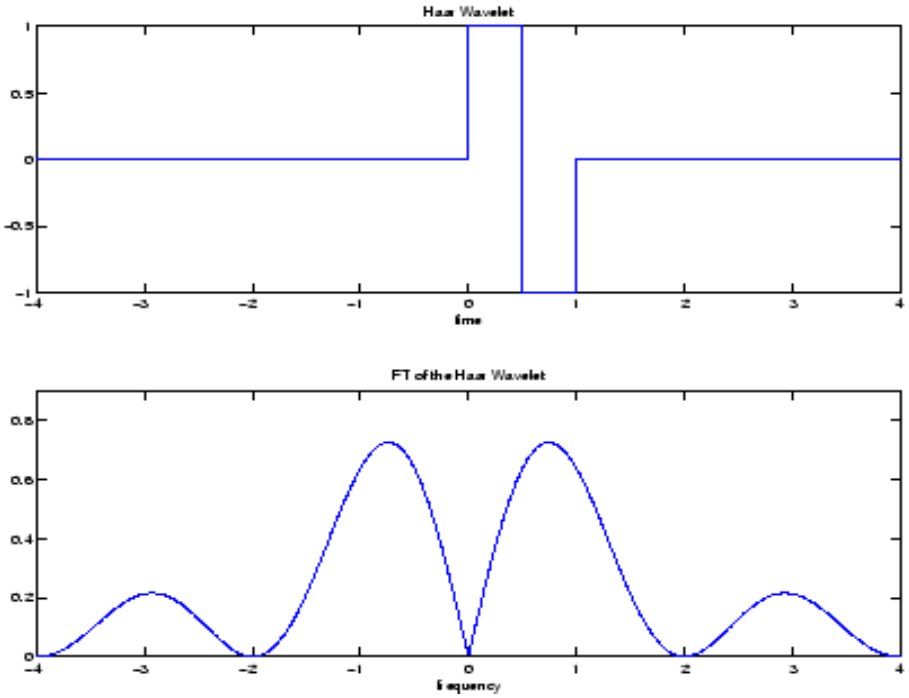


Fig 6.4: Haar Wavelet Transform Function and Frequency Spectrum

The DWT for an image $f^n[x, y]$ as a 2D signal will be obtained from a 1D-DWT. The scaling function is obtained by multiplying two 1D functions $\phi(x, y) = \phi(x)\phi(y)$. For the 2D case, there are three wavelet functions that scan details in $f^n[x, y]$ in *horizontal*

(6.16), *vertical* (6.17) and *diagonal* (6.18) directions respectively (Talukder and Harada, 2007):

$$f_{\phi}^{1}[x, y] = \phi(x)\varphi(y) \tag{6.16}$$

$$f_{\phi}^{2}[x, y] = \varphi(x)\phi(y) \tag{6.17}$$

$$f_{\phi}^{3}[x, y] = \varphi(x)\varphi(y) \tag{6.18}$$

With regards to (6.16) – (6.18): f_{ϕ}^{n} is the compressed image and *n* = the directional processing of each image frame. This is fed through a 2D-IDWT reconstruction filter bank (6.11) and (6.12) with each filter being either HPF or LPF for separating horizontal, vertical and diagonal components. The transformation of the 2D image $f^{n}[x, y]$ is a 2D generalization of the original image, with an average value of pixels in each row having detail coefficients. Each row is then treated as an image matrix, with the result of a lower number of averaged coefficients depending on the complexity of each row. In order to complete the process the transformation is repeated recursively. Operations of averaging and differencing are performed to arrive at a new matrix representing the same image in a more concise compressed manner (Talukder and Harada, 2007).

6.8.2 The Daubechies Wavelet

The Daubechies wavelet transforms (see Fig 6.5 and Fig 6.6) are defined in the same way as the Haar wavelet transform by computing the running averages and differences via scalar products with scaling signals and wavelets (Khare and Tiwary, 2007). The only difference between them consists of how the scaling signals and wavelets are defined. The Daubechies family of wavelets are ortho-normal wavelets and makes discrete wavelet analysis practicable. There are nine basic functions in this family of wavelets and are written as dbN, where N is the order of the wavelet within the family (Wang *et a.l*, 1997). Daubechies wavelets are more complicated than the Haar wavelet and are continuous. They are more computationally intensive to use than the Haar wavelet but

produces smoother end results because of the continuous nature (Wang *et al*, 1997 and Elfouly *et al*, 2008).

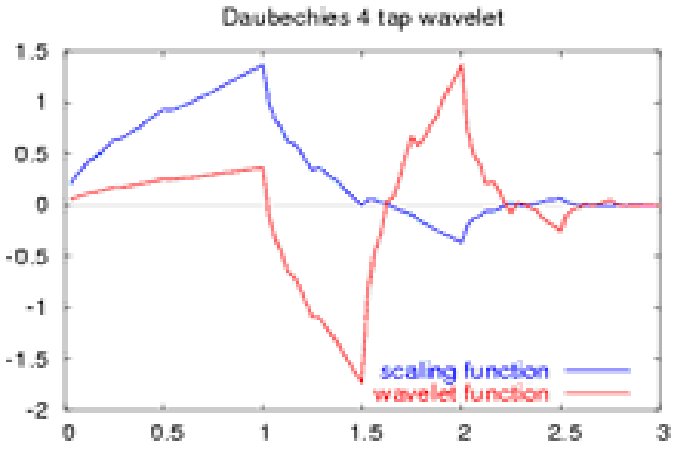


Fig. 6.5 Daubechies Wavelet (db4)

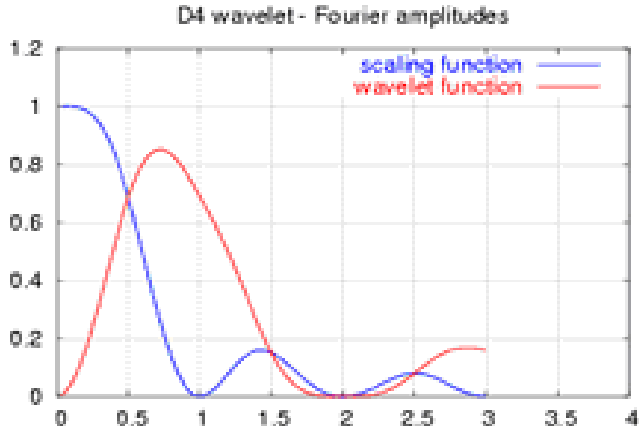


Fig. 6.6 Frequency Spectrum of Daubechies Wavelet(db4)

(6.19) and (6.20) are scaling functions applied in vector space V^j.

$$\phi(x) = 2^{j/2}\phi(2^j x - k) \tag{6.19}$$

$$\varphi(y) = 2^{j/2}\varphi(2^j y - k) \tag{6.20}$$

where k = 0, 1, 2,…….. k *-1*. The scaling function is obtained by multiplying $\phi(x,y) = \phi(x)\phi(y)$ as in Haar wavelet compression and can be translated

to $f_{\phi}^{n}[x,y] = \varphi(x)\phi(y)$ for any image frame. The remaining end processing compression algorithms are the same as that for the Haar described in section 6.8.1.

6.8.3 Biorthogonal Wavelets

This family of wavelets exhibit properties of linear phase which is needed for signal and image reconstruction and analysis. There are twenty eight basic wavelets in this family for image decomposition and reconstruction. They are excellent for image coding and de-noising. Many elegant quantization and entropy coding techniques have been developed that exploit the multi-resolution nature of this wavelet family to enhance the low bit rate coding of images (Winger and Venetsanopoulos, 2001). The properties of the biorthogonal wavelets have duality in any subspace because of its symmetry and separate decomposition and reconstruction functions in each member of its family of wavelets (Pragada and Jayanthi, 2008). Once again the scaling functions are obtained by multiplication for a vector space V^{j} which can be translated to an image matrix $f^{n}[x,y]$. The coefficients of (6.21) and (6.22) can be obtained by the inner product of the appropriate dual function by applying (6.23) and (6.24).

$$\phi(x) = \sqrt{2}\sum_{k} h_k \phi(2x-k) \text{ and } \psi(x) = \sqrt{2}\sum_{k} g_k \phi(2x-k) \tag{6.21}$$

$$\hat{\phi}(x) = \sqrt{2}\sum_{k} \hat{h}_k \hat{\phi}(2x-k) \text{ and } \hat{\psi}(x) = \sqrt{2}\sum_{k} \hat{g}_k \hat{\phi}(2x-k) \tag{6.22}$$

$$h_k = (\hat{\phi}(x), \phi(x)) = \sqrt{2}\int_{-\infty}^{\infty} \hat{\phi}(2(x-k)\phi(x)dx \text{ and } g_k = (\hat{\phi}(x)\psi(x) = \sqrt{2}\int_{-\infty}^{\infty} \hat{\phi}(2x-k)\psi(x)dx$$

(6.23)

$$\hat{h}_k = (\phi(x), \hat{\phi}(x)) = \sqrt{2}\int_{-\infty}^{\infty} \phi(2(x-k)\,\hat{\phi}(x)\,dx \text{ and } \hat{g}_k = (\phi(x), \hat{\psi}(x) = \sqrt{2}\int_{-\infty}^{\infty} \phi(2x-k)\hat{\psi}(x)dx$$

(6.24)

With regards to (6.21) – (6.24):

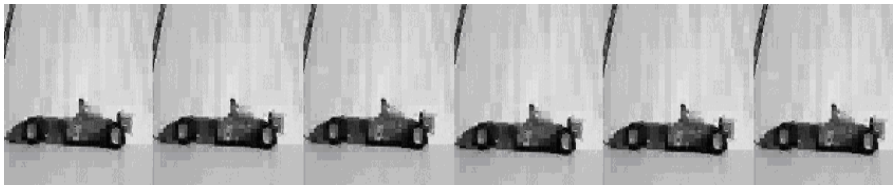
h_k and $\hat{h}_k$ are the coefficients of the subspace V^j; by applying (6.23) and (6.24) again coefficients g_k and $\hat{g}_k$ for subspace W^j are obtained; the dual multi-resolution with dual subspaces are V^j and W^j and can be defined for decomposition and reconstruction of the image frames $f^n[x, y]$.

6.9 Image Compression Preprocessing for ANN Motion Detection System

Chapter 5 discussed three motion detection methods using DSP, ANN ensembles and a hybrid DSP+ANN ensemble system. From the Table 5.1 and Graph 5.2, we observe that the ANN ensemble is slower than the pure DSP system. When only DSP is used the processing speed is superior to that of the ANN ensemble system but image quality is compromised (Moorgas and Govender, 2008).

To improve the processing speed of the hybrid system, we conducted tests to compare the performance of our hybrid system using DCT, and the Haar, Daubechies (db4) and biorthogonal (bior 1.3) wavelets for image compression. The conditions under which these tests were conducted are the same as was mentioned for all the other tests in the previous discussions. The results of the tests are given in Fig 6.7 – Fig 6.10 for the car, Fig 6.11 – Fig 6.14 for the ball and Fig 6.15 – Fig 6.19 for the plant.

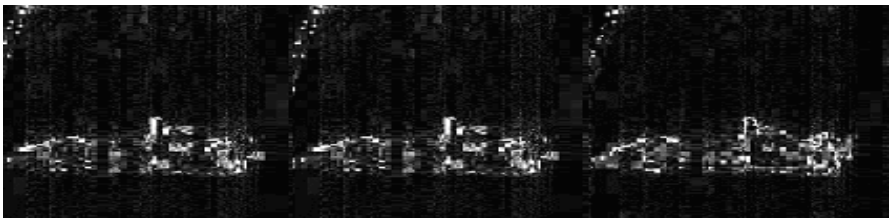
(a) Captured Image Sequences



(b)DCT Compressed Images

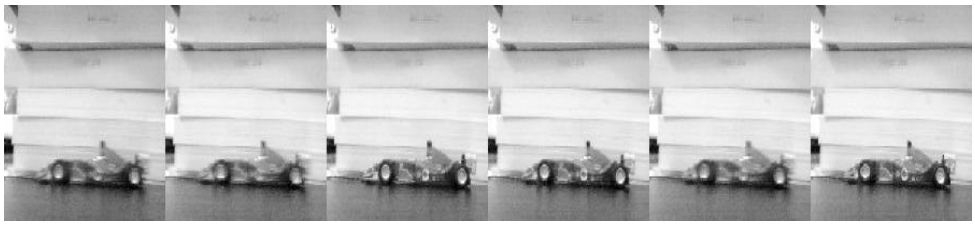
(c) Noise Induced at 0.09 salt & pepper

(d) ANN Motion Detection

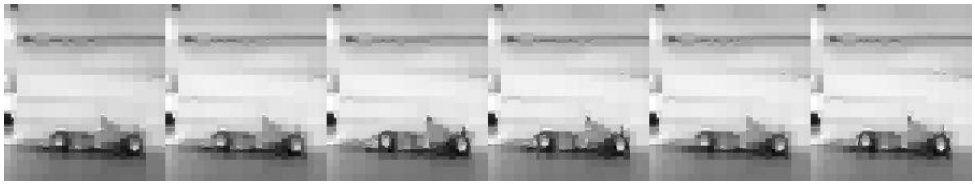


(e) Enhanced Images

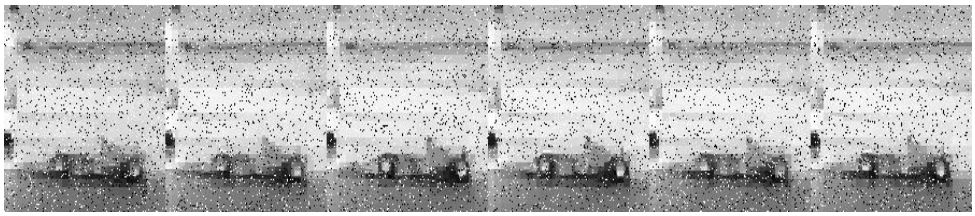
Fig. 6.7 DCT Image Compression Preprocessing



(a) Captured Image Sequences

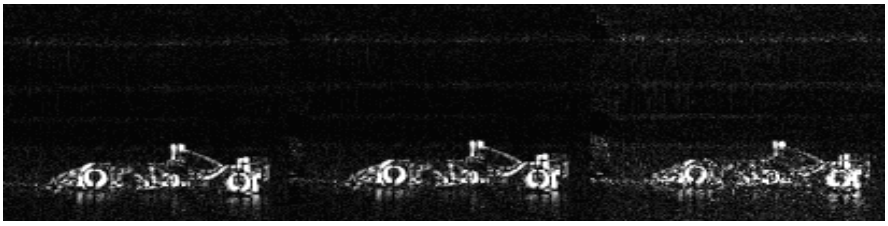


(b) Haar Wavelet Compressed Images



(c) Noise Induced at 0.09 Salt & Pepper

(d) ANN Motion Detection



(e) Enhanced Images

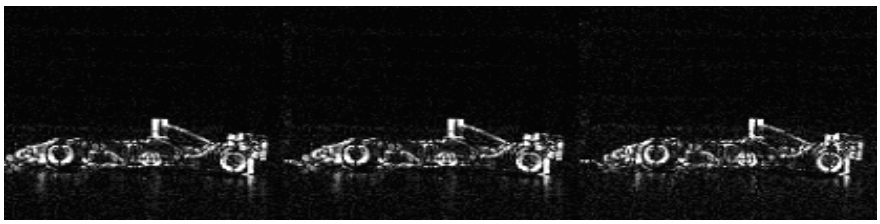
Fig. 6.8 Haar Wavelet Compression Preprocessing

(a) Captured Image Sequences

(b) Daubechies (db4) Wavelet Compressed Images

(c) Noise Induced at 0.09 Salt & Pepper

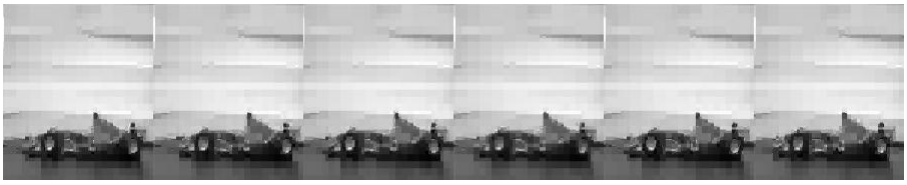
(d) ANN Motion Detection



(e) Enhanced Images

Fig. 6.9 Daubechies (db4) Wavelet Compression Preprocessing

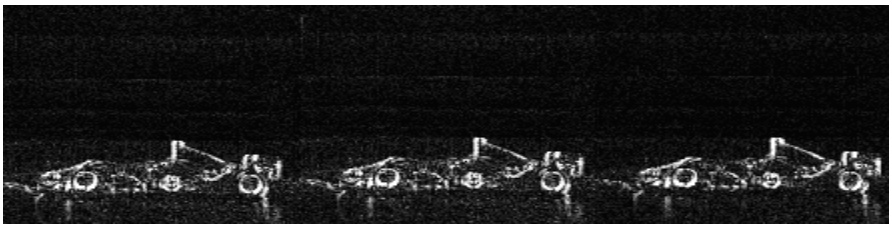
(a) Captured Image Sequences



(b) Biorthogonal (bior 1.3) Wavelet Compressed Images

(c) Noise Induced at 0.09 Salt & Pepper

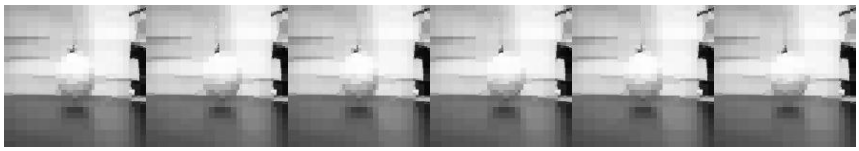
(d) ANN Motion Detection



(e)Enhanced Images

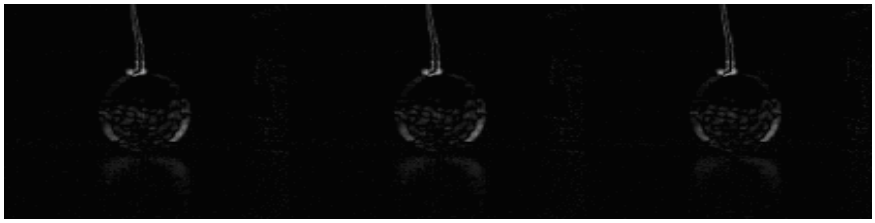
Fig. 6.10Biorthogonal (bior 1.3) Wavelet Compression Preprocessing

(a) Captured Image Sequences

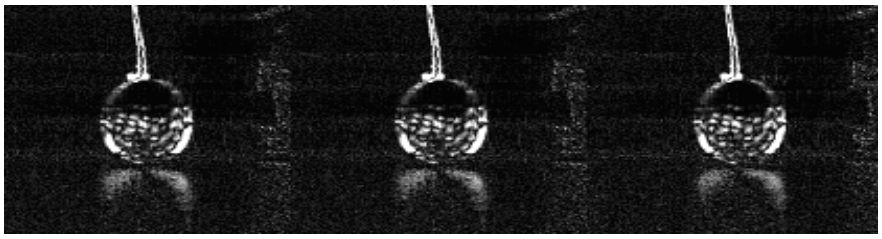


(b) DCT Compressed Images

(c) Noise Induced at 0.09 salt & pepper

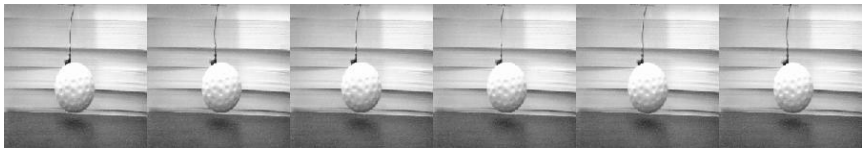


(d) ANN Motion Detection

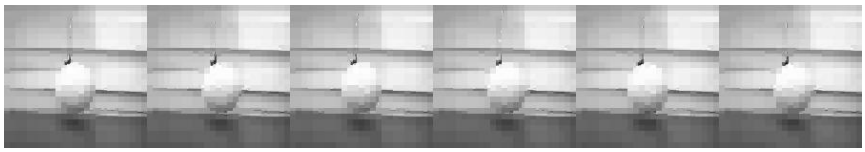


(e) Enhanced Images

Fig. 6.11DCT Compression Preprocessing

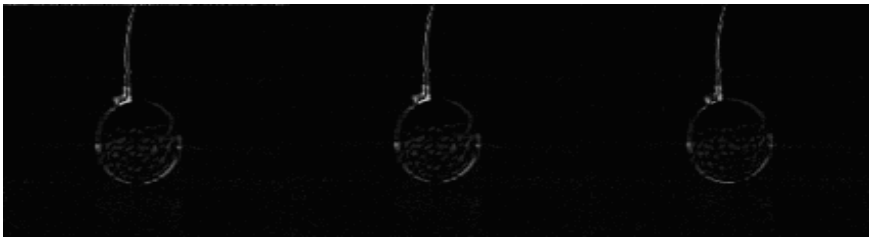


(a) Captured Image Sequences

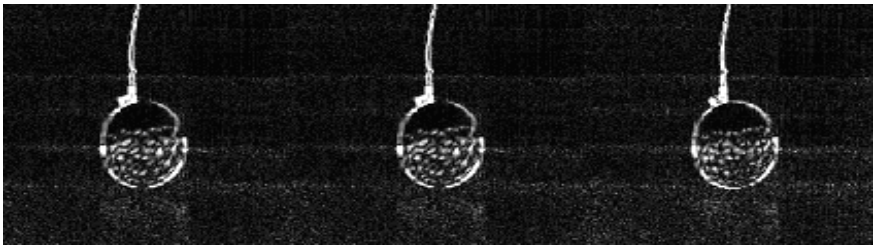


(b) Haar Wavelet Compressed Images

(c) Noise Induced at 0.09 Salt & Pepper

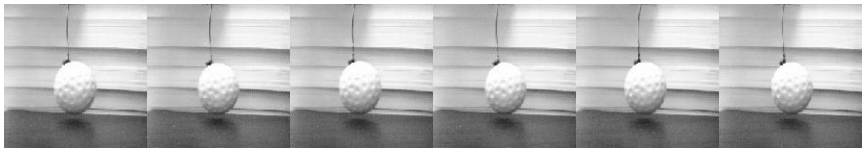


(d) ANN Motion Detection

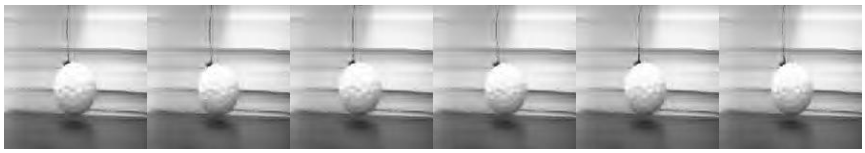


(e) Enhanced Images

Fig. 6.12 Haar Wavelet Compression Preprocessing



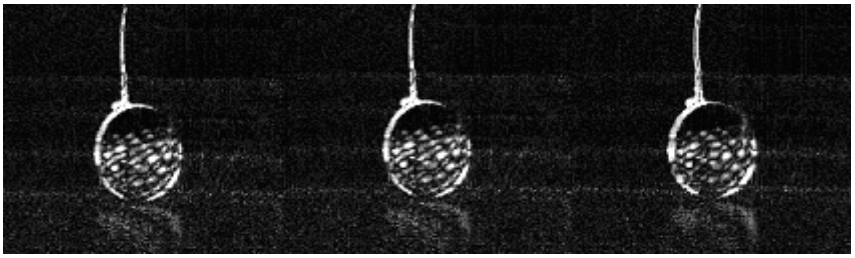
(a) Captured Image Sequences



(b) Daubechies (db4) Wavelet Compressed Images

(c) Noise Induced at 0.09 Salt & Pepper

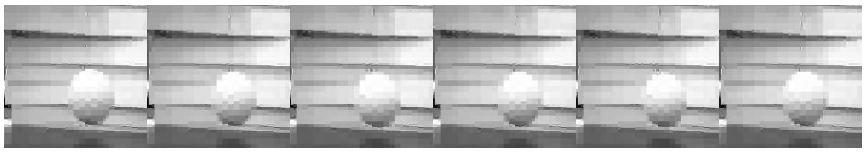
(d) ANN Motion Detection



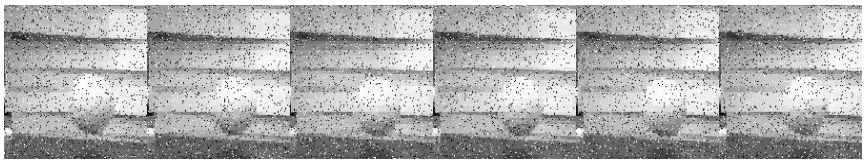
(e) Enhanced Images

Fig. 6.13 Daubechies (db4) Wavelet Compression Preprocessing

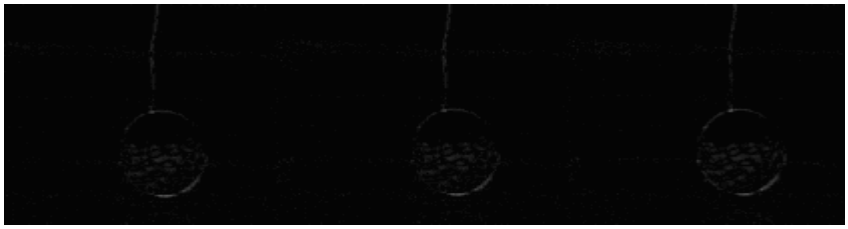
(a) Captured Image Sequences



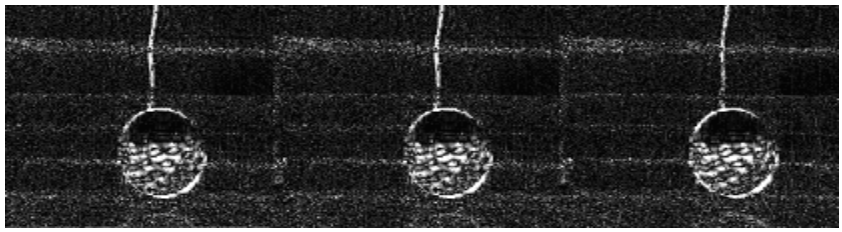
(b) Biorthogonal (bior 1.3) Wavelet Compressed Images



(c) Noise Induced at 0.09 Salt & Pepper



(d) ANN Motion Detection



(e) Enhanced Images

Fig. 6.14 Biorthogonal (bior 1.3) Wavelet Compression Preprocessing

(a) Captured Image Sequences

(b) DCT Compressed Images

(c) Noise Induced at 0.09 salt & pepper

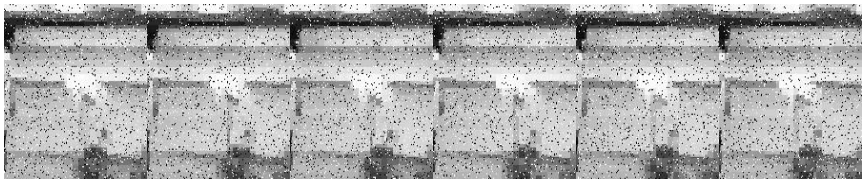
(d) ANN Motion Detection

(e) Enhanced Images

Fig. 6.15 DCT Compression Preprocessing

(a) Captured Image Sequences

(b) Haar Wavelet Compressed Images



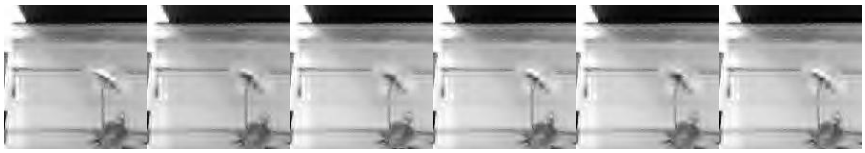
(c) Noise Induced at 0.09 Salt & Pepper

(d) ANN Motion Detection

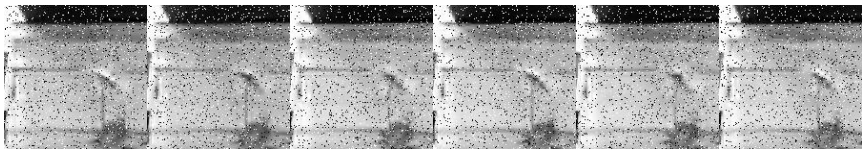
(e) Enhanced Images

Fig. 6.16 Haar Wavelet Compression Preprocessing

(a) Captured Image Sequences

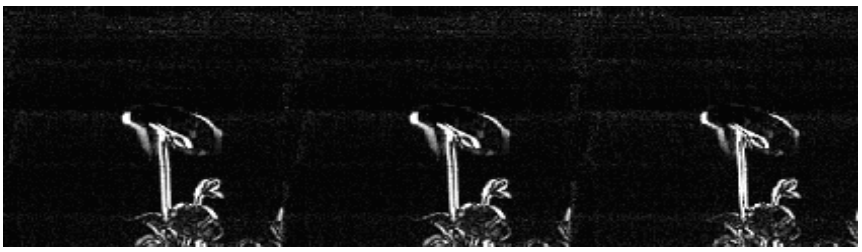


(b) Daubechies (db4) Wavelet Compressed Images



(c) Noise Induced at 0.09 Salt & Pepper

(d) ANN Motion Detection

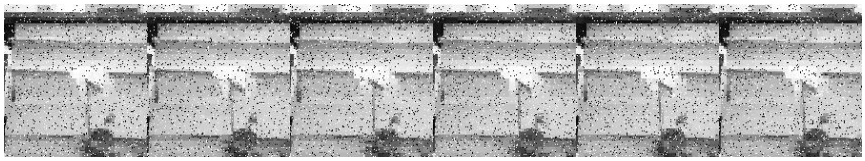


(e) Enhanced Images

Fig 6.17 Daubechies (db4) Wavelet Compression Preprocessing

(a) Captured Image Sequences

(b) Biorthogonal (bior 1.3) Wavelet Compressed Images



(c) Noise Induced at 0.09 Salt & Pepper

(d) ANN Motion Detection

(e) Enhanced Images

Fig 6.18 Biorthogonal (bior 1.3) Wavelet Compression Preprocessing

6.10 Noise Responses for ANN Motion Detection System Compressed Image

The noise responses from the image frame sequence tests are shown in Graph 6.1. For Graph 6.1(a), Fig 6.1(b) and Fig 6.1(c), the corresponding respective image frame sequence samples are listed in Table 6.1, Table 6.2, Table 6.3, Table 6.4 and Table 6.5.

Figure Number	Compression Type	Objects
Fig 6.7 (a)	DCT	Car
Fig 6.8 (a)	Haar	Car
Fig 6.9 (a)	Db4	Car
Fig 6.10 (a)	Bior (1.3)	Car
Fig 6.11 (a)	DCT	Ball
Fig 6.12 (a)	Haar	Ball
Fig 6.13 (a)	Db4	Ball
Fig 6.14 (a)	Bior (1.3)	Ball
Fig 6.15 (a)	DCT	Plant
Fig 6.16 (a)	Haar	Plant
Fig 6.17 (a)	Db4	Plant
Fig 6.18 (a)	Bior (1.3)	Plant

Table 6.1 Images compressed in 1st image frame sequence

Figure Number	Compression Type	Objects
Fig 6.7 (b)	DCT	Car
Fig 6.8 (b)	Haar	Car
Fig 6.9 (b)	Db4	Car
Fig 6.10 (b)	Bior (1.3)	Car
Fig 6.11 (b)	DCT	Ball
Fig 6.12 (b)	Haar	Ball
Fig 6.13 (b)	Db4	Ball
Fig 6.14 (b)	Bior (1.3)	Ball
Fig 6.15 (b)	DCT	Plant
Fig 6.16 (b)	Haar	Plant
Fig 6.17 (b)	Db4	Plant
Fig 6.18 (b)	Bior (1.3)	Plant

Table 6.2 Images compressed in 2nd image frame sequences

Figure Number	Compression Type	Objects
Fig 6.7 (c)	DCT	Car
Fig 6.8 (c)	Haar	Car
Fig 6.9 (c)	Db4	Car
Fig 6.10 (c)	Bior (1.3)	Car
Fig 6.11 (c)	DCT	Ball
Fig 6.12 (c)	Haar	Ball
Fig 6.13 (c)	Db4	Ball
Fig 6.14 (c)	Bior (1.3)	Ball
Fig 6.15 (c)	DCT	Plant
Fig 6.16 (c)	Haar	Plant
Fig 6.17 (c)	Db4	Plant
Fig 6.18 (c)	Bior (1.3)	Plant

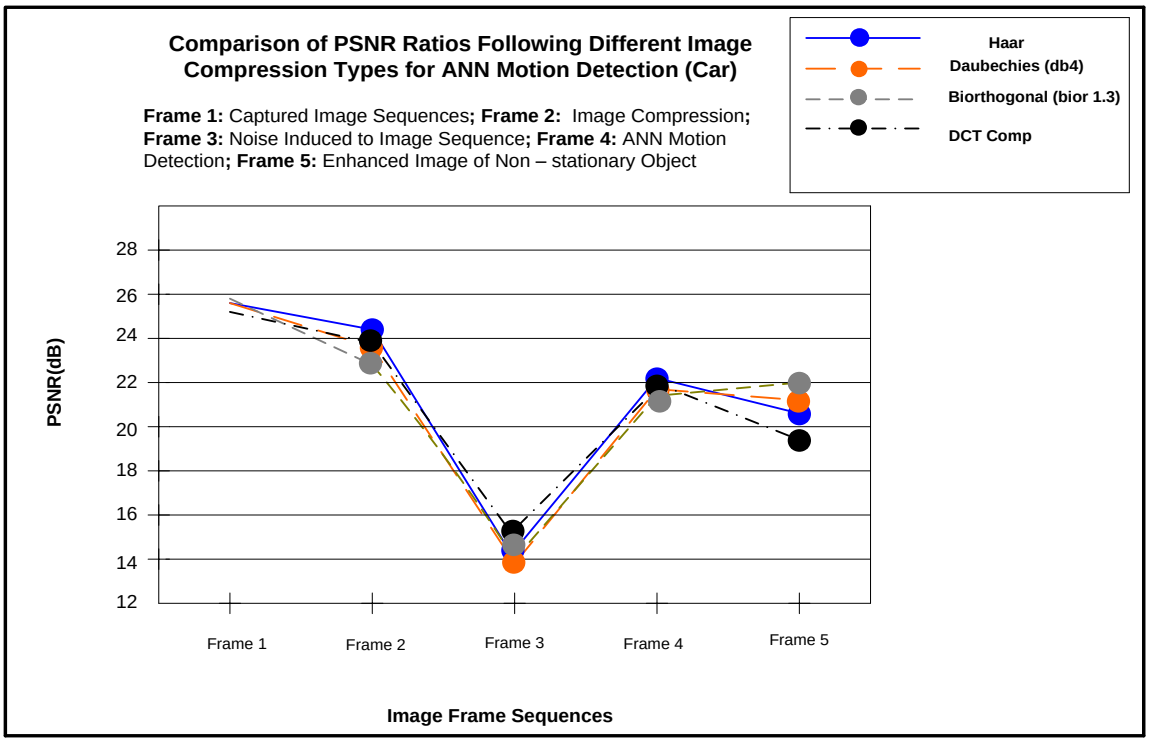
Table 6.3 Noise Induced at 0.09 Salt & Pepper in 3rd Image Frame Sequences

Figure Number	Compression Type	Object
Fig 6.7 (d)	DCT	Car
Fig 6.8 (d)	Haar	Car
Fig 6.9 (d)	Db4	Car
Fig 6.10 (d)	Bior (1.3)	Car
Fig 6.11 (d)	DCT	Ball
Fig 6.12 (d)	Haar	Ball
Fig 6.13 (d)	Db4	Ball
Fig 6.14 (d)	Bior (1.3)	Ball
Fig 6.15 (d)	DCT	Plant
Fig 6.16 (d)	Haar	Plant
Fig 6.17 (d)	Db4	Plant
Fig 6.18 (d)	Bior (1.3)	Plant

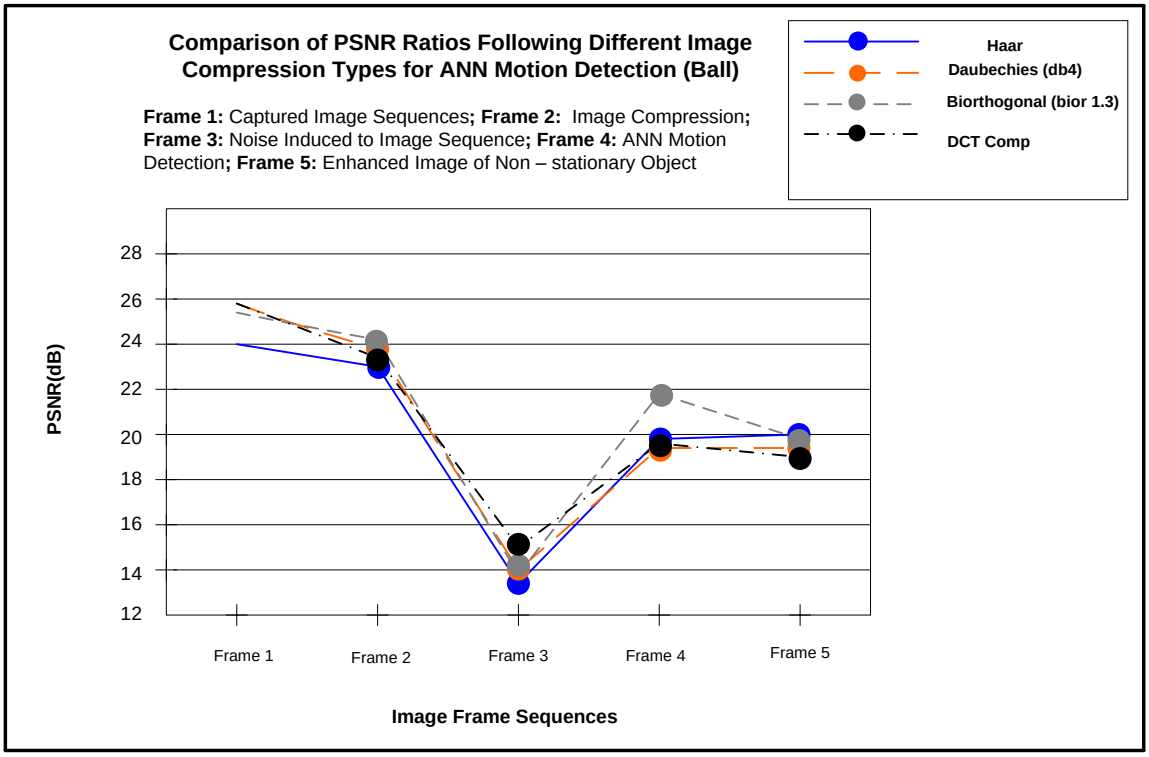
Table 6.4 ANN Motion Detection and Filtering in 4th Image Frame Sequences

Figure Number	Compression Type	Object
Fig 6.7 (e)	DCT	Car
Fig 6.8 (e)	Haar	Car
Fig 6.9 (e)	Db4	Car
Fig 6.10 (e)	Bior (1.3)	Car
Fig 6.11 (e)	DCT	Ball
Fig 6.12 (e)	Haar	Ball
Fig 6.13 (e)	Db4	Ball
Fig 6.14 (e)	Bior (1.3)	Ball
Fig 6.15 (e)	DCT	Plant
Fig 6.16 (e)	Haar	Plant
Fig 6.17 (e)	Db4	Plant
Fig 6.18 (e)	Bior (1.3)	Plant

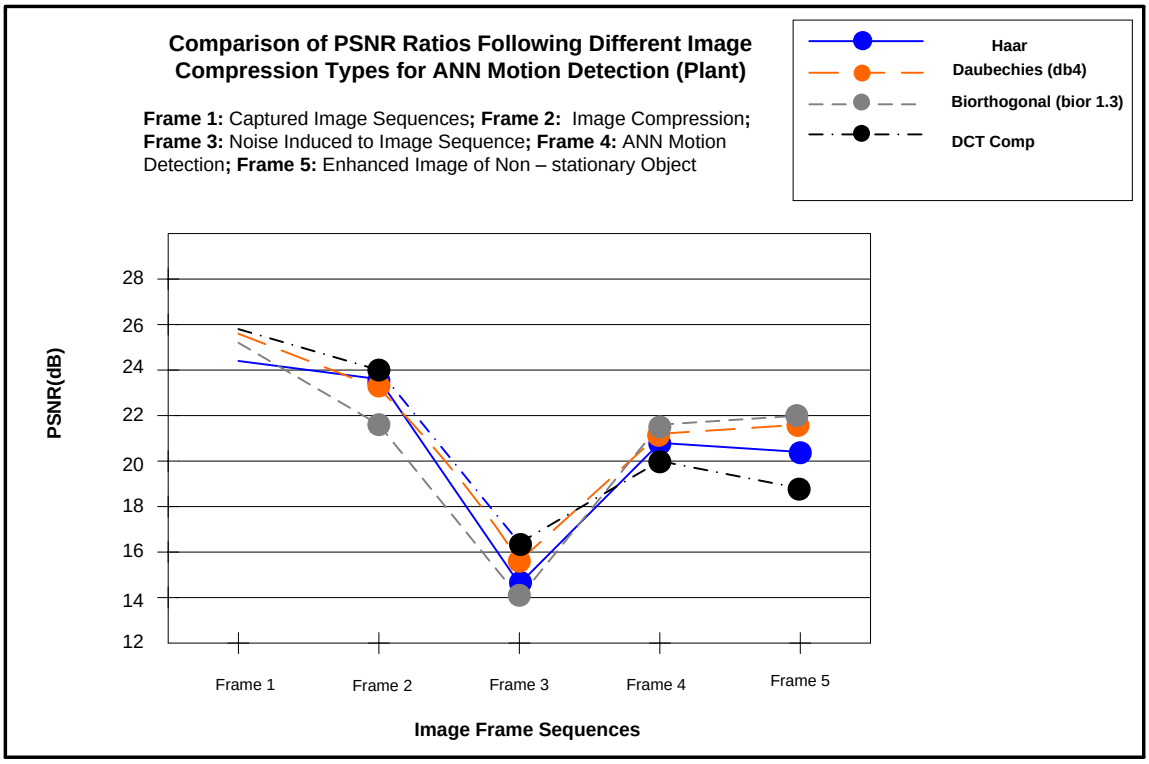
Table 6.5 Image is Enhanced in 5^{th} Image Frame Sequences



(a) PSNR of Motion Detection with Image Compression Preprocessing (Car)



(b) PSNR of Motion Detection with Image Compression Preprocessing (Ball)



(c) PSNR of Motion Detection with Image Compression Preprocessing (Plant)

Graph 6.1 PNSR for ANN motion detection system following image compression

6.11 Average Processing Time of the ANN Motion Detection System using Image Compression as Pre-processing

CAR Motion	No Preprocessing	Preprocessing with DCT Compression	Preprocessing with Haar Wavelet	Preprocessing with Daubechies Wavelet	Preprocessing with Biorthogonal Wavelet
Without noise	7 min	4 min	4.536 min	4.72min	4.5min
With induced noise	7 min	4.75 min	4.138 min	4.912min	4.865min

Table: 6.6 Average Processing Time of ANN Motion Detection System (Using P4: 3GHz, 2GB RAM) (Car motion)

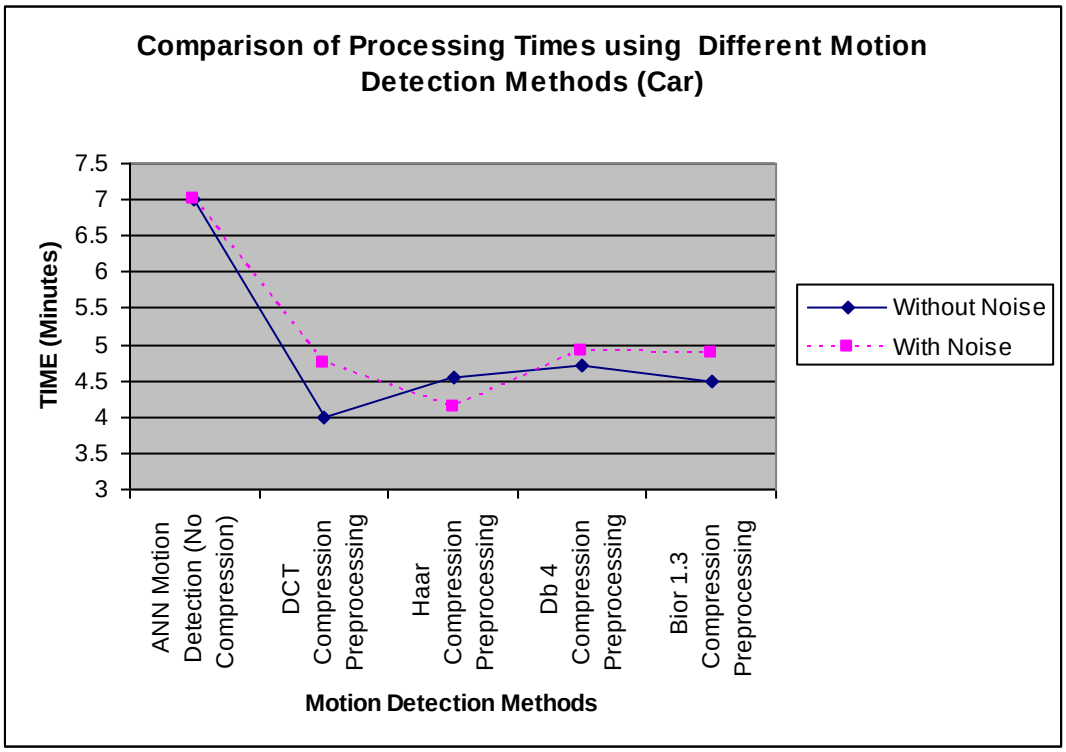
Ball Motion	**No Preprocessing**	**Preprocessing with DCT Compression**	**Preprocessing with Haar Wavelet**	**Preprocessing with Daubechies Wavelet**	**Preprocessing with Biorthogonal Wavelet**
Without noise	10.85min	9.05 min	5.15 min	5min	5.6min
With induced noise	10.12 min	8.66 min	4.90 min	.66min	6.05min

Table: 6.7 Average Processing Time of ANN Motion Detection System (Using P4: 3GHz, 2GB RAM) (Ball motion)

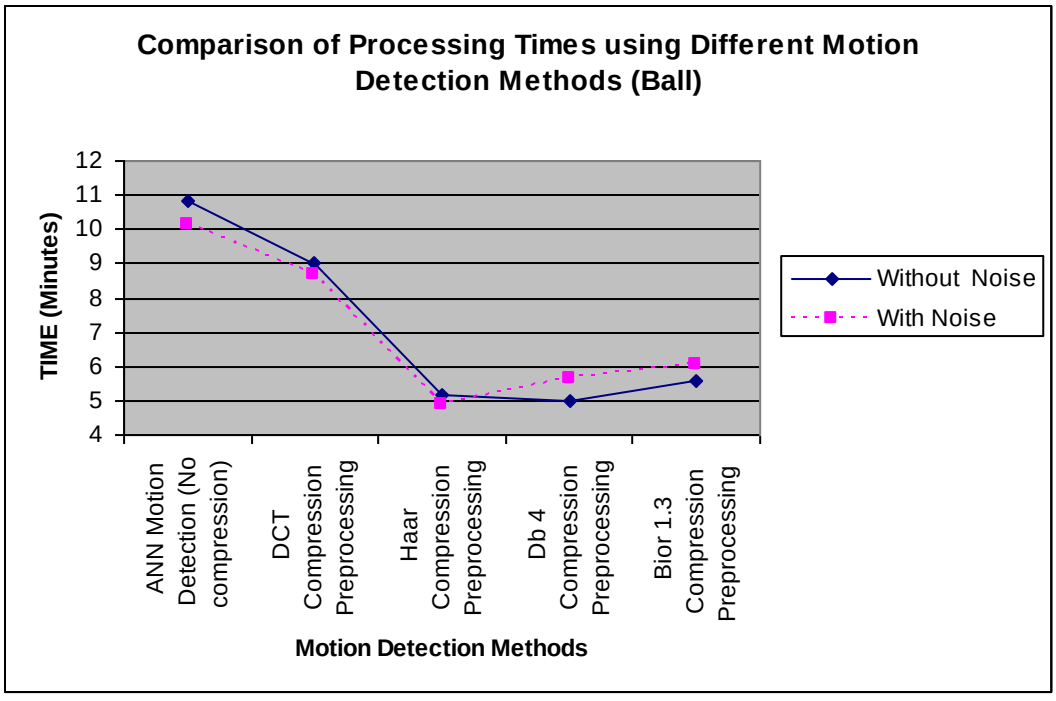
Plant Motion	**No Preprocessing**	**Preprocessing with DCT Compression**	**Preprocessing with Haar Wavelet**	**Preprocessing with Daubechies Wavelet**	**Preprocessing with Biorthogonal Wavelet**
Without noise	14.98min	9. min	5.7 min	6.4min	6.15min
With induced noise	14 min	8.05 min	5.97 min	8.5min	7min

Table: 6.8 Average Processing Time of ANN Motion Detection System (Using P4:3GHz, 2GB RAM) (Plant motion)

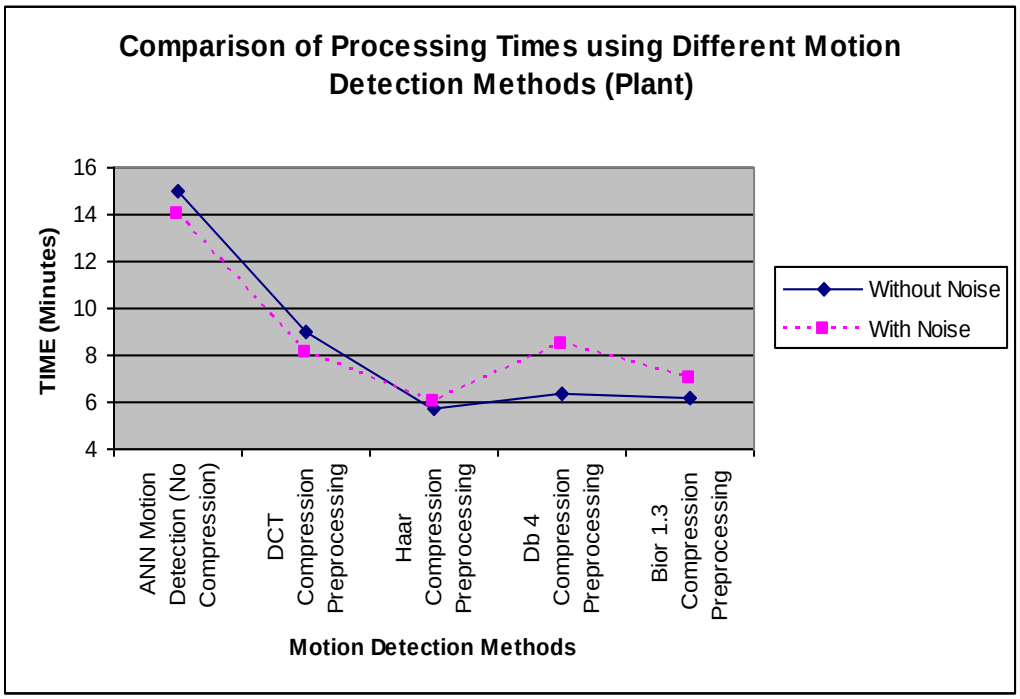
6.12 Average Processing Time Graphs of the ANN Motion Detection System using Image Compression as Pre-processing



(a) Average Processing Time of ANN Motion Detection System (Car)



(b) Average Processing Time of ANN Motion Detection System (Ball)



(c) Average Processing Time of ANN Motion Detection System (Plant)

Graph 6.2 Average Preprocessing Times of ANN Motion Detection System with Image Compression as Preprocessing

6.13 Image Quality with Image Compression as Preprocessing

With regards to the PSNR responses in Graph 6.1, the biorthogonal (1.3) method of image compression together with the inherent noise immunity ANN's leads to a superior performance when compared to the Haar wavelet and the Daubechies (db4) method. This family of wavelets exhibit linear phase properties which is required for image reconstruction. There are two wavelets used for each category, one for decomposition and the other for reconstruction of images making them suitable for image compression (Misiti *et al*, 2000). This verifies the results of Pragada and Sivaswamy (2008). The Haar wavelet is the simplest WT of all the wavelet families and resembles a step function. It can be interpreted a discrete function and is not continuous over large intervals (Raviraj and Sanavullah, 2007).

Modifications to the Haar and Daubechies methods have been proposed (Khare and Tiwary, 2007; Nielsen *et al.*, 2006; Raviraj and Sanavullah, 2007). These modifications improve the performance of the standard Haar and Daubechies image compression

algorithms (Gransermann and Miikkulainen, 2005; Raviraj and Sanavullah, 2007). For this study only the standard Haar and Daubechies compression algorithms were used to improve the speed and quality of the image frames during motion detection, extraction and filtering.

The DCT was one of the first image compression standards for images, and is recently beginning to be overtaken by the WT as a preferred method of image compression (Menegaz, 2006). For this study only the standard DCT was used. In recent research modifications to the DCT algorithm have been proposed and competes with the flexibility and superior image quality of wavelet compression (Nadarajah and Kotz, 2007; Dai *et al.*, 2005).

6.14 Comments on the Processing Speed of the ANN System for Compressed Images

With regards to Table 6.1, Table 6.2, Table 6.3, Graph 6.2 (a), Graph 6.2 (b) and Graph 6.2 (c) the average speed of the Haar and Daubechies methods is marginally better than the biorthogonal transform. This can be attributed to the fact that the biorthogonal WT has two wavelets in each category of its family. This adds to the complexity of the biorthogonal algorithm when compared to the Haar and Daubechies WT. The Haar and the Daubechies only have a single wavelet in each category. This makes the compression algorithm simpler and much faster to process. The Haar algorithm has the fastest average speed of the three because it is a step function and uses a simple compression algorithm (Raviraj and Sanavullah, 2007; Elfouly *et a.l*, 2008; Talukder and Harada, 2007).

6.15 Summary and Conclusion

This chapter has compared the performance of the ANN motion detection system for different methods of image compression. The comparison was based on the PSNR ratio and processing speeds of the system. From the results presented in this chapter it can be clearly observed that including image compression reduces the processing speed of the ensemble based ANN motion detection system (see Table 6.1, Table 6.2 and Table 6.3). Although the image quality was lower as compared to the results produced by the pure

ANN system (see Chapter 4), the image quality of the ANN based motion detection system using image compression outperforms that of the pure DSP based motion detection system. From the PSNR analysis in Graph 6.1, it is evident that using the biorthogonal wavelet algorithm with the ANN motion detection system produces the best performance when compared to the other wavelets used in this study.

Chapter 7

Summary of Study, Strengths and Conclusions

7.1 Introduction and summary of results

This study has proposed an ANN ensemble based motion detection system. The system utilizes dedicated ensembles for processing individual image sequences, and exploits the inherent noise immunity, generalization and adaptability of ANN systems. The ANN motion detection system was also compared to a conventional motion detection system using DSP image processing. Following the tests and the results of these tests, the following can be concluded:

- Pure DSP systems are faster than ANN based systems but exhibit poor PSNR ratios
- ANN systems are slower than DSP but produce an image of superior quality as is evident from the excellent PSNR figures
- The hybrid ANN + DSP system exploits the excellent speed characteristics of DSP, and the excellent noise immunity, generalization and adaptability characteristics of ANN systems
- Image compression reduces the ANN processing time by approximately 50% without sacrificing image quality

7.2 Shortcomings of DSP Systems

Some short-comings of DSP systems include sampling errors, aliasing and the use of complex filters.

Sampling Errors and Aliasing

Sampling of an analog signal occurs at twice the signal's highest frequency. Sampling produces quantization errors which occurs when the sampled signal's amplitude is adjusted to its nearest available digital representation. Increasing the sampling rate reduces the quantization errors. Quantization errors lead to over-sampling of the analogue signal. Oversampling results in distortion of the original signal during signal reconstruction.

Under-sampling of the original signal will cause overlapping between the new signals and the original signal, leading to aliasing. Complex and expensive anti-aliasing filters are used to overcome aliasing in DSP systems (Mitra, 1988). Variations from the ideal design of the filter may have significant effects on DSP system performance. For example, the shortcomings in the design of the ant-aliasing filter means that the system will spend more time compensating for the filter and leads to poor phase response, resulting in harmonic distortion (Stewart, 1991; Thompson, 1989).

Output Filtering of DSP Systems

The output from the DA converter in a DSP system is a stepped waveform. This 'stepped effect' is a distortion of the original signal and must be removed during signal reconstruction. A low-pass filter or reconstruction filter is used to remove this 'stepped effect'. The design of this filter is also critical in producing an efficient DSP system (Mitra,1998).

DSP systems are manufactured with discrete AD and DA converters, anti-aliasing filters and programmable smoothing filters. This makes DSP systems expensive and algorithmically intensive when compared to ANN systems. ANN systems on the other hand possess an inherent ability to reject noise, and have the ability to adapt to changes in the input data, thereby reducing computational and design burdens.

7.3 Significance of using ANN System's

ANN systems are fundamentally simple filters that possess excellent noise immunity characteristics which ensures excellent PSNR figure over that of DSP. In this study the

comparison between DSP and ANN system performance for motion detection is based on the PSNR ratio and computational speed.

PSNR Ratio

This study has exploited the excellent noise immunity, generalization ability and adaptability characteristics of ANN's for the design of a motion detection system. Dedicated networks have been designed and trained to behave as 'experts' or ensembles for processing single frames of a complex image. From the results of the study (see Graph 4.1 (c), Graph 5.1(a) and Graph 6.1 (c) repeated here for convenience) it is clearly evident that the ANN ensemble based motion detection system produces superior image quality over conventional DSP motion detection systems.

Computation Speed

DSP methods have superior processing speeds but do not perform well in the face of noise (Moorgas and Govender, 2008). Configuring the ANN into an ensemble reduces the ANN's processing time without compromising image quality. The computation speed of the DSP system *versus* that of the ANN motion detection system is given in Table 5.1, Table 6.3, Graph 5.2 and Graph 6.2 (repeated here for convenience). Combining DSP and ANN system's into a hybrid system exploited the positive aspects of DSP, namely its speed, and the filtering properties of ANN's for enhanced image quality.

7.4 Conclusion

This study has demonstrated that using ANN's in ensemble configuration enhances the computation speed of ANN based systems. Ensembles utilize a strategy which involves the creation of many individual decision making neural network sub-systems which combine their outputs to enhance performance in terms of error reduction, improved resolution of results, and faster processing speeds when compared to standard ANN systems.

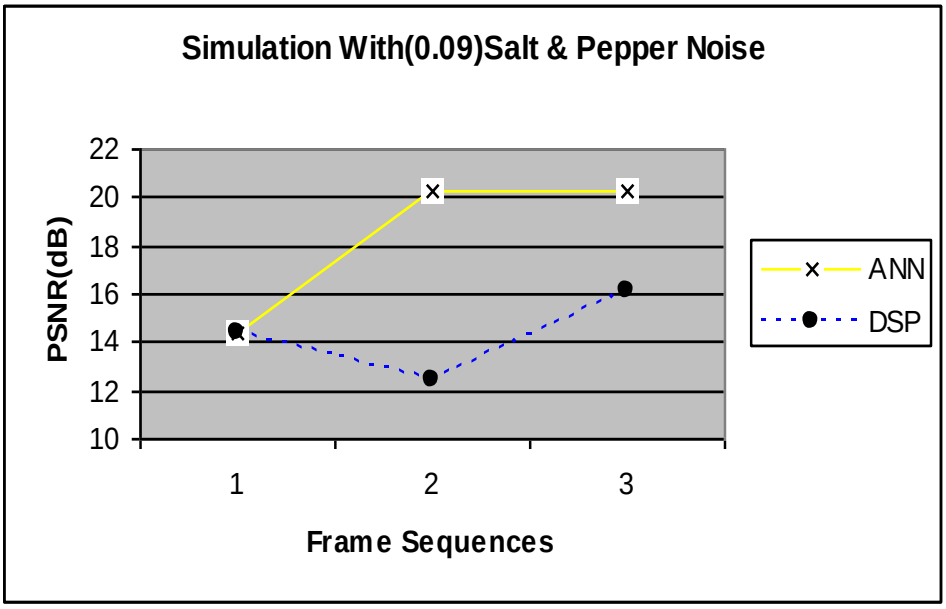
The inherent noise immunity, generalization, adaptability and simplicity of ANN's were exploited to overcome the shortcomings of conventional DSP. Utilizing ANN's in ensemble configurations reduced the data supplied to the ANN for processing, and hence contributed significantly towards reducing the processing time of the ANN system without compromising image quality.

	DSP Motion Detection System	**ANN Motion Detection System**	**ANN + DSP Motion Detection System**
Without noise	11.75s	7 min	32.115secs
With induced noise	17.33s	7 min	34.795secs

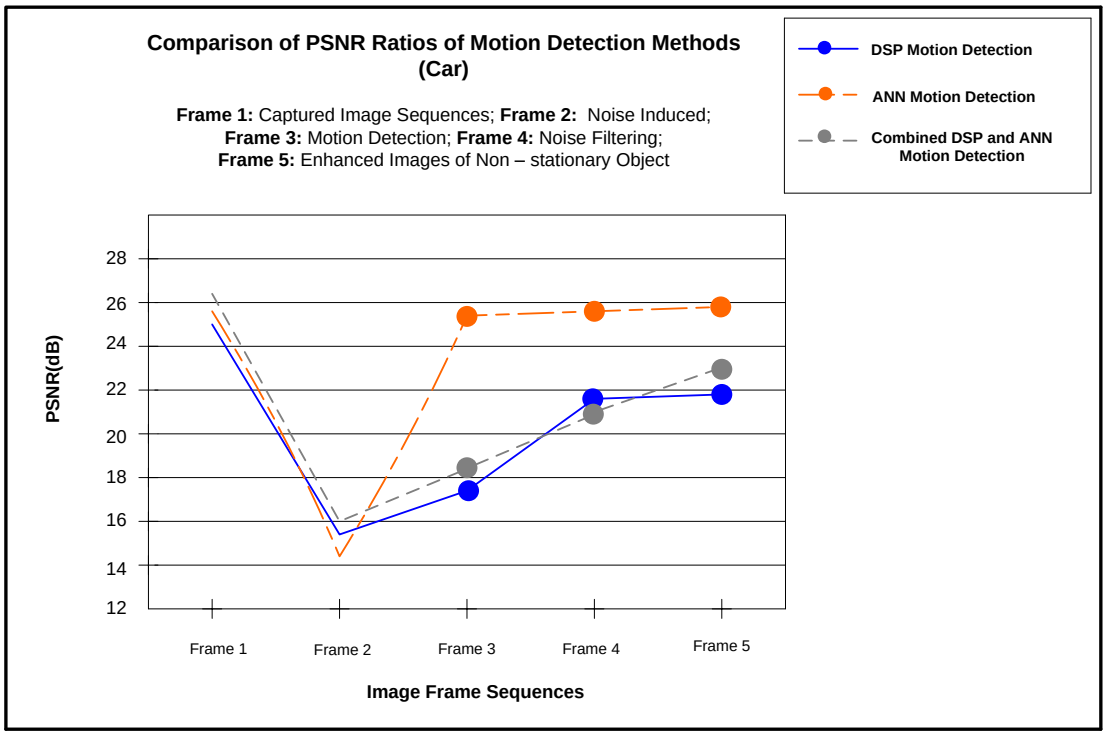
TABLE 5.1: Average System processing time of DSP and ANN Motion Detection Systems (Using P4: 3GHz, 2GB RAM)

	No Preprocessing	**Preprocessing with DCT Compression**	**Preprocessing with Haar Wavelet**	**Preprocessing with Daubechies Wavelet**	**Preprocessing with Biorthogonal Wavelet**
Without noise	14.98 min	9 min	5.7 min	6.4min	6.15mins
With induced noise	14 min	8.05 min	5.97 min	8.5min	7mins

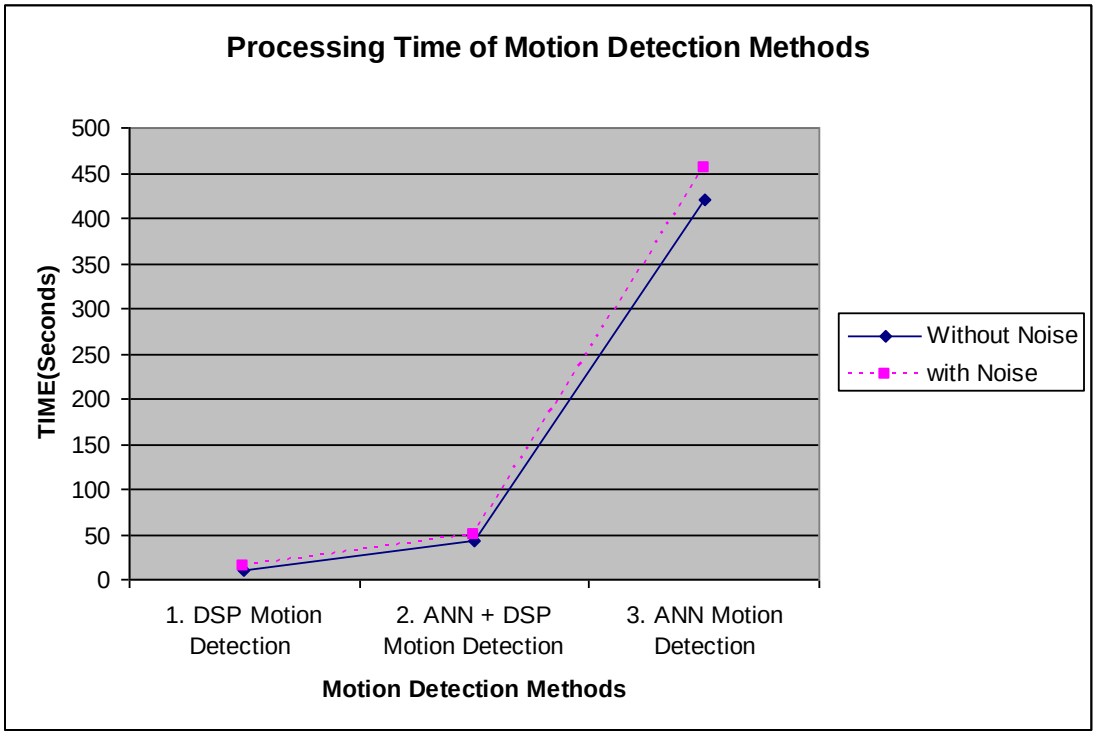
Table: 6.3 Average Processing Time of ANN Motion Detection System (Using P4: 3GHz, 2GB RAM)



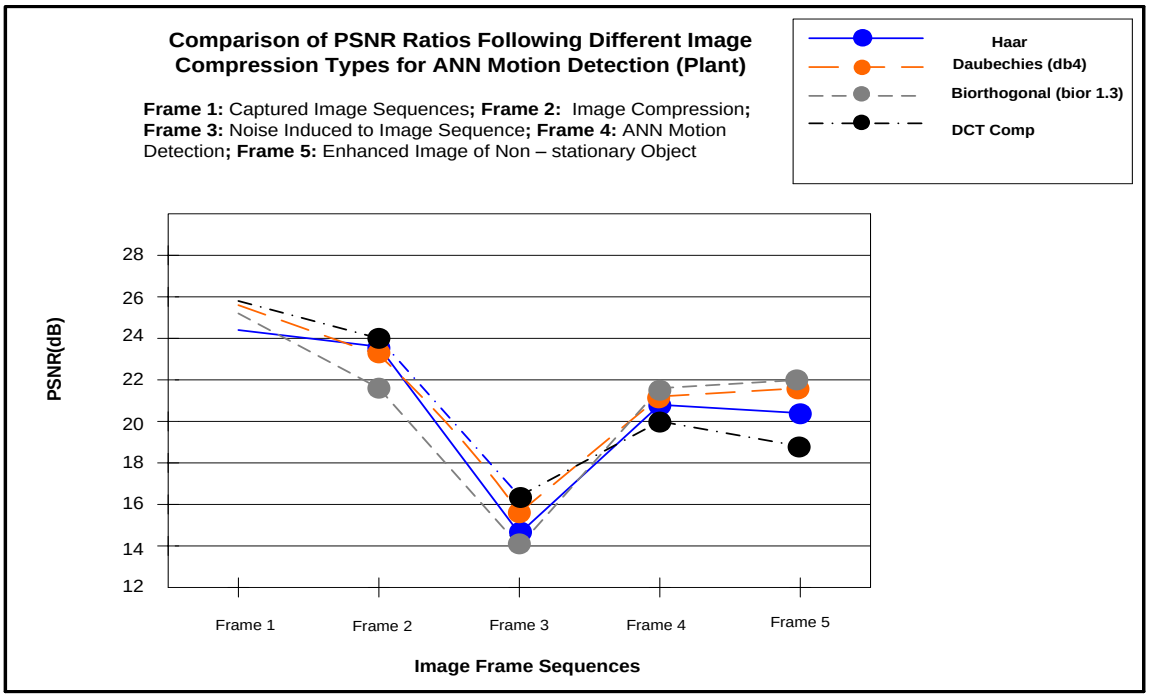
Graph 4.1 (c) PSNR Response of DSP versus the ANN system



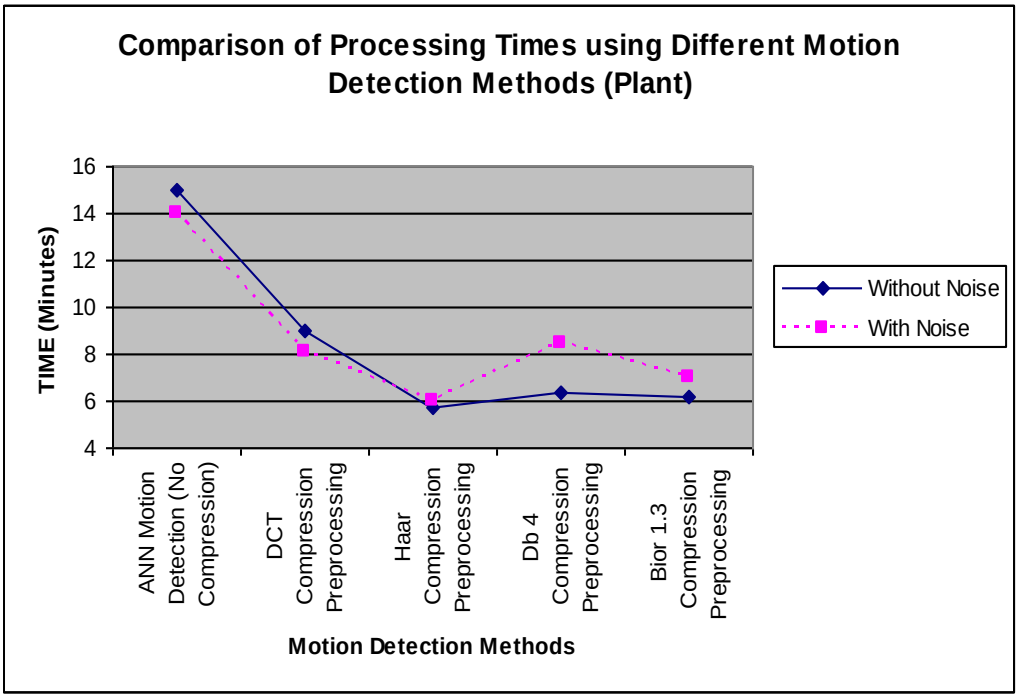
Graph 5.1(a) PSNR of Hybrid Motion Detection System (Car)



Graph 5.2 Processing Times of Motion Detection Methods



Graph 6.1 (c) PNSR of ANN Motion Detection using Image Compression as Preprocessing



Graph 6.2 Average Preprocessing Times of ANN Motion Detection System with Image Compression

References

[1] Ahmed J., Jafri M.N., Khan M.I., "Design of a Neural Network for Real-Time Object Tracking", Proceedings of World Academy of Science, Engineering and Technology, Vol.6, June 2005 *PWASET* ISSN 1307-6884.

[2] Allwein E., Schapire R.E., and Singer Y., "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers", Journal of Machine Learing, Vol. 1, pp 113 – 141, 2000.

[3] Asamoah F., "Discrete Wavelet Analysis of Two – Dimensional Signals", International Journal of Electrical Engineering Education 39/2, 2008.

[4] Alam M., Badawy W., Dimitrov V. and Jullien G., "An Efficient Architecture for Lifted 2D Biorthogonal DWT", Journal of VLSI Signal Processing 40, 335-342, 2005, Springer Science + Business Media, Inc.

[5] Bitten J.R. and Dr. Osorio F.S., "Adaptive Filters for Image Processing based on Artificial Neural Networks", Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing, *IEEE* (SIBGRAPI '00). ISBN 0-7695-0878-2/00, 2000.

[6] Buckheit J.B. and Donoho D.L., "Wavelab and Reproducible Research Technical Report", Department of Statistics, Stanford University, Stanford Califonia, USA, 1995.

[7] Bins L.S., Garcia Fonseca L.M., Erthal G.J. and Mitsuo It F., "Satellite Imagery Segmentation: A Region Growing Approach", Vol.8 Simposio Brasilerio de Sensoriamento Remoto, Salvador, Brasil, Journal of International Institute of Space Research (INPE), pp 677 – 680, 1996.

[8] Corbalan L., Osella Massa G., Russo C., Lanzarini L., De Giusti A., "Image Recovery Using a New Nonlinear Adaptive Filter Based on Neural Networks", III-LIDI, Faculty of Computer Sciences, National University of La Plata, Argentina.

[9] Cheng L., Liang D.L. and Zhang Z.H., "Popular Biorthogonal Wavelet Filters via a Lifting Scheme and its Application in Image Compression," *IEE* Prc-Vis Image and Signal Processing, Vol. 150, August 2003. Proceedings online no. 20030557.

[10] Desa S.M and Salih Q.A, "Image Subtraction for Real Time Object Extraction", *IEEE* Proceedings of the International Conference on computer Graphics, Imaging and Visualization, pp xx-xx, 2004.

[11] Dasarathy B.V. and Sheela B.V., "Composite Classifier Systems Design: Concepts and Methodology", Proceedings *IEEE*, vol. 67 Number 5, pp 708-713, 1979.

[12] Dumontier C., Luthon F. and Charras J, "Real Time DSP Implementation for MRF-Based Video Motion Detection", *IEEE* Transactions on Image Processing, Vol.8 No.10, pp. 1341 – 1347, Published Oct.1999.

[13] Dai Q, Chen X, and Lin C**,** "Fast Algorithm for Multidimensional DCT to DCT Computation Between a Block and Its Associated Subblocks", *IEEE*, 1053-587X, pp 3219 – 3225, 2005.

[14] Dietterich T.G. and Bakiri G., "Solving Multiclass Learning Problems via Error – Correcting Output Codes", Journal of Artificial Intelligence Research, Vol.2, pp. 263 – 286, 1995.

[15] Dremin I.M., "Wavelets: Mathemathics and Applications", Physics of Atomic Nuclei, Vol.68, No 3, 2005 pp 508 – 520. From Yadernaya Fizika, Vol.68, No. 3, 2005 pp 537 – 549.

[16] Elfouly F.H., Mahmoud M.I., Dessouky I.M., and Deyab S., "Comparison Between Haar and Daubechies Wavelet Transformations on FPGA Technology", International Journal of Computer Information, Systems Science, and Engineering 2;1 *WASET* 2008.

[17] Falomir H., Muschietti M.A., Santangelo E.A. and Solomin J., "A Calculation with a Biorthogonal Wavelet Transformation", Journal of Mathematics and Physical Science, Vol. 35 April 1994, American Institute of Physics 1994 pp 1939 – 1950.

[18] Gonzalez R.C, Woods R.E. and Eddins S.L., "Digital Image Processing Using Matlab", Prentice Hall, Published, 2004.

[19] Gonzalez R.C. and Woods R.E., "Digital Image Processing", Pearson Prentice Hall, Third Edition, Published 2008.

[20] Govender P., Singh R., Paramanund C., Naicker A., Bright G., "Industrial Quality Assurance Using Artificial Intelligence", Published, 2006.

[21] Garcia-Cabrera L., Garcia-Salina M. J., Luque-Escamilla P. L., Martinez-Aroza J., Gomez Lopera J. F., Roman-Roldan R. "Median –Type Filters with Model-Based Preselection Masks, Image and Vision Computing", ISSN 0262-8856 *Elsevier, Oxford*, Vol.14, pp 741-752, 1996.

[22] Goswami J.C. and Chan A.K., "Fundamentals of Wavelets, Theory, Algorithms, and Applications", Published 1999 John Wiley & Sons, Inc, ISBN 0-471-19748-3.

[23] Gransermann U. and Miikkulainen R., "Effective Image Compression Using Evolved Wavelets", University of Texas, Department of Computer Sciences, USA, 2005.

[24] Hagan M.T., Demuth H.B., and Beale M.H., "Neural Network Design", Boston, MA: PWS Publishing,1996.

[25] Hansen L.K. and Salamon P., "Neural Network Ensembles", *IEEE* Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, Number 10, pp 993- 1001, 1990.

[26] Ham F.M. and Kostanic I., "Principles of Neurocomputing for Science and Engineering", Mc Graw Hill, Published, 2001.

[27] Hussain A. and Campbell D.R., "Binaural Sub-band Adaptive Speech Enhancement Using Artificial Neural Networks", Elsevier Science, pp 177 – 186, 1998.

[28] Heath M., Sarkar S., Sanocki T. and Bowyer K., "Comparison of Edge Detectors, A Methodology and Initial Study", Computer Vision and Image Understanding, Volume 69, Number 1, January 1998, pp 38-54, Article number IV960587.

[29] Jordan M.J. and Jacobs R.A., "Hierarchical Mixtures of Experts and the EM Algorithm", Neural Computation, Vol. 6, Number 2, pp 181-214, 1994.

[30] Khare A. and Tiwary U.S., "Daubechies Complex Wavelet Transform Based Technique for Denoising of Medical Images", International Journal of Image and Graphics, Vol 7, No.4 (2007) 663-687, World Scientific Publishing Company.

[31] Liu M., Jiang X., Kot A. C., "Nonlinear Fingerprint Orientation Smoothing by Median Filter", *IEEE* Information, Communications and Signal Processing, Fifth International Conferernce, Vol. Issue 2005 pp 1439-1443.

[32] Mitra S.K., "Digital Signal Processing, A Computer –Based Approach", McGraw Hill, Published, 1998.

[33] Murta L.O.Jr., Ruiz E.E.S., Pazin-Filho A., Scmidt A., Almeida-Filho O.C., Simoes M.V., Marin-Neto J.A. and Maciel B.C., "Automated Grading of Left Ventricular Segmental Wall Motion by an Artificial Neural Network Using Color Kinesis Images", Brazilian Journal of Medical and Biological Research (2006) 39: 1-7 ISSN 0100-879X.

[34] Moorgas K.E. and Govender P., "DSP Systems vs ANN Ensembles for Motion Detection and Filtering", Conference Proceedings, World Congress of Engineering and Computer Science, WCECS 2008, pp 1129-1134.

[35] Majhi B., Sa P.K., and Panda G.K., "ANN based Adaptive Thresholding for Impulse Detection", Proceedings Signal Processing, Pattern Recognition, and Applications 2006 ACTA Press.

[36] Misiti M., Misiti Y., Oppenheim G., Poggi J.M., "Wavelet Toolbox for Use with Matlab", The Mathworks Inc, September 2000.

[37] Menegaz G., "Trends in Medical Image Compression", Current Medical Imaging Reviews, 2, 165-185, Bentham Science Publishers Ltd, 2006.

[38] Najarian K., and Splinter R., "Biomedical Signal and Image Processing", Taylor & Francis CRC Press, Published, 2006.

[39] Narendra K.S., and Pathasarathy K., "Identification and Control of Dynamical Systems Using Neural Networks", *IEEE* Trans on Neural Networks. Vol 1.No 1. 3, 1990.

[40] Nagan K.A., Meier T., and Chai D., "Video Coding Principles and Techniques", Volume 7, *Elsevier*, Published, 1999.

[41] Nelwamondo F.V. and Marwala T., "Fuzzy Artmap and Neural Network Approach to Online Processing of Inputs with Missing Values", Trans, Africa Research Journal, Vol 98, No.2 pp 45 – 51, June 2007.

[42] Nadarajah S. and Kotz S., "Popular DCT Models", International Journal of Wavelets, Multiresolution and Information Processing, Vol.5, Number 5, pp 725 - 733, 2007, World Scientific Publishing Company.

[43] Nielsen M., Kamavuako E.N., Andersen M.M., Lucas M.F. and Farina D., "Optimal Wavelets for Biomedical Signal Compression", Medical Bio Engineering Computer (2006) 44: pp561-568 DOI 10.1007/s11517-006-0062-0.

[44] Polikar R., "Ensemble Based Systems", *IEEE* magazine circuits and systems 2006, 1531-636X/06/. pp 21 – 45.

[45] Pongpadpinit W. and Pearmain A, "Recovery of Motion Vectors for Error Concealment Based on an Edge Detection Approach", *IEE* February 2006, Proceedings Image and Signal Processing, Vol. 153, Number 1, pp 63 – 67.

[46] Pragada S. and Sivaswamy J., "Image Denoising Using Matched Biorthogonal Wavelets", Sixth Indian Conference on Computer Vision, Graphics & Processing, 978-07695-3476-3/08, 2008 IEEE, DOI 10.1109/ICVGIP.2008.95.

[47] Raviraj P. and Sanavullah M.Y., "The Modified 2D - Haar Wavelet Transformation in Image Compression", Middle-East Journal of Scientific Research 2 (2) 73-78, 2007, ISSN 1990-9233, IDOSI Publication, 2007.

[48] Roushdy M., "Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter", GVIP Journal, Volume 6, Issue 4, pp 17-23, 2006.

[49] Riazifar N. and Yazdi M., "Effectiveness of Contourlet vs Wavelet Transform on Medical Image Compression: A Comparative Study", Proceedings of World Academy of Science, Engineering and Technology (PWASET) Vol. 37, January 2009 ISSN 2070-3740.

[50] Sarfraz E.M., "Computer Aided Intelligent Recognition Techniques and Applications", Wiley, Published, 2005.

[51] Sato K., Aggarwal J.K., "Temporal Spatio-Velocity Transform and Its Application to Tracking and Interaction", *Elsevier*, Computer Vision Understanding 96, pp. 100 – 128, 2004.

[52] Shapiro L.G. and Stockman C. G., "Computer Vision", Prentice Hall, Published 2001. ISBN 0-13-030796-3.

[53] Singh S., Kumar V., and Verma H.K., "Optimization of Block size for DCT-based Medical Image Compression", Journal of Medical Engineering & Technology, Vol. 31, No. 2, March/April, pp 129 – 143, 2007.

[54] Stewart R. W., "Digital Signal Processing: Technology and Marketing for Audio Systems", IEE Colloquium on Digital Audio Signal Processing, Digest No 07, pp Chap 5, 1-6, 1991.

[55] Thompson C.D., "A VLSI Sigma –Delta A/D Converter for Audio and Signal Processing Applications", Proceedings of ICASSP, Glasgow, UK, May 1989.

[56] Talukder K.H. and Harada K., "Haar Wavelet Based Approach for Image Compression and Quality Assessment of Compressed Images", IAENG International Journal of Applied Mathematics, 36:1, IJAM 36 1 9, Online publication 1 February 2007.

[57] Viet T., Patra J.C., and Meher P.K., "WMicaD: A New Digital Watermarking Technique Using Independent Component Analysis", *Eurasip Journal on Advances in Signal Processing*, Vol 2008, Article ID 317242.

[58] Vozikis G. and Jansa J., "Advantages and Disadvantages of the Hough Transform in the Frame of Automated Building Extraction", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences Journal. Vol. 37 Part B3b, Beijing, 2008, pp 719 – 724.

[59] Van Der Heijden F., Duin R.P.W., De Ridder D. and Tax D.M.J., "Classification, Parameter Estimation and State Estimation: An Engineering Approach Using Matlab", Wiley, Published 2004.

[60] Wu B.F., Lin S.P. and Chiu C.C., "Extracting Characters from Real Vehicle Licence Plates Out-of-Doors", Proceedings *IET* pp xx-xx, 2007.

[61] Wang Y., Wen Z., Nashed Z., and Sun Q., "On Direct Methods for Time-Limited Signal and Image Reconstruction and Enhancement", International Journal of Wavelets, Multiresolution and Information Processing, Vol.5, Number 1, pp 51- 68, 2007, World Scientific Publishing Company.

[62] Wang J.H. and Chiu H.C., "HAF: An Adaptive Fuzzy Filter for Restoring Highly Corrupted Images by Histogram Estimation", Proceedings National Science Council. ROC(A), Vol.23 Number 5, pp 630 – 643, 1999.

[63] Wang J.Z., Wiederhold G., Firschein O and Wei S.Z., "Content Based Image Indexing and searching using Daubechies Wavelets", International Journal on Digital Libraries, Springer-Verlag 1997, Vol. 1 pp311- 328.

[64] Winger L.L. and Venetsanopoulos A.N., "Biorthogonal Nearly Coiflet Wavelets for Image Compression", Elesevier, Signal Processing: Image Communication 16 (2001), pp 859 – 869. 0923-5965/01/, PII: S0923-5965(00)0047-3

[65] Yazdi S.H, Lotfizad M. and Fathy M., "Car Tracking by Quantised Input LMS, QX-LMS Alorithm in Traffic Scences", Proceedings *IEE* conference, pp37-44, February 2006.

[66] Zuo F. and De With P.H.N., "Cascaded Face Detection Using Neural Network Ensembles", *Eurasip Journal on Advances in Signal Processing*, Vol 2008, Article ID 736508.

[67] Zhang C., Wang X., Zhang H., and Duanmu C., "An Anti-Noise Algorithm for Enhancing Global and Local Contrast for Infrared Images", International Journal of Wavelets, Multiresolution and Information Processing, Vol.5, Number 1, pp 101-112, 2007, World Scientific Publishing Company.

Appendix A: Publications and Awards

A1. International Conference Paper:

Moorgas K.E. and Govender P., "DSP Systems vs ANN Ensembles for Motion Detection and Filtering", *Proc.- IAENG World Congress of Engineering and Computer Science*, pp 1129-1134, 2008.

A2. IEEE Book Chapter:

Moorgas K.E. and Govender P., 'ANN Ensembles *vs* DSP for Mobile Detection and Filtering', *IAENG Trans. On Electrical and Electronic Engineering*, Vol.1, IEEE Computer Society

A3. Award

Best Student Paper Award

IAENG INTERNATIONAL ASSOCIATION OF ENGINEERS

The World Congress on Engineering and Computer Science 2008

San Francisco, USA, 22-24 October, 2008

presents this

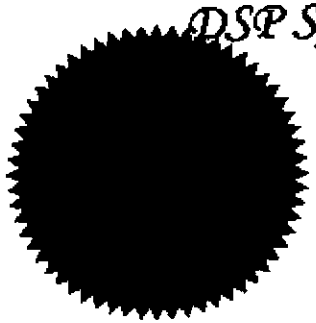
Best Student Paper Award of
International Conference on
Signal Processing and Imaging Engineering 2008

to

Kevin Emanuel Moorgas, and Poobalan Govender

for the paper entitled

DSP Systems vs ANN Ensembles for Motion Detection and Filtering



Anna Lee, Assistant Secretary, IAENG
2 December 2008