APPLICATION OF PATTERN RECOGNITION TO PROJECTIVE 3D IMAGE PROCESSING PROBLEMS

Mariana Liana Dănăilă

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, in fulfilment of the requirements for the degree of Master of Science

Johannesburg, March 2000

Declaration

I declare that this dissertation is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Mariana Liana Dănăilă

31 March, 2000

Abstract

This dissertation presents the development and performance of a few algorithms used for automated scene matching. The objective is to recognise and predict the location of a template (reference image) inside a degraded scene image (sensed image). A set of perspective, projective optical images of relatively well defined man-made objects located in areas of varying background is used as database. Perturbations to the grey levels of the image cause artefacts that easily destroy the unique match location and generate false fixes. Therefore, suitable enhancement and noise removal techniques are applied first. Several different types of features are investigated to decide upon those that are best suited to describe the original content of the scene. Statistical features, such as invariant moments are chosen for one of the algorithms, *Multiband Image using Moments (MBIMOM)*. The second one, *Spatial Multiband Image (SMBI)* algorithm, uses the spatial correlation of the pixels within a neighbourhood as initial descriptive features. Each algorithm uses either *Principal Components* transform or *Maximum Noise Fraction* transform for dimensionality and noise reduction. A normalised correlation coefficient of 1.00 was achieved by the SMBI algorithm. The final design of the algorithms is a trade-off between speed and accuracy.

To my parents, Dida and Ştefan, my sister Aida, and my fiancé, Roger

Acknowledgements

Foremost I thank my former supervisor, Dr. V. Nedeljkovič, for his unwavering support and enthusiasm throughout the course of my postgraduate academic career. I am indebted to him for his expertise and foresight, his patience and dedication.

I thank Professor F. M. Mahomed for his invaluable advice and for undertaking to supervise this dissertation at the eleventh hour after unforseen circumstances rendered the former arrangements impossible.

I thank Dr. N. Pendock for his expert direction in the practical aspects of image processing which helped me assimilate the theoretical and computational sides of the field in a meaningful and enriching way.

I wish to thank Dr. E. Momoniat and Mr. K. Mitchell for their pertinent comments and suggestions.

I am also indebted to Mr. G. B. Grobbelaar, Department of Electrical Engineering, for his useful suggestions about the coding of the algorithms.

I am grateful to Mr. D. B. Walker (Eskom-TRI, South Africa) and Mr. R. Dalais (BCE, Mauritius) who generously supplied the original optical images which were used to test the algorithms.

Special thanks go to Mr. J. Everitt (Anglo American Corporation-GSD, South Africa) for proof-reading this dissertation.

I am also indebted to Mrs. L. Rouessart, Faculty of Science, for her kind and constant advice.

Lastly, I would like to thank the members of the Department of Computational and Applied Mathematics, in particular, Professor D. Block, Mr. G. Buric, Mr. J. Faulkner, Dr. A. R. Gardiner, Mrs. R. Gray, Mr. S. Hlubi, Mr. B. Maistry, Professor D. P. Mason, Ms. N. Roussos, Professor D. R. Taylor, Mr. D. Wilson and Professor C. Wright (mentioned in alphabetical order) for their unceasing support.

Contents

List of figures

List of Tables

Abbreviations

1D	One dimensional signal
2D	Two dimensional signal
3D	Three dimensional signal
ATR	Automatic (or aided) target recognition system
CDF	Cumulative distribution function
CWT	Continuous wavelet transform
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DWT	Discrete wavelet transform
FIR	Finite impulse response filter
FFT	Fast Fourier transform
FLIR	Forward looking infrared image
FT	Fourier transform
IDWT	Inverse discrete wavelet transform
IU	Image understanding
LSI	Linear space invariant system
MBIMOM	Multiband Image algorithm using moments
MNF	Maximum noise fraction transform
MSE	Mean square error
NAPC	Noise-adjusted principal components transform
PCA	Principal component analysis
PDF	Probability density function
PSF	Point spread function
RGB	Red, green and blue colormap
SAR	Image recorded by synthetic aperture radar
SLAR	Image recorded using side-looking airborne radar
SMBI	Spatial multiband algorithm
SNR	Signal-to-noise power ratio
STFT	Short time Fourier transform
SVD	Singular value decomposition

1 Introduction

1.1 Background

One of the oldest and most complicated image forming systems is the human beings' visual system. The combination eye-brain system acts partially as a high-resolution sensor. Despite its sophisticated design, the human visual system has several limitations. The eye is sensitive only to a certain range of wavelengths. It does not have the capability to perform at extended range. It does not perceive complete signal information when the illumination is very poor, and is deficient in transmitted-attenuated atmospheres. Its performance decreases after a long period of continuous collection of signal data and in the presence of repetitive tasks. The human eye can easily be deceived, and cannot perform in the presence of dangerous, hostile conditions.

As a response to these shortcomings, humans have developed intelligent sensors, able to record signals beyond their natural abilities, in order to accomplish complex tasks. These intelligent sensors are also designed to register large amounts of data. Despite the fact that no computer-based image system can compete with the overall performance of human vision in flexibility or speed, there are specific tasks when the computer surpasses any human's abilities. The processing and analysis of large amounts of data by humans, in a repetitive manner, over a long period of time, are prone to errors. In addition, the data must be processed accurately and the invariance to changes kept. In many cases, mapping images into other spaces (that are beyond normal human experience, e.g., Fourier or Hough spaces) results in extracting hidden data. Often, it is useful that absolute measurements are performed instead of relative comparisons. The use of statistical techniques allows extracting valuable information that is, otherwise, embedded in chaotic and noisy data, in order to identify underlying trends and similarities. Hence, the need to convert the signals recorded by sensors into an appropriate form that can be interpreted and processed by digital computers.

Most objects encountered in the physical world are three-dimensional. A group of these objects may be called a *scene*. A two-dimensional projection or picture can be generated by viewing an image of a scene. A scene refers not only to the collection of objects that form that scene, but also to their relationships in the three-dimensional environment. Therefore, a scene is composed of a set of discrete objects and their relationships with one another, while an image is created by a set of discrete regions.

Images can be classified into several categories, based on their form or the method used to generate them. Amongst such a wide variety of images, this dissertation limits itself to the description, manipulation, analysis and interpretation of visible images, that is, those images that can be perceived and seen by the eye.

1.2 Data acquisition

Regarding the method of generating images, the study is conducted using optical images, formed with lenses, more precisely recorded by a photographic camera. The images used are *perspective as well as projective optical images of relatively well defined, man-made objects located in areas of various backgrounds*. The general method of acquiring such an image type is depicted in figure 1.1, and is similar to that of figure 4.25 from Hall (1979).

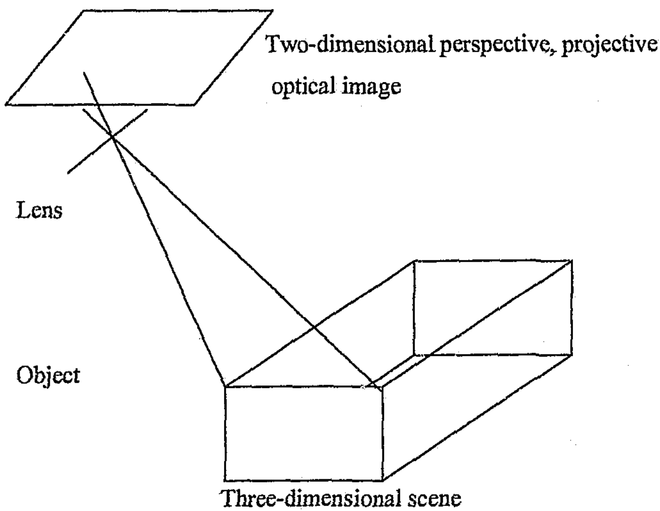


Figure 1.1 Method of generating a perspective, projective, two-dimensional optical image of a scene

The purpose of figure 1.1 is not solely to describe the method of acquisition of an optical image, which is a trivial, well-known process, but to emphasise the types of optical images that were used. The decision to use optical images was made due to lack of access to a database generated by intelligent, imaging or non-imaging sensors. Therefore, the effort was concentrated on expressing the image content using features that are invariant to various changes and not to exploit the information generated by a multisensor approach. This can be considered a limitation of this dissertation. Once the image is generated, the next step is to transform it into a suitable form for computer analysis. The images that were used are black-and-white images that have only one value of brightness at each point. The basic element (the smallest) that composes a digital image is called a pixel. A digital image can be defined as "a sampled, quantized function of two dimensions that has been generated by optical means, sampled in an equally spaced, rectangular grid pattern, and quantized in equal intervals of amplitude" (Castleman, 1996). Non-optical devices, for example, radar, can also lead to digital images. Therefore, a quantified value, associated with a pixel in a digital image, represents the brightness of the original scene at the point (represented by that pixel), and is known as a *grey level*. The set of all possible grey levels in a digital image represents the grey scale. This is one of the commonly used forms for image data representation for use with a computer. The image is equivalent to a two-dimensional matrix of samples, of which each element describes the brightness of an image element having particular coordinates. Although the source of the images is usually continuous, both spatially and in grey scale, after digitisation, the input data are both discrete and quantified. It is also desirable that the images that are used for computer processing and analysis manifest global uniformity and local sensitivity to variations.

1.3 Statement of the problem

A very significant problem in image analysis is the detection of change or presence of an object in a given scene. These problems occur in sensing for monitoring growth patterns of urban areas, weather prediction using satellite images, diagnosis of disease from medical images, target detection from radar images, automation using robot vision and so on.

The subject of this dissertation is automated scene matching. In broad terms, automated scene matching refers to the ability to locate or match a region of an image with a corresponding region of another view of the same scene, often taken with a different sensor at a different view angle, without the intervention of a human operator.

A subset of a scene image, called a template, is selected with the goal to find all possible locations of the template within scene image. Undesired perturbations such as noise, blurring, rotational and translation errors, changes in illumination or scale, or an incomplete set of data as initial signal image, may degrade the original scene image. One of the perspective, projective, two-dimensional, non-degraded images of relatively well defined man-made objects located in areas of various backgrounds that was used to perform scene matching, is displayed in figure 1.2, together with its designated template. Any subset of the image may be chosen as a template, but a subset characterised by a distinctive signature generates a better similarity measure.

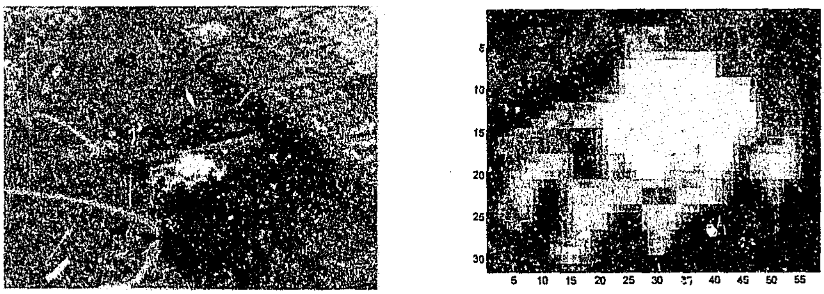


Figure 1.2 Image “UKF.bmp” together with its designated template, used for scene matching process

The process to detect which part of an image is similar to another one, based on the information contained in a scene is a decisional process that necessitates the use of various methods developed by *pattern recognition*. Pattern recognition, which is an independent mathematical discipline, when applied to digital image processing, means that objects in an image can be detected, measured and classified by automatic or semi-automatic methods.

Scene matching is not a new subject of research amongst a bewildering range of topics tackled by different researchers in the field of image processing. An impressive collection of techniques and algorithms has evolved over time. Almost all the scene matching systems may be considered to be composed of five building blocks (see figure 1.3).

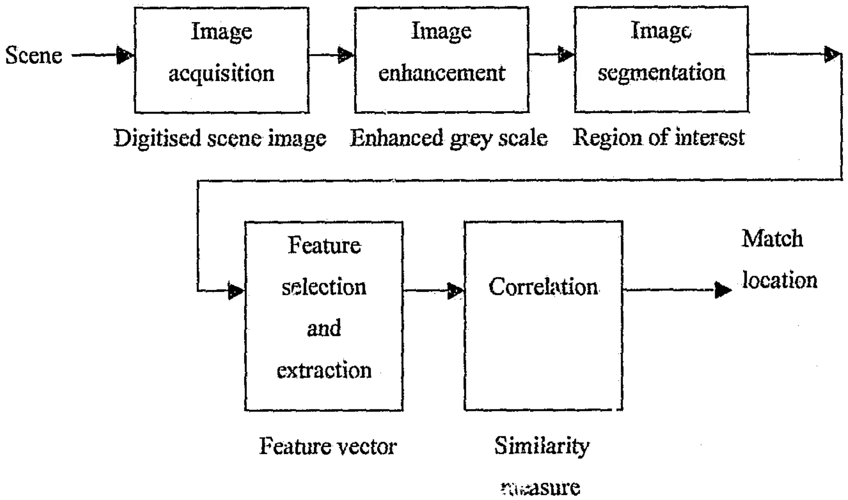


Figure 1.3 Scene matching system

This dissertation does not describe or analyse various sensors, their characteristics or performances, and does not detail different sampling methods. A shortcoming of many scene matching algorithms is that they require a reasonable standard of the input image. Another problem is that some of the algorithms in use fail to detect the true match location, in the case of templates whose content is similar in energy with other parts of the image, or in the case of a medium or high cluttered[1] environment. As a result, most of the algorithms generate multiple false fixes. Also, most of the algorithms are computationally very expensive.

Due to all the above mentioned weak points manifested by various algorithms, it was decided to attempt the design of a set of scene matching algorithms that are robust to realistic differences, between the template and scene image. They must generate a unique

---

[1] "Clutter is a term borrowed from radar that has been expanded to include all signals in a scene of no interest to the observer. Currently, clutter is classified by trained observers as low, medium, or high, but we have been unable to quantify a scene's clutter content". - Ratches *et al.* (1997)

match for any pair of template-sensed image. Also, the scene matching process must be entirely automatic and the computational time required to perform the matching considerably low.

The methodology followed to achieve the above listed goals is different compared with the standard procedure. The pre-processing steps were reduced to a minimum. This approach resulted in an increased computational speed, without a loss in performance and accuracy. Attention was devoted to represent the original image content by using the most suitable features, whose descriptors are invariant to various perturbations. The best features enhance within-class pattern similarity and between-class pattern dissimilarity. Then, the selected features, which are a projection of the original representation space, were transformed onto a smaller dimensional subspace. The orientation of the new subspace that preserves the best of the information available in the original, complete space was found, using either *Principal Components Transform* or *Maximum Noise Fraction Transform.*

Image enhancement techniques are generally used to increase the visual appearance of the original image for human perception. In an automatic process, this step is not necessary. When a significant difference between the dynamic range of the original image and the sensed one was noticed, this was corrected by applying histogram processing. The goal was to adjust the grey scale, spreading the distribution of grey levels, wider and more evenly.

Images degraded by various perturbations generate poor results and increase the probability of false fixes. Therefore, special attention was paid to decrease the undesired effect of the noise, using the most suitable denoising techniques.

It was decided not to implement image segmentation. The reason was that, while image segmentation is the process of subdividing an image into disjoint regions, (normally corresponding to objects of interest and the background upon which the objects reside), the purpose of this research is to use any subset of the original image as a template to perform scene matching. Image segmentation would therefore have limited this project. The immediate effect was that the total computational time that is required by the algorithms to perform scene matching decreased.

Amongst a wide variety of feature descriptors, only the most suitable to describe the particular content of a natural scene, containing man-made objects and that are invariant to various perturbations were selected and used. It was decided to use *invariant moments* as feature descriptors for the first algorithm, named *Multiband Image using Moments*, MBIMOM. The second algorithm, *Spatial Multiband Image*, SMBI uses the original grey level values and the spatial correlation of the pixels within a neighbourhood as initial feature descriptors.

The feature descriptors represent an image that is still corrupted by noise, its elements are correlated and contain redundant or irrelevant information. Therefore, each algorithm uses either Principal Components transform or Maximum Noise Fraction transform for dimensionality and noise reduction. Dimensionality and noise reduction are performed automatically, in an optimum mean square error sense.

Finally, the normalised correlation coefficient was selected to measure the similarity between the scene image and its chosen subset, the template image. The algorithms were tested on a set of images, degraded by various perturbations, and their performance and accuracy assessed.

A short description of the content of each of the following chapters is given below.

Chapter 2 describes the image enhancement process. The attention was focused on image enhancement by point operation, that is, histogram processing. Effective illustrations depict the process of image enhancement.

Chapter 3 gives a detailed description of noise and its undesirable effects on an image. It investigates a wide range of filters and techniques aimed to reduce the noise. The results of applying a broad range of classical filters and relatively new techniques, such as *wavelet transform* and the improvement on image quality, as a consequence, are illustrated and commented upon. The choice of most suited techniques to reduce a certain type of noise, which degrades an image, is made based on the experimental results obtained. It also explains why denoising techniques in the frequency domain were not considered suitable for this project.

Chapter 4 presents a large range of image features and associated descriptors and states the reasons why invariant moments were selected as feature descriptors for one of the algorithms. It incorporates extensive experiments set to prove the invariance of these descriptors to various changes. The necessary mathematical background is provided.

Chapter 5 is concerned with dimensionality reduction techniques. Two main techniques were used, Principal Component transform and Maximum Noise Fraction transform. It also sets the mathematical background used to transform a monochromatic image into a "multispectral" image in a transformed feature space.

Chapter 6 provides a history of previous work done on the scene matching problem and the shortcomings of these techniques. Then, it describes two algorithms, each in two versions, proposed for solving the scene matching problem. It contains the description of a broad spectrum of experiments aimed to verify the robustness and accuracy of the algorithms that were tested for various scenarios. It incorporates and displays the results by means of graphical tools and tables. A full interpretation of the results is given. New ways of increasing the computational speed of the algorithms are also suggested.

Chapter 7 closes with the dissertation's conclusions and recommendations for further research.

Appendix A includes the mathematical derivation of the Principal Components transform.

Appendix B encapsulates the mathematical approach used to project an image from the original image space onto the Maximum Noise Fraction transformed space.

Appendix C contains all relevant code used. The code was written in MATLAB^{R}, version 5.2. The code for MATLAB's built-in functions that were used is not listed. The code written requires the image processing, signal processing and wavelet toolboxes.

Appendix D displays the database used to test the proposed algorithms.

2 Image enhancement by point operation

Image enhancement techniques are designed to improve the quality of an image as perceived by a person. At the same time, the principal objective of image enhancement procedures is to process an image in such a way that the resultant one is more suitable for a desired application. This can be done either using *spatial domain* methods or *frequency domain* methods. This chapter deals only with enhancement methods in the spatial domain based on point processing. Point processing methods modify the grey level of a pixel using only the brightness values, and neglecting its connections with the adjacent pixels.

2.1 Histogram processing

Amongst different enhancement techniques that will be described and used afterwards, histogram processing is one of the simplest as it is only based on the intensity of single pixels. The term *basic techniques* is used to include all the point operations that can be carried out on an analogue or digital image (Marion, 1991).

A point operation can be defined as being one which transforms an input image $A(x,y)$ into an output image $B(x,y)$ in accordance with a law f, which may or may not be linear and is consistent for the whole image:

$$B(x,y) = f[A(x,y)]. \tag{2.1}$$

For digitised images, each starting pixel is converted to another pixel at the same geometrical location, which derives its grey level from the original pixel using law f. These kinds of operations handle dynamic range and contrast. The histogram is considered to be a point operation that enhances the general contrast of the image, spreading the distribution of grey levels wider and more evenly, ideally to an equal number of pixels per grey level.

In order to improve the visual contrast of a digitised image, the grey scale can be adjusted so that the histogram of the resulting image has a pre-determined shape. The histogram of an image is used to display the distribution of grey values in the image, in other words, to plot the number of pixels at each of the 256 possible brightness levels. An 8-bit image has 256 (2^8) grey levels ranging from absolute black to absolute white. This increase in visual contrast is an artificial enhancement due to a more judicious choice of the relative intensities represented by the quantisation levels of the original image. Although the process does not provide enhancement in an information-theoretic sense, the data are simply better presented, resulting in a remarkable increase in visual clarity. In contrast, it must be remembered that histogram methods do not exploit the important fact that points from the same object are usually spatially close, due to surface coherence.

Histogram manipulation is the first step taken by researchers in image processing. Over many years, different histogram techniques have been developed and implemented, depending on the problem at hand. Some histogram processing techniques are more suitable for aerial pictures, others more suitable for satellite imagery or medical pictures. Amongst different histogram processing techniques that are described in the literature, the most suitable for this work was considered histogram equalisation and histogram specification. Later in this document, in chapters 4 and 6, based on the experimental results, it was decided that histogram equalisation is the best suited technique that must be used when the sensed image is degraded by intensity changes. Subsequently, the results generated by images degraded by intensity changes, without applying histogram equalisation and implementing this technique, will be compared and analysed.

2.1.1 Histogram equalisation

Histogram modelling is usually introduced in the literature using continuous, rather than discrete process functions (Gonzales and Woods, 1992; Marion, 1991; Fisher *et al.*, 1996), a procedure that facilitates the understanding of the process. In the previous section, histogram processing was introduced as a family of point operations defined by given equations. In equation (2.1), f may take any form (linear, non-linear, monotonic, non-monotonic, and so on), but it is important that f be consistent for all points of the image. The assumption is that pixel values are continuous quantities that have been normalised. Under these circumstances, they belong to the interval [0,1], with 0 representing black and 1 representing white. The forward transformation function, f, which produces a level s for every pixel value r in the original image is continuous within this interval. Furthermore, it must be assumed that the transfer function may also be written in terms of intensity density levels, for example:

$$D_B = f(D_A) \tag{2.2}$$

is single-valued and increases monotonically in the interval [0,1]. These assumptions will allow the inverse function to be defined as:

$$D_A = f^{-1}(D_B) \tag{2.3}$$

where the inverse function must satisfy the same conditions as the forward transformation described by equation (2.1). Figure 2.1 illustrates an example of such a transformation.

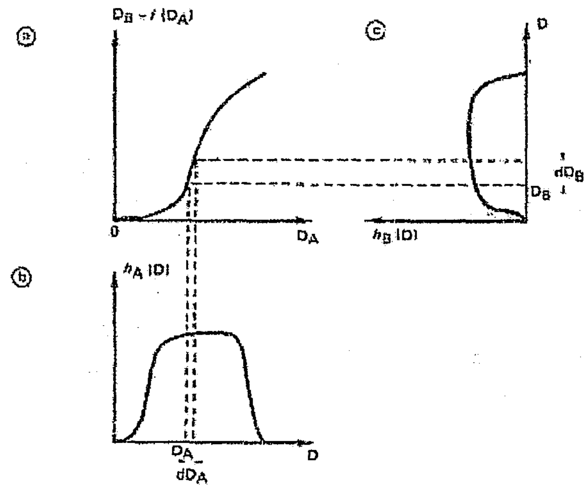


Figure 2.1 Transformation of the histogram by a strictly monotonic point operation

a) law $D_B = f(D_A)$

b) histogram of input image $h_A(D)$

c) histogram of output image $h_B(D) = \dfrac{h_A(f^{-1}(D))}{df/dD}$

Figure 2.1 is reproduced from Marion (1991). All pixel values from the input image with densities between D_A and $D_A + dD_A$ will generate points on the output image with densities ranging between D_B and $D_B + dD_B$. The surface areas $h_A(D_A)dD_A$ and $h_B(D_B)dD_B$ are therefore equal:

$$h_B(D_B)dD_B = h_A(D_A)dD_A. \tag{2.4}$$

If the histogram is regarded as a continuous probability density function $p_B(D_B)$ and is known, as well as the direct transformation *f*, the probability density function of the transformed grey levels can be defined as:

$$p_B(D_B) = \frac{p_A(f^{-1}(D_A))}{\left| {df}/{dD_A} \right|} \tag{2.5}$$

The input image *A(x,y)* is characterised by probability density $p_1(a)$ and distribution function $F_1(a)$. From this it is possible to obtain an output image *B(x,y)* such that its density $p_2(b)$ is constant between 0 and D_m.

The probability density function of the transformed grey levels can be defined as:

$$f(a) = D_m \int_0^a p_1(u)du = D_m F_1(a). \tag{2.6}$$

The quantity under the integral is known as the cumulative distribution function (CDF). Thus, for D_m, function *f* equals the distribution function of the input image. Therefore, transforming an image by its cumulative histogram yields an output histogram that is flat.

Because the images are presented in a digitised form, the concepts introduced previously are re-defined for the discrete form. In the case of grey levels that take on discrete values, the probabilities can be expressed as:

$$p_r(r_k) = \frac{n_k}{n} \quad \text{with } 0 \le r_k \le 1 \text{ and } k = 0, 1, ..., L\text{-}1 \tag{2.7}$$

where L represents the number of levels, $p_r(r_k)$ is the probability of the kth grey level, n_k is the number of times this level appears in the image, and n denotes the total number of pixels of the image.

The discrete form of transformation function f can be expressed as:

$$s_k = f(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} p_r(r_j) \text{ with } 0 \le r_k \le 1 \text{ and } k=0,\ 1,\ \ldots,L\text{-}1 \tag{2.8}$$

and the transformation function $f(r_k)$ may be computed directly from the image by using equation (2.8). Applying histogram equalisation more than once to the same image does not change the visual clarity of the image. During the first pass of histogram equalisation, the histogram is flattened and will subsequently remain unchanged.

Two of the images used for this work, named "UKF.bmp" and "UKC.bmp", are displayed in figures 2.2 and 2.3 respectively, together with their original histograms. Looking at the histogram of the image "UKF.bmp", it is apparent that this is quite narrow, which means that the image has low contrast features. In comparison, the histogram of the image "UKC. bmp" shows a wider distribution of grey levels, which means that the image has a good dynamic range. These conclusions can be verified directly through the appearance of the images to the naked eye.

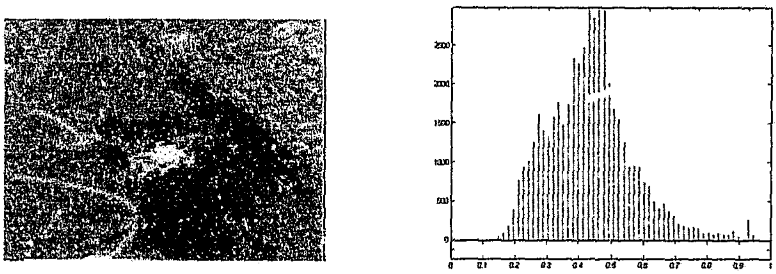


Figure 2.2 The original image "UKF.bmp" and its histogram

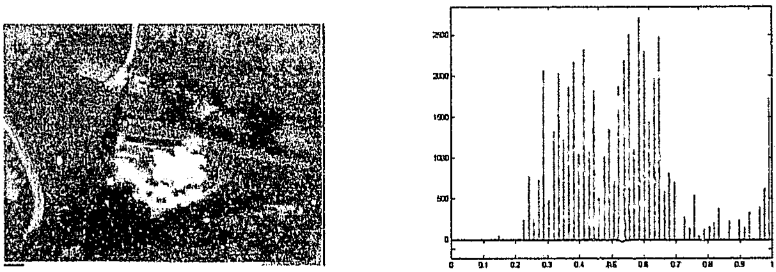


Figure 2.3 The original image "UKC.bmp" and its histogram

Therefore, it can be expected that applying histogram equalisation to these two images, the results will consist of a more dramatic improvement in the observed image 'quality' of "UKF.bmp", in comparison with the image "UKC.bmp". This is predictable, as the histogram of the former image (see figure 2.2) is more narrow, compared to the latter (see figure 2.3). Thus, histogram equalisation is more effective when applied to the image "UKF.bmp", that is, to images characterised by a narrow dynamic range.

Figures 2.4 and 2.5 display the same two images after histogram equalisation was applied, together with their new histograms.

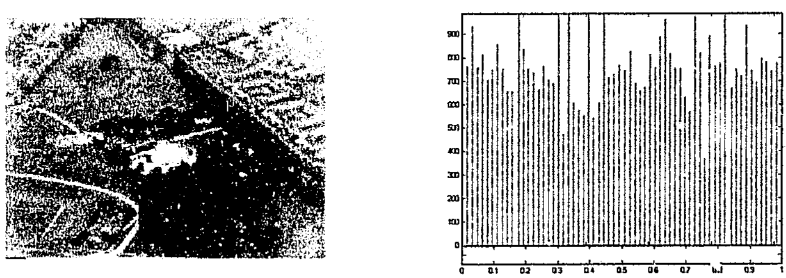


Figure 2.4 Image "UKF.bmp" after histogram equalisation and its histogram

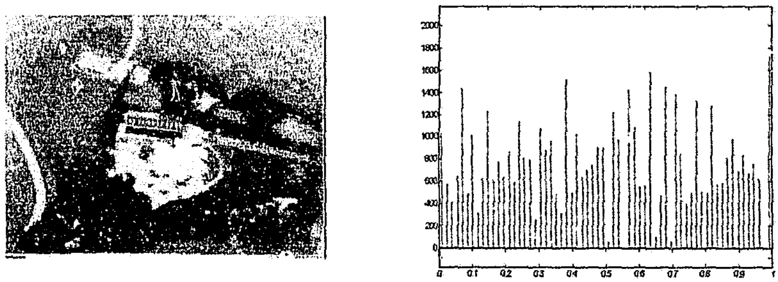


Figure 2.5 Image "UKC.bmp." after histogram equalisation and its histogram

If one is looking for details that are not visible or clear because the picture has a poor dynamic range, histogram equalisation is the logical step to be taken. This will visually enhance the entire image.

Sometimes it is necessary to specify certain histogram shapes, for example, to enhance certain density ranges for which greater detail is required. This can be accomplished by using *histogram specification,* which maps a given intensity distribution *a(x,y)* into a desired distribution *c(x,y)*, using a histogram equalised image *b(x,y),* as an intermediate stage. The reader interested in more details about histogram specification should consult Gonzales and Woods (1992).

A particular case of histogram specification, where the desired shape of the histogram follows a hyperbolic form, is *histogram hyperbolisation.* This type of histogram uses the fact that human perception performs a non-linear transformation of the light intensity. The aim of histogram hyperbolisation is to apply a transformation *J(I)* to the pixel value *I(x,y),* such that all perceived brightness levels, *B*, are equally probable. This procedure is applicable to automatic recognition of different templates, if the sensitivity profile of the sensor is known. This topic is extensively treated by Frei (1977) and Hummel (1977).

If there is more than one image of the same scene, it is possible to perform more point operations. They are called *basic algebraic operations* (addition, subtraction, etc.), which are commonly carried out between two or more images. The value attributed to each pixel in the output image depends only on the point with the same coordinates in each of the images that are processed, without regard for neighbouring pixels.

Subtraction is mainly used to discover differences between images. An effective way to remove the features that do not change, while highlighting those that do, is to subtract one image from another. This technique is used in reconnaissance photography to monitor the appearance or disappearance of targets in a complex scene (Russ, 1992). The parts of the picture that are essentially unchanged in the two images cancel out, except for minor variations, with the condition that lighting and geometry of view are consistent.

3 Noise reduction

Noise is a random signal, always unwanted, that interferes with the attempt to extract valuable information. In a broader way, noise can be defined as an additive or possibly multiplicative contamination of an image. Both image enhancement and image restoration are concerned with the development of techniques that include the operation of eliminating noise disturbances contained in the image. Noise disturbances may be due to spurious object reflections, inaccuracies in the sensing mechanism, disturbances introduced during transmission, irregular granularity in the recording film, or motion of the medium between the camera and the object. Real picture data are always distorted by noise, which is, generally, broadband in comparison with the undistorted picture signal. In the case of this work, noise might also introduce erroneous peaks, which may lead to false conclusions regarding the location of the template. Therefore, it is advisable to attempt removing or reducing the noise, contained in the images, prior to subsequent processing. To achieve this goal, a good understanding, description, estimation and the best suited methods of removing noise must be employed. This is the purpose of this chapter.

3.1 Noise description and modelling

As mentioned in the previous chapter, the images used for this work are optical images. Noise sources that are usually responsible for corrupting optical images can be divided into three categories. Firstly, images recorded on photographic film, as in the case of images used for this project, are subject to degradation by *film grain noise*. Secondly, the conversion of images, from optical to electrical form, that was performed on these images is a statistical process, because each image element received a finite number of photons. This type of degradation is usually called *photoelectronic noise*. Thirdly, electronic amplifiers that process a signal, introduce another type of noise, commonly called *thermal noise*. *Electronic noise* due to the random thermal motion of electrons in resistive circuit elements is usually modelled as spectrally *white Gaussian noise.* Electronic noise can be ignored, as sensor noise and quantisation noise are much greater. *Photoelectronic noise,* at low light levels, is often modelled as a random process governed by a *Poisson distribution.* At high light levels, the Poisson distribution approaches the Gaussian distribution. Finally, the intrinsic process of creating a picture, using an optical device, gives rise to a random process whose underlying distribution is a Poisson one. *Film grain noise* inherently exists in the process of photographic recording and reproduction. This type of noise can also be modelled effectively as a zero-mean *Gaussian* process. The reader interested in more details about different types of noise affecting optical images should refer to Castleman (1996), Marion (1991), Kuan *et al.,* (1985). If an image taken by an optical device is transmitted to be processed, another type of noise that can occur is *binary noise*, due to transmission errors. From the above description, it can be noticed that the image signal is degraded by a mixture of different types of noise, some signal dependent, others signal independent. For the purpose of this work, in which restoration of degraded images is not the core task, the assumption that can be made is that noise and signal are statistically independent. This implies that noise can be considered uncorrelated with the original image signal.

The task of restoring a degraded image can be treated in one of two basic ways. One way is to model and characterise the sources of degradation (blurring and noise) and implement a process designed to remove or reduce their effects. This is known as an estimation approach and is presented later in section 3.2. Another way is to develop a mathematical model of the original image and fit the model to the degraded image, with the condition that a great deal of prior knowledge about the image is available. In this case, it is rather a task of detection. From a different perspective, image restoration can be done using either continuous mathematics or discrete mathematics. The development can be carried out in *spatial domain* or *frequency domain* and the implementation can be performed in spatial domain (via convolution) or frequency domain (via multiplication).

Some types of degradation act only on the grey level of image points, without introducing any spatial blurring. They are often called *point degradations.* The degradation process can be modelled as an operator (or system), H, which together with an additive term, $n(x,y)$, known as noise, operates on an input image $f(x,y)$, generating the degraded image, $g(x,y)$.

Before mathematically describing the degradation process, it is useful to introduce the concept of linear systems. When the input to such a system is an impulse $\delta(x,y)$ centred at the origin, the output $g(x,y)$ is called the system's impulse response. Moreover, a system whose response remains the same, irrespective of the position of the input pulse, denotes a space invariant system. A linear space invariant (LSI) system can be described completely by its impulse response $g(x,y)$ as follows:

$$g(x,y) = H[f(x,y)] \tag{3.1}$$

where $f(x,y)$ and $g(x,y)$ are the input and output images, respectively. An LSI system must satisfy the following relationship (Jain *et al.*, 1995):

$$H[a \cdot f_1(x,y) + b \cdot f_2(x,y)] = aH[g_1(x,y)] + bH[g_2(x,y)] \tag{3.2}$$

where $f_1(x,y)$ and $f_2(x,y)$ are the input images, while $g_1(x,y)$ and $g_2(x,y)$ are the output images corresponding to f_1 and f_2. Here, a and b are constant scaling factors. For $a=b=1$, it can be concluded that, if H is a linear operator, the response to a sum of two inputs is equal to the sum of the two responses. Furthermore, an operator described by equation (3.1) is defined to be *space invariant* if:

$$H\left[f\left(x-x',y-y'\right)\right] = g\left(x-x',y-y'\right) \tag{3.3}$$

for any $f(x,y)$ and any x' and y'.

For this kind of system, the output $g(x,y)$ is the convolution of $f(x,y)$ with the impulse response $h(x,y)$, and for continuous functions can be defined as:

$$g(x,y) = f(x,y) * h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')h(x-x',y-y')dx'dy'. \tag{3.4}$$

Using the concept of LSI, the degraded image can be represented, taking also into account the noise, in the following manner:

$$g(x,y) = f(x,y) * h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x',y')h(x-x',y-y')dx'dy' + n(x,y). \tag{3.5}$$

In the continuous spatial frequency domain the corresponding equation is:

$$G(u,v) = H(u,v)F(u,v) + N(u,v) \tag{3.6}$$

where $F(u,v)$ and $G(u,v)$ are the Fourier transforms of the input and output images respectively, $H(u,v)$ is the filter transfer function and $N(u,v)$ designates the uncorrelated noise.

Once the mathematical model has been established, the next step is to provide the best possible estimate of the original picture $f(x,y)$. Before image restoration can be accomplished, some prior knowledge about the impulse response $h(x,y)$ is necessary, although it is sometimes possible to determine the degradation function, $h(x,y)$, experimentally, from the degraded image. It is also required to know the statistical characteristics of the noise and how the noise is related to the signal image. One of the best ways to restore an image to its former appearance, subject to the above degradations, is to design a filter that computes a signal, $\hat{f}(x,y)$, from the distorted image signal, which optimally approximates the original signal, $f(x,y)$, with respect to a defined error criterion. This can be achieved by using Wiener filter, that is the classical linear noise reduction filter.

3.2 Wiener filter

The system shown in figure 3.1 will be used to model image degradation and restoration processes in this section. The image $f(x,y)$ is blurred by a linear operation $h(x,y)$, and noise $n(x,y)$ is added to form the degraded image $w(x,y)$. The degraded image is convoluted with the restoration filter, $g(x,y)$, in this case the Wiener filter, to produce the restored image, $\hat{f}(x,y)$. The convolution operation can be visualised with functions being reflected, shifted, multiplied and integrated or it can be visualised as sinusoidal decomposition followed by multiplication and resummation (Castleman, 1996).

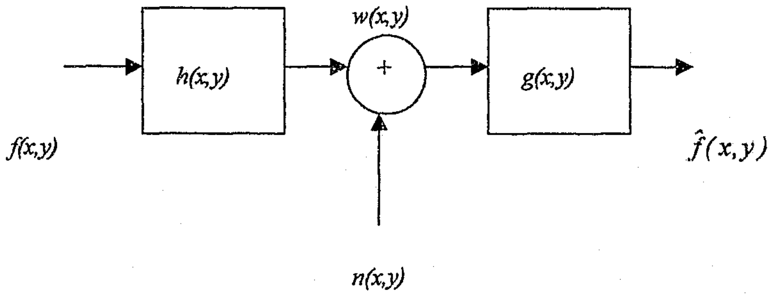


Figure 3.1 A continuous image restoration model

Before starting to present the optimal filter, it is necessary to introduce an objective criterion of optimality. When one is looking at an estimator, it is normal to assess how 'good' one estimator is compared to another. The most frequently used measure of the quality of an estimator is *the mean square error* (MSE). The error signal at the output of the filter can be defined as:

$$e(t) = f(t) - \hat{f}(t) \tag{3.7}$$

and denotes the amount by which the actual output differs from the desired one. As a measure of the average error, *the mean square error* is given by:

$$MSE = E\{e^2(t)\} = \int_{-\infty}^{\infty} e^2(t)dt = E\left(\left\|f - \hat{f}\right\|^2\right) \tag{3.8}$$

where E designates the integral expectation operator. Before presenting and discussing the transfer function that characterises the Wiener deconvolution filter, it is necessary to define a few functions (operations) that are incorporated in the final form of the Wiener transfer function.

In the case of the operation of self-convolution of a random function, if one term in the product is not reflected, the autocorrelation function is obtained instead as:

$$R_f(\tau) = f(t) * f(-t) = \int_{-\infty}^{\infty} f(t) f(t+\tau) dt \tag{3.9}$$

where * denotes convolution operation, τ denotes that the input signal is shifted in time by this constant amount and t represents time. The autocorrelation function is always even and has a maximum for $t = 0$; moreover, every function has a unique autocorrelation function. The Fourier transform of the autocorrelation function, $f(t)$, using equation (3.9), is:

$$P_f(s) = F\{R_f(\tau)\} = F\{f(t) * f(-t)\} = F(s)F(-s) = F(s)F^*(s) = |F(s)|^2 \tag{3.10}$$

where the superscript (*) represents the complex conjugate of a function and $F(s)$ represents the Fourier transform of a random function $f(t)$. The quantity described by equation (3.10) is called the *power spectral density function* or *power spectrum* of $f(t)$. For a function that is real, its power spectrum is real and even and the function $f(t)$ has a unique power spectrum. Another useful quantity, called *signal-to-noise power ratio* (SNR), as a function of frequency, is defined as:

$$SNR = \frac{|S(s)|^2}{|N(s)|^2} = \frac{P_s(s)}{P_n(s)}. \tag{3.11}$$

Using figure 3.1 to model the degraded image, the two-dimensional transfer function of the optimal deconvolution filter, using the MSE as an optimality criterion, is given by the following expression:

$$G(u,v) = \frac{H^*(u,v)P_f(u,v)}{|H(u,v)|^2 P_f(u,v) + P_n(u,v)} = \frac{1}{H(u,v)} * \frac{|H(u,v)|^2}{|H(u,v)|^2 + P_n(u,v)/P_f(u,v)} \tag{3.12}$$

where P_f (u,v) and P_n (u,v) represent the power spectra of the signal and the noise, respectively. The full derivation of the transfer function of the Wiener filter can be found in Hall (1979).

Analysing equation (3.12), it can be concluded that knowledge about the degradation function, $H(u,v)$, noise and signal power spectral density functions is required. If signal and noise are considered as good approximations for random variables, being ergodic random variables, their autocorrelation functions are known, as are their power spectra.

The degradation function, $H(u,v)$, can be measured from the degraded image itself, or is known *a priori.*

If the signal and noise are uncorrelated, an assumption that was made at the beginning of this chapter, this will reduce the transfer function of the Wiener filter to a more manageable form, in terms of computed power spectra, as:

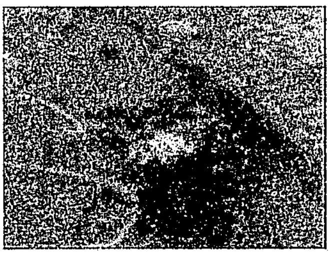
$$H_0(s) = \frac{P_s(s)}{P_s(s) + P_n(s)} \qquad for\ s \neq 0 \tag{3.13}$$

with $P_s(s)$ and $P_n(s)$, representing the power spectra for signal and noise, respectively.

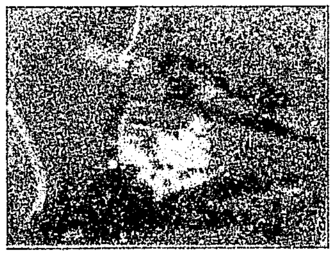
Despite the fact that the Wiener filter presents an optimal method for the deconvolution transfer function, it has three weak points. Firstly, the human eye has a tendency to be more tolerant to errors in dark areas and high-gradient areas than elsewhere, while MSE weights all errors equally, resulting in the image restored by Wiener filter being not very appealing for a human viewer. It is smoothed more than desired. Secondly, the classical Wiener deconvolution cannot deal with the case of a spatially variant blurring transfer function (for example, in the case of motion blur that involves rotation). The assumption made at the beginning of this chapter, that the analysis is conducted on point degradation systems is a desired one, under these circumstances. Thirdly, the technique cannot handle the case of non-stationary signals and noise. The fact is that most images are highly non-stationary, containing large flat areas, separated by sharp transitions. Even if images are non-stationary in a global sense, they can be assumed locally stationary. An image is stationary if the local computed power spectrum is the same (or almost the same) over the entire image (for example, the power spectrum of a white noise process is constant, see Shanmugan and Breipohl, 1988). This implies that the local power spectrum (calculated over a small window) changes slowly as one moves the window within the image, and this is the assumption made when the Wiener filter was applied in this work.

The original images were contaminated by a mixture of different types of noise of unknown statistical parameters and percentage, and possibly blurred. Therefore, it was decided to degrade them further, this time adding a simulated, controlled amount of noise and blurring, of fixed, known parameters. This was done with the intention to analyse the behaviour of different filters in the presence of a known type of noise and blurring. The Wiener filter, implemented using the algorithm described in *Matlab-Image processing toolbox* was applied to images "UKF.bmp" and "UKC.bmp". To simulate a spatially invariant blurring process, the images were first filtered using a Gaussian filter with main lobe width equal to 0.5 (the description of this type of filter is given in section 3.3.1). Afterwards, the images were degraded further by adding different types of nois *white Gaussian noise, salt and pepper noise* and *white speckle noise*, of known parameters. A short description of all types of noise used to degrade the images during different experiments is given in the following paragraph, together with their chosen statistical parameters. These values are kept the same for all experiments. A more detailed description of different types of noise can be found in Starck *et al.*, (1998, pp. 46-55).

White Gaussian noise can be described as a zero mean, stationary Gaussian random process *N(t)* with a power spectral density that is flat over a very wide range of frequencies (Shanmugan and Breipohl, 1988). The term *white* refers to the spectral shape of the process. A process that is not governed by a Gaussian distribution may also have a flat, white, power spectral density. For all experiments, the noise variance was chosen to have a numerical value equal to 0.01.



a)



b)

Figure 3.2 a) Image "UKF.bmp" blurred and degraded by white Gaussian noise
b) Image "UKC.bmp" blurred and degraded by white Gaussian noise

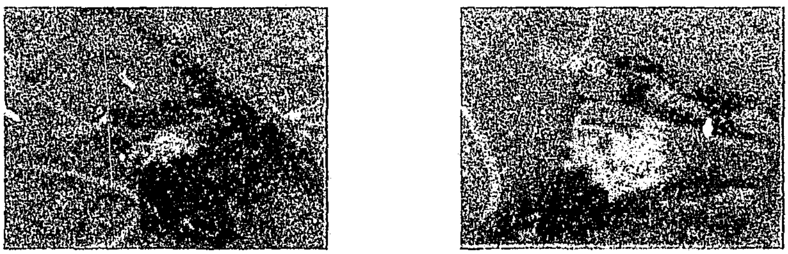
Salt and pepper noise consists of random occurrences of both black and white intensity values and was chosen of noise density equal to 0.05. This affects approximately a number of pixels equal to noise density x product (size of the degraded image). Figure 3.3 displays both images degraded by salt & pepper noise of the above parameters.

a) b)

Figure 3.3 a) Image "UKF.bmp" blurred and degraded by salt and pepper noise

b) Image "UKC.bmp" blurred and degraded by salt and pepper noise

Speckle noise added to the original images, first blurred by a rotationally symmetric Gaussian lowpass filter, is uniformly distributed, multiplicative random noise, with mean zero and variance 0.04. The original images degraded by the above-described operations are depicted in figure 3.4.



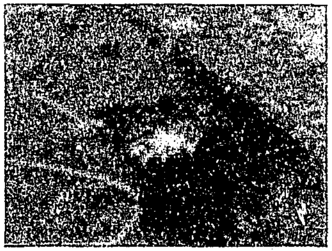
a) b)

Figure 3.4 a) Image "UKF.bmp" blurred, with speckle noise added

b) Image "UKC.bmp" blurred, speckle noise added

Once the images have been degraded, the Wiener filter can be used to remove different types of noise. Obviously, all the filters that will be described later in this chapter, including the Wiener filter, will act on the added noise and degradation due to blurring introduced on purpose, as well as on those degradations already existent in the original images. It seems that the original images are not degraded by blurring, but they contain a mixture of noise, similar to that described in section 3.1. Recall that they are optical images. This means that the original images will contain film grain noise, photoelectronic noise, electronic noise, etc.

The results of applying Wiener filter to degraded images are displayed in figure 3.5.



a) b)

Figure 3. 5 The results of denoising procedure, using the Wiener filter

a) Image "UKF.bmp" (blurred and degraded by Gaussian noise) with noise power of 0.0106

b) Image "UKC.bmp" (same degradation type), with noise power of 0.0115

The noise power estimated by Wiener filter that was calculated using a neighbourhood of size 3 x 3 represents the additive white Gaussian noise power, after Wiener filter was applied.

Let the original images, before degradation by blurring and noise be considered as reference images. A correlation coefficient equal to one can be associated to them, when compared with the degraded images. Other correlation coefficients, of values smaller than one, can be associated with the images degraded by different types of noise and blurred. Restoring the images using the Wiener filter will increase the previous correlation coefficients. The difference between the correlation coefficients before and after applying the Wiener filter can provide a good measurement of the improvement in the quality of images. Figure 3.6 displays a chart that is somehow controversial. It is known that the Wiener filter performs better in the presence of Gaussian noise. While this is true for one of the images, it is not for the other one, where the best result was obtained for removing speckle noise. Therefore, it can be concluded that the restoration process using this type of filter is data dependent and the assumption that the signal and noise are uncorrelated is not correct. Furthermore, it should not be forgotten that some of the noise contained in the original images was also removed. According to Weeks, Myler and Wenaas (1993), the process of filtering via a circularly symmetrical Gaussian noise generates the transformation of uncorrelated Gaussian noise into correlated Gaussian noise. The reader is asked to recall that this filter was used to simulate a blurring effect on the original images. Therefore, it acted upon the Gaussian noise contained intrinsically by the original images prior to furth. degradation. It can be concluded that one of the images, that is, image "UKC. bmp" was initially degraded by a higher amount of Gaussian noise than image "UKF. bmp". When a circularly symmetrical Gaussian filter was applied to the former image, it changed the initial Gaussian noise into correlated Gaussian noise. Hence, a poorer performance was obtained by applying the Wiener filter to this image (corrupted further by Gaussian noise) as compared to the image "UKF. bmp", which was degraded in the same manner. Normalised moments (described in section 4.2) up to the fifth order were calculated for the images, before and after applying the circularly symmetrical Gaussian filter. It was noted that a discrepancy in their numerical values, significant only at the level of second decimals, occurs. Therefore, for the purpose of this work, namely scene matching, it can be assumed that no radical changes occur in the probability distribution functions characterising the noise before and after applying the circularly symmetrical Gaussian filter.

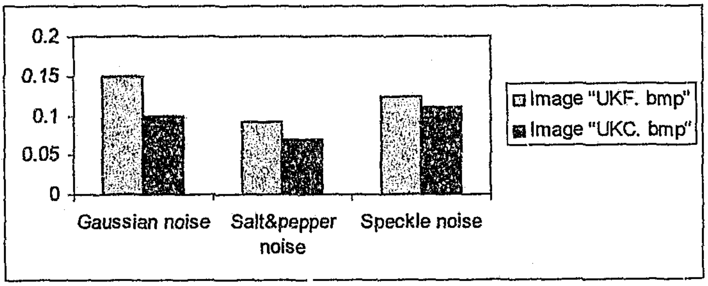


Figure 3.6 Improvement of the 'quality' of degraded images, expressed as a difference of correlation coefficients, after applying Wiener filter

Another way to assess the performance of different techniques used to remove noise is to compare the statistical parameters of the noise, before and after applying different filters. This is another use of Wiener filter: the ability to estimate different parameters for noise. If the signal and noise are uncorrelated, the local variance of the observed image is equal to the sum of the variances for signal and noise. This means:

$$\sigma_w^2(x,y)=\sigma_s^2(x,y)+\sigma_n^2(x,y) \tag{3.14}$$

where the variances for signal and noise are computed over a small local window centred on *(x,y)*. If it is assumed that the noise is locally white, then the power (mean square amplitude or variance) can be considered proportional to the local mean grey level. This implies that the noise power spectrum and variance are related in the following manner:

$$P_n(u,v,x,y) = P_n(0,0,x,y) = \sigma_n^2(x,y) = N_0 \mu_w(x,y) \tag{3.15}$$

where N_0 is a constant and $\mu_w(x,y)$ is the average grey level calculated over a small local window centred on *(x,y)*. More details about this topic can be found in Castleman (1996).

From the tests conducted, using images containing no signal and only noise of known parameters, it was established that this way of estimating the noise is quite an accurate procedure and will be used for one of the final algorithms that requires an estimation of noise. The Wiener filter is not the only way to remove the noise that degrades an image. More filters are presented in the following section.

3.3 Spatial filtering

When it comes to implementing filters, this can be performed either by applying spatial filtering or frequency domain filtering. In order to perform spatial filtering, the use of spatial masks is required, and the masks themselves are called *spatial filters*. Using equation (3.4), in the case of discrete functions, the output image generated by the input image convoluted with the impulse response of an LSI system, can be reformulated as:

$$h[i,j] = f[i,j] * g[i,j] = \sum_{k=1}^{n}\sum_{l=1}^{m} f[k,l]g[i-k,j-l]. \tag{3.16}$$

For the f and h images, the convolution becomes the computation of weighted sums of the image pixels. The impulse response *g[i, j]*, is called a *convolution mask*. When a mask is applied to an image, for each pixel *[i,j]*, the final value *h[i,j]* is calculated by translating the convolution mask to pixel *[i,j]* in the image, and then computing the weighted sum of the pixels in the neighbourhood of *[i,j]*. The individual weights are the corresponding values in the convolution mask. Convolution is a spatially invariant operation, since the same filter weights are used throughout the image.

The concept behind linear filters is that the transfer function and the impulse or point spread function (PSF) (the inverse of the optical transfer function is called the *point spread function*) of a linear system are inverse Fourier transforms of each other. Linear filters can be classified as lowpass, highpass and bandpass filters, depending on the frequencies that the filters allow to be passed without alteration. The filters that generated satisfactory results and were selected as the best methods to perform the denoising procedure are described in the following sections, together with the experimental results generated. The filters that are not used for pre-processing the images are mentioned briefly, for completeness. The mask size selected for all the filters tested was 3 x 3, 5 x 5, 7 x 7, 9 x 9 and 11 x 11.

Apart from linear filters, non-linear adaptive smoothing filters are used quite often. The popularity of adaptive noise filters may be attributed to several factors: ease of implementation, conceptual simplicity, edge retention in the imagery, applicability to pipeline processing, and computational efficiency when compared to Fourier methods (Mastin, 1985).

Smoothing filters are used for blurring and noise reduction. Various filters may be used to reduce different types of noise. Linear smoothing filters are good tools for removing Gaussian noise, as Gaussian processes are generally synonymous with linearity. In this case, the linearity is spatial, that is, the implementation of the filter uses the weighted sum of the pixels in successive windows. However, if the same pattern of weights is used in each window, this means that the linear filter is spatially invariant and can be implemented using a convolution mask. If the filter is not a weighted sum of pixels, it is a non-linear filter. Non-linear filters can be spatially invariant, too, if the same calculation is carried out, regardless of the position in the image.

3.3.1 Lowpass spatial filtering

Lowpass filters attenuate or even eliminate high-frequency components in the Fourier domain, keeping unchanged low frequencies, as the name of the filter suggests. High-frequency components characterise edges and other sharp details in a picture, so the effect of applying a lowpass filter is that the image will appear blurred, but, at the same time, a part of the noise will be removed. A lowpass filter requires that all the coefficients of the spatial mask have positive values, in order to follow the shape of a lowpass impulse response. Another requirement for the convolution mask is that the sum should be scaled in order to not exceed the valid grey-level range. For this class of filters, the impulse response is simply the average value under the mask, therefore, these filters are sometimes known in the literature as *neighbourhood averaging filters*. The local averaging operation will replace the value of each pixel with the mean of all the pixel values in the local neighbourhood, following the equation:

$$h(i,j)=\frac{1}{M}\sum_{(k,l)\in N} f[k,l] \tag{3.17}$$

where M is the total number of pixels in the neighbourhood, N. The size of the neighbourhood will dictate the amount of filtering, which means that the bigger the size of the filter, the better the degree of filtering. Conversely, the use of larger filters results in a loss of details contained in the image. The size of the mask cannot exceed that of the image, or area of interest.

A crucial question to be answered is how to judge the performance of a certain type of filter. The correlation coefficient is a measure of the closeness of a linear relationship between two variables (Snedecor and Cochran, 1980). Considering the original image as being matrix A and the image after applying the filter as matrix B, the correlation coefficient between these two matrices can be defined as:

$$r=\frac{\sum_{n_1}\sum_{n_2} A(n_1,n_2)B(n_1,n_2)}{\sqrt{\sum_{n_1}\sum_{n_2} A^2(n_1,n_2)\sum_{n_1}\sum_{n_2} B^2(n_1,n_2)}}, \tag{3.18}$$

It must be mentioned that both matrices must have the same size.

The original images may be considered as being characterised by a unit correlation coefficient, r. When two sets of images are compared, the set of the original images and the set formed by the degraded images, the degraded set can be characterised by a correlation coefficient r_d. The relation between the two coefficients is $r_d<r$. A filtered image can be described by another correlation coefficient, r_f, that is expected to be related to the previous ones as $r_d<r_f<r$.

An averaging filter was applied to images, with the expected result of smoothing some of the noise. Applying an averaging filter to the degraded images, resulted in an improvement in terms of image quality. However, other filters generated better results. Therefore, it was decided, based on experimental results, not to use this type of filter for denoising.

Another type of filter that was tested was a *rotationally (circularly) symmetric Gaussian lowpass filter*, with the main lobe width equal to 0.5. It can be classified as a linear smoothing filter whose weights are chosen according to the shape of a Gaussian function. These kinds of filters are very efficient for removing noise drawn from a normal distribution. The two-dimensional zero-mean discrete Gaussian function,

$$g[i,j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \tag{3.19}$$

is used as a smoothing filter, where the main lobe width σ represents the standard deviation (in pixels).

It was found through experimentation that Gaussian filters are efficient tools for removing speckle noise. Gaussian filters are also efficient at removing Gaussian noise, but not as good for the reduction of salt and pepper noise. When a Gaussian filter is applied to images degraded by different types of noise and blurred, it can be noticed that the performance decreases considerably. This phenomenon happens because the filter acting on a blurred image is acting as a point spread function, due to its chosen main lobe size. Due to excessive blurring generated by applying a Gaussian filter, it was decided not to use Gaussian filters for removing any type of noise. However, a rotationally symmetric Gaussian lowpass filter with main lobe width equal to 0.5 was applied to all the images used for scene matching tests, unless otherwise specified. This was done to generate new images simulating errors due to blurring.

Despite the fact that Wiener filter, averaging filter or Gaussian filter have been successful in removing Gaussian or speckle type of noise, they are not able to provide the best results for removing salt and pepper noise. It is well known from the literature (Castleman, 1996; Richards, 1986; Russ, 1992) that the specific filter that performs reasonably well in the presence of salt and pepper noise is the median filter. The experiments conducted in the following section confirm this statement. The median filter is discussed in the next section.

3.3.2 Median filters

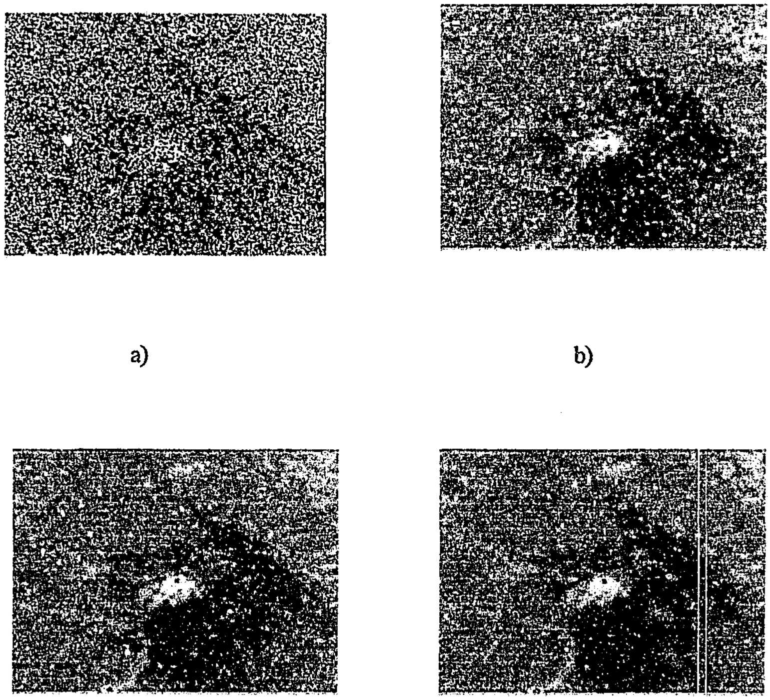
Another class of operations that combine neighbouring pixels may be characterised by comparing and selecting. They are called *rank value filters*. In order to achieve this goal, it is necessary to take all the grey values of the pixels that lie within the filter mask and sort them by ascending grey value. All rank value filters use this sorting technique. The filter operation that selects the medium value is called the *median filter*.

If the primary objective of using a filter is to remove noise, instead of blurring the edges, it is better to use a median filter rather than a lowpass filter. The approach taken by a median filter is to replace each pixel value with the median of the grey values in the local neighbourhood of that pixel. Median filters are very efficient in removing salt and pepper and impulse noise, and generally, strong, spike-like components of the noise, while preserving the sharpness of edges. Median filters work in successive image windows in a way similar to linear filters. However, the result cannot be referred to as a weighted sum, therefore they are not linear filters, and they do not work on the basis of a convolution operation (Jain, 1989). The principle behind a median filtering operation is as follows:

a) the values of the pixel and its neighbours, that are under the mask, are sorted in ascending order by grey level;

b) the median value is selected and assigned to the pixel.

The minimal degradation of edges in median filtering allows the method to be applied repeatedly. During this process the fine details are erased and large regions take on the same brightness values. However, the edges remain in place and are well defined. Figure 3.7a) shows a severely degraded image "UKF.bmp" treated with salt and pepper noise of density 0.5. Passing a median filter of mask size 3 x 3 over the image three times, has reduced the noise considerably, while retaining sharp edges, with negligible degradation of the image. The results are shown in figure 3.7b), c), and d).



a) b)

c) d)

Figure 3.7 Removal of salt and pepper noise using a median filter:

a) image "UKF.bmp" degraded by salt and pepper noise with density of 0.5

b) application of median filtering to image a) using mask size 3 x 3

c) application of median filtering to image b) using mask size 3 x 3

d) application of median filtering to image c) using mask size 3 x 3

The median filter is well suited for the removal of impulse-like noise because pixels corresponding to noise spikes are atypical of their neighbourhood and will be replaced by the most typical pixel (Russ, 1992). Median filters perform poorly when the noise is Gaussian, because in this situation there are no atypical noise pixels. Another type of median filter, and quite popular, is a *Tukey median filter*, a non-linear filtering technique that reduces noise without much image degradation (Castleman, 1996).

The median filter was applied to the same images, "UKF.bmp" and "UKC.bmp", degraded by different types of noise, having statistical parameters mentioned in section 3.2. The size of the mask was set between 3 and 11, with the result that a mask size of 3 x 3 has provided the best filtering, in a reasonable amount of time. If the mask size is chosen to take all integer numbers, it can be noticed that the results of a median filter will produce a saw-tooth shaped graph. The median filter, using even mask sizes, produces much poorer results than in the case of odd mask sizes. The reason is that by setting the mask size to even values, some of the median values may not be integers. Under these circumstances, the median filter will discard these values. Therefore for best results, the mask size must be an odd number.

To analyse the effect of this filter for different types of noise, the same technique of examining the correlation coefficients was used. The results are displayed in figure 3.8.

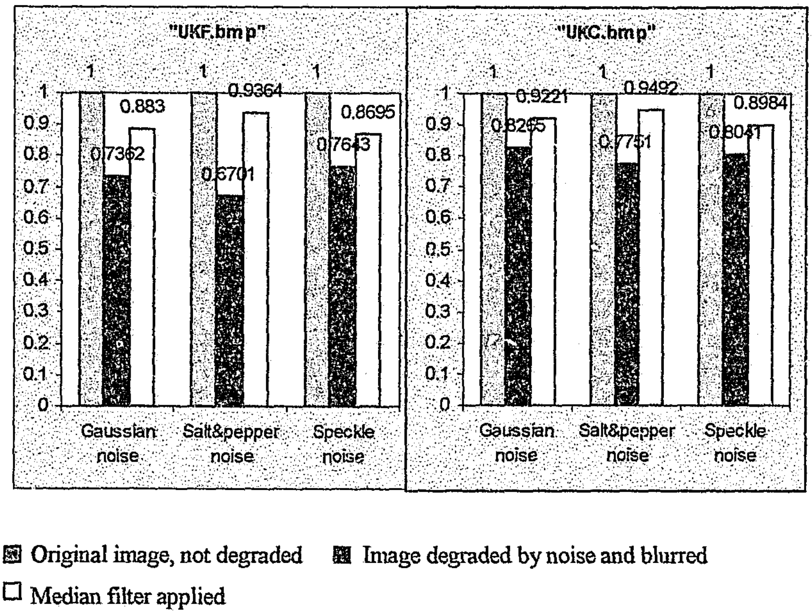


Figure 3.8 The results of applying a median filter of mask size 3 x 3 to degraded images

The reason median filters perform well in the presence of impulse-spike type noise can be explained by the fact that pixels distorted by transmission errors have lost their original grey values. While linear smoothing does not eliminate this erroneous information, but carries it on to the neighbouring pixels through the convolution process, rank value filters detect these atypical values and eliminate them.

Other types of filters sometimes used are *sharpening filters*. The primary goal of using sharpening filters is to highlight small details in a picture or to enhance details that have been blurred. Often, image sharpening is employed to pre-process images that are used for autonomous target detection in smart weapons. Because the filters chosen to denoise the degraded images have preserved the edges, it was not necessary to use sharpening filters.

3.4 Filtering in the frequency domain

An image can be viewed as a matrix of points with each point represented by a certain grey level. It can also be regarded as a combination of two-dimensional Fourier series with different frequencies. Thus, a picture can be represented uniquely in the frequency domain. It has been noticed (Duda and Hart, 1973) that low frequency Fourier components correspond to a homogeneous object and background, and high frequencies are introduced by the occurrence of a sharp edge or noise. Therefore, lowpass or highpass filters can be employed to smooth the picture, or to enhance the sharp edges.

The foundation of frequency domain techniques is the *convolution theorem*. The usual approach to developing the mathematical background of the Fourier transform begins with a one-dimensional waveform and then expands to two dimensions, that is an image. The Fourier transform and its properties are broadly described by several researchers (Castleman, 1996; Richards, 1993; Jain, 1989). Therefore, there is little need to concentrate on this topic. What is of more interest is to use concepts from the field of linear system theory and link them with the properties of the Fourier transform.

If one recalls that if $g(x,y)$ is an image formed by the convolution of an image $f(x,y)$ and a linear, position invariant operator $h(x,y)$, (Gonzales and Woods, 1992), then:

$$g(x,y) = h(x,y) * f(x,y). \tag{3.20}$$

Applying the convolution theorem, the frequency domain relation becomes:

$$G(u,v) = H(u,v)F(u,v) \tag{3.21}$$

where G, H, and F are the Fourier transforms of g, h, and f. There are many image enhancement problems that can be expressed in the form of equation (3.21). The procedure for enhancement in the frequency domain is straightforward. It is necessary to compute the Fourier transform of the image to be enhanced, multiply the answer by a filter transfer function, and take the inverse transform to produce the enhanced image.

In most cases, an image has the majority of its energy in the low- and mid-frequency of its amplitude spectrum. When the frequency is higher, the information of interest is very often embedded in noise. Hence the need to design a filter that reduces the amplitude of high-frequency components, thereby reducing the visible effect of noise.

From equation 3.21 it can be seen that $F(u,v)$ is the Fourier transform of an image to be smoothed. The procedure is to select a filter transfer function $H(u,v)$ that generates $G(u,v)$, by attenuating the high-frequency components of $F(u,v)$. The inverse transform will produce the desired smoothed image $g(x,y)$.

A broad description of lowpass filters, bandpass and bandstop filters, as well as high frequency filters can be found in Castleman, (1996), Gonzales and Woods, (1992). Amongst all the filters described, which one should be used? Those filters operating in spatial domain or those operating in frequency domain? A further consideration relates to computer processing time, which is a key requirement for the solution to scene matching problem offered by this dissertation. The analysis done by Richards (1986) will justify the choice taken during this work.

In both cases, the frequency domain process and spatial filtering are performed using only sets of multiplication and addition. Therefore, in order to evaluate the computer processing time, it is sufficient to consider a comparison using the number of multiplications and additions involved in achieving the desired result. It is not necessary to analyse the additions, because they are faster than multiplications for most computers. If, to an image of size $k \times k$ pixels, a template of size $M \times N$ is applied, the total number of multiplications necessary to move this template over the image, for every image pixel is:

$$N_C = MNK^2. \tag{3.22}$$

The number of multiplications required in the frequency domain is:

$$N_F = 2K^2 \log_2 K + K^2. \tag{3.23}$$

Under these circumstances, the cost comparison gives:

$$\frac{N_C}{N_F} = \frac{MNK^2}{2K^2 \log_2 K + K^2} = \frac{MNK^2}{K^2(2\log_2 K + 1)}. \tag{3.24}$$

If this figure (cost comparison coefficient) is less than unity, it is more efficient to perform the enhancement using spatial filtering procedure by using a template operation. Of course, this brief comparison does not take into account program overheads, but it is a reasonable rough estimation of the time involved in applying each of the two approaches. Table 3.1, reproduced from Richards (1986), reflects a number of values of N_C/N_F for various image and template sizes.

Table 3.1 Time comparison of geometric enhancement by template operation compared with the Fourier transformation approach - based upon multiplication count comparison - described by equation (3.24)

Template size					
	3 x 3	3 x 5	5 x 5	5 x 7	7 x 7
Image size	Cost comparison coefficient				
16 x 16	1.00	1.67	2.78	3.89	5.44
64 x 64	0.69	1.15	1.92	2.69	3.77
128 x 128	0.60	1.00	1.67	2.33	3.27
256 x 256	0.53	0.88	1.47	2.06	2.88
512 x 512	0.47	0.79	1.32	1.84	2.58
1024 x 1024	0.43	0.71	1.19	1.67	2.33
2048 x 2048	0.39	0.65	1.09	1.52	2.13
4096 x 4096	0.36	0.60	1.00	1.40	1.96

Source: Richards (1986)

The conclusion that can be drawn from the above table is that the bigger the size of the template, and the smaller the size of the image, the bigger the cost comparison coefficient, described by equation (3.24). The images used during this work are usually 256 x 256 or smaller and the spatial filters applied, with good results, were of mask size of 3 x 3 or 5 x 5. Therefore, the cost comparison coefficient has a value less than unity. These results, definitely decide in favour of choosing filters in spatial domain, in order to remove noise, or enhance different desirable features (edges, for example). Bearing in mind all the requirements and constraints of this work, the template approach was chosen; from the previous experiments it can be noticed that 3 x 3 templates are sufficient to achieve the desired results.

3.5 Wavelet transforms

In the preceding sections of chapter 3, different types of filters were investigated and applied. The result was that some filters provided better results than others, in the presence of a certain type of noise. But, it is possible to find a procedure that will provide reasonably good results in terms of reducing the noise, while preserving the sharp features (edges), for all types of noise. This procedure uses *wavelets transform*. They have proved to possess powerful abilities for denoising and compressing an image, whilst retaining the sharp features of the image.

3.5.1 Overview of continuous and discrete wavelet transforms

A wavelet is a waveform of effectively limited duration that has an average value of zero (Misiti *et al.*, 1996). Wavelet transform, in broader terms, can be defined as a new multiresolution decomposition tool for continuous and discrete time signals. The kernel of wavelet transform is obtained by dilation and translation of a prototype (mother) bandpass function. Using wavelet transforms a signal can be decomposed into the sum of a lower resolution (coarser) signal plus a detail, similar to the dyadic sub-band tree in the discrete time case. Further, each coarse approximation can be decomposed again into a coarser signal and a detail signal at that resolution level. According to this approach, the signal can be represented by a lowpass (coarse) signal at a certain scale (corresponding to the level of the tree) and a sum of detail signals at different resolutions. The wavelet analysis, during this work, will be restricted to transform real valued, measurable, square integrable functions of two-dimensions.

A basic objective in signal analysis (one-dimensional signal, 1D, or a two-dimensional one, 2D) is to use a transformation that represents the signal characteristics in time and frequency simultaneously. Using Fourier analysis one can break down a signal into sinusoids of different frequencies. Similarly, wavelet analysis performs a decomposition of the signal into shifted and scaled versions of the original wavelet. In other words, Fourier analysis is a mathematical technique for transforming the description of a signal from a time-based one to a frequency one. Through Fourier analysis it is possible to decompose the signal into frequency components and determine the relative strength of each component, but it does not show when the signal exhibits a particular frequency characteristic. If the signal is a stationary one, Fourier analysis provides all the necessary information. But, most of the real world signals, including images, are non-stationary or transient; hence the need for another transformation that would be able to provide needed information about the behaviour of a particular signal in time. For this type of signal Fourier analysis is not suitable.

Gabor (1946) tried to reach a compromise between the time- and frequency-based representation of a signal. The technique, known as *short-time Fourier transform* (STFT), moves a fixed-duration window over the time function and extracts the frequency content in that interval. The precision of this approach is limited by the size of the window and once the size of time window is chosen it is kept constant for all frequencies. This procedure could be suitable for signals that are locally stationary, but globally nonstationary (Akansu and Haddad, 1992). A windowing technique using a variable size window would correct this drawback. Wavelet analysis allows the use of a variable size window, depending on the information wanted.

Before introducing more concepts, it is necessary to clarify the notation used for this section. The terms Z, R and R^+ denote the sets of integers, real numbers and positive real numbers, respectively. $L^2(R)$ represents the Hilbert space of measurable, square integrable functions. The notation $f(x) \in L^2(R)$ means that the function is of finite energy:

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty. \tag{3.25}$$

In wavelet analysis a set of basis functions are created by dilating and translating a prototype function, $\psi(x) \in L^2(R)$, which is called a basic wavelet. This is an oscillatory function, usually centred upon the origin, which quickly converges to zero as $|x| \to \infty$. If the basic wavelet $\psi(x)$ is translated $\psi(x-b)$ and scaled $\frac{1}{\sqrt{a}}\psi\left(\frac{x}{a}\right)$, similarly to the procedure used for FT, this will generate a set of 1D-wavelet basis functions as:

$$\psi_{a,b}(x)=\frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right) \qquad \text{with } a\in R^{+} \text{ and } b\in R. \tag{3.26}$$

Depending on the scale parameter, *a*, the wavelet function $\psi(t)$ dilates or contracts in time, inducing the corresponding dilation or contraction in the frequency domain. Hence, the wavelet transforms provide a flexible time-frequency resolution. Therefore, the forward *continuous wavelet transform* (CWT) of *f(x)* with respect to the wavelet $\psi(x)$ can be expressed as the inner product:

$$W_f(a,b)=\langle f,\psi(a,b)\rangle=\int_{-\infty}^{\infty} f(x)\psi_{a,b}(x)dx. \tag{3.27}$$

The inverse continuous wavelet transform is:

$$f(x)=\frac{1}{C_\psi}\int_{0}^{\infty}\int_{-\infty}^{\infty} W_f(a,b)\psi_{a,b}(x)db\frac{da}{a^2}. \tag{3.28}$$

Generally, the forward transform does the decomposition by determining the coefficients. The inverse transform does the reconstitution by summing the basis functions, weighted by those coefficients. The CWT $W_f(a,b)$ of 1D function *f(x)* is a function of two variables. In the same manner, for functions of two variables, this transform will increase the dimensionality by one. For a function of two variables it can be defined in the same way, that is, in terms of its direct and inverse CWT. More details concerning this topic can be found in Castleman (1996). The coefficients of CWT are the time-scale view of the signal, therefore the goal to express the signal characteristics, in terms of time and frequency (scale, actually), was reached. The advantage of using CWT is that it can operate at any scale, depending on the issue to be solved. But, the problem with CWT is that it has two weak areas: redundancy and impracticality (Akansu and Haddad, 1992). Hence, the need to develop another transform, which is able to handle the signal in a more efficient way.

The solution to these drawbacks of CWT is *discrete wavelet transform (DWT)* which is obtained by sampling the parameters a, b to obtain a set of wavelet functions in terms of discrete parameters.

When a basic wavelet is scaled and translated to form a set of basis functions, the scale and translation factors are specified by integers and not real numbers. The basis functions are formed by binary scaling (shrinking by factors of two) and dyadic translations (a shift by the amount $k/2^j$) of the basic wavelet. This results in DWT being more restrictive than CWT.

A function $\psi(x)$ is an orthogonal wavelet if the set $\{\psi_{j,k}(x)\}$ of functions defined as:

$$\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi\left(2^j x - k\right) \quad \text{with } -\infty < j,k < \infty \text{ and } j,k \in Z \tag{3.29}$$

forms an orthonormal basis of $L^2(R)$, with j representing the dilation and k the translation. If the function $f(x)$ and the basic wavelet are restricted to be zero outside the interval [0,1], the family of orthonormal basis functions can be written using a single index, n, instead of j and k:

$$\psi_n(x) = 2^{\frac{j}{2}}\psi\left(2^j x - k\right), \text{ assuming that } \psi_0(x) = 1 \tag{3.30}$$

where j and k are connected to n as $n = 2^j + k$, with j the largest integer such that $2^j \leq n$, $j=0,1,\ldots$, $k=0,1,\ldots,2^j-1$. The inverse transform is defined as:

$$f(x) = \sum_{n=0}^{\infty} c_n \psi_n(x). \tag{3.31}$$

The transform coefficients are given by the inner product:

$$c_n = \langle f(x), \psi_n(x) \rangle = 2^{\frac{j}{2}} \int_{-\infty}^{\infty} f(x)\psi\left(2^j x - k\right)dx. \tag{3.32}$$

Everything can now be linked using multiresolution signal analysis, in which the function $f(x) \in L^2(R)$ must be seen as a limit of successive approximations. Each successive approximation is a smoothed version of $f(t)$, corresponding to different resolutions. The smoothing is realised by convolution with a lowpass kernel, called the scaling function, $\phi(t)$. From this point one can develop a DWT. It is necessary to define a discrete lowpass filter impulse response, $h_0(k)$, (sometimes called a scaling vector) that obeys certain conditions, see Castleman (1996). From the impulse response, the scaling function, $\phi(t)$, can be generated.

Furthermore, an ideal halfband bandpass filter, $h_1(k)$, can be constructed, and ultimately, the basic wavelet, $\psi(t)$. If the scaling vector has a finite number of nonzero entries, then the scaling function, the basic wavelet and the resulting wavelets will have compact support, meaning that they are zero outside a short interval on the time axis. The scaling function can be defined in the following manner:

$$\phi(t) = \sum_k h_0(k)\phi(2t-k) \tag{3.33}$$

and can be built as a weighted sum of half scale copies of itself, with $h_0(k)$ as weights. With the scaling function and scaling vector defined, a discrete highpass impulse response, called the *wavelet vector*, can be expressed as:

$$h_1(k) = (-1)^k h_0(-k+1). \tag{3.34}$$

This was the last step before defining the basic wavelet:

$$\psi(t) = \sum_k h_1(k)\phi(2t-k) \tag{3.35}$$

that is, the orthonormal wavelet set, described by equation (3.29). Therefore, the design of the wavelet is a process involving few steps; first finding a sequence of $h_0(k)$, then constructing the corresponding scaling function. It is possible to design a wavelet the other way around, choosing an orthonormal scaling function and determining $h_0(k)$, then calculating the wavelet vector using equation (3.34) and finally the basic wavelet as in equation (3.35). For a more detailed treatment of multiresolution analysis, the reader should consult Akansu and Haddad (1992).

3.5.2 Wavelet analysis

The theoretical foundation of wavelets, presented before, has paved the way for a more explicit approach in dealing with the practical side of implementation of wavelets and the way they operate. Most of the content of the images is localised in the low- and mid-frequency components. High frequency components describe small details, edges and noise. If the image can be decomposed into low- and high-frequency components, high-frequency components filtered, and afterwards the image reconstructed, this will eliminate most of the noise, with a secondary effect of slightly blurring the edges.

In wavelet analysis the procedure is based on approximations and details. The approximations are the high-scale, low-frequency components of the signal, while the details represent the low-scale, high-frequency components. Figure 3.9 depicts the graphical representation of the filtering process, using wavelets, in its simplest form.

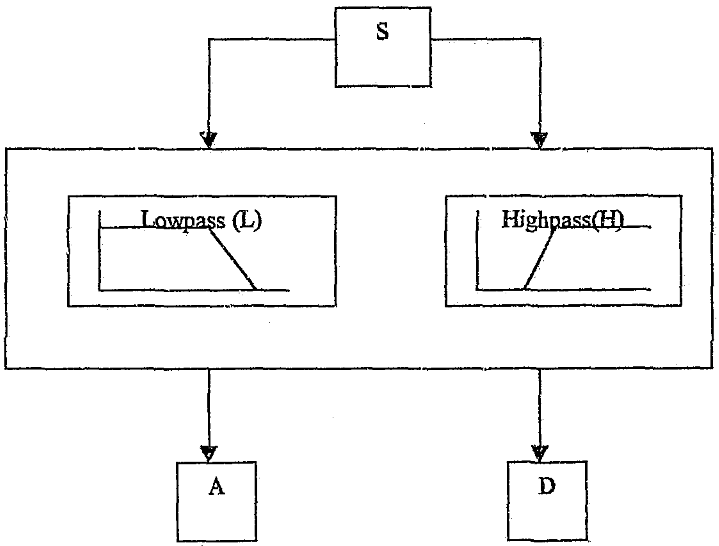


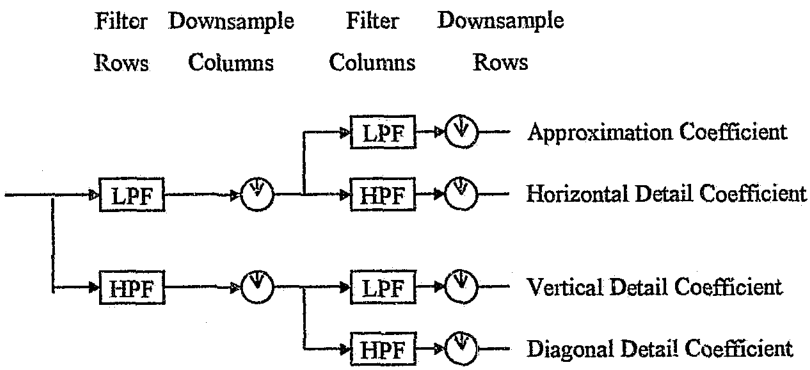
Figure 3.9 Graphical representation of one decomposition level of the signal

If this procedure is implemented, this filtering will unfortunately increase the size of the data. This problem can be overcome if downsampling (known also as subsampling or decimation) is performed. The result is that every second sample of the data is discarded. This can be modelled as multiplying the signal by a subsampling function first, with the effect that odd-numbered samples are set up to zero, and then discarding the odd-numbered samples. This procedure will result in no loss of information, if performed correctly. When it is necessary to recover the signal, upsampling is performed with the effect that the encoded low-band signal is changed by inserting the zero-valued odd numbered samples. Practically, the process depicted by figure 3.9 provides DWT coefficients. What is important is the fact that the detail coefficients (cD) contain mainly noise, while the approximation coefficients (cA) contain much less noise than the original signal. The decomposition process can be iterated, with the effect that successive approximation coefficients will be decomposed at different lower resolution levels, a procedure known as wavelet decomposition tree. The choice of the number of decomposition levels should either be based on the nature of the signal, or on a suitable criterion. It must not be ignored that a larger number of decomposition levels places a computational burden on the system.

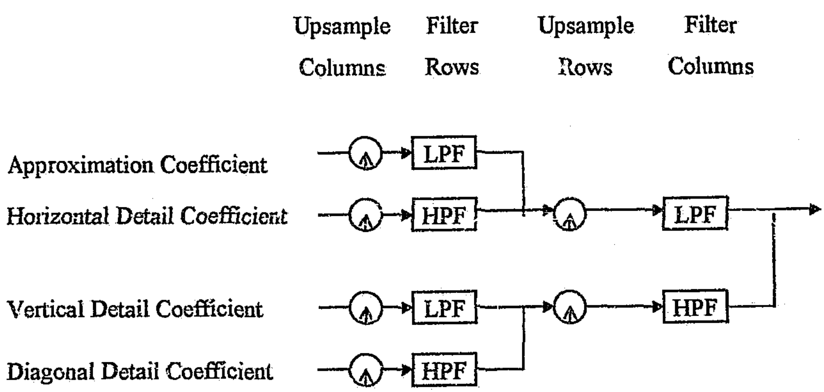
The discrete wavelet transform can be used to analyse or decompose signals. After the analysis or denoising is performed, the signal must be reconstructed without loss of information. This process is called reconstruction or synthesis and the mathematical procedure that performs this operation is known as the *inverse discrete wavelet transform* (IDWT). The signal is reconstructed from the wavelet coefficients through upsampling and filtering. The perfect reconstruction of the signal is possible by a careful choice of the filters used for the decomposition and reconstruction phases. This approach allows not only a perfect reconstruction, but, it, also, determines the shape of the wavelet used to perform the analysis.

The low- and highpass decomposition filters (L and H) together with their associated reconstruction filters (L' and H') form a system called *quadrature mirror filters*. The wavelet waveform that will reconstruct the signal accurately must have a shape dictated by the quadrature mirror decomposition filters. While the wavelet function $\psi(x)$ is determined by the highpass filter, which produces the details of the wavelet decomposition, the lowpass quadrature mirror filters determine the scaling function $\phi(t)$. They are also associated with the approximations of the wavelet decomposition.

Therefore, there are three steps involved in wavelet analysis: decomposing the signal to obtain the wavelet coefficients, modifying the coefficients according to the desired aim (compressing or denoising) and assembling the signal back from the coefficients. For the 2D wavelet transform there are three wavelet functions corresponding to the horizontal, vertical and diagonal detail coefficients and one scaling function corresponding to the approximation coefficients. The direct and inverse 2D wavelet transform pair is represented graphically in figure 3.10.



a) Decomposition phase (DWT)



b) Reconstruction phase (IDWT)

Figure 3.10 Wavelet analysis performed on a 2D signal at level one

a) decomposition phase (DWT)

b) reconstruction phase (IDWT)

The same procedure can be applied for each level, according to how many levels are chosen.

Applying wavelet analysis, the signal is split into approximation and detail coefficients. The approximation is then split again into a second level approximation and detail, and the process is continued at each level. Sometimes, there are not the approximation coefficients that contain the significant information, but rather the detail coefficients. This is the reason why it is desirable to decompose both the detail and approximation coefficients at each level. This goal can be achieved using wavelet packet analysis that is a generalisation of wavelet analysis. The experiments conducted using wavelet packet analysis, which resulted in extra computation, instead of wavelet analysis, did not provide better results in terms of denoising the image. Therefore, wavelet packet analysis was not pursued further.

3.5.3 Wavelet denoising procedure

The noise contained in an image can be described using the model proposed by Misiti *et al.*, (1996):

$$s(i,j) = f(i,j) + \sigma e(i,j) \qquad \textit{with } i,j=0,\dots,m-1 \tag{3.36}$$

where $e(i,j)$ is a white Gaussian noise of variance one and the noise level, σ, is supposed to be equal to one. The goal is to eliminate the noise part of the image $s(i,j)$ and recover the original image $f(i,j)$. There are a few standard steps that must be applied when an image is denoised, and they are listed below:

a) decompose the image; this is performed choosing a wavelet, the number of levels for the decomposition, followed by decomposing the image at the chosen level;

b) threshold detail coefficients, deciding what type of thresholding is best suited for each level;

c) reconstruct the image based on the original approximation coefficients and the thresholded detail coefficients of levels from 1 to N.

At this point a few questions must be answered: what wavelets are the best to use, which type of threshold should be chosen and how many decomposition levels are required? The wavelet basis must be chosen to best represent the signal. Compact support wavelets result in *finite impulse response* (FIR) filters for the DWT and good space localisation. A wavelet with N vanishing moments is able to suppress a polynomial of order N. This means that if a signal is modelled as an Nth order polynomial representing useful signal and noise and is decomposed with a wavelet that has N or more vanishing moments, then the polynomial portion of the signal will not affect the detail coefficients. Also, it is desirable to use a symmetric analysing wavelet for images, because DWT filters will have a linear phase. It is possible to use one wavelet for decomposition and another one for reconstruction. The advantage is that they possess different properties, with the condition that they be biorthogonal for perfect reconstruction. The two different wavelet bases, $\psi(x)$ and $\tilde{\psi}(x)$ are dual to each other and the wavelet families $\{\psi_{jk}(x)\}$ and $\{\tilde{\psi}_{jk}(x)\}$ are biorthogonal:

$$\left\langle \psi_{jk}, \psi_{lm} \right\rangle = \delta_{jl}\delta_{km}. \tag{3.37}$$

This results in the following functions for decomposition phase:

$$c_{j,k} = \left\langle f(x), \tilde{\psi}_{j,k}(x) \right\rangle \text{ and } d_{j,k} = \left\langle f(x), \psi_{j,k}(x) \right\rangle \tag{3.38}$$

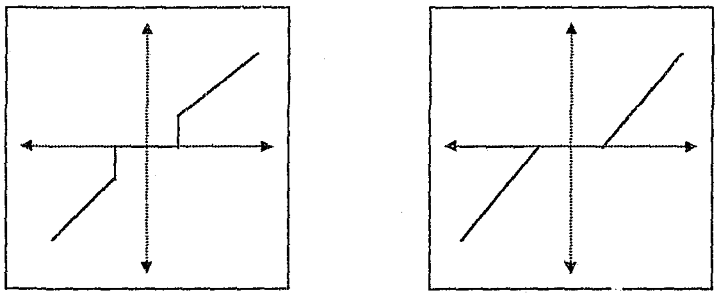
and for the reconstruction phase:

$$f(x) = \sum_{j,k} c_{j,k}\psi_{j,k}(x) = \sum_{j,k} d_{j,k}(x)\tilde{\psi}_{j,k}(x). \tag{3.39}$$

The biorthogonal wavelets transform permits the use of symmetric (even or odd) wavelets having compact support. Regarding the number of levels used, it was already mentioned that the bigger the number of levels, the better the signal representation, but, at the same time, more computation is required. Generally, increasing the number of levels results in a higher signal to noise ratio, because the higher frequencies are removed. Experimentally, it was determined that a number of levels between 3 and 5 generate a good balance between the amount of noise removed and the computational demand.

Through trial and error, it was established that using four levels of decomposition provide the best results.

There are two types of threshold: hard and soft. Hard thresholding can be described as setting to zero the elements whose absolute values are lower than the threshold. Soft thresholding, usually used for denoising, is similar to hard thresholding, with the difference that all the coefficients above the threshold are shifted towards zero, avoiding the discontinuities in the wavelet coefficient values. The two types of thresholding are represented in figure 3.11.



a) b)

Figure 3.11 Different types of thresholding:

a) hard thresholding;

b) soft thresholding

The next step is to practically implement the denoising algorithm. Only the detail coefficients should be thresholded. This is because these coefficients contain the higher frequency components of the signal, including noise. Wavelet thresholding relies on the assumption that the noise power is significantly smaller than the signal power. This means that most of the noise resides in the small wavelet coefficients. By setting these coefficients to zero, the noise is removed without damaging larger wavelet coefficients that represent image details. Conventional image restoration techniques, generally, result in blurring the image, while a wavelet restoration preserves most of the image edges. This is important for this work because the image details are required to uniquely

This is important for this work because the image details are required to uniquely describe the real image features and form a sharp correlation peak. If the threshold is set up too high, then wavelet coefficients representing the signal will also be removed and this will result in damaging the image details. The threshold selection rule is important because there is no human intervention to monitor the image denoising algorithm in the process of automated scene matching. The threshold selection rule should be robust to changes in image features and noise.

Starck and Murtagh (1994) describe a thresholding technique that satisfies these requirements. Standard denoising algorithms use the same threshold for each detail level and orientation. However noise behaviour in the wavelet space is dependent on the analysing wavelet and the detail level because alterations caused by the DWT filters mean that the noise in the wavelet coefficients is coloured. This technique begins by modelling the behaviour of noise in the wavelet domain. This is done by performing the DWT on a noise image of zero mean and unit variance. The standard deviation of the detail coefficients is then found as a function of level and orientation (horizontal, vertical and diagonal), so that there is a standard deviation curve for each orientation, $N(n,o)$ (where o is the orientation and n is the level number).

This model does not need to be calculated each time the denoising algorithm is used. The model is dependent only on the analysing wavelet and can be found and stored before the final stage of the scene matching algorithm. Once the unit variance model is known, it can be modified to describe the behaviour of the noise that contaminates the real image in the wavelet domain. This is done by scaling the model values by the standard deviation of the real image noise (this can be robustly estimated from the standard deviation of the first level detail coefficients).

$S(i,o) = (\sigma_o)N(i,o)$, where σ_o is the standard deviation of the real image noise in orientation o.

Now a threshold $T(i,o)$, for each level and orientation can be chosen as a scaled version of the noise standard deviation model of that level and orientation as $T(i,o) = (k)\, S(i,o)$.

k was chosen being equal to the value 1.2 to remove approximately 75% of the noise. The choice of k is a trade-off between noise removal and detail removal. This threshold selection technique was tested on a number of different images and the denoising results were found to be consistent.

This wavelet denoising algorithm was applied to both, the sensed and template images, before performing the correlation. Thresholding inevitably removes some of the signal. The threshold removes similar signal portions from the sensed and template images, and thus one obtains two similar images with a minimum of information loss. The threshold values used for the template image must be the same as the values found for the scene image.

Two wavelet families were selected for denoising because of their properties. The *Symlet wavelets* with three vanishing moments, *sym3* were used due to their near-symmetry properties and compact image representation properties. Amongst the biorthogonal wavelet family, the pair *bior1.3* provided the best results, in terms of denoising and computational time. This set of wavelets is characterised by linear phase, needed for image reconstruction. It is known that symmetry and exact reconstruction are incompatible, if the same FIR filters are used for decomposition and reconstruction (Misiti *et al.*, 1996). That is why two wavelets are used, instead of one, (refer to equations (3.38) and (3.39)), one for analysis and the other for synthesis.

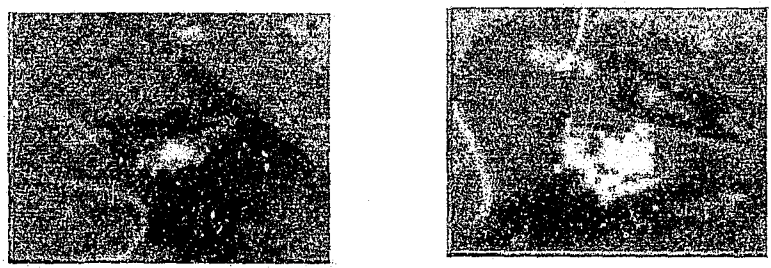
Also, the wavelets are related by duality as:

$$\int \tilde{\psi}_{j,k}(x)\psi_{j',k'}(x)dx = 0 \text{ , as long as } j \neq j' \text{ and } k \neq k' \tag{3.40}$$

and also:

$$\int \tilde{\phi}_{0,k}(x)\phi_{0,k'}(x)dx = 0 \text{ , as long as } k \neq k' \tag{3.41}$$

Figure 3.12 displays the results of denoising the original images, blurred and degraded further by adding white Gaussian noise. The pair of *bior 1.3* wavelets was used, four decomposition levels were selected, soft thresholding was applied and $k=1.2$



a) b)

Figure 3.12 Denoised images using "bior1.3" wavelets pair

a) image "UKF.bmp" initially degraded, after denoising using wavelets

b) image "UKC.bmp" initially degraded, after denoising using wavelets

It can be noticed that the edges were quite well preserved during denoising process. For the first image an improvement, in terms of correlation coefficients between the original image and the reconstructed one, of 13 % was achieved, while for the second image an improvement of 10 % was obtained. Expressing this in a different way, for the degraded image "UKF.bmp" the noise power decreased from 0.0106 to 0.0029, whilst for the degraded image "UKC.bmp" the noise power was reduced from 0.0115 to 0.0037 (noise power was calculated using Wiener filter). The difference in results between using Symlet wavelets or biorthogonal wavelets is not significant, while the computational time is more or less the same, depending on each image. It was noticed from the experiments conducted that wavelets do not handle images degraded by salt and pepper type of noise very well.

3.6 Choice of denoising technique

After extensive experiments regarding the choice of the best procedure that must be employed to denoise an image, a documented decision can now be made. Two approaches can be taken.

If the type of noise that degraded an image is known, then the filter that delivered the best results for that type of noise should be used. If the noise and degradation of the image are not known, or a mixture of different types of noise degraded the signal, then the procedure that best preserves edges and that is computationally least demanding should be implemented.

Different sensors introduce various types of noise when an image is recorded. It is known that in the case of *forward looking infrared images* (FLIR) the dominant noise that degrades the useful signal is salt and pepper type of noise. Therefore, it is advisable to use a median filter to denoise the image due to its specific properties as described in section 3.3.2. Ratches *et al.,* (1997) also recommend a median filter being best suited for this type of noise.

If the sensor used to record an image is acting in the microwave range of wavelengths, for example, using the technique of *side-looking airborne radar* (SLAR) or *synthetic aperture radar* (SAR), it is known that the noise that corrupts the signal is mostly speckle noise. The reader interested in more details about the above mentioned types of sensors should refer to Richards (1986). In this case, Wiener filter or wavelets are the best techniques to reduce the noise, while preserving the edges.

Optical images are usually degraded by noise following a Gaussian or Poisson distribution. The noise incorporated in the useful signal is Gaussian noise, Poisson noise or a combination of both. The noise was modelled as Gaussian noise. Wiener filter, especially if the image is blurred, or a denoised algorithm based on wavelets transform are the best choices.

A short description of different sensors, together with their performance characteristics, can be also found in Bhanu and Jones (1993).

If the noise affecting the signal is unknown or a mixture of diverse types of noise, the best choice is to employ wavelets for denoising the signal, with the desired characteristics of preserving the edges and being a low computational burden for the system. The same range of results was obtained for other degraded images, using the above selection of filters and denoising techniques.

4 Feature selection

It was stated in the introduction that the goal of this work is that a number of designed algorithms distinguish between objects of different types and perform the matching between a set of objects, embedded in a natural scene, and a chosen template. Therefore, it is vital to decide which characteristics of the objects must be measured to produce descriptive parameters that will best describe the features of various objects. The representation and description of objects or regions is a compulsory step in the process of automatic scene matching and the quality of the features is of paramount importance for the success of scene matching algorithms.

A proper selection of the features is important, since only these features will be used to identify the objects. The intuitive idea that as many features as possible may be the best answer is erroneous. As long as the descriptors do not reveal independent properties of the same object, the only result will be an increased computational time without improved performance of the system. Therefore, one must seek a small set of reliable, independent and discriminating features.

Feature selection can be viewed as the process of eliminating some features (those that do not possess a high degree of generality) and combining those that are related, until the feature set becomes manageable, and the performance achieved is adequate.

The choice of a particular descriptive method over another one is dictated solely by the problem under consideration. It is necessary to identify descriptors that are able to illustrate general, but, at the same time, essential differences between different objects, and provide a robust stability to changes in location, size and orientation, without being strongly affected by spurious contamination of the signal image. It is, therefore, highly desirable that the features selected display properties of algebraic and geometric invariance. Another requirement is that the resultant feature vectors must be characterised by low dimensionality, such that they do not place a burden on the system, in terms of computational time and storage capacity.

Myers (1996) proposed a classification of the features selected to describe an object or the content of a scene in four categories:

1. visual features;
2. transform features;
3. statistical features;
4. algebraic features.

4.1 Short presentation of different descriptors

Visual features are intuitive and obvious to human inspection of an image. They characterise the form of the object and are known, usually, as regional or shape descriptors. All the information about the shape of an object is contained in its boundary pixels. Human intervention is eliminated, therefore the shape descriptors must be described or approximated, mathematically, in such a manner as to make them suitable for computer use. Also, the choice must be made such that these visual features should not be restricted to a certain type of image. They may be extracted from any type of image such as indexed, intensity, binary or RGB images.

Morphological operators work on binary images represented by a matrix. Despite the fact that binary images can be stored in a much more compact form, the images used during this work are grey level images, because they provide a more versatile and richer support, suitable for natural scenes. Another representation of the boundary of a binary object on a discrete grid, known as *chain code*, was considered. It is also suitable for binary images, and it is translation invariant, but neither rotational, nor scale invariant. Therefore, morphological operators were not considered suitable for this work, due to the types of images used.

Elementary geometrical parameters such as *area*, defined as the number of pixels contained within the boundary, and *perimeter*, measured as the length of the boundary of an object or region were discarded because they are not invariant under scale changes, even if they are independent of the orientation of the object. The *compactness* or *circularity* of a region, described as the quotient between the squared perimeter and area, was not suitable because, although it is insensitive to scale changes, it introduces rotational errors.

At the first glance, the *principal axes* of a region appear to be a more suitable option. They represent the eigenvectors of the covariance matrix calculated by using the pixels within the region as random variables (Gonzales and Wintz, 1987). The maximal region spread is given by the two eigenvectors of the covariance matrix and the measure of the degree of spread is contained in the corresponding eigenvalues. This descriptor is insensitive to rotation, but not to scale changes. One way to overcome this drawback is to use the ratio of the large to small eigenvalues as descriptors. Other measures used as region descriptors are the mean and median of grey levels, strongly affected by intensity changes. The success of most of the above descriptors relies heavily on a very accurate and successful segmentation process. If this condition is not accomplished, the desired results cannot be reached.

In the same category, *topological descriptors* should be mentioned. The number of holes, H, and connected components, C, in a figure are used to define the *Euler number, E:*

$$E = C - H. \tag{4.1}$$

The *Euler number* is used to describe a binary image and is defined as the total number of objects in the image minus the number of holes in those objects. It can be calculated using either four or eight connected neighbourhoods. This descriptor could be a perfect candidate for character recognition, for example.

The problem is that the appearance of a natural scene containing man-made objects is complex and various. Therefore, the use of regional descriptors should be avoided. When they are used, this must be done cautiously and preferably by using the regional or topological descriptors as additional features rather than the dominant ones. They could prove to be valuable descriptors to supplementary characterise almost identical objects embedded in a natural scene or in certain regions.

Texture is another intrinsic feature of a certain object or region. The main approaches to describe the texture of a region are statistical, structural and spectral. Measures of texture calculated using only histograms are limited by the fact that they do not display any additional information regarding the relative position of the pixels with respect to each other.

Nevertheless, it is necessary to pay special attention to a particular case of boundary (regional) descriptors, known as *Fourier descriptors*. The discrete Fourier transform (DFT) can be employed as the tool to generate these quantitative descriptors. If N points on the boundary of the region are available, that region can be visualised as being in a complex plane, with the ordinate considered being the imaginary axis and the abscissa the real axis. Under these circumstances, the coordinates of each point lying on the contour become complex numbers. Starting the process at an arbitrary point on the contour and tracing around it once will generate a sequence of complex numbers. The DFT of this sequence is defined as the Fourier descriptor of the contour. The attractive characteristic of this procedure is that simple manipulations of this frequency representation of shapes generate the desired invariance to position, size and orientation for these features.

Fourier descriptors can be formulated for continuous or sampled curves. Because it is not easy to generate a boundary curve with equidistant samples, the alternative is to extract subpixel accurate object boundary curves directly from the grey scale images. This is not an easy task and it still remains a challenging research problem. These types of Fourier descriptors are, usually, denoted as *cartesian Fourier descriptors*. Mathematically they can be derived in the manner described below.

The boundary closed curve is described parametrically by taking the path length, p, from a starting point (x_0, y_0) as a parameter. The continuous boundary curve is represented by $x(p)$ and $y(p)$. These two curves can be combined into one curve, that is cyclic, using the complex function $z(p)=x(p)+iy(p)$. Considering the perimeter of the curve P, then the following relation is true:

$$z(p+nP)=z(p), \quad n \in Z. \tag{4.2}$$

At the same time, a periodic curve can be expanded into a Fourier series, whose coefficients are given by:

$$\hat{z}_u=\frac{1}{P}\int_0^P z(p)\exp\left(\frac{-2\pi i u p}{P}\right)dp, \quad u \in Z. \tag{4.3}$$

The periodic curve can be reconstructed from the Fourier coefficients as:

$$z(p) = \sum_{u=-\infty}^{\infty} \hat{z}_u \exp\left(\frac{2\pi i u p}{P}\right). \tag{4.4}$$

The coefficients $\hat{z}_u$ are known as the Fourier descriptors. They are always paired. The first coefficient gives the centroid of the boundary, the second one describes a circle, and so on. They can be easily computed from sampled boundaries. For a detailed treatment of this topic the reader is referred to Jähne (1997).

Amongst different attractive properties of Fourier descriptors, it must be mentioned that the computational time required to generate them is low. When more coefficients are calculated, a more detailed description of the boundary curve is obtained. Only the complete set of Fourier descriptors constitutes a unique shape description. The invariance to the scale, translation and rotational changes is contained in the first two Fourier descriptors, fact that constitutes a desirable attribute.

On the negative side, they are quite sensitive in the presence of noise. This means that the noise perturbs the phases of the Fourier descriptors coefficients, and the coefficients of lower amplitude could be seriously affected, as mentioned by Hall (1979). Slight shifts in orientation and/or starting point due to noise could drastically affect the final results. This would not have been a major problem, as the images should be carefully denoised, prior to the extraction of Fourier descriptors. The main concern and reason for not implementing Fourier descriptors as invariant features to characterise different objects or regions were generated by contradictory reports in the literature about their abilities in dealing with occluded objects. In the case of natural scenes containing man-made objects, this is more a normal situation than a special one.

While Hussein (1991) reported that Fourier descriptors fail when they are employed to characterise occluded objects, Jähne (1997) proposed a solution by using Fourier descriptors to describe non-closed boundaries. To close them, the curve must be traced backward and forward. Such open curves are easily recognised by the fact that their area is zero. The assumption is that through a careful choice of the starting point that must belong to the boundary, this solution is feasible, but makes the process more difficult and possibly faulty. It can be concluded that the use of Fourier descriptors creates a framework for scale, translation and rotation invariance. Due to the fact that they use only the boundary line, they should be invariant to intensity changes, as well. However, they restrict the object description only to the shape of it and do not reflect any spatial distribution of the grey level values within the object. Nevertheless, they can be successfully employed in the process of classification and reconstruction of non-clustered objects, with simpler, geometrical shapes.

Moving now to the category of transform features, it can be remarked that there are many mathematical operators that transform a two-dimensional signal image into another n-dimensional image. Features that were difficult to describe in the original measurement space become manageable, easy to interpret, and display interesting properties in the new feature space. The *Fourier transform*, the *wavelets transform* allow the extraction of useful features that are difficult or impossible for a human operator to obtain. Both transforms were previously treated, separately.

Other transforms including *Walsh, Hadamard, Slant* (Gonzales and Woods, 1992) were not considered suitable for the problem investigated. The basis vectors of these transforms are characterised by the fact that their basis functions spread over the entire image, thus losing the locality of the image. For example, if an image contains few independent objects, they will be simultaneously decomposed into global patterns. As a result, they will no longer be recognisable as individual objects in the transforms (Jähne, 1997). Therefore, it was decided not to use the abovementioned transforms, as they inadequately preserve features which characterise individual objects within a scene.

Special attention was paid to the *discrete cosine transform* (DCT), due to its interesting compressing properties. Unlike the discrete Fourier transform, DCT is real valued. It was supposed that if the coefficients of the DCT could compress discriminating information to uniquely characterise an object, they could be used as features and the matching process could be performed. Unfortunately, the experimental results did not provide good support for using this transform as a suitable descriptor.

An image can be regarded as a two-dimensional random signal, characterised by a particular probability density function. As a result, a wide range of statistical descriptors emerged. Therefore, features such as *auto-correlations, covariances, means* may be used as descriptors. Also *histogram features* could possibly be employed, but they are strongly dependent on intensity changes. *Grey level co-occurrence matrices* may be used to provide information related to contrast, homogeneity, entropy and correlations. One may estimate the *first-order entropy* in a picture by computing a normalised histogram for each level p_i, $i=1,2,\ldots,2^M$ (Hall, 1979) and calculating the first-order entropy as:

$$H = -\sum_{i=1}^{2^M} p_i \log_2 p_i \tag{4.5}$$

The most prominent statistical descriptors are, probably, *the invariant moments*. They exhibit rotational, translation and scale invariance. They are not very sensitive to intensity changes and in the presence of noise. They were selected as the main features descriptors for one of the final scene matching algorithms and are described extensively in the following section. Covariance matrices of the data were also used as an intermediate step when algebraic transforms were implemented for decorrelating the data and performing dimensionality reduction of the features, as in the case of Principal Components Analysis or Maximum Noise Fraction.

Finally, algebraic features represent intrinsic attributes of an image and evolved due to the possible matrix representation of a signal image. Because an image can be represented as a matrix, feature extraction is possible through various transforms or matrix decomposition. As an example of algebraic transforms, those based on eigenvector decomposition, such as *Karhunen-Loève transform*, *Hotelling trace criterion* should be mentioned. Principal Components Analysis constitutes an integral part of two of the final algorithms, and is therefore treated separately later. Hu (1962) used *algebraic invariants* as the foundation when deriving his invariant moments.

Another special class that must be remembered is that of *singular values*, derived through *Singular Value Decomposition* (SVD) of the image matrix. These features comprise properties of algebraic and geometric invariance, display good stability and deserve a more detailed treatment, due to their characteristics.

Any real rectangular matrix of size $M \times N$ $(M>N)$ may be transformed into a diagonal matrix through an orthogonal transform. The process of diagonalising such a matrix is called *singular value decomposition*. For any real rectangular matrix, A, there are two orthogonal matrices U and V and an $N \times N$ diagonal matrix Δ containing the singular values of matrix A, along its diagonal. This means that matrix A can be expressed as:

$$A = U \Delta V^T \tag{4.6}$$

where the columns of U and V are the eigenvectors of AA^T and A^TA, respectively. Symbol 'T' signifies the transpose of a matrix, and the diagonal matrix, Δ, is given by:

$$\Delta = U^T A V = diag(\lambda_1, \lambda_2, \cdots, \lambda_k, 0, \cdots 0) \tag{4.7}$$

The rank of matrix A is, by definition, equal to the number of non zero eigenvalues, *rank (A)* = k. The singular values of the matrix A can be used as the feature vector for an image, constructed in the following manner:

$$f = [\lambda_1, \lambda_2, \ldots \lambda_k, 0, \ldots, 0] \tag{4.8}$$

Equation (4.6) is known as the *forward transform*, while equation (4.7) represents the *inverse transform* of a unitary transform pair. In the case of A being a symmetric matrix, the two orthogonal matrices U and V are equal. At the same time, because Δ is a diagonal matrix, it has at most k nonzero elements, a fact that determines a loss-less compression of at least a factor of k, a property that is fundamental for data reduction. Despite the fact that the compression power of SVD seems so conspicuous, it must be mentioned that every time an image is compressed into few unique, diagonal elements, the kernel matrices U and V are individual for each image. Consequently, they must be stored together with the singular values. One amendment that can be made is to store the same kernel matrices for a similar class of images, for an approximate reconstruction. Another amendment is to select only a few singular values, and, as a result, to keep only the corresponding columns of the orthogonal matrices for storage. However, it is recommended to apply SVD with caution, and decide carefully which of the small singular values can be thresholded to zero. More details about this topic can be found in Press *et al.*, (1986).

Hong (1991) proved that singular values features are invariant to translation, rotation, scale, intensity and transposition. Furthermore, the singular value feature vector exhibits a strong stability, which makes the pre-processing steps more relaxed. If the singular value feature vector is extracted from an unprocessed image, one can still obtain high performance recognition. These features can be extracted from the whole image, overlapping blocks or distinct blocks, depending on the requirements of the problem.

The shortcomings of this descriptor are that the method is computationally extremely demanding and large storage space is required for the orthogonal matrices U and V. Moreover, no fast transform exists and the eigenvectors of each $M \times N$ matrix must be computed prior to the transformation itself (Castleman, 1996). A major restriction of the problem under consideration is that for the scene matching process the computational time must be as low as possible. This fact made the choice of this descriptor inadequate.

The successful recipe for a superior performance of the recognition system is to incorporate a combination of some of the previously mentioned descriptors. There may be some overlap between different types of features, but as long as they describe different properties of the same object, they constitute a desirable set of features.

The advantage of an automatic, computerised procedure is that some of the features are beyond the visual abilities of a human operator, due to the fact that they cannot be visualised. Due to a wide spectrum of features available, it is impossible to explore all of them. Only those that were considered reliable, suitable and manageable were implemented to accomplish a good correlation performance.

4.2 Moment representation

Recognition of visual patterns, independent of position, scale, intensity, and orientation in the visual field is not only an important problem of scene analysis, but also the core of the problem in hand. A set of mathematical features, that meet, up to a certain extent, these constraints, are *moments*, as mentioned in the previous section. Such features contain information that is based upon a global, rather than a local description of the image. Moreover, the advantage of using such features is that they are less sensitive to noise than local features. This property of invariance is highly significant in compressing the data, which is needed for object recognition. The use of invariant moments was first proposed by Hu (1962), who derived a set of seven invariant moments, for the second and third order moments, initially called *moment invariants*.

The mean of the product $x^p y^q$ is, by definition, a joint moment of the random variables x and y of order $p+q = n$. The two-dimensional $(p+q)$th order moments of a density distribution function $f(x,y)$ are defined in terms of Riemann integrals, in the case of continuous random variables, as:

$$m_{pq} = E\{x^p y^q\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy \qquad p,q=0,\ 1,\ 2,\ldots \tag{4.9}$$

where E denotes *expected value* or *mean* of a continuous random variable x, and is defined as:

$$\eta_x = E\{x\} = \int_{-\infty}^{\infty} x f(x)\,dx \tag{4.10}$$

From equations (4.9) and (4.10), the immediate result is that $m_{10} = \eta_x$, $m_{01} = \eta_y$, and they are the first-order moments, while $m_{20} = E\{x^2\}$, $m_{11} = E\{xy\}$, $m_{02} = E\{y^2\}$ are the second-order moments (Papoulis, 1991). As p and q take on all positive integer values, they generate an infinite set of moments. Furthermore, due to a uniqueness theorem (Hu, 1962) regarding moments, which states that if $f(x,y)$ is piecewise continuous (bounded function) having nonzero values only in a finite part of xy plane, moments of all orders exist and the set m_{pq} is uniquely determined by $f(x,y)$. Conversely, that particular set of moments m_{pq} uniquely determines $f(x,y)$. This implies that every unique shape corresponds to a unique set of moments.

The *joint central moments* of x and y are the moments of $x-\bar{x}$ and $y-\bar{y}$, defined as:

$$\mu_{pq} = E\{(x-\bar{x})^p (y-\bar{y})^q\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,d(x-\bar{x})\,d(y-\bar{y}) \tag{4.11}$$

where $\bar{x} = \dfrac{m_{10}}{m_{00}}, \bar{y} = \dfrac{m_{01}}{m_{00}}$.

All these moments, defined by equation (4.11), are related to the centre of mass. This is why they are, often, denoted as central moments.

For a digital image, in the case of discrete random variables, equation (4.11) can be rewritten as:

$$\mu_{pq} = \sum_{x}\sum_{y}(x-\bar{x})^p(y-\bar{y})^q f(x,y) \qquad (4.12)$$

where the image intensity function *f(x,y)* is expressed by a matrix with *x* and *y* representing the discrete locations of the image pixels. Actually, any pixel-based features can be used to calculate object moments (Jähne, 1997). Using equation (4.12), the central moments of order 3, in terms of the ordinary moments, can be easily calculated. The full calculation for the central moments of order 3 can be found in Hall (1979). They are as follows:

$$\begin{array}{ll}
\mu_{00} = m_{00} & \mu_{11} = m_{11} - \bar{y}m_{10} \\
\mu_{10} = 0 & \mu_{30} = m_{30} - 3\bar{x}m_{20} + 2m_{10}\bar{x}^2 \\
\mu_{01} = 0 & \mu_{12} = m_{12} - 2\bar{y}m_{11} - \bar{x}m_{02} + 2\bar{y}^2 m_{10} \\
\mu_{20} = m_{20} - \bar{x}m_{10} & \mu_{21} = m_{21} - 2\bar{x}m_{11} - \bar{y}m_{20} + 2\bar{x}^2 m_{01} \\
\mu_{02} = m_{02} - \bar{y}m_{01} & \mu_{03} = m_{03} - 3\bar{y}m_{02} + 2\bar{y}^2 m_{01}
\end{array} \qquad (4.13)$$

The zero-order moment is the area or "total mass" of a binary or grey value object and the first-order central moments are zero by definition. They are computed using the centre of mass as the origin, and are translation invariant (Castleman, 1996).

It is also necessary to use parameters that do not depend upon the size of the object. This constraint derives from the fact that it is required to compare objects that are observed from different distances, as in the case of this project. If an object *g(x)* is contracted/dilated by a factor α, then $g'(x) = g(x/\alpha)$ and its moments are scaled by $\mu'_{pq} = \alpha^{p+q+2}\mu_{pq}$. Therefore, taking a step further, the *normalised central moments*, denoted η_{pq}, can be normalised with the zero-order moment, μ_{00} to achieve scale invariance:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \text{where} \quad \gamma = \frac{p+q}{2} + 1 \quad \text{for } p+q=2,3,\ldots \qquad (4.14)$$

and they are invariant to scale changes (Jain, 1989). Because the zero-order moment of an object gives the area of the object, the normalised moments are scaled by the area of the object.

Shape analysis that goes beyond area measurements starts with the second-order moments. From the second and third-order normalised central moments, a set of *seven invariant moments* can be derived:

a) for first-order moments, $\mu_{01} = \mu_{10} = 0$, always invariant.

b) for second-order moments, $(p+q = 2)$, the invariant moments are:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{4.15}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

c) for third-order moments $(p+q = 3)$, the invariants are:

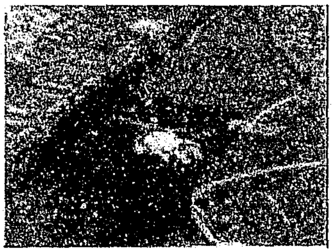
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4.16}$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) + [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03}) + [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

ϕ_7 is known as skew invariant, and is useful in distinguishing mirror images. It is invariant under rotation, but changes the sign under reflection. The derivation of this set of invariant moments is based on the theory of algebraic invariants. Therefore, this set of moments is invariant to translation, rotation, and scale changes. Because the magnitudes of these moments reflect the shape of the object, they can be successfully used to identify different visual patterns.

A critical problem that arises is that the uniqueness of the shape of an object is spread out over an infinite set of moments (Castleman, 1996). Therefore, a large set of features may be required to distinguish between similar shapes. But, higher-order moments are quite sensitive to noise. How many moments are necessary to be computed for discriminating shape features, and, at the same time, not to be affected by noise? The answer to this question will be suggested through various experiments. Generally, only the first few moments are required to differentiate between signatures of clearly distinct shapes (Gonzales and Woods, 1992).

The invariance manifested by invariant moments in the presence of scale, translation and rotational errors was checked experimentally using the original optical image "UKF.bmp". The robustness of invariant moments was also checked in the presence of noise, intensity changes, and for an incomplete initial signal image. To simulate scale changes, the size of the original image was reduced by 50%, increased by 100% and 400%, respectively. Shifting the original image ten pixels to the right and ten pixels down simulated translation errors.

Figure 4.1 displays the original image mirrored and half scaled.



a)

b)

Figure 4.1 Image "UKF.bmp"

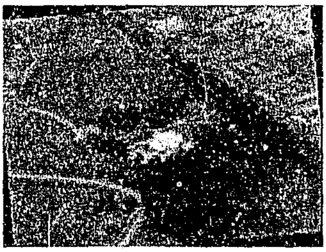
a) mirrored

b) half scaled

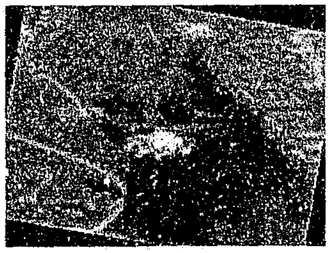
Because the invariant moments have very large dynamic range, including both ± signs, it is more practical to display the logarithm of the magnitudes of ϕ, as suggested by Hsia (1981). Therefore, only for the benefit of display, the seven invariant moments can be expressed as $\psi(i)=\log|\phi(i)|$, with $i=1,2,\ldots,7$.

Figure 4.2 depicts the original image rotated by $+5^0$and -10^0 . Same images will be used for the final stage of scene matching, when the robustness of different algorithms is tested for rotational errors. Actually, for this section, the experiments were conducted on a larger set of images, rotated by $\pm5^0$,$\pm10^0$ and $\pm45^0$. As expected, the invariant moments calculated for the set of images created to simulate scale changes, translation and rotational errors were in the same order of magnitude with the invariant moments' values calculated for the original image. Moreover, it has been noticed that the difference between the invariants computed for the original image and the corresponding invariant moments generated by all changed images is less than one unit. A slight discrepancy in the numerical results appeared when the image was rotated by $\pm 45^0$. A self imposed constraint of this work is that the algorithms proposed for scene matching must be performed on images that, possibly, suffer rotational errors within the range (0, $\pm 10^0$). The experimental results comply with this demand.

Therefore, it can be concluded that, based not only on the theoretical statements, but also through experimentation, invariant moments do not change under rotation or translation. They are, also, invariant in the presence of scale changes.



a)



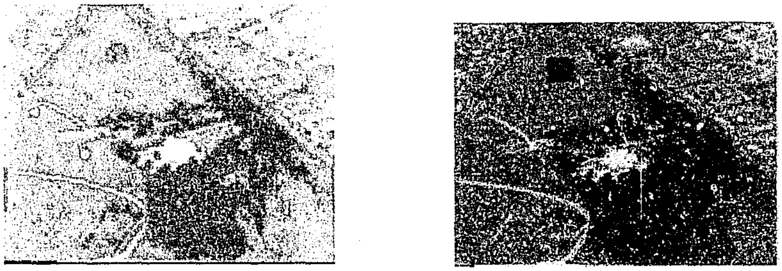
b)

Figure 4.2 Image "UKF.bmp"

a) rotated by $+5^0$

b) rotated by -10^0

What is of more interest, is to analyse the behaviour of invariant moments in the presence of different types of noise, contrast changes and when data is missing from the original image matrix. The images degraded by noise and blurring are displayed in figures 3.2-3.4. Figure 4.3 presents the original image affected by intensity changes and when a section of it was removed to simulate missing data. Contrast changes were simulated by subtracting the same constant value (0.3) from the intensity grey levels of all pixels of the original image. This generated a darker image. Adding same constant value to the original grey level values, generated a brighter image. These changes represent 30% from the theoretical range of grey level values (intensity images of class double cover the range from 0.0 (black) to 1.0 (full intensity, white)).



a) b)

Figure 4.3 Simulation of degraded images

a) brighter image

b) missing data

Table 4.1 contains the invariant moments' values calculated for the original image, modified by contrast changes, after histogram equalisation was applied to the changed images and invariant moments calculated for the image reflecting missing data. They are displayed in order from $\psi(1)$ to $\psi(7)$.

Table 4.1 Invariant moments calculated for the original image and images simulating intensity changes and missing data

Invariant moments $\psi(i)$ orig. image	Image brighter (+0.3)	After hist.eq. applied	Image Darker (-0.3)	After hist.eq. applied	Missing data
-0.8410	-1.3918	-0.9216	0.3665	-0.9632	-0.8290
-4.0327	-5.2438	-3.9038	-1.1576	-4.0868	-3.9915
-9.5521	-12.5836	-7.9385	-3.5043	-8.6048	-9.7537
-8.4625	-11.0889	-7.0047	-3.1825	-7.5251	-8.7308
-17.6551	-23.1506	-14.4763	-7.1044	-15.6960	-18.5698
-10.9639	-14.4326	-9.2500	-3.7888	-9.8175	-11.0574
-18.0559	-23.4317	-19.1696	-6.7146	-16.4176	-18.1536

Analysing this table, it can be concluded that invariant moments are not greatly affected by missing data, with the condition that only a reasonable number of elements from the image matrix is missing.

For the case of intensity changes, the experimental results are in agreement with Maitra (1979) who demonstrated that invariant moments are not contrast invariant. The contrast-changed image introduces a non-uniform scaling effect on ϕ. Hence, the need to transform the images affected by intensity changes such that they become as similar as possible with the original ones, in terms of their dynamic range. It can be noticed that all the moments, especially higher invariant moments modify their values. This suggests that it is necessary to pre-process the image before calculating the moments, applying histogram equalisation. The result is that the pre-processed image has a new dynamic range, following as close as possible that of the original image. Therefore, after histogram processing the new calculated moments exhibit close numerical values to those of the original image. This can be noticed particularly for lower order moments.

Another approach would be to retain only the first two invariant moments as descriptive features for images. It is known that an image is statistically characterised by its first two moments, namely the *mean* and *autocorrelation* (Keshavan and Shrinath, 1977). Higher moments, or statistics involving higher powers of the input data are, almost always, less robust than lower moments or statistics that involve only linear sums or counting. The third moment, known as *skewness*, is a measurement of the degree of asymmetry of a distribution around its mean. While the mean and standard deviation are dimensional quantities, the skewness is defined in such a way that makes it non-dimensional. It is a number that characterises only the shape of the distribution. The *kurtosis* or the fourth moment is also a non-dimensional quantity that measures the relative peakedness or flatness of a distribution, relative to a normal distribution. Therefore the advice of Press *et al.*, (1986) is that these two moments and others, of higher order, should be used cautiously, or not at all, if this approach does not involve any loss of valuable information.

Table 4.2 displays the invariant moments calculated for degraded images by different type of noise and blurring, before and after denoising the images.

Table 4.2 reflects a surprising stability of invariant moments in the presence of noise and blurring. It is very important that the set of invariant descriptors used is relatively stable for noise-degraded images. Any image is affected by noise, irrespective of the high quality of the sensor used to record it. A total removal of noise is not possible.

Table 4.2 Invariant moments for degraded images by noise and blurring, before and after denoising

Invariant moments $\psi(i)$ orig. im.	Degraded by Gaussian noise and blurring	Image denoised by Wiener filter	Degraded By salt&pepper Noise and blurred	Image denoised by median filter	Degraded by speckle noise and blurred	Image denoised by wavelets
-0.8410	-0.8405	-0.8401	-0.8525	-0.8380	-0.8384	-0.8391
-4.0327	-4.0279	-4.0244	-4.0694	-4.0173	-4.0238	-4.0225
-9.5521	-9,7467	-9.7338	-9.7424	-9.5339	-9.5441	-9.6359
-8.4625	-8.4708	-8.4690	-8.5807	-8.4207	-8.4275	-8.4368
-17.6551	-17.8829	-17.8662	-17.8667	-17.5827	-17.6066	-17.6611
-10.9639	-10.9224	-10.9236	-11.1745	-10.9575	-10.9611	-10.9767
-18.0559	-17.9735	-17.9735	-18.4983	-17.9855	-17.9822	-18.0534

All the experiments conducted proved that Hu's set of seven invariant moments do not change when the object (image) is contracted/expanded, and are rotationally and translation invariant. They have proved to be robust in the presence of noise, not very sensitive to missing data error and can be improved for intensity changed-images, if the pictures are pre-processed, as suggested. All these facts enhance the decision to use them as one of the suitable feature descriptors set. Another advantage of using this set of features is that they can be extracted from any original grey scale, enhanced grey scale, pre-processed or binary image. The calculation of this set of invariant descriptors was performed using program code called *mominv7,* listed in appendix C.

Reddi (1981) suggested that radial and angular moments could be successfully used to characterise an object, with the aim of performing image identification, classification or scene matching, instead of Hu's set of invariant moments. These moments are also invariant to rotation, translation, reflection and size changes, and can be derived without using algebraic invariants. Their calculation relies on expressing the intensity function $f(x,y)$ in polar coordinates. However, they seem to be more difficult to calculate, without providing any special characteristics in comparison with Hu's moments. Therefore, the popular choice to use the set of seven invariant moments to perform scene matching was made due to their ease of calculation.

Once the seven invariant moments were computed, they were stored as the elements of the invariant moment vector of size 7 x 1. One shortcoming of this approach is that there is some correlation between the moments of different orders. A badly estimated or corrupted *n*th order invariant moment would result automatically in the degradation of other moments. Also, all the redundant or irrelevant information was captured in the feature vector. The solution to this limitation is to represent the geometrical image characteristics by a matrix of invariant moments vectors, rather than a single invariant moment vector. This being the case, each column moment vector would represent localised image information and not global information. The idea behind this procedure is to map each pixel of the scene image into a feature vector that contains all the information (statistical, topological, etc.) about the original pixel and its immediate neighbourhood. This procedure was implemented in the following manner. If $x(i,j)$ is an image pixel, then its neighbourhood can be represented by:

$$\begin{aligned} N(x(i,j)) = \{&x(i,j), x(i-1,j), x(i+1,j), x(i,j-1), x(i,j+1), \\ &x(i-1,j-1), x(i-1,j+1), x(x(i+1,j-1), x(i+1,j+1)\} \end{aligned} \tag{4.17}$$

and denotes the set containing neighbouring points of $x(i,j)$, including itself.

Adjacency is an important, powerful binary relationship in scene description. All the eight neighbouring points of point $x(i,j)$, described by equation (4.17), may be considered adjacent to $x(i,j)$, as defined by Hall (1979).

Let $\Phi_k(N(x(i,j)))$ be the operator mapping the neighbourhood of $x(i,j)$ into some $y_k(i,j)$, that is,

$$y_k = \Phi_k(N(x(i,j))), k = 1,2,\ldots.d. \tag{4.18}$$

Therefore, each pixel of the scene image is mapped into a vector, and the original image is transformed into a set of *d* single band images (or one multiband image with *d* bands). The same mapping is performed for the pixels of the template image. The maximum number of band images, *d*, is equal to the number of invariant moments calculated. One can visualise a multiband image being similar to a multispectral image. Instead of different wavelength bands of the electromagnetic spectrum generating the set of images for the same scene, various multidimensional features, in this case the invariant moments, are those creating the set of images. This means that similar techniques to those used for multispectral images could be applied to a multiband image.

Regarding the size of the sliding window, this can be any size. The only condition is that it must produce more data points (*n*) than its maximum rank, or for a window size of $N \times N$, $n \rangle\rangle N$. The bigger the size of the sliding window, the better the description of pixel $x(i,j)$ in relation to its immediate neighbourhood, but a longer computational time is required. Therefore, a compromise must be reached between accuracy of the results and the necessary computational time. It is desirable that the computational time necessary to calculate invariant moments be minimised. Experimentally, it was established that a sliding window size of 11 x 11 is a good compromise between accuracy and computational time. The results sustaining this statement are discussed in section 6.2. The program code *momovl*, listed in appendix C, implements the mapping of feature vector invariant moments onto a matrix of invariant moments vectors.

The invariant moments were extensively used for scene matching or pattern classification and/or recognition, as well as target identification. The experiments conducted by Dudani, Breeding and McGhee (1977), who used invariant moments as features for aircraft identification, serve as classic examples of using moments to perform various decisional tasks. Also, Soland and Narendra (1979) achieved good results when they designed a recognition classifier using moments as features.

The advantage of the above procedure is that it is very flexible and a multiband image can be created using any other statistical or topological descriptors, depending on the problem to be solved.

The following problem to be solved is that the multiband image created through the above-mentioned method must be processed further because the information contained is still redundant, due to inter-band correlation. Also, despite the different noise reduction techniques applied previously, the true data are still embedded in noise. In order to reduce the dimensionality of the data and remove the noise further, two types of transforms are used. One is called *Principal Component Analysis* (PCA), the other is *Maximum Noise Fraction* (MNF) and both are described in the following chapter.

5 Feature extraction and dimensionality reduction

The multiband image has already been formed. It is necessary to process it further for the reasons explained in section 4.2. The use of redundant or irrelevant information contained in the bands used to perform the correlation would have the effect of increasing the computational time, while noisy features would generate a smooth peak, with lower height. When the correlation is performed, the algorithms could also generate false peaks. Two main techniques were investigated for reducing the features' dimensionality and are presented hereafter.

5.1 Principal Component Analysis (PCA)

Hotelling developed a mathematical procedure, called the method of *principal components*, also known as *Karhunen-Loeve* (K-L) *expansion* or *eigenvector transformation,* which develops a linear transformation that removes the correlation among the elements of a random vector and is characterised by a dimension-reducing capability. Using this transform, the discriminatory information compression is achieved by approximating the representation pattern vector by a small number of the terms of the expansion. It was proved that, for any given number of terms, the mean square error (MSE) between the approximation and the original pattern is minimal in comparison with the error incurred by truncating any other orthogonal expansion of the pattern vector (Devijver and Kittler, 1982). Therefore, most of the intrinsic information about the pattern vector, including the discriminatory one, is compressed into a few terms of the Karhunen-Loeve (K-L) expansion.

Let $[X_1, X_N]$ be a p x N matrix of pixel vectors. A random pixel vector X is selected and the value of the (i,j)th pixel vector is denoted by $X(i,j)$. The matrix of pixel vectors is characterised by a mean vector, m_X, and a covariance matrix, C_X. It is possible to subtract the mean vector from each data point. This results in translating the coordinates from the initial origin to the centroid of the data. In the new coordinates, the mean vector is zero and the covariance matrix becomes equal to the *scatter matrix* (*autocorrelation matrix*, that gives essentially the same information about the dispersion of the distribution, but is not normalised). The diagonal elements, C_{ii}, are the variances of the individual random variables, while the off-diagonal elements, C_{ij}, (for $i \neq j$, with $j = 1, \ldots, p$) are their covariances. If C_{ij} entries are equal to zero, this means that the random variables are uncorrelated. Analysis of multivariate data $X_1, \ldots, X_N$ is greatly simplified when most or all of the variables $X_1, \ldots, X_p$ are uncorrelated, which means that the covariance matrix C_x is diagonal or nearly diagonal (Lay, 1994).

The full proof for the Principal Compor nts transform is contained in appendix A.

The covariance matrix, C_x, may be expressed in a diagonal form, using the unique set of *eigenvalues* (sometimes called the *characteristic values*) and *eigenvectors* or *characteristic vectors* of the matrix, generated by its characteristic equation. This is the way to construct the diagonalisation of the covariance matrix. It is fundamental to the development of the principal components transformation to find a new coordinate system in the multispectral vector space in such a way that the data can be represented without correlation. In other words, such that the covariance matrix in the new coordinate system is diagonal (Richards, 1986).

The next step is to define a linear transformation, (the standard matrix of the transformation is chosen to be an orthogonal p-square matrix, denoted by **A**), that generates a new vector Y, from any vector X, such that:

$$Y = A(X - m_X). \tag{5.1}$$

This equation is known as the *Karhunen-Loeve transform*.

It can be shown that the mean of the new random vector Y, m_Y is equal to zero. The idea behind the whole procedure is to find a way such that the covariance matrix of Y is diagonal. The covariance matrix of the new vector Y can be proved to be equal to:

$$C_Y = \Delta \tag{5.2}$$

where Δ is a diagonal matrix, with the eigenvalues of C_X arranged such that $\lambda_1 \geq \lambda_2 \geq ...\lambda_p \geq 0$. Therefore, the covariance matrix, C_Y, is equal to the diagonal matrix of the eigenvalues of the covariance matrix of vector X, C_X. The off-diagonal elements of the new covariance matrix are zero, which means that the elements of the new vector Y are decorrelated. Furthermore, the two covariance matrices have the same eigenvalues. The effect of applying equation (5.1) is to transform a vector X from the p-dimensional data space into a corresponding vector Y in the new feature space. The eigenvectors (the columns of U, where U is an orthogonal matrix whose columns are the unit eigenvectors corresponding to the eigenvalues of C_X) are a new basis for the transformation. Therefore, the new pixel vector $Y(i,j)$, can be expressed as :

$$Y(i,j) = U^T (X(i,j) - m_X). \tag{5.3}$$

This new coordinate system shows clearly that *Hotelling transform* is a rotation transformation that aligns the data with the eigenvectors, and this is the mechanism that decorrelates the data. Since C_Y is, by definition, a covariance matrix and is diagonal, its elements will be the variances of the pixel data in the respective transformed coordinates. Due to the way the eigenvalues were arranged on the main diagonal of the matrix, the data exhibits maximum variance in y_1, the next largest variance in y_2, and so on, with the smallest variance in y_p . The covariance matrix used to generate the principal component transformation matrix is a global measure of the variability of the original data image vectors and Karhunen-Loeve transform provides a method for decomposing these image vectors into a number of mutually uncorrelated components.

Another property of the Hotelling transform is that of reconstruction of the original vector X from the newly-created one, Y. This results in another appealing property of this transform, that of reducing the number of the components of the original vector to any specified accuracy. Recalling equation (5.1) and the fact that the rows of A are orthonormal vectors,

$A^{-1} = A^T$ and $A = U^T$, any vector X can be recovered from its corresponding Y, using the relation:

$$X = A^T Y + m_X = UY + m_X . \tag{5.4}$$

Therefore, instead of using all the eigenvectors of C_X, one could form the matrix A_m from only m eigenvectors corresponding to the m largest eigenvalues of the covariance matrix C_X, generating a transformation matrix of order m x p. Hence, the reconstructed vector $\hat{X}$ is described as:

$$\hat{X} = A_m^T Y + m_X = \sum_{k=1}^{m} \sum_{l=1}^{m} Y_{kl} B_{kl} \tag{5.5}$$

in which the reconstructed image was expressed as a sum of weighted basis images B_{kl} that are the individual eigenvectors in the reconstruction matrix and the weights Y_{kl} are the Hotelling-transformed versions of the original image pixels $X(i,j)$. It follows that the mean square approximation error between the original X and the approximate image $\hat{X}$, using only the first m eigenvectors, can be expressed in terms of the transformed data and the basis images:

$$e_{MS} = E\left\{\left\|X - \hat{X}\right\|^2\right\} = E\left\{\left\|\sum_{k=1}^{p} \sum_{l=1}^{p} Y_{kl} B_{kl} - \sum_{k=1}^{m} \sum_{l=1}^{m} Y_{kl} B_{kl}\right\|^2\right\} = E\left\{\left\|\sum_{k=m+1}^{p} \sum_{l=m+1}^{p} Y_{kl} B_{kl}\right\|^2\right\}.$$

Equation (5.6).

It could be observed that the error is zero if $m = p$, meaning that all the eigenvectors are used in the reconstruction. But, the purpose is to use only the first m eigenvectors to reconstruct the original image in order to discard some of the redundant information and noise. The mean square error between the original multiband image and the reconstruction obtained using only the first m eigenvectors can be shown to be equal to the sum of the eigenvalues corresponding to the eigenvectors not used in the reconstruction (Gonzales and Wintz, 1987). It has been shown that the minimum square error (MSE) between the original vector and its approximation is minimised by selecting coefficients with the largest variances (Devijver and Kittler, 1982). In order to decide how many eigenvectors should be used in the reconstruction of the original data set of vectors, the eigenvalues should be scrutinised. It can be proved that an orthogonal change of variables, in the case of matrix U, formed by the eigenvectors of covariance matrix C_X, does not change the total variance of the data:

$$Tr\,(C_X)=Tr(\Delta)=\lambda_1+\lambda_2+\ldots\lambda_p \quad (Tr \text{ refers to the trace of a matrix}). \tag{5.7}$$

The ratio between each eigenvalue of the covariance matrix C_X and the total variance of the data shows at any moment the contribution of each principal component (the unit eigenvectors of matrix U, corresponding to respective eigenvalues,defined as above) to the data. Therefore, if the eigenvalues of the covariance matrix are calculated, it can be established how much of the information data is contained in the first component, and so on. In this way, it can be decided how many principal components should be used for an accurate estimation of the original data. Usually, the eigenvalues vary considerably in magnitude, and the smaller ones can be ignored without the introduction of significant error. This is a good way to discard redundant or irrelevant information and, at the same time, remove some of the noise. It is also necessary to establish a criterion regarding which eigenvalues must be kept to reconstruct the image containing relevant information and which eigenvalues to neglect, according to the requirements of the problem in hand.

It was found that using the eigenvectors corresponding to the first few largest eigenvalues produced satisfactory correlation results. The ideal situation would be to have the best truncation index found automatically. Plotting the eigenvalues, it can be observed that there is a sharp drop in magnitude between the eigenvalues containing useful information and those representing noise. One procedure to find the index where the eigenvalue sequence should be truncated, is to find the derivative of this curve. The point where the derivative exhibits a minimum can be used as the point to truncate the eigenvalue sequence.

A simpler method is to use the properties of principal components for finding the point where to discard the unwanted eigenvalues. The mean square error can be expressed in terms of variance as the following, using the fact that the basis images are orthogonal:

$$e_{MS}=\sum_{k=m+1}^{p}\sum_{l=m+1}^{p}\sigma^2{}_{ykl} \tag{5.8}$$

where σ^2_{yki} are the variances of the discarded coefficients. At the same time, the variances are equal to the eigenvalues, according to equation 5.7.

A logical step is that, knowing the variance of the noise, determine how many eigenvectors should be used in the reconstruction of the original image, in such a way that noise is excluded. Usually a signal image is modelled as the sum of a noiseless signal and a Gaussian contamination of the original signal image.

$$X = S + \sigma_n^2 n \tag{5.9}$$

where S is the original noiseless signal image, X is the real image and n represents a white Gaussian noise of variance σ_n^2. Therefore, the mean square error between the real image and the noiseless one can be expressed as:

$$e_{MS} = E\left\{\|X - S\|^2\right\} = \sigma_n^2 . \tag{5.10}$$

It is now possible to decide which eigenvectors should be used in a Hotelling transform, in such a way that the noise could be excluded. Let X_m be the principal component reconstruction of X, the original image, using only the first m eigenvectors. If it is desired that X_m should approximate S, the noiseless image, then the mean square error between X and X_m is equal to the variance of the noise σ_n^2. This statement can be expressed as a condition on the number m of the eigenvalues used in the Hotelling transform (Gonzales and Woods, 1992):

$$e_{MS} = \sum_{k=m+1}^{p} \lambda_k \approx \sigma_n^2 . \tag{5.11}$$

To conclude, the sum of the eigenvalues excluded from the reconstruction should be approximately equal to the noise variance, this procedure ensuring that the eigenvalues containing noise will be excluded. Because a multiband image is used, the noise variance must be scaled by the number of bands. If the noise variance is not known, it can be accurately estimated using different methods that are described in the following section. The truncation procedure, performed in an optimum mean square error sense, described above, was implemented for the final algorithms used to perform the scene matching.

Despite all the advantages exhibited by principal components, it happens occasionally that this method does not posses the ability to order the components in terms of decreasing image quality. The principal component transform of a multiband image results in another multiband image, whose bands are decorrelated and ordered according to their signal variance. In the case where the original image is noisy, the noise will contribute to the signal variance and it is possible that a band containing a large amount of noise have a larger variance than one containing more useful information. This is the reason why another transform, known as *Maximum Noise Fraction* (MNF) was developed. This transform orders the bands according to image quality, most often measured by *signal-to-noise ratio* (SNR).

5.2 Maximum Noise Fraction (MNF)

Maximum Noise Fraction (MNF), introduced by Green *et al.*, (1988), is equivalent to the principal components when the noise variance is the same in all bands, and is reduced to a multiple linear regression when the noise manifests in only one band. The appealing property of MNF is that the noise can be removed in an effective way from the "multispectral" data by transforming the original image matrix to the MNF space, smoothing or discarding the noisy components and re-transforming the data into the original space.

When applying MNF, instead of choosing new components to maximise the signal variance, the procedure designed for PC, the new components are chosen in such a way as to maximise the signal-to-noise ratio (SNR). Hence, the optimal ordering of the bands in terms of image quality. This transform maximises the noise fraction in consecutive components that are uncorrelated over the data set. Equivalently, this approach maximises the SNR in the components taken in reverse order.

MNF transform is equivalent to a two-phase transformation in which the data are:

a) transformed such that the noise covariance matrix is the identity matrix of unit variance and uncorrelated, in other words, the noise is whitened and decorrelated;

b) classical principal components transform is applied to the noise whitened data resulted from the previous step.

Consider a multivariate data set of p bands with grey levels $X_i(x)$, $i=1,\dots,p$, where x gives the coordinates of the sample. It is assumed that:

$$X(x) = S(x) + N(x) \quad \text{with} \quad X^T(x) = \{X_1(x),\dots,X_p(x)\} \tag{5.12}$$

when $S(x)$ and $N(x)$ are the uncorrelated signal and noise components of the corrupted signal $X(x)$. The covariance matrices can be expressed as:

$$Cov\{X(x)\} = \Sigma = \Sigma_S + \Sigma_N \tag{5.13}$$

with Σ_S and Σ_N the covariance matrices of the signal and noise, respectively. Even if an additive model was assumed, the technique can be applied to multiplicative noise by first taking the logarithms of the observations. The noise fraction of the ith band is defined as the ratio between the noise variance and the total variance for this band, $Var\{N_i(x)\} / Var\{X_i(x)\}$.

The MNF transform results in new p uncorrelated bands, which are a linear transformation of the original data set and can be written in matrix form as:

$$Y(x) = A^T X(x) \text{ where } Y^T(x) = (Y_1(x),\dots Y_p(x)) \text{ and } A = (a_1,\dots a_p). \tag{5.14}$$

An important property of the MNF transform is that, because it depends on SNR, it is invariant to scale changes for any band; another useful quality is that it orthogonalises $S(x)$, $N(x)$ and $X(x)$.

The coefficients of the linear transform A can easily be found by solving the eigenvalue equation:

$$A\Sigma_N\Sigma^{-1} = \Delta_N A \tag{5.15}$$

where Δ_N is a diagonal matrix of the eigenvalues λ_i ,such that the noise fraction in Y_i *(x)* is given by these eigenvalues. But, recall that these eigenvalues are ordered such that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$, which implies that the MNF transformed data will be arranged in bands of decreasing noise fraction. The eigenvectors are normed, therefore $A^T \Sigma A$ is equal to an identity matrix. The advantage of this approach is that MNF transformed bands are ordered by noise fraction, which allows a filtering of high noise fraction bands (containing less signal to be distorted by the denoising process), followed by a reversed transform.

When MNF is reformulated as a natural scaling where the noise in each band is equal in magnitude and is uncorrelated with the noise in any other band, the transform was called the *noise-adjusted principal components* (NAPC), introduced by Lee *et al.,* (1990). The NAPC transform requires an estimate of the noise covariance matrix, Σ_N that is assumed to be real, symmetric and positive definite. Then, the PC transform or eigenvector matrix of Σ_N is computed, in order to weight the data covariance matrix, Σ_D. The PC transform matrix of lately weighted matrix is also calculated. The steps taken to implement the NAPC transform are contained in appendix B.

The NAPC transformed image is the set of pixel vectors, **Y***(i,j)*, produced from the original pixel vectors, ***X****(i,j)*, by the transform:

$$
\begin{aligned}
Y(i,j) &= H^T(X(i,j)-m_X) = (FG)^T(X(i,j)-m_X) = G^T F^T(X(i,j)-m_X) = \\
&= G^T \Delta_N^{-1/2} E^T (X(i,j)-m_X)
\end{aligned} \tag{5.16}
$$

with $\boldsymbol{H} = \boldsymbol{FG}$, $\boldsymbol{H}$ representing the NAPC transform matrix, using the definition of the renormalisation matrix $\boldsymbol{F} = \boldsymbol{E}\,\Delta_N^{-1/2}$.

Notice the similarity between the PC transformed pixels, given by equation (5.16) and the NAPC transformed image pixels. Comparing equations (5.3) with (5.16), it can be noticed that for the transform matrix for PC, $A = U^T$, the role of the matrix U is played by the transform matrix $\boldsymbol{H}$, and that $U = \boldsymbol{E}\ \Delta_N^{-1/2}\ \boldsymbol{G}$.

Instead of performing two PC transforms, one to whiten the noise, and another to decorrelate noise-adjusted data, Roger (1994) showed that the NAPC transform can be implemented by a single matrix, $\boldsymbol{H}$, which simultaneously diagonalises the covariance matrices Σ_D and Σ_N :

$$H^T \Sigma_D H = \Delta_{adj} \quad \text{and} \quad H^T \Sigma_N H = I \tag{5.17}$$

The existence of matrix $\boldsymbol{H}$ can be proved by solving the polynomial equation:

$$|\Sigma_D - \lambda \Sigma_N| = 0 \tag{5.18}$$

where λ are the roots of the above equation, known as *the generalised symmetric eigenvalue problem.*

If Σ_N is an accurate estimation of the noise covariance matrix, $H^T\ \Sigma_N\ H = I$, which means that the variance of the noise in each transformed band is equal to unity and uncorrelated between transformed bands. Due to this result, it can be concluded that the variance of each transformed band equals its SNR. Moreover, $H^T\ \Sigma_D\ H = \Delta_{adj}$, with the roots, λ, of this diagonal matrix, Δ_{adj} obtained by solving the polynomial equation (5.18), implies that the bands will be ordered by their variances, that is, their SNR's. In the event that the noise covariance matrix, Σ_N, is ill-conditioned, it is advisable to apply NAPC transform, instead of the fast computationally method proposed by Roger.

As in the case of PC transform, not all the MNF transformed bands carry valuable information. Theoretically, the transformed bands used as features in the correlation process should not contain any noise, and also redundant information should have been discarded. By plotting the eigenvalues (roots) of the generalised symmetric eigenvalue problem, it can be decided which of the bands could be discarded without loss in performance. Again, if this decision can be automated, it will speed up the entire process and eliminate human intervention. The method used to truncate the eigenvalues series for the PC transform can be extended to the MNF transform. The roots generated by the generalised symmetric eigenvalue problem are the same with the eigenvalues of the noise-whitened data covariance matrix. Remember that the noise corrupting this data is of unit variance, therefore the desired mean square error in an MNF reconstruction process intended to remove noise should be one. This means that the sum of the eigenvalues, λ_k, generated by solving the polynomial equation (5.18), excluded from the reconstruction should be one. Using equations (5.07)-(5.10), the following result is obtained:

$$\sum_{K=m+1}^{p} \lambda_k \approx 1 \tag{5.19}$$

Because a multiple band image is used, the above condition for the sum of the eigenvalues should be scaled by the number of bands. The above method, together with the one described for truncating the eigenvalues generated by PC transform, were tested on different images, degraded by different amounts of noise, and it was found that they consistently generated accurate results.

The primary problem encountered when applying MNF or its reformulated version NAPC, is that only a correct estimation of the noise covariance matrix will secure the success of these methods. There are a number of ways to calculate this matrix, two of which are outlined below, as they were used during this work. One method, proposed by Switzer and Green (1984) uses near-neighbour differences to estimate the noise covariance matrix by:

$$N(x) = Y(x) - Y(x + \delta) \tag{5.20}$$

where δ is an appropriately chosen step length, and assumes that the signal at any point of the image is strongly correlated with the signal at neighbouring pixels, while the noise exhibits a weak spatial correlation.

$$S(x) = S(x + \delta) \tag{5.21}$$

This procedure, known as *minimum/maximum autocorrelation factors* (MAF), was applied for estimating the noise covariance matrix for one of the final algorithms. More details are discussed in chapter 6.

Another method used to estimate Σ_N makes use of the DWT. Remember from section 3.5 that if the DWT of each band is found, then the detail coefficients of the wavelet transforms will contain the noise contaminating that band. To obtain an accurate estimation of the noise, the details coefficients are thresholded in a similar manner to the one used for wavelet denoising, described in section 3.5.3. The difference is that the coefficients below the threshold and not above it, are used to reconstruct a noise image. The threshold can be calculated in the same way described in section 3.5.3. Reconstruction of the noise image adopts the same approach used for wavelet denoising. The dissimilarity is that the lowest resolution approximation coefficient is set to zero. It was established through experiments that a threshold set up to 2.5 standard deviations of the noise provided consistent and accurate estimates for the noise, in the case of different images. The experimental numerical value for the threshold is in accordance with the *3-sigma clipping* method proposed by Starck *et al.,* (1998). They suggest that a robust estimation of the noise variance should be obtained only on the basis of values within three standard deviations of the mean.

The assumption that was made is that the contaminating signal is white Gaussian noise. The reasons for using the Gaussian (normal) probability density function so often to model the noise is partly because of the *central limit theorem* and partly due to its relatively simple analytical form. The central limit theorem states that a random variable that is determined by the sum of a large number of independent causes tends to have a Gaussian probability distribution (Shanmugan and Breipohl, 1988), and this could be considered a good description for the noise.

The next requirement is to estimate the noise covariance between bands, which implies that an estimate of the noise for each band is required. However, it is not necessary to use the entire band in the wavelet noise estimation method. Even downsampling the band, before performing this estimation to reduce the computational time, did not result in a significant loss of accuracy. This was the method used to estimate the noise covariance matrix required by the implementation of MNF transform with the final algorithms. The results and comparison are provided in chapter 6.

Gordon (1998) proposed a generalisation of MNF transform, in which a new data set could be created by appending up to q powers of the original data set, similar to the generalised principal component transform. Thus, the generalised maximum noise fraction transform leads to a polynomial filter. Although this method claims better denoising results, the computational demands made it inadequate for this project.

6 Scene matching

The problem of using all relevant information generated by the previously described processes for the purpose of interpreting the contents of an image is usually known as *image understanding* (IU) or *scene analysis*. Image understanding is an attempt to extract knowledge about a three-dimensional (3D) scene from two-dimensional (2D) images (Kanatani, 1990). One of the important goals of IU is to develop techniques for automated 3D-scene interpretation and this is, broadly speaking, the final aim of this work.

Key applications of IU are automatic target recognition (ATR), autonomous navigational guidance, photointerpretation, cartography, natural resource analysis, weather prediction, environmental studies, automation for manufacturing and quality control, robot vision and the list continues.

A common problem encountered in most of these applications is *image registration*. Image registration is the process of matching a sensor image to a reference image to determine the relative offset of the images. A basic matching problem consists in locating a reference subset in a larger set of data, which contains some variation of the reference function, subject to a given criterion. The smaller the variation between the reference subset and the data set or subset to which it is matched, the better the match. Variations of the problem include template matching for object recognition, matched filtering for signal detection, image registration, change detection, cartography feature location, correlation guidance and scene matching.

6.1 Overview

Attention will be focused on scene matching that refers to the ability in locating or matching a region of an image with a corresponding region of another image, often recorded using a different sensor, at another time and from a different view angle. Usually, the general problem involves corrections of geometrical and radiometric distortions. Because the reference image was chosen to be a subset of the sensed picture, no geometrical distortion exists between the two images. Scene matching has many applications such as sensor alignment, navigation updating and missile lock-on-after launch.

As already stated, the purpose of this work is to match a given image subregion at each location in a larger image. Given a pictorial description (template) of a region in a scene, the goal is to determine which region in another image is similar. The simplest approach to this problem is called *template matching*.

At each location there are four possible scenarios:

1) the true matching position is recognised by the algorithm;
2) a wrong position is classified as a match, that is, *false fix*;
3) the true matching position is skipped;
4) a wrong position is skipped.

"A reliable, robust and accurate algorithm must make only decisions one and four" (Hall *et al.*, 1980).

There are two fundamental approaches to the scene matching process:

- Correlation procedure;
- Feature matching procedure.

The correlation procedure is a variation of the cross-correlation procedure, in which the template is shifted relative to the scene and a translation-offset parameter is computed for each location. The method most used for the similarity detection is correlation. The resultant correlation measurement may be expressed in many forms: correlation coefficient, mean square error, mean absolute difference.

The feature matching procedure extracts designated sets of features and their associated descriptors to perform the comparison. This method uses only the extracted features, which may be stored as a new image or in a tabular format.

The approach adopted during this dissertation is a combination between the two above-mentioned methods. In order to determine the trans on offset parameters, it is necessary to first establish a measure of similarity at each test location. Few methods that have been developed for detecting similar regions and performing scene matching, utilise the *basic correlator*, the *statistical correlator*, the *sequential similarity comparison* and *hierarchical matching* (Hall, 1979). The basic correlator method forms a correlation measure between the two picture functions and determines the correspondence location by calculating the location for maximum correlation. It is, therefore, necessary to find a characteristic feature from the first image inside the second, within a certain search range. Within this delimited range, the position of optimum similarity between two images is searched for. The similarity measure must be robust against changes in illumination. The correlation approach is insensitive to intensity changes between the two images (Jähne, 1997). This means that two spatial feature patterns are equal if they differ only by a constant factor that reflects the difference in illumination. This is the reason why the correlation approach was chosen as a similarity measure.

Having been given two images, the template image matrix T of size J x K, with the picture elements assuming one of L grey levels, $0 \leq L \leq 255$, and the scene image matrix S of size M x N, whose picture elements have the same grey level scale range, the cross-correlation function between the reference (template) image and the sensed (scene) picture is defined as:

$$C(u,v) = \sum_{j=1}^{J}\sum_{k=1}^{K} T(j-u, k-v) S(j,k),$$
$$\textit{with} \quad 1 \leq u \leq M-J+1 \quad \textit{and} \quad 1 \leq v \leq N-K+1 \qquad (6.1)$$

where (j,k) are indices in a J x K point template area, T, that is located within an M x N point search area, S, and (u,v) is the reference location. The correlation will have a maximum value when the two images are in registration (Bezaie and Srinath, 1984). It is clear that the correlation function $C(u,v)$ must be calculated for all $(M\text{-}J+1) * (N\text{-}K+1)$ possible translations of the template (window) area within the scene (search) area. This process generates the location of the maximum. Usually, the cross-correlation function has a broad peak that makes the scene matching process rather difficult. It has been proved by Pratt (1974) that the normalised correlation measure $R(u,v)$ at the reference location (u,v), defined as:

$$R(u,v)=\frac{\sum_{j=1}^{J}\sum_{k=1}^{K}S(j,k)T(j-u,k-v)}{\left[\sum_{j=1}^{J}\sum_{k=1}^{K}S^2(j,k)\right]^{1/2}\left[\sum_{j=1}^{J}\sum_{k=1}^{K}T^2(j-u,k-v)\right]^{1/2}} \tag{6.2}$$

generates a sharper peak. The above normalised correlation coefficient also manifests scale invariance properties. It can be concluded that the location of the template within the scene image is indicated by the peak value of the normalised correlation coefficient.

The problem with this approach is that it requires a large amount of computation to be performed, when the window and search areas are large. One method to improve the speed of the correlation procedure is to use a fast correlation technique by implementing the fast Fourier transform, that decreases the amount of correlation computations, but still requires computations at each of the *(M-J+1)*(N-K+1)* locations. A solution to the problem is to reduce the computation time for the matching. Two types of methods have been developed for this purpose: sequential methods and hierarchical approaches. Barnea and Silverman (1972) proposed a sequential similarity detection algorithm. The basic principle behind the sequential methods is to accumulate the level of mismatch between the selected template and each location in the image. Because mismatches grow faster than matches, the area with a lower chance of matching can be detected at an earlier stage of the algorithm and rejected. Using this approach the computational time can be decreased considerably.

Due to the fact that a comparison is required after each computation, it appears that a normal correlation with no threshold would be faster. The algorithm is also a non-linear processing algorithm, for which system performance and sensor noise conditions must be evaluated by simulation or system tests.

Another alternative is to use a hierarchical search method. In this approach the resolution of images and templates is reduced by averaging or pyramids. The first stage involves performing the matching over the reduced resolution images, and if a promising area is detected for a pre-established threshold, the matching is completed at higher resolution within the same region. This approach improves the efficiency of the matching algorithm, but can also increase the probability of a false fix. It was mentioned that the performance of the above methods (sequential or hierarchical) could become extremely poor in the presence of images degraded by non-white noise (Liu and Caelli, 1988).

Therefore, it was decided, based on the above considerations, that the normalised correlation coefficient, given by equation (6.2), be used as a similarity measure. In addition, correlation appears to be a "natural solution for the mean-square-error criteria" and the "analog-optical methods implement correlation easily" (Barnea and Silverman, 1972).

Once the similarity measure is established, the next step is to decide upon the algorithms, which are implemented to perform the scene matching. Depending upon the practical application of scene matching, certain requirements must be kept in mind for the purpose of deciding which approaches must be adopted. As already mentioned, the practical output of this work is to identify the location of a template in a noisy natural scene that incorporates man-made objects. The template image, also denoted as the *reference image*, is a subsection of the noiseless scene, known as the *sensed image*. The real image, recorded by a sensor, in this case an optical device, is subject to degradation caused by noise, blurring, rotational errors, scale changes, occlusion and intensity changes, up to a limit that still allows the recognition of the original scene by the naked eye. Standard approaches to the scene matching problem extract features from the reference and sensed images and perform a similarity measure on these features. When the correlation between the real, noisy scene image and the reference one is performed, multiple peaks are formed.

The objective of this work is to overcome this inconvenience by designing scene matching algorithms that are robust to degradation of the real image and generate a unique match for any pair of template-sensed images. Also, the process must also be automatic, and characterised by a high computational speed.

In an outdoor real world situation, characterised by dynamic environmental conditions, where the problem complexity is significantly higher than in the indoor scenario, it is not possible to detect and recognise all reference images under all conditions, using a single algorithm. In order to deal with the change of the scene due to different degradations, one needs to use inherently parallel algorithms (not a parallel version of a serial algorithm), that will generate and agree upon the same match, while discarding different false fixes.

Another problem that arises is which information from the original picture should be used. The original input picture contains the maximum available image information, but the direct use of the entire image as feature space would result in high dimensionality, making the algorithms computationally very expensive. In practice, the images are usually transformed by different methods to be projected onto the feature space, in this way reducing the computational burden. Therefore it was decided to use two algorithms; one using invariant moments as features for performing the correlation, the other employing the information provided by the raw image and not requiring feature extraction. There are also two different approaches on which the two algorithms operate. One relies only on the relative luminance and shape, while the other makes use of the relative location and relation of the pixel to its neighbours, in addition to the pixels' grey level values. In this way, the primary difficulty of scene matching algorithms, that of providing poor discrimination between objects of different shapes, but of similar size or energy is surpassed.

The database used to assess the performance of *Multiband Image using Moments* (MBIMOM) algorithm and *Spatial Multiband Image* (SMI) algorithm for automated scene matching, comprises a set of nine perspective, projective optical images of relatively well defined man-made objects located in areas of various background. Each of the algorithms was implemented in two versions, using either PC or MNF transform. If one were to present the results of each different experiment, the size of this dissertation would have been unnecessarily long. The procedure was to select and present only those results that were meaningful for comparison purposes. Most of the parameters that required optimisation, for example, size of a kernel, size of a sliding window, were optimised and only the optimal solutions were used for the final results. In some cases, in order to emphasise various tendencies, more than one result for a particular solution is given, for example, various rotational angles for the calculation of invariant moments, number of bands selected to reconstruct a PC or MNF transformed matrix.

In some cases, if an experiment generated a set of results, only the best was displayed, while the others were commented upon. The results were presented, where possible, in a graphic format accompanied by a detailed discussion. Only the most representative images were selected for display, together with their significant templates. The complete set of images used to test the algorithms are displayed in appendix D. Some of the images comprise similar content, for example, "UKF.bmp", "UKC.bmp" (and a few others, displayed in appendix D), but significantly different view angles and altitudes from where they were recorded, generated different looking images. No attempt was made to perform any geometrical corrections. Therefore, they were considered to be independent images, and treated as such. More than one template was selected for some of the images and the true match location searched, but only one representative template was presented for each image. The two algorithms used for performing automated scene matching are presented in two separate sections. The results generated by similar sets of experiments, required to assess the performance and accuracy of the algorithms are described, displayed and commented upon for each algorithm, when appropriate. Both algorithms are described below, having been applied, their performances are discussed and compared.

Faster computational solutions to automated scene matching problem are also described at the end of the chapter.

6.2 Multiband Image Algorithm using Moments (MBIMOM)

The first algorithm to be discussed and implemented for performing scene matching utilises invariant moments as feature descriptors. The algorithm is implemented in two versions: one using PC transform, the other using MNF transform, both with the aim of reducing the dimensionality of the initial data and removing further noise. Extensive information and experiments about the use of invariant moments as features were already presented in section 4.2, while sections 5.1 and 5.2 described the theoretical principles and implementations of the two transforms, PCA and MNF, respectively.

To clearly illustrate the entire process, it is necessary to describe now the algorithm operates and draw a block diagram. The algorithm was named *Multiband Image using Moments* (MBIMOM). Nedeljkovič (1997) first described one of the versions of this algorithm, using MNF transform.

After the original images were pre-processed in order to reduce the noise, using the appropriate filters described in chapter 3, or by applying histogram processing if it is necessary, the algorithm starts by forming multiband images using the single band template image and the scene image respectively, as initial data. Therefore, the scene image, which is the real image, and the template image, that is considered the synthetic picture, independently feed the algorithm. The template image is a subset of the original non-degraded scene image. For each image, a set of seven invariant moments is calculated and used as features, employing the "raw" data contained in an $n \times n$ neighbourhood, where n denotes the size of the sliding window used to calculate the invariant moments. After the multiband images were formed, they are processed further, using either PC or MNF transforms. In this way, the bands become mutually uncorrelated. Redundant, irrelevant information or noise distorting the original data are discarded. Before performing the correlation, the bands transformed by PC or MNF are analysed and an automatic truncation of the eigenvalues sequence, in an optimum mean square error sense, is performed. Finally, the transformed template image is correlated with the transformed scene image and the maximum value of the normalised correlation coefficient indicates the location of the template within the scene image. Figure 6.1 visualises the block diagram of this algorithm.

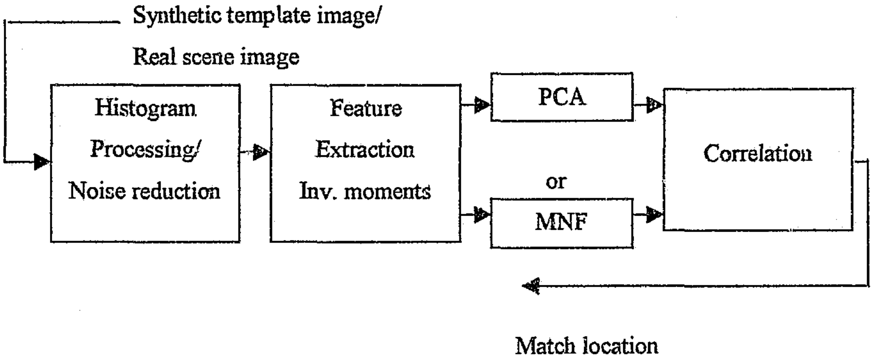
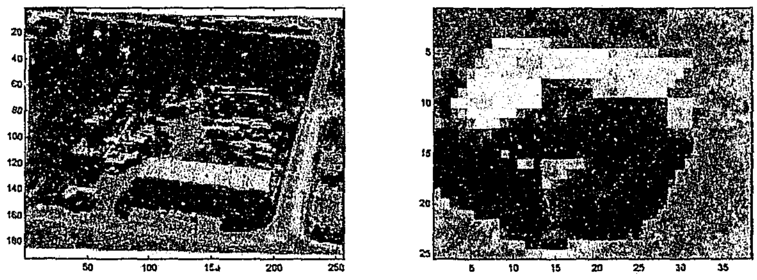


Figure 6.1 Block diagram of MBIMOM algorithm

Once the algorithm was designed, it was necessary to validate its robustness in the presence of various perturbations of the original signal image through an extensive set of experiments. Its performance is assessed by the maximum numerical value obtained for the normalised correlation coefficient. The accuracy of the algorithm is indicated by the fact that it is able to generate the true match location for the template image or generate a false fix. The computational speed is monitored by taking into account the computational time required to perform the scene matching from the initial phase of extracting the feature descriptors until the last stage, that of correlation. When the results are evaluated two characteristics are taken into account: the value of the correlation peak and the rightness of the match location.

This algorithm and the one presented in the next section were tested on all images incorporated in appendix D. However, only the images displayed in figures 2.2 and 2.3 are used for discussions. The third image, displayed in figure 6.2, together with its designated template, was also selected for analysis because it contains similar objects embedded in a natural scene. If one of these objects is chosen to represent the content of a template, it is vital to check whether the algorithms proposed are able to detect the true match location. For objects whose contents are similar, in terms of energy or size the algorithms in use often generate multiple false fixes. It was mentioned at the beginning of this chapter that the algorithms proposed must generate a unique match location for any pair template-sensed image. All three images, presented in figures 2.2, 2.3 and 6.2a) are referred to as synthetic scene images. They are sometimes named original images.



a) b)

Figure 6.2 Optical image BCE, headquarters, Bambous, Mauritius, "BCE.bmp"

a) original image with its designated template

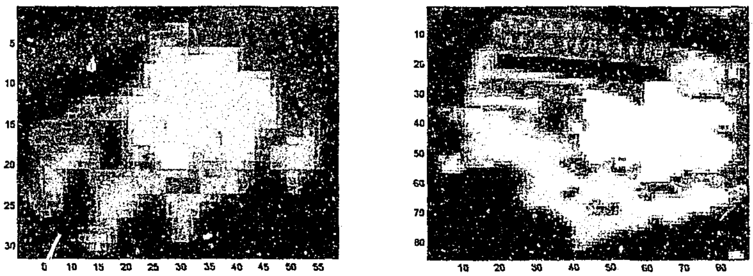
b) template displayed

The algorithm is first tested for images that could be possibly recorded by different sensors, introducing white Gaussian noise and blurring, without the presence of any other distortions. Therefore, the synthetic pictures were blurred using a 3 x 3 Gaussian filter with main lobe width of 0.5 and corrupted further by adding white Gaussian noise of variance equal to 0.01. These degraded images, "UKF.bmp" and "UKC.bmp" were displayed in figure 3.2. The designated template images selected to be the reference images are displayed in figures 6.2b) and 6.3.

The size image matrices for all synthetic images selected is 193 x 256. The templates designated for the three scene images are characterised as following:

a) template "UKFT.bmp" for scene image "UKF.bmp" is of size 31 x 58, with coordinates of the top left corner (97, 98);
b) template "UKCT.bmp" for scene image "UKC.bmp" is of size 85 x87, with coordinates of the left upper corner (74, 79);
c) template "BCET.bmp" for scene image "BCE.bmp" is of size 25 x 38, with coordinates of the left top corner (135, 29).

The axes attached to the images are in matrix coordinates. The coordinate system origin is at the upper-left corner. The *i*-axis is vertical and is numbered from top to bottom. The *j*- axis is horizontal and is numbered from left to right (Matlab™, 1997).



a) b)

Figure 6.3 The templates selected for performing scene matching:

a) template "UKFT.bmp"

b) template "UKCT.bmp"

All algorithm codes were implemented in Matlab^R, version 5.2b, using a Pentium II 350 MHz processor, with 256 MB RAM.

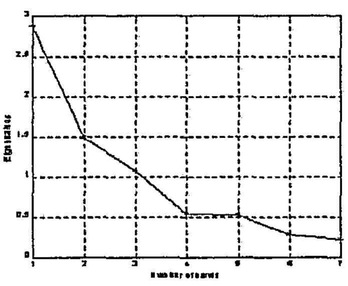
The first decision to be made before testing the algorithm is to decide upon the size of the sliding window used to extract the invariant moments. The same window size must be maintained for the calculation of invariant moments for the template and the sensed image. The same sliding window size is also used for performing the correlation. The best compromise between the computational time, performance measured by the correlation peak amplitude and accuracy is obtained with a sliding window size of 11 x 11. This value was established through experimentation.

All the following experiments were conducted, independently, to assess the MBIMOM algorithm implemented in two versions: using either PC or MNF transforms.

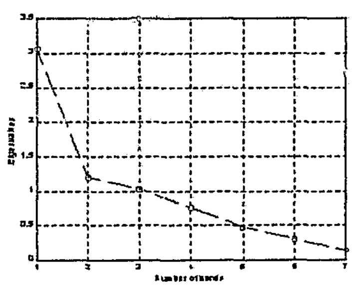
If all seven bands are used for the correlation process, the computational time is high, without the benefit of increased performance. It was mentioned in chapter 5 that some of the bands contain redundant or irrelevant information and noise. If fewer bands are used, this results in discarding some of the redundant information and noise, with a gain in performance and computational speed. However, it is important not to sacrifice performance in favour of only decreasing the computational time. There are two methods to select the number of bands used in correlation. One of them is to scrutinise the eigenvalues and decide which of them are associated with bands containing mostly noise and redundant information. Usually, a drop in the series of characteristic values signifies that the associated bands can be discarded, without loss of valuable signal information. This is an empirical approach and rather dangerous for the accuracy of the algorithm. It can generate a higher correlation peak, at the expense of delivering a false fix. This is the reason why this method was not used. Another method is to automatically truncate the eigenvalues sequence, in an optimum mean square error sense. This method was described in sections 5.1 and 5.2 respectively, for PC and MNF transforms and was adopted during this work. The beneficial result of performing an automatic truncation of eigenvalues sequence is that the computational speed increases, and well the correlation peak amplitude. Through this procedure the true match location is also delivered.

Before implementing the MBIMOM algorithm, all three degraded images were denoised using *Symlet* wavelets with three vanishing moments, *sym3*, four decomposition levels, soft thresholding and $k = 1.2$. Despite the fact that the associated templates are not degraded by various perturbations, they are always denoised, using the same method, for similarity reasons.

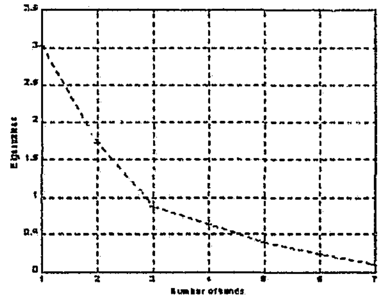
The eigenvalues associated with seven band covariance matrix image reflect the contribution of each principal component to the total variance of the data. Figure 6.4 displays the eigenvalues generated by PC transform, in the case where all three images were used. The covariance matrix of the scene (sensed) image is formed by using the invariant moments previously calculated. The invariant moments calculation and correlation were performed using a sliding window size of 11 x 11 for all images.



a)



b)



c)

Figure 6.4 Eigenvalues generated by solving the characteristic equation associated with each covariance matrix, PC transform

a) for image "UKF.bmp"

b) for image "UKC.bmp"

c) for image "BCE.bmp"

Table 6.1 displays the eigenvalues sequence and reflects the influence of the number of bands selected for correlation on the value of the maximum normalised correlation coefficient. It also incorporates the fractions of the total data variance that are "captured" by different characteristic values. The eigenvalue contribution found in table 6.1 was calculated in the following manner: a fraction of the total data variance is reflected by the corresponding eigenvalue(s), and the quotient λi / $Trace(\Lambda)$, $i=1,2,...,7$ (using equations (5.4) and (5.7)), the result expressed in percents), measures the contribution of (a) particular eigenvalue(s) to the total variance of the data.

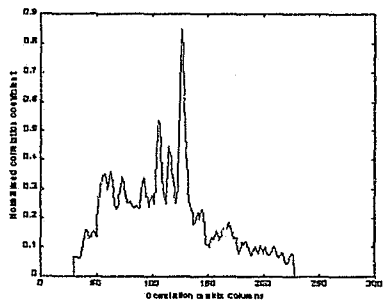
The automatic truncation of eigenvalues series, done in an optimum mean square sense must be in accordance with equation (5.11). This implies that the sum of the eigenvalues discarded, scaled by the number of bands, must be approximately equal to the noise variance. The noise variance of the correlation matrix, formed using the invariant moments calculated for each scene image, was estimated using Wiener filter, as described in section 3.2. For images "UKF.bmp" and "UKC.bmp" this procedure selected only the first five bands to be used for the reconstruction of PC transformed images. For image "BCE.bmp" the first four bands were selected.

Table 6.1 Eigenvalues and correlation peak values as a function of number of bands selected for correlation

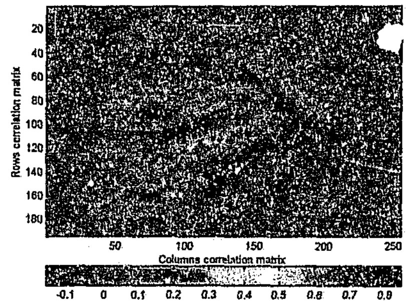
Image "UKF.bmp", PC transform used			Image "UKC.bmp", PC transform used			Image "BCE.bmp", PC transform used		
Eigenvalues			Eigenvalues			Eigenvalues		
λ_1	3.0600		λ_1	2.8724		λ_1	3.0310	
λ_2	1.1984		λ_2	1.4907		λ_2	1.7395	
λ_3	1.0380		λ_3	1.0697		λ_3	0.8749	
λ_4	0.7644		λ_4	0.5362		λ_4	0.6322	
λ_5	0.4838		λ_5	0.5275		λ_5	0.4014	
λ_6	0.3100		λ_6	0.2800		λ_6	0.2320	
λ_7	0.1454		λ_7	0.2235		λ_7	0.0889	
$\sum_{i=1}^{7}\lambda_i$	7.0000		$\sum_{i=1}^{7}\lambda_i$	7.0000		$\sum_{i=1}^{7}\lambda_i$	6.9999	
Number of bands	λ_i contr.	*R(u,v)*	Number of bands	λ_i contr.	*R(u,v)*	Number of bands	λ_i contr.	*R(u,v)*
5 bands	93.49%	0.8464	5 bands	92.81%	0.9128	4 bands	89.68%	0.7389
7 bands	100%	0.8472	7 bands	100%	0.9025	7 bands	100%	0.6500

Applying the automatic truncation of the eigenvalues sequence, resulted in an increase of maximum correlation coefficient values and a decrease in computational time.

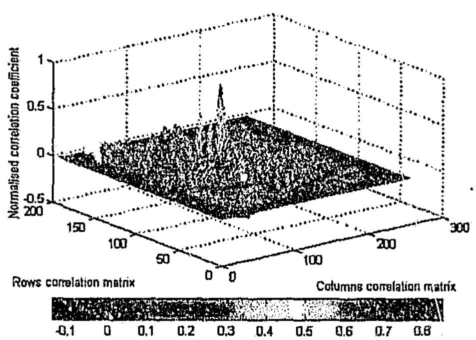
Figures 6.5, 6.6 and 6.7 display, using various visual representations, the correlation results in the case of each image, with all seven bands used and with PC transform implemented. It can be noticed that all main correlation peaks are narrow and distinctive, with a strong tendency of falling off sharply. In the case of image "BCE.bmp", where false peaks surrounded the main correlation peak, the algorithm also calculated the true match location. For images "UKF.bmp" and "UKC.bmp", the correlation peaks amplitudes are similar, in terms of numerical values, with those obtained by Hall (1979). For image "BCE.bmp", the value of the maximum normalised correlation coefficient does not reach an impressively high value, due to the similar content of the template to other parts of the scene and because the original image is of poor quality. Remember that it was degraded even further.



a)



b)



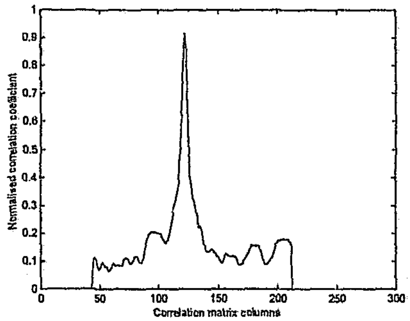
c)

Figure 6.5 Correlation results for the degraded image "UKF.bmp", PC transform, seven bands used

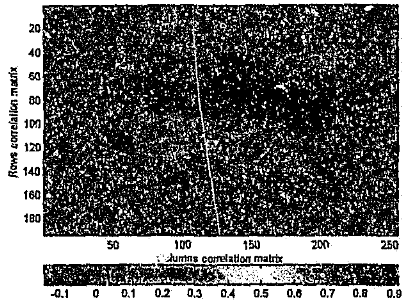
a) plot of normalised correlation coefficients across the correlation matrix columns

b) correlation matrix with associated colour bar

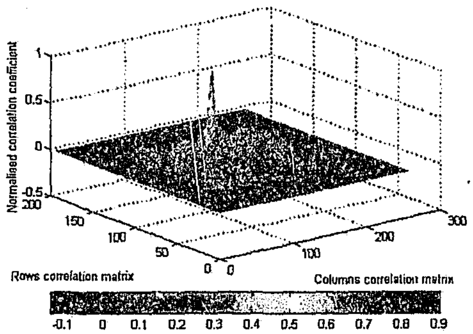
c) 3D visualisation of the correlation matrix



a)



b)



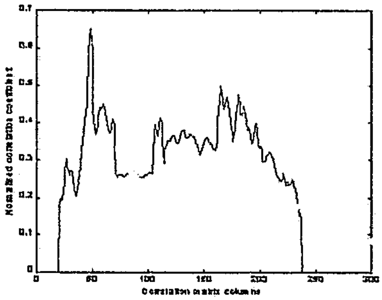
c)

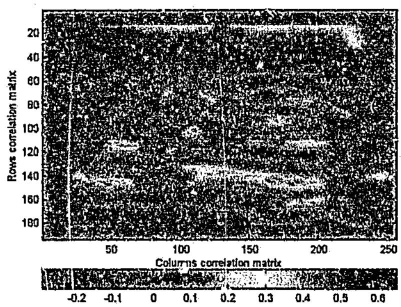
Figure 6.6 Correlation results for the degraded image "UKC.bmp", PC transform, seven bands used

a) plot of normalised correlation coefficients across the correlation matrix columns

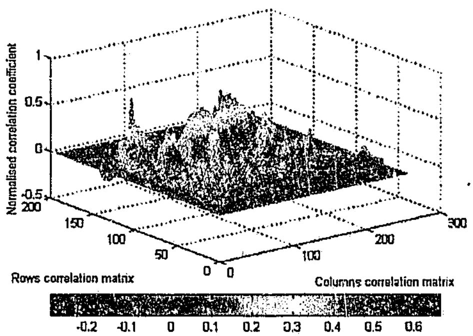
b) correlation matrix with associated colour bar

c) perspective plot of the correlation matrix





a) b)



c)

Figure 6.7 Correlation results for the degraded image “BCE.bmp”, PC transform, seven bands used

a) plot of normalised correlation coefficients across correlation matrix columns

b) correlation matrix with associated colorbar

c) spatial representation of correlation matrix

The values of the correlation matrix elements lay within the closed interval [-1,1]. This is in accordance with Cauchy-Schwarz inequality, and can be expressed as: $-1 \leq R(u,v) \leq 1$. They were generated according to equation (6.2). This means that a numerical value of this coefficient equals to one, or in the vicinity of this value, indicates a maximum positive linear correlation between the template and the reference location (u,v) within the scene image. A value close to zero gives a clear indication that the template cannot possibly be found at that reference location.

For both situations, using either all seven bands or only those selected by the automatic truncation of the eigenvalues series, the coordinates of maximum normalised correlation coefficients belong to the designated template and are as follows:

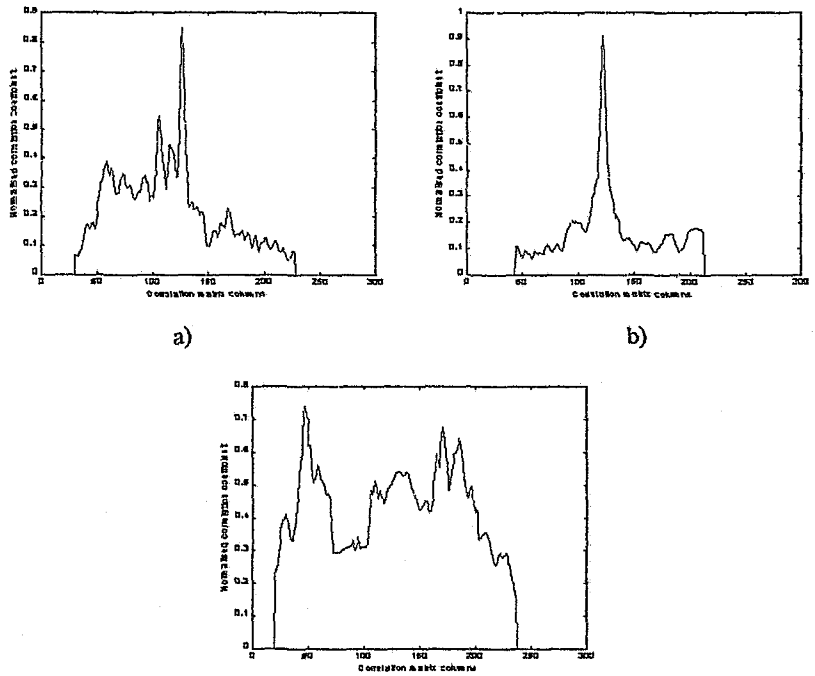
a) for image "UKF.bmp", the location of maximum $R(u,v)$ was (112,127) in matrix coordinates;
b) for image "UKC.bmp", the location of maximum $R(u,v)$ was (116,122) in matrix coordinates;
c) for image "BCE.bmp", the location of maximum $R(u,v)$ was (147,48) in matrix coordinates.

Different numerical values for the maximum normalised correlation coefficients for various images indicate a strong data dependency manifested by the correlation process.

It was proved that the truncation of eigenvalues, performed in an optimum mean square error sense, increases the correlation performance, decreases the computational time, and eliminates human intervention. It also does not generate false fixes.

Based on these considerations, the rest of the experiments will incorporate the automatic truncation of eigenvalues sequence when PC transform is used.

Figure 6.8 displays the plots of normalised correlation coefficients, for all images, when automatic truncation of eigenvalues series is executed.



a) b) c)

Figure 6.8 The normalised correlation coefficients for the degraded images, PC transform applied, selected number of bands used in the correlation

a) image "UKF.bmp", five bands selected

b) image "UKC.bmp", five bands selected

c) image "BCE.bmp", four bands selected

Comparing figure 6.8 with figures 6.5a), 6.6a) and 6.7a), it can be concluded that the shape of the correlation peaks does not change very much when only a selected number of bands is used for correlation. Image "BCE.bmp" constitutes an exception. The correlation peak becomes broader, using only four bands for correlation. This is due to a larger amount of intrinsic data information that is not used for correlation.

PC transform orders the bands in decreasing order of their contribution to the total signal variance. MNF transform was designed to order the bands in accordance with their SNR values, that is, in terms of image quality. Hence, the possibility to discard the bands containing mainly noise, without affecting the performance of the algorithm. The procedure was explained in section 5.2. It is important to mention that the fast computational method of *noise-adjusted principal components*, proposed by Roger (1994), was implemented, but for ease of communication, is referred to as MNF transform. A necessary condition for this implementation to be possible, is that the noise covariance matrix, Σ_N, must not be ill-conditioned. According to Gerald and Wheatley (1994), a quantitative measure of the degree of ill-condition of the coefficient matrix is given by the condition number, expressed as the product of two matrix norms:

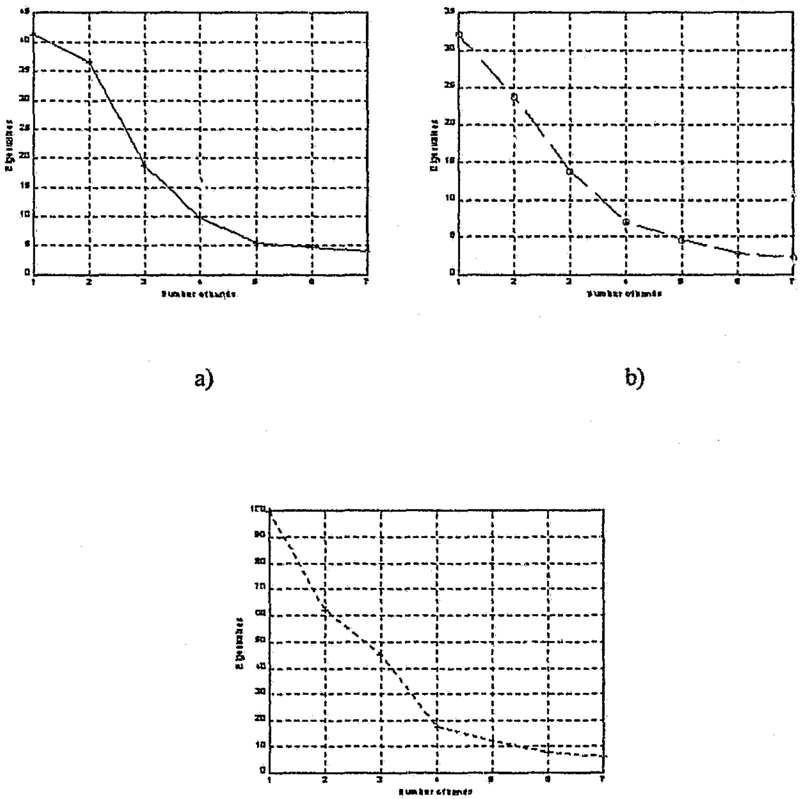
$$condition(\Sigma_N) = \|\Sigma_N\| \times \|\Sigma_N^{-1}\| \tag{6.3}$$

with the two norms expressed as matrix ∞-norms. If the condition number is extremely large, the matrix is ill-conditioned. This restriction was checked for all scene images and none of the noise covariance matrices was found to be ill-conditioned. Therefore, the above-mentioned transform was implemented.

The success of MNF transform is entirely dictated by a good estimation of the noise covariance matrix. Two methods were used for estimating the noise covariance matrix. One uses near-neighbour differences introduced in section 5.2, and described by equations (5.20) and (5.21). This method did not provide satisfactory results for all pictures tested. This is due to the fact that this method, known as minimum/maximum autocorrelation factors (MAF), relies on the assumption that the signal at any point is strongly correlated with the signal of neighbouring pixels, while the noise displays a weak spatial correlation, which could not always be the case. Moreover, when a shifted image is subtracted from its original version, the difference image contains not only the noise, but also intrinsic information about the signal image. The second method used to estimate the noise covariance matrix uses wavelets and was described in section 5.2. Due to better results generated, this method was adopted to estimate the noise covariance matrix required by the implementation of MNF transform.

The algorithm with MNF transform was applied to all synthetic images, first ensuring a measure for comparing the performance of the algorithm when applied to degraded scene images. The same sliding window size of 11 x 11 was used to extract the invariant moments and perform the correlation. By solving the generalised symmetric eigenvalue equation, expressed by equation (5.18), a set of eigenvalues for each scene image is obtained. The two covariance matrices involved in the calculation are those formed by calculating invariant moments for the degraded scene image and the estimated noise covariance matrix of the same scene image.

Figure 6.9 displays the eigenvalues obtained for the degraded images "UKF.bmp", "UKC.bmp" and "BCE.bmp".



a) b)

c)

Figure 6.9 Eigenvalues generated by solving the generalised symmetric eigenvalue problem for:

a) degraded image "UKF.bmp"
b) degraded image "UKC.bmp"
c) degraded image "BCE.bmp"

Table 6.2, similar to table 6.1, displays the eigenvalues for all three images and correlation results, using all seven bands of the images and only a limited number of bands. As in the case of PC transform, it is possible to truncate the eigenvalues sequence in an optimum mean square error sense. This implies that only the matching eigenvectors, associated with the selected eigenvalues, are used for the reconstruction of MNF transformed images, used then to perform the correlation.

Regarding the truncation of the series of characteristic values, for MNF transform, this must be done in accordance with equation (5.19), which indicates that the scaled sum of discarded eigenvalues must be approximately smaller or equal to one.

Table 6.2 Eigenvalues generated by generalised symmetric eigenvalue problem; Maximum normalised correlation coefficients for degraded images

Image "UKF.bmp", MNF transform used		Image "UKC.bmp", MNF transform used		Image "BCE.bmp", MNF transform used	
Eigenvalues		Eigenvalues		Eigenvalues	
λ_1	41.2955	λ_1	32.1256	λ_1	99.5815
λ_2	36.4510	λ_2	23.8134	λ_2	62.0287
λ_3	18.7455	λ_3	13.8765	λ_3	44.7379
λ_4	9.7986	λ_4	7.0645	λ_4	17.4038
λ_5	5.5393	λ_5	4.6100	λ_5	12.0540
λ_6	4.7044	λ_6	2.8684	λ_6	7.6005
λ_7	4.0817	λ_7	2.3022	λ_7	5.7476
Number of bands	$R(u,v)$	Number of bands	$R(u,v)$	Number of bands	$R(u,v)$
6 bands	0.7573	5 bands	0.8667	6 bands	0.5443
7 bands	0.7867	7 bands	0.8632	7 bands	0.5606

Figure 6.10 displays the plots of the normalised correlation coefficients for all three degraded images in both cases: all bands used for correlation and only the number of bands selected by the automatic truncation of the eigenvalues sequence.

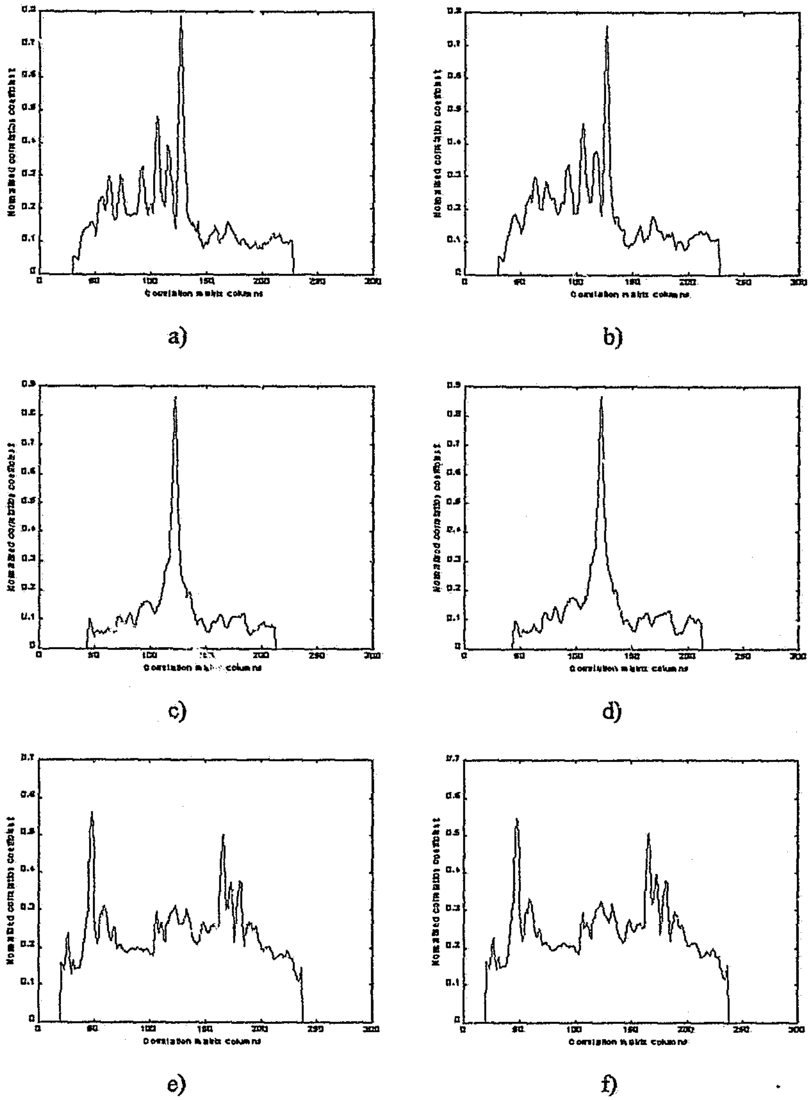


Figure 6.10 Plot of normalised correlation coefficients generated by MBIMOM algorithm, version MNF, applied to the degraded images

a) degraded image "UKF.bmp", seven bands used in the correlation

b) degraded image "UKF.bmp", six bands used in the correlation

c) degraded image "UKC.bmp", seven bands used in correlation

d) degraded image "UKC.bmp", five bands used in the correlation

e) degraded image "BCE.bmp", seven bands used in correlation

f) degraded image "BCE.bmp", six bands used in the correlation

From table 6.2 and figure 6.10, it can be concluded that MBIMOM algorithm with MNF transform generates reasonably good results, identifying the correct match location for all designated templates.

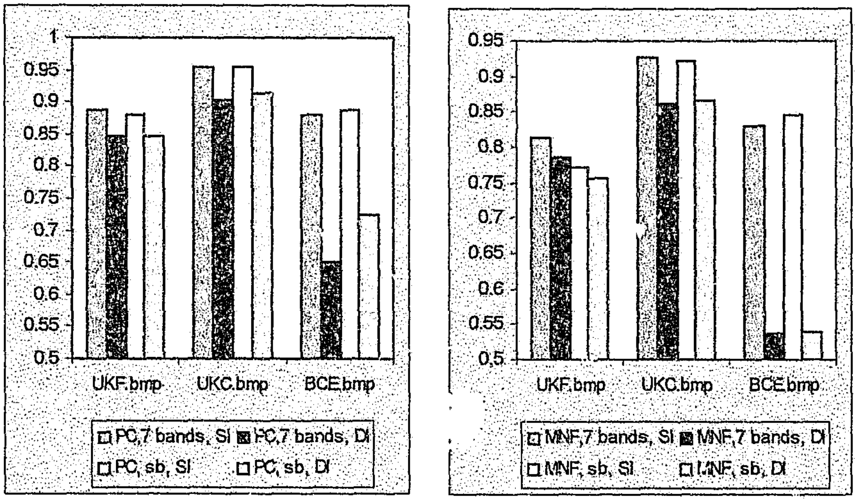
When noise reduction is performed through MNF transform, this involves discarding the components that are dominated by noise, and reconstructing the image back into MNF transformed space. This process is always associated with some loss of information, therefore the correlation peaks are lower than the case where PC transform is used. However, it can be noticed that all correlation peaks are sharp and distinctive.

The unexpected phenomenon is that truncating the eigenvalues does not generate spectacularly better results, as predicted. This implies that the last eigenvalues are not associated with bands with lower SNR power. Therefore, a part of the useful signal image was also discarded.

Recall that the noise covariance matrix estimates the noise affecting the covariance matrix formed by using invariant moments. It was proved in section 4.2 that they are sensitive in the presence of noise. Also, if the first invariant moments were distorted by noise, they created a cascading effect for the subsequent ones, as mentioned by Press *et al.*, (1986). This limitation, combined with an inaccurate estimation of the noise covariance matrix, resulted in the undesired effect that, instead of obtaining better results through an automatic truncation of eigenvalues sequence, MNF transform produced approximately the same results or poorer. However, the true match locations were calculated correctly for each template.

Finally, figure 6.11 displays the results obtained for the correlation process, performed by the MBIMOM algorithm in both cases: using either PC or MNF transform. It displays the correlation peaks amplitudes when the algorithm was applied to the synthetic scene images and to the degraded scene images. The figure incorporates, also, the results obtained by using all seven band images and only m band images, where m, with $m < 7$, represents the number of bands selected through an automatic truncation of the eigenvalues series, for PC and MNF transforms.

Within the chart legend, symbol "SI" signifies synthetic images, "DI" degraded images, and "sb" denotes a selected number of bands used for correlation.



a) b)

Figure 6.11 A comparison between the results generated by the MBIMOM algorithm

a) PC transform applied

b) MNF transform applied

A new set of experiments must be designed to further check the robustness of the MBIMOM algorithm in the presence of various distortions. When an image is recorded, it occurs fequently in real world situations that its quality is affected by different errors. The MBIMOM algorithm should be insensitive to realistic differences between the sensed and synthetic images. Therefore, it was decided to verify the performance and accuracy of the algorithm in the presence of the following errors:

a) scale changes, that is, the recorded image is dilated/ stretched;
b) translation errors generated by shifting the original image;
c) rotational errors;
d) intensity of grey scale changes, that is, contrast changes;
e) incomplete set of initial data, that is, missing data from the original image;
f) a combination of the above mentioned errors.

For the following experiments, the synthetic images were transformed in such a way as to simulate different adverse conditions. The algorithm was assessed using both transforms. All bands were used for correlation; this was decided due to unsatisfactory results generated by the automatic truncation of eigenvalues series for MNF transform. The template for each picture is kept the same size as the original, and at the same location inside the scene image, unless otherwise specified. They were selected from each transformed picture to preserve similarity. The same sliding window size is kept to calculate the invariant moments and perform the correlation, of size 11 x 11.

All the experiments described in section 4.2 to verify the robustness of invariant moments in the presence of various perturbations were conducted again. This time with the purpose of checking how the MBIMOM algorithm, which uses invariant moments as descriptive features, performs under various undesired, controlled changes.

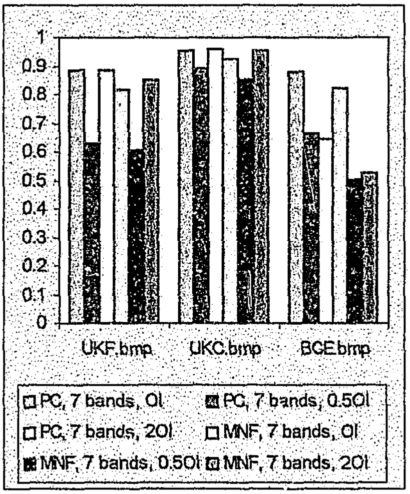
All the graphs displayed later contain the results generated by the algorithm, in both versions, with PC or MNF transform implemented, applied to the synthetic (original) images and degraded ones, as described in each case. This approach was adopted for ease of comparison.

For the purpose of the first experiment, each of the three synthetic images was resized to 50% and 100%, respectively. This implies that the corresponding templates were resized accordingly. Figure 6.12a) visualises the results of this test. The comments upon the correlation results follow after the below experiment.

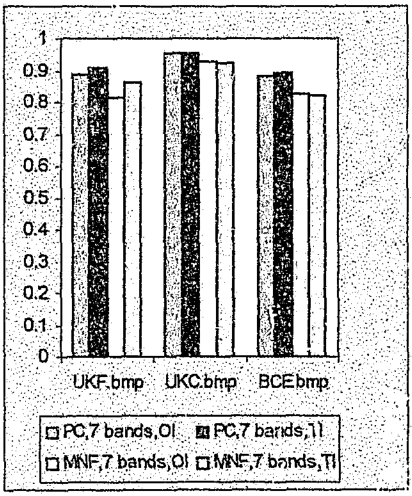
Another experiment is directed towards verifying whether MBIMOM algorithm is robust in the presence of translation errors. The original images were shifted ten pixels to the right and ten pixels down, and due to this operation, the original size of the images was increased. The size of the modified images was not kept identical to the synthetic ones, as this would have generated images with sections of the original content discarded. Figure 6.12b) displays the correlation results for the images distorted by translation errors.

When images are shifted, rotated or parts of them discarded deliberately to simulate incomplete set of data, the process generates new image matrices. Some of their entries must be filled with new coefficients. Two approaches are commonly used: one is to introduce intensity grey level values close to zero, but not zero (in order to avoid division by zero during different steps of the algorithm). The second approach is to allocate the value of one, to all newly created coefficients. The second approach was adopted, each time when appropriate, in order to avoid the possibility of changing the image matrices into ill-conditioned ones. This can be explained in the following manner: if a quantity has an error in it resulting from rounding-off, the division of that quantity by a number near zero amplifies the rounding-off error, that is, the closer the divisor is to zero, the greater the amplification. (Anton, 1987).

Depending upon the approach taken, the final numerical results would vary because the total variance of the original signal image was modified due to this operation. Therefore, the values of normalised correlation coefficients will reflect not only the changes due to different simulated errors, that is, translation, rotational, incomplete set of initial data, but, also, the changes in the original signal, due to the above-described procedure. However, what is more important is to verify that the algorithm is capable of generating the match location. Moreover, the correlation peaks amplitudes generated must be of realistic values.



a)



b)

Figure 6.12 Maximum normalised correlation coefficients generated by MBIMOM algorithm for images:

a) degraded by scale changes

b) degraded by translation errors

Within the chart legend OI, 0.5 OI and 2 OI means full, half, and double size of the synthetic (original) image, respectively. "TI" inside the second chart legend signifies translated images.

It can be noticed that, when the images were changed to half size, compared to full size, the maximum correlation coefficient values, for both transforms, are lower than for double size images. This is explained by the fact that a part of the intrinsic information is discarded when the size of a picture is reduced. For images bigger than the original ones, the re-scale takes places by retaining all the original information, hence higher values for the correlation peaks. Concerning the significance of the results, they reflect not only the effect of scale changes, but also the degradation of images introduced by bicubic interpolation method, used to reduce aliasing effects. In the presence of translation errors, the algorithm proved being robust and accurate. The algorithm generated, in each case, narrow, distinctive correlation peaks and detected the true match location. Therefore, it can be concluded that the proposed algorithm, MBIMOM is robust in the presence of scale and translation errors.

.The goal of the following experiment is to check the robustness and accuracy of the MBIMOM algorithm in the presence of rotational errors. Therefore, all three images previously used were rotated by $+5^0$ and -10^0 (with counter-clockwise direction representing a positive rotation). Due to the fact that the rotation was performed using the bicubic interpolation method to reduce aliasing effects, it can be considered that the resultant images suffered a higher degradation, the combined effect of interpolation method and rotational errors. The sizes of the new images have increased. For images rotated by $+5^0$, the new image matrices are of size 217 x 273, while for the images rotated by -10^0 the image matrices size has became 237 x 289. This approach was adopted with the goal of preserving all original information and still performs the rotation. Recall that the empty positions of new matrices generated by rotational process were filled with numerical values equal to one.

It is important to comment further on the practical side of the rotational process Certainly, when images are rotated, trying to retain the same coordinates of the templates, inside the scene images, results in choosing new templates with different contents. This means that it should not have been possible to assess the performance of the algorithm for similar templates. Therefore, it was decided to maintain, approximately, the same contents of the new selected templates. The sizes of rotated templates were kept equal to the sizes of the original templates. The new coordinates for rotated templates were found through trial and error.

A good measurement of the similarity of two images, in terms of their intrinsic information, is given by comparing their total variances. This can be calculated in two ways and both approaches must lead to the same result. In one approach, the total variance of the data is the sum of the variances on the diagonal of the covariance matrix attached. This can be expressed as:

$$\{total\ variance\} = tr(C_t) \tag{6.4}$$

where C_t represents the covariance matrix of the old/new template. In the second approach, the eigenvalues generated by the characteristic equation solved for the template covariance matrix are summed:

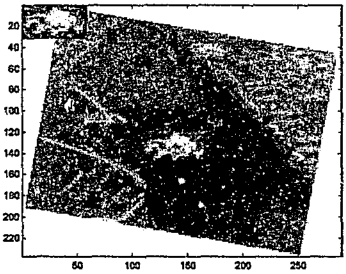
$$\{total\ variance\} = tr(C_t) = \lambda_1 + \lambda_2 + \ldots + \lambda_n \tag{6.5}$$

with n representing the highest size (rows or columns) of the template matrix.

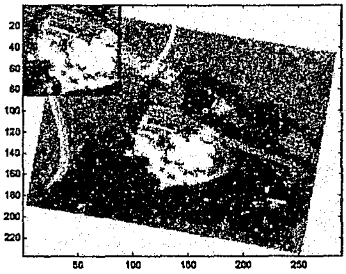
The ratio between the total variance of the rotated template and the total variance of the original template measures the information content of the original template signal data captured by the rotated template. Each new template was chosen such to incorporate between 95 % and 99 % from the original template signal content. The algorithm, in two versions, using either PC or MNF transform generated the same coordinates for the new match locations for the designated templates.

The images displayed in figure 6.13 represent the original images rotated by -10^0, together with the corresponding templates. A red star, inside the rotated image marks the position of the maximum correlation peak, for each image.

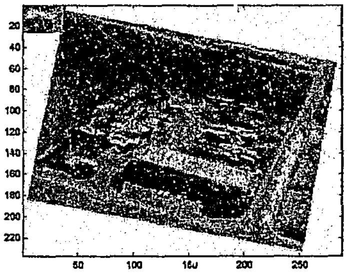
The numerical values generated by the MBIMOM algorithm in the presence of rotational errors for the correlation peaks amplitudes are shown in figure 6.14. Inside the chart legend, "RI" signifies the rotated image.



a)



b)



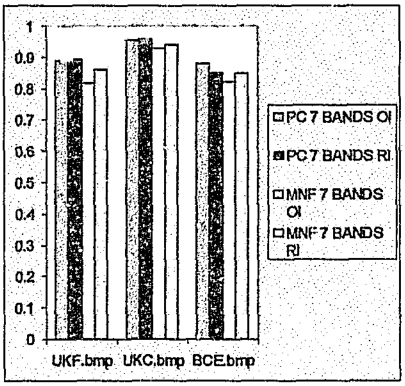
c)

Figure 6. 13 Images rotated by -10^0 and the corresponding templates

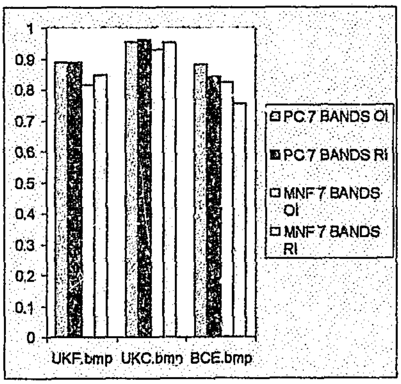
a) image "UKF.bmp"

b) image "UKC.bmp"

c) image "BCE.bmp"



a)



b)

Figure 6.14 Values of maximum correlation coefficients for rotated images

a) images rotated by +5^0;

b) images rotated by –10^0

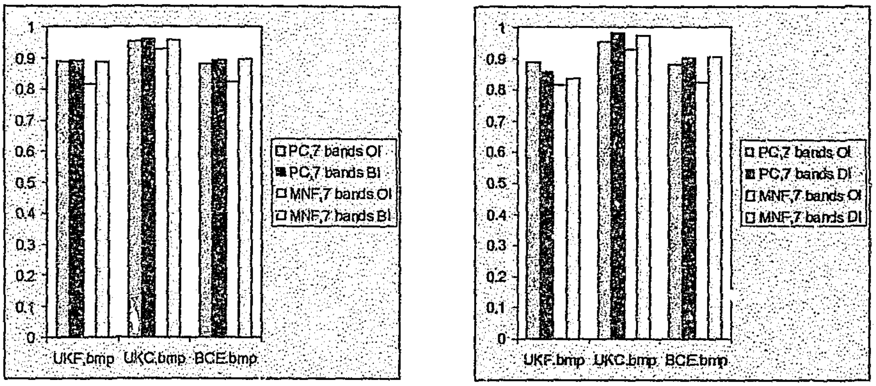
Analysing the graphs one can observe the good results generated by this experiment. The explanation can be found in the special properties exhibited by PC and MNF transforms. PC transform is used to map signals into a new coordinate system, whose orthogonal eigenvectors are pointing in the directions of the largest signal variances as basis vectors. The new axes generated by this transform will be directed towards the largest and second largest variances of the coordinates of the original image points. The new axes should be close, if not identical, for the same images at various different angles. MNF transform, which is similar to PC transform, with the difference that it orders the eigenvalues in terms of SNR power instead of signal variances, possesses the same property for the new axes.

Therefore, any algorithm incorporating PC or MNF transforms should be invariant to small rotational errors. For some images, the numerical values of normalised correlation coefficients were slightly higher that those generated by the original images. This is explained by the fact that the signal variances of rotated images were changed, due to incorporating new matrix entries equal to ones, when the rotational process created the new matrix images, 'framing' the original images.

The next experiment evaluates the performance of the MBIMOM algorithm when the original images are subjected to contrast changes. The images were transformed so as to generate 30 % brighter pictures than the original, and 30 % darker images than the originals. The process was described in section 4.2. The images brighter than the originals generated high and narrow correlation peaks, due to a wider dynamic range, hence a higher signal variance.

In the case of darker images, the algorithm generated only false fixes, due to low contrast features. The solution to this problem was to pre-process the images by applying histogram equalisation. Section 2.1.1 described thoroughly how this method works. When histogram equalisation was applied to darker images, the dynamic range was broadened, generating a good performance of the algorithm. It also produced very sharp peaks, in each situation. Histogram specification may be also used to obtain a similar effect.

Figure 6.15 shows the correlation results obtained by using the MBIMOM algorithm, applied to brighter and darker images, for all pictures used previously.



a) b)

Figure 6.15 Correlation results generated by the MBIMOM algorithm applied to all images subject to intensity changes

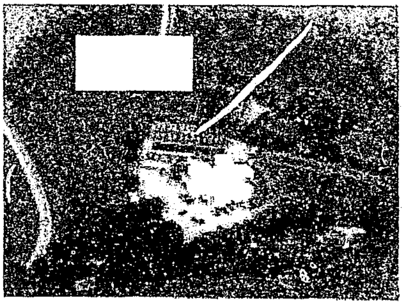
a) images 30% brighter than the originals

b) images 30% darker than the originals

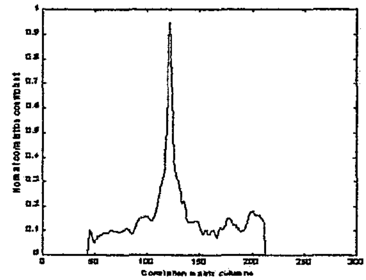
The symbol "BI" within the legend represents the bright images, symbol "DI", dark images.

For darker images, it is compulsory to apply histogram processing as the first step, before running the algorithm, to generate a more balanced distribution of grey levels. Therefore, it is advisable that before processing the sensed image, to compare its histogram with that of the reference image. If there is a large discrepancy between the two histograms, it is necessary to apply histogram processing to the sensed image, as the first step, otherwise the algorithm will fail to identify the true match location. Histogram processing can be applied as the first step of preparing the real image for the algorithm, in all situations, but at the price of obtaining a broader correlation peak.

The following experiment emphasises the robustness of the algorithm when it uses an incomplete set of initial data to perform scene matching. To simulate this situation, a section from image "UKC.bmp" was removed. The size of this section was 35 x 75, with coordinates of the top left corner of (20,45). Figure 6.16 displays the new image and the plot of normalised correlation coefficients across the correlation matrix columns, when PC transform was used. All seven bands were retained for performing the correlation.



a)



b)

Figure 6.16 Image "UKC.bmp" tested for missing data

a) the image with a part of it missing
b) the plot of normalised correlation coefficients, PC transform applied, seven bands used for correlation

The highest value for the normalised correlation coefficient was *R(u,v) = 0.9412,* with PC transform and *R(u,v) = 0.9232,* for MNF transform applied. The true match location was calculated and the correlation peak sharp (see figure 6.16). It can be concluded that the algorithm is not affected by missing data, as long as they do not belong to the designated template.

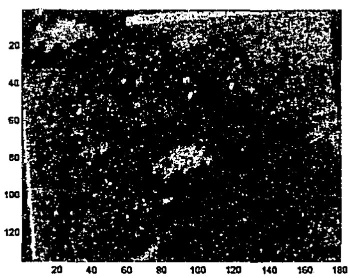
Finally, the last experiment is conducted to investigate the performance of the algorithm tested for one of the most severe possible real world scenario. The following operations were performed to degrade the synthetic image "UKF.bmp". The following set of operations simulate a sensed image of poor quality and test the performance and accuracy of the proposed algorithm for extremely severe conditions.

1. The original image was shifted 10 pixels to the right and 5 pixels down to simulate translation errors; new image size became 202 x 260;
2. The new image was resized to 135 x 180 to induce scale changes;
3. It was rotated by $+5^0$ for the purpose of introducing a rotational error. The size of the new image was kept as in step 2. Keeping the same size for the image matrix, 135 x 180, generated an incomplete initial data set,also.
4. The image was transformed such that to become 20% darker than the synthetic image. Contrast changes were simulated this way.

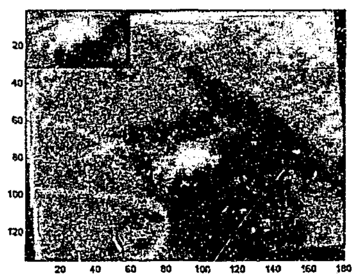
5. The image was degraded further by blurring it using a Gaussian filter of main lobe equal to 0.5;
6. Finally, the last step was to add white Gaussian noise of variance equal to 0.01.

After step 4, the template was selected. The subset used as template was kept the same size as the original template, but with the coordinates of the top left corner (69,69). The new template conveys 82,36 % from the total signal variance of the original template.

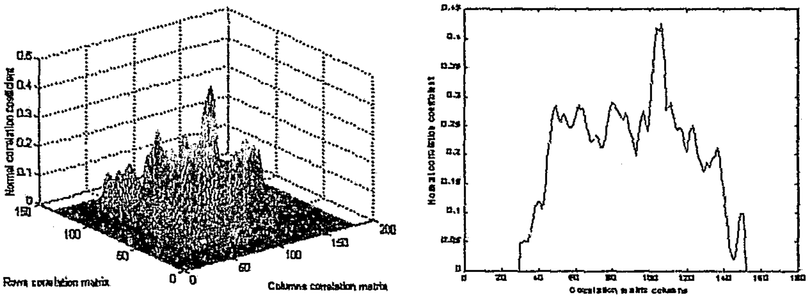
Figure 6.17a) displays the degraded image obtained through the implementation of the above mentioned operations, together with its template. One would expect that in the presence of such harsh degradation, the algorithm would not be able to deliver the true match location. Before applying the MBIMOM algorithm, both, the image obtained through steps 1 to 6, and the template, selected after step 4, were pre-processed. Histogram equalisation was applied to increase the dynamic range, followed by denoising using wavelets, as described at the beginning of this section. It was not necessary to denoise the template, but this step was performed to keep similarity. Figure 6.17b) displays the degraded image, together with its template after all pre-processing techniques were performed. A red star marks the location of the maximum normalised correlation coefficient. Finally, figures 6.17c) and 6.17d) display the graphical results of the correlation process. For PC transform used and all seven bands kept for correlation, the numerical value for $R(u,v) = 0.4247$, while for MNF transform, the algorithm generated a numerical value for the maximum normalised correlation coefficient $R(u,v) = 0.3043$. It can be noticed that the correlation peak is wider and much lower, surrounded by multiple false correlation peaks. However, the algorithm found the true match location.



a)



b)



c) d)

Figure 6. 17 Image "UKF.bmp" tested for a variety of adverse conditions

a) the degraded image, together with its template displayed

b) th ·ı re-processed image and template, the location of the maximum normalised correlation coefficient

c) spatial representation of the correlation peaks;

d) plot of normalised correlation coefficients across correlation matrix columns

After such a diversity of tests, used to check the performance of the MBIMOM algorithm, it can be concluded that:

- The MBIMOM algorithm using PC transform generated very good correlation results, comparable with those reported in the literature, and for some images even higher.
- The automatic truncation of the eigenvalues sequence, for PC transform, performed in an optimum mean square error sense, increased the correlation peaks amplitude and decreased substantially the computational time.
- The MBIMOM algorithm using MNF transform generated numerical results lower than those obtained by using PC transform, but comparable with those reported in the literature. It must be emphasised that the correlation results obtained by different researchers were generated using other scene matching methods.
- The automatic truncation of eigenvalues series for MNF transform did not generate better results, compared with those obtained by using all image bands. This is due to an inaccurate estimation of the noise covariance matrix, as mentioned.

- The algorithm was able to perform under a variety of adverse situations, always delivering the true match location for the designated templates. Therefore, it can be used as a suitable tool for performing scene matching, at least in the case of optical images.
- The MBIMOM algorithm is not very sensitive to different degradations of the image, with the condition that the right sequence of pre-processing steps is implemented, if necessary;
- The algorithm generated a unique match location for each pair of sensed-template images.
- It is computationally quite expensive, due to the calculation of invariant moments.
- The robustness and accuracy of the MBIMOM algorithm in the presence of severe perturbations of the original image is due to its specific design that uses invariant feature descriptors and transforms that exhibit special properties, such as PC and MNF transforms.
- The accuracy of the algorithm may be expressed as the probability of correctly discriminating a template in the image from background and system noise, known as *probability of detection* (Ratches *et al.*, 1997). The algorithm scored a detection probability equal to one. Expressed in a different manner, the algorithm generated a *false alarm probability* equal to zero. This measures the probability of an error in detection.
- Another measure used for scene matching problem is so-called *consistency*. This is a measure of how often a given scene matching algorithm implementation gives the same declaration for images having the same content. The algorithm may be considered consistent, based on the experimental results.

A second algorithm was developed for scene matching and is presented afterwards. The reasons for developing an independent second algorithm are presented hereafter. From the experience and the results obtained by different researchers in the field of image processing and pattern recognition, it was concluded that it is always a wise approach if more than one non-equivalent way of processing data can be found. Therefore, it is worthwhile developing as many imaging techniques as possible, provided that each is viable in its own right. Another more logical reason for developing more than one, independent algorithm, is that no single technique is likely to be superior in all circumstances. An even more important reason is that improved results can often be obtained by combining different methods.

6.3 Spatial Multiband Image Algorithm (SMBI)

The second algorithm presented, implemented and its performance and accuracy assessed under various conditions, uses a similar procedure to the MBIMOM algorithm, but does not rely on feature extraction for performing scene matching. Hence, a much shorter computational time is required. It is called the *Spatial Multiband Image* algorithm (SMBI), for reasons that will unfold shortly. Willis (1997) was the first to develop one version of the algorithm, using MNF transform. This algorithm, compared with the previous one, uses not only the information contained in the grey level pixel, but also makes use of pixel's relation to its neighbouring pixels, that is, the spatial connection of the pixel with its neighbours. Subsequently, the resulting correlation peak is unique, without the presence of any false peaks. A sliding window of chosen size, $n \times n$, with $n << M$ and $n << N$, where $M \times N$ denotes the size of the sensed (scene) image matrix, moves across it, extracts and generates neighbourhood vectors from the entire image. The same operation takes place for the template image. The generated vectors from the sensed image are later correlated with the similar vectors generated from the template image.

As already mentioned in section 6.1, the original image matrix contains the maximum amount of information. It contains not only the brightness values of the pixels, but also the spatial correlation of the pixels with their neighbours. This is exactly what this algorithm uses; the spatial correlation of the pixels within an $n \times n$ vicinity. The neighbourhood should be chosen as small as possible, due to computational constraints, but big enough to take advantage of the spatial correlation neighbourhood. However, using the 'raw' image as feature space would increase the dimensionality of the data, resulting in a very expensive computational procedure. It would also generate poor results, due to redundant or irrelevant information and noise. In order to overcome these disadvantages, the algorithm incorporates either Hotelling or MNF transform. This way the pixels are decorrelated, and the noise removed. These transforms operate on a one-dimensional vector, while the neighbourhood is two-dimensional. The next step is to convert the $n \times n$ neighbourhood into a vector of length n^2. Concatenating the data either column by column, in which situation the correlation between columns is ignored, or row by row, which implies that the correlation between rows is neglected, generates a new n^2 vector. Either way, using a sliding neighbourhood window of the above size generates a resulting vector. One can "visualise" the new transformed image as an n^2 band image created from the pixel neighbourhood. This approach will generate $(M-n+1)(N-n+1)$ vectors, for the whole sensed image. The same procedure is applied to the template image.

The covariance matrix of this new generated matrix of neighbourhood vectors, C_X, (refer to equation (A.6)) is formed and used to perform a PC or an MNF transform. When a PC transform is used, the eigenvector basis generated by solving the characteristic equation for the covariance matrix is further used to reconstruct the scene and template images, required for the final stage of the correlation. When MNF transform is implemented, the eigenvector basis is generated by diagonalising the covariance matrix, Σ_D, through a generalised symmetric eigenvalue equation (refer to equation (5.18)). Therefore, it is necessary to estimate the noise covariance matrix, Σ_N. This estimation is done based on either near neighbour difference or on wavelets transform. Both methods were presented in sections 5.2 and 3.5.3 respectively, and also used for the previous algorithm. There is a difference in the way the noise covariance matrix is estimated for this algorithm, compared to the previous noise covariance matrix estimation for the MBIMOM algorithm. For the SMBI algorithm the noise image is found first and the vectors are extracted afterwards. The noise image is obtained by thresholding the degraded scene image using wavelets. Detail coefficients below the threshold are used in the wavelets reconstruction of the image. Then, the sliding neighbourhood window extracts vectors of length n^2 from the wavelets reconstructed noise image. The use of distinct windows to extract the neighbourhood noise vectors decreases the computational time, without affecting the noise estimation accuracy. The next step is to create the noise covariance matrix from the thresholded noise vectors image (using wavelets).

The MNF transformation matrix must be generated using only the eigenvectors matching the eigenvalues above the noise. Both, scene and template images, must be reconstructed using only these eigenvectors, in order to preserve similarity. Following this procedure, redundant information and noise are discarded from the original signal data. These results in an improved correlation performance and a shorter computational time. These MNF transformed multiband images are then used to perform the correlation in the same way as for the MBIMOM algorithm. To make everything clearer, the description of the way the SMBI algorithm works can be summarised as follows:

1. the sensed (scene) image is transformed into a matrix of neighbourhood vectors, using a designated sliding neighbourhood window;
2. the same procedure is applied to the template image that is selected from the synthetic, non-degraded scene image;
3. the covariance matrix, C_X, is calculated from the matrix generated at step 1 and used to perform a PC transform on the sensed image data and template image data; alternatively, the covariance matrices, Σ_D and Σ_N, are calculated from the matrix generated at step 1 and estimated noise covariance matrix respectively, and used to perform an MNF transform on the scene image data and template image data[1];
4. PC or MNF transformed denoised multiband scene and template images are used for correlation.

Figure 6.18 depicts the block diagram for SMBI algorithm.

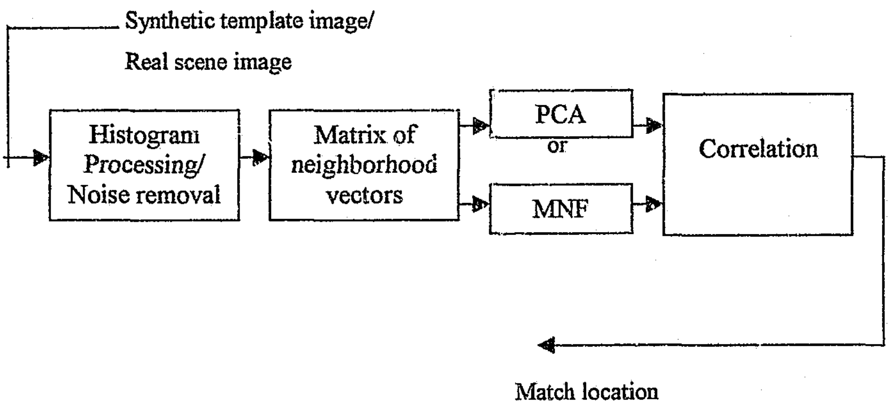


Figure 6.18 Block diagram for the SMBI algorithm

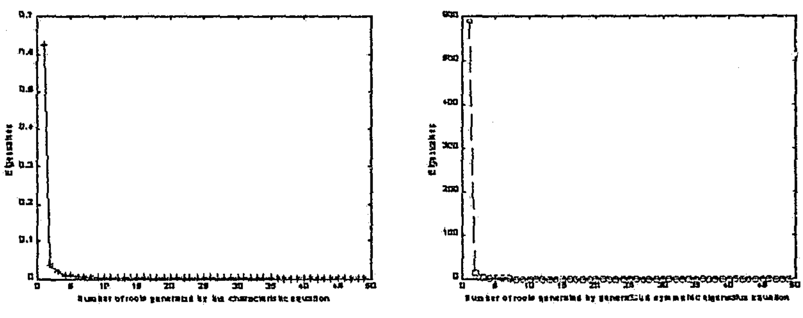
After the algorithm was presented theoretically, it was necessary to check its performance, accuracy and robustness. It is necessary, also, to assess the computational speed of the algorithm. All the experiments conducted to assess the MBIMOM algorithm will be replicated, using the same synthetic and degraded images. The templates used are those displayed in figures 6.2b), 6.3a) and 6.3b), respectively.

[1] The MNF transformation matrix should be formed using the eigenvectors that lie above the noise.

The first task is to establish the size of the sliding neighbourhood window, which must be chosen so as to provide the best compromise between the quantitative performance of the algorithm and the computational time required. A very small window size does not exploit the relation of the pixel with its neighbours. A very big window size is computationally expensive. It was found through experimentation that the best sliding neighbourhood window size is 7 x 7.

Regarding the number of bands used to recreate the image, using either PC or MNF transform, brings back into discussion the number of eigenvectors that must be kept for the reconstruction of PC or MNF transformed images. For a sliding neighbourhood window size $n \times n$, the size of the scene data covariance matrix is $n \times n$ by $n \times n$. Solving the characteristic equation or generalised symmetric eigenvalue equation will generate n^2 eigenvectors. This means that, if all the eigenvectors are used in the reconstruction, the computational time for the correlation process will be unnecessarily high, without an improvement in terms of the correlation peak amplitude. Therefore, all the following experiments use the number of bands selected by the automatic truncation of the eigenvalues sequence, performed in an optimum mean-square error sense, as described in sections 5.1 and 5.2 and applied for the previous algorithm.

The large difference in magnitude between the first eigenvalues and the following ones, for both transforms is significant, see figure 6.19. Therefore, it is expected that the automatic truncation will retain only one or possibly two eigenvectors to reconstruct the PC or MNF transformed images. This will result in a significant low computational time.



a) b)

Figure 6.19 Eigenvalues generated for degraded image "UKF.bmp"

a) using PC transform;

b) using MNF transform

The truncation of the eigenvalues sequence requires the estimation of noise variance for PC transform. This was done in two ways, using either wavelets or Wiener filter. Both methods generated values in the same order of magnitude, with the result that the automatic truncation of the eigenvalues retained the same number of eigenvectors for the reconstruction of the PC transformed image. The final result was that the correlation peaks amplitudes were identical up to numerical values, irrespective of the method used to estimate the noise variance. Figure 6.20 displays the correlation peak amplitudes generated by the SMBI algorithm, for PC and MNF transforms implemented. The same images, degraded by white Gaussian noise and blurred, used for the MBIMOM algorithm, were investigated. The degraded images and the designated templates were first denoised using the same method as for the previous algorithm. Regarding the chart legend, PC/MNF signifies the transform applied, while OI/DI represents the synthetic image/degraded image, respectively. More comments about the implementation of MNF transform will follow shortly.

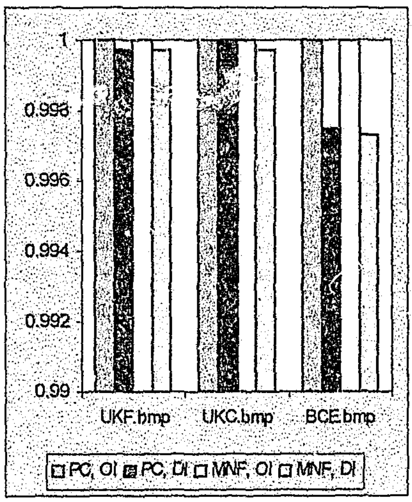
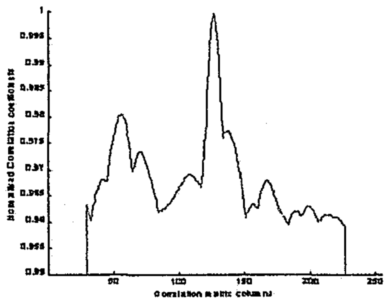
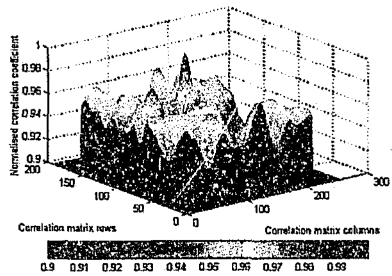


Figure 6.20 A comparison between the correlation results obtained for synthetic and degraded images, PC/MNF transforms used and automatic truncation of eigenvalues series performed

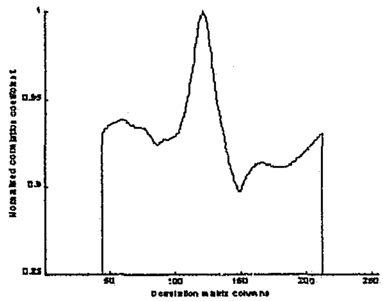
The plots of the normalised correlation coefficients across the correlation matrix columns, when PC transform is used, are shown in figure 6.21, for each of the degraded images. The notation *PC(m)* or latter *MNF(m)* designates the number of bands, that is, eigenvectors used to generate a PC/MNF transformation matrix.


Correlation matrix columns

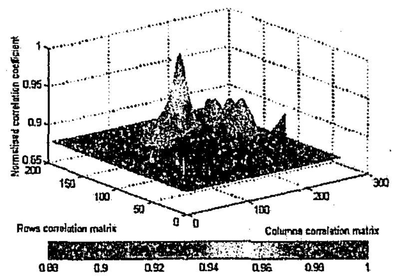
a)


Normalised correlation coefficient
Correlation matrix rows
Correlation matrix columns

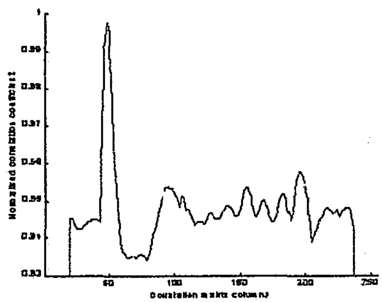
b)


Correlation matrix columns

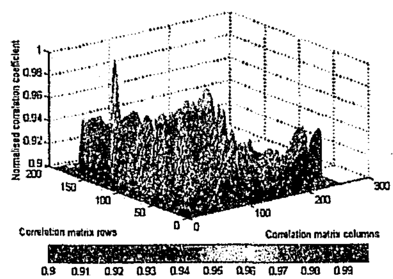
c)


Normalised correlation coefficient
Rows correlation matrix
Columns correlation matrix

d)


Correlation matrix columns

e)


Normalised correlation coefficient
Correlation matrix rows
Correlation matrix columns

f)

Figure 6.21 Normalised correlation coefficients generated by the SMBI algorithm, with PC transform implemented

a) a plot of correlation coefficients across the correlation matrix for the degraded image "UKF.bmp", PC(1)
b) 3D representation of $R(u,v)$ for the same image
c) a plot of correlation coefficients across the correlation matrix for the degraded image "UKC.bmp", PC(2)
d) 3D representation of $R(u,v)$ for the same image
e) a plot of correlation coefficients across the correlation matrix for the degraded image "BCE.bmp",PC(2)
f) 3D representation of $R(u,v)$ for the same image

It can be concluded that the SMBI algorithm, version PC transform, generates a unique, high correlation peak, for each of the degraded images. Remarkably, for all synthetic images the amplitude was equal to one. For the degraded image "UKC.bmp", the algorithm generated a maximum correlation peak amplitude equal to one, also. For all images, the coordinates of the maximum correlation peaks were identical with those calculated by the MBIMOM algorithm.

When MNF transform replaces PC transform, it should be remembered that the first step is to estimate the noise covariance matrix. This was done using two methods: near-neighbour differences or wavelets, both described in section 5.2. Using the method near-neighbour differences to estimate the noise covariance matrix results in choosing more eigenvectors to recreate the MNF transformation matrix, with the immediate result of increased computational time and lower values for the correlation peaks (see figure 6.22). It was decided that the following experiments should employ wavelets to estimate the noise covariance matrix, for their better performance.

Analysing the results displayed by figures 6.20 and 6.22, a very good performance generated by the algorithm with MNF transform applied is evident. For all synthetic images, the correlation peak amplitude reached maximum values, while for the degraded images, MNF transform generated similar values to those obtained by using PC transform. Also, the number of eigenvectors selected automatically was small. Because this algorithm uses the 'raw' data as input, it was expected that the maximum normalised correlation coefficient would be equal to one, for the synthetic images. This is because no initial processing of the data was performed. On the whole, both sets of results, using either PC transform or MNF transform were particularly good, and this algorithm is recommended as a suitable candidate for performing scene matching, at least for sensed optical images.

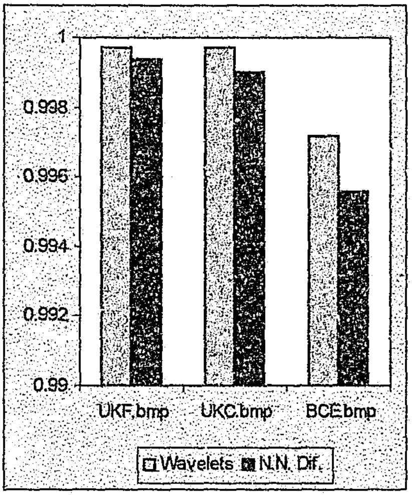


Figure 6.22 Comparison of correlation results generated by the SMBI algorithm, version MNF transform, with noise covariance matrix estimated by near-neighbour differences (N.N.Dif) and wavelets

However, the robustness of the SMBI algorithm must be tested further for various perturbations of grey level values. All the experiments conducted using the MBIMOM algorithm were replicated for the SMBI algorithm. In most cases, the interpretation of the results is identical to that given for the previous algorithm. Therefore, it is not necessary to interpret the same output generated by the experiments again. The synthetic images were changed in the same manner as for the previous algorithm, to obtain new scene images simulating various perturbations. As a consequence, only the results obtained by using this algorithm, in two versions, with PC and MNF transforms used, are displayed.

The first experiment was conducted to verify the robustness of this algorithm to scale changes. The numerical values generated by the algorithm applied to images distorted by scale changes are displayed in figure 6.23. As expected, when the images were smaller than the originals, the numerical values of the maximum normalised correlation coefficients decreased for reasons already explained in section 6.2. In all situations, the algorithm generated the same coordinates for the location of correlation peaks, identical to those generated by the previous algorithm. The significance of the acronyms used within the chart legend is identical to that in figure 6.12a).

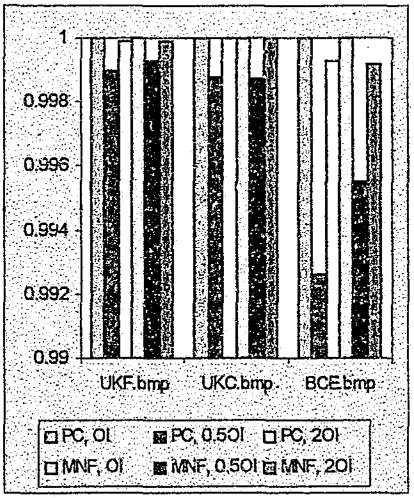


Figure 6.23 Correlation peak amplitudes generated by the SMBI algorithm tested for scale changes

The functioning of the algorithm is not affected by translation errors, and the numerical results reflect this insensitivity, with same interpretation of the results as for the previous algorithm. For all shifted images , *R(u,v)* reached maximum value, in both versions, with PC or MNF transforms used.

As already expected and proved for the MBIMOM algorithm, the SMBI algorithm is also very stable in the presence of rotational errors, due mainly to special properties, previously mentioned in section 6.2, exhibited by PC and MNF transforms. For all images, rotated by $+5^0$ and -10^0 respectively, *R(u,v)* was equal to one.

When the algorithm is tested for its stability in the presence of contrast changes, a different phenomenon happens, compared with the previous algorithm. For brighter images, the results are those expected, $R\ (u,v) = 1$. When the SMBI algorithm is applied to images whose dynamic range is poor, it was expected that it would fail, and the image must be processed first by increasing its dynamic range through histogram equalisation.

This was not necessary because the algorithm was able to detect the match location, even if the pixels' brightness was considerably lower.

Applying histogram processing before the SMBI algorithm acts on the data, changes the initial structure of the signal data and the correlation peak amplitude generated is lower than without pre-processing the image.

When a portion of the image "UKC.bmp" is removed (as in the identical experiment conducted for the previous algorithm), the performance of the SMBI algorithm is not affected, generating the following values for the correlation peaks: $R(u,v) = 0.9994$, for PC transform, and $R(u,v) = 0.9997$, for MNF transform.

Finally, when the algorithm is required to perform on the image degraded so as to simulate one of the worst real world possible scenarios, (recall the similar experiment set for the other algorithm), the true match location is found again, inside the image "UKF.bmp", at position (84,98), and the numerical values obtained were $R(u,v)=0.9859$, for PC transform, and $R(u,v) =0.9854$, for MNF transform.

After extensive tests, it was decided that SMBI algorithm, presented in both versions, is able to solve the scene matching problem, under various adverse conditions, manifesting an impressive stability and robustness.

- The SMBI algorithm using either PC or MNF transform generated impressively good correlation peaks values, always in the range (0.99, 1). This is due to its specific design that creates a unique correlation peak for any pair of given template-match images. Either PC or MNF transform generated similar results, in terms of numerical values for the correlation peaks amplitudes.
- The automatic truncation of the eigenvalues sequence, performed in an optimum mean square error sense is a compulsory step for this algorithm, due to a large number of bands created by the algorithm. The result is a much higher computational speed.
- The algorithm was able to perform under a wide variety of adverse conditions, always calculating the true match location.
- The SMBI algorithm is very robust in the presence of different degradations of the image. It requires only a denoising pre-processing step.

- The algorithm is not computationally expensive because it does not require feature extraction stage.
- The robustness and accuracy are ensured by its specific design, which uses all the information contained in the initial data set and the use of PC or MNF transform.
- The probability of detection scored by the SMBI algorithm is equal to one, as in the case of the previous algorithm. The algorithm manifests, also, a high consistency.

It was proved that both algorithms generate good results when they are applied to blurred images corrupted by white Gaussian noise. The effect of other types of noise that may degrade the sensed image on the performance and accuracy of both algorithms, it was not analysed. It is known that optical images are degraded mainly by noise following a Gaussian distribution, a Poisson distribution, or a combination of both. However, it is necessary to assess the performance of the proposed algorithms when other types of noise degrade the images. Therefore, the synthetic images were changed by adding different types of noise. Consequently, a new set of images was generated. The statistics that describe different types of noise added to the original images are as follows:

a) Gaussian white noise of variance equal to 0.01;
b) Salt and pepper noise of density 0.05;
c) Speckle white noise of variance equal to 0.04.

The images corrupted by white Gaussian noise were pre-processed using wavelets, as described in section 6.2. The images degraded by salt and pepper type noise were filtered using median filter with a kernel size of 3 x 3. Finally, the images corrupted by white speckle noise were filtered using Wiener filter with a mask size of 3 x 3. After the pre-processing steps were performed, each of the algorithms was used to perform the correlation. As before, both versions of the algorithms were tested. The numerical results generated by the correlation process performed by MBIMOM and SMBI algorithms are displayed in figure 6.24. It can be noticed that the MBIMOM algorithm using PC transform generates better results for all images, versus using MNF transform. The SMBI algorithm is almost insensitive to the influence of various types of noise, in both versions, PC and MNF transforms. Also, the results obtained for three different images are consistent, in terms of correlation peak amplitudes, for each of the algorithms. It can be concluded that the algorithms are able to perform on images degraded by various types of noise. It is necessary to reduce the noise, using appropriate techniques, prior to applying the algorithms. It can be concluded that the algorithms are able to handle a wide range of images, not only optical images. It is known that FLIR images are corrupted mainly by salt and pepper noise, while SAR images are corrupted by speckle noise.

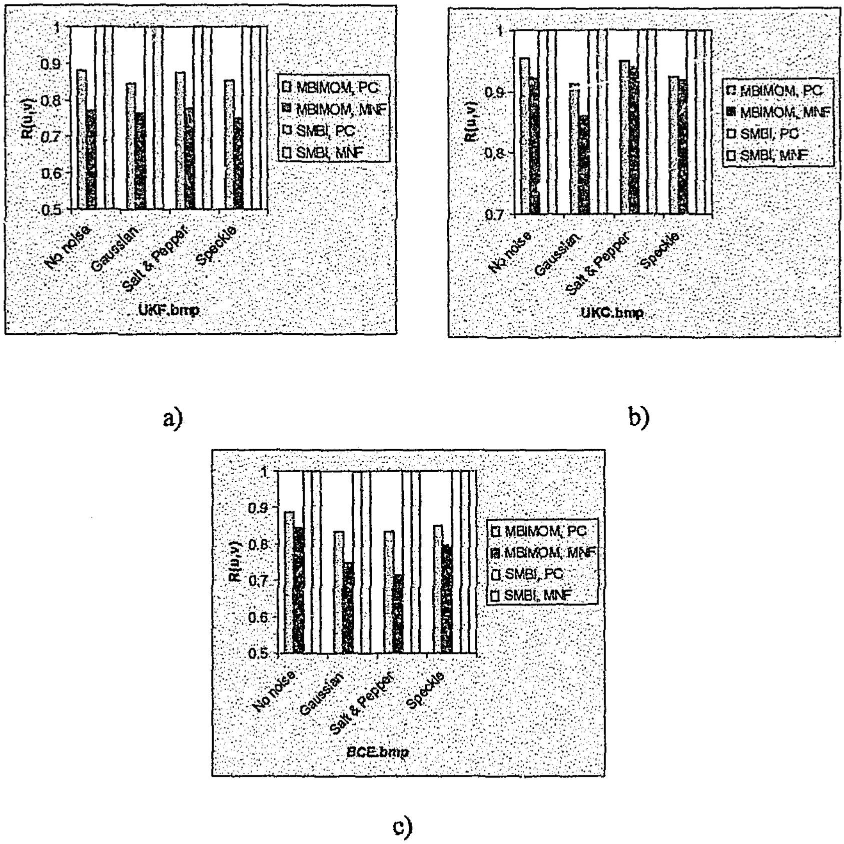


Figure 6.24 Normalised correlation coefficients generated by MBIMOM and SMBI algorithms for images degraded by different types of noise

This experiment simulated the situation when different sensors are used to record same scene content. It also proved that one of the initial goals, that of designing algorithms that can perform scene matching on various images, recorded by different sensors, has been reached. Therefore, they possess a wide range of applicability and generality; the algorithms are not sensor dependent.

Unfortunately, none of the algorithms was able to perform in real time (30 TV frames per second), which would have been the ideal case.

Computationally, it is possible to improve the speed by operating the matrices more efficiently, vectorising various variables when possible, and coding the data in such a way that would make it possible to use the whole set of data simultaneously. Recoding the algorithms using C++ is another viable solution. While Matlab is an interpreted language, C++ is compiled, therefore C++ may be optimised. One should thus expect some improvements from a C++ implementation, which is done via an available Matlab-to-C++ converter.

An alternative option of increasing the computational speed of the algorithms would be to specify a different way in which they operate. They should act on images that are transformed in such a way that they preserve the original signal data, but on a reduced scale. The following solutions that are presented in the next section were not implemented, but they are considered, from a theoretical point of view, to surpass the only shortcoming of previously investigated algorithms, that of not operating in real time.

6.4 Faster computational solutions to scene matching problem

It is well known that scene matching performed by using correlation is computationally a very demanding process. A match is declared by selecting the position of maximum cross-correlation between the template and each possible shift position of the scene image. For a template size of $J x K$ and a scene image of size $M x N$, there are $(M-J+1)*(N-K+1)$ possible test locations. *Sequential Hierarchical Scene Matching*, proposed by Wong and Hall (1978), is computationally more efficient because it uses two sets of structured images, one for the template and another for the scene image, at different resolution levels. Reducing the resolution for each search level results in details of the image content being minimised and only salient characteristics of the objects in the image being retained. For a full description of hierarchical search techniques the reader should refer to Hall (1979). Applying this algorithm, a logarithmic computational efficiency can be gained because the number of search positions is reduced from $(M-J+1)*(N-K+1)$ to $K \log(M-J+1)*(N-K+1)$, with K a constant. The second benefit is that the algorithm can be used with virtually any feature extraction algorithm, and the correlation measure chosen to best suit the problem to be solved. However, extensive tests must be conducted to check the accuracy of this method. It was mentioned in section 6.1 that the method may generate multiple false fixes, especially for sensed images degraded by non-white noise.

Another way to speed up the scene matching process is to use various representations of original images, in such a way that, while fully preserving the original signal content, it still represents the data in a more compact form. One way to represent the original images is to use a pyramidal structure, as suggested by Starck *et al.*,(1998). A pyramid is a set of images that was created by recurrently smoothing and downsampling the original image. What comes to mind immediately, is to use wavelets transform. There are a few ways to decimate the original image:

a) using Mallat's two dimensional algorithm, where the detail signal is obtained from three wavelets (horizontal, vertical and diagonal wavelets);
b) using Feauveau *quincunx* analysis, that undersamples the image by two in one direction, alternatively, by keeping one pixel out of two, then again, even and odd, at each level; this method generates only one wavelet image at each scale.

The problem is that these methods are not invariant under all changes and some of them are computationally expensive. Starck *et al.*,(1998) emphasised that there is no ideal wavelet transform algorithm, and the selected one should depend on the application and signal content. Therefore, they proposed few methods that appear to be more suitable for scene matching, under the general name of *multiresolution based on the median transform.* It would be more desirable to represent the pixel structure only at the first scale and a positive structure not to create negative values in the multiresolution space. It is also desirable that the computational demand be minimal. *Non-iterative pyramidal median transform with exact reconstruction* seems to satisfy the above requirements. The median transform is non-linear and performs robust smoothing. Complete details about the description of this method and its implementation may be found in Starck *et al.*, (1998). A combination of this algorithm with possibly modified versions of the MBIMOM or SMBI algorithm could be a solution to the computational speed issue, while preserving accuracy and robustness.

7 Conclusions

There is a tremendous database of images that can be created from natural scenes and scenes incorporating man-made objects. To analyse all possible images and scenarios and deal with them would have been impossible. However, the results generated by some experiments can be generalised for the scene matching problem, with respect to the algorithms developed during this dissertation.

There is no single solution to the successful scene matching problem. This is due to the enormous variability of the input scene. Each method must be chosen, judged and adapted in such a way as to maximise the use of initial information contained in the input signal and minimise the manipulation of the data. This is the most desired approach because it is well known that each processing step tends to reduce the "strength" of the original data and place a computational burden on the system. This was also the approach adopted for the work presented in this dissertation.

This dissertation attempted to offer a solution to the automated scene matching problem, when the input images are perspective and projective optical images of natural scenes, which contain man-made objects. The objective was that the algorithms proposed must generate a unique match for any pair of template-sensed images, even if the input images are characterised by a low SNR and distorted by various perturbations.

To the author's best knowledge the algorithms proposed in this dissertation were not previously mentioned or published in any book or journal. They were designed with the primary goal to offer an alternative solution to the automated scene matching problem and were tested for the first time during this work.

The qualities exhibited by both algorithms highly recommend them as a fast, reliable, robust and accurate solution to the scene matching problem. Compared to other algorithms developed over the years, the proposed algorithms are characterised by a much higher probability of detection. Ratches *et al.*,(1997) describe a wide range of tests and results generated by various algorithms based on template-matching approaches evaluated in clutter environments. While a large set of algorithms, some of them based on more advanced template-matching approaches and primitive model-based algorithms, tested in 1990 and 1992, manifested a probability of detection between 0.82 and 0.95, the proposed algorithms, the MBIMOM and SMBI algorithms, generated a probability of detection equal to one. The comparison mentioned before may be considered a fair one, bearing in mind that the proposed algorithms are template-matching based and were tested on a medium cluttered environments.

Both algorithms identified and located a designated template, independent of each other. There were no false fixes generated by any of the algorithms, despite the fact that the images tested were distorted by a wide range of perturbations, and represented clutter environments.

Higher correlation peak amplitude, in the case of both algorithms, is ensured if the adequate sequence of pre-processing steps is applied. Images recorded by various sensors are contaminated by spurious, undesirable signals. The process of digitising an image introduces further degradation. A noise-free image would be the output of an ideal sensor and digitising process. In practice there are no such systems. When the image is contaminated by noise, it is advisable to first reduce the level of degradation of the image and increase its 'quality'. A degraded input image generates poor results, in terms of correlation peak amplitude and can also create false fixes. A variety of denoising techniques was investigated, with the result that those best suited for a certain type of noise were used.

The features selected to be used for matching must display algebraic and geometrical invariance. The choice of invariant feature descriptors, that is, the invariant moments for one of the algorithms and the use of the complete initial information for the other had a great contribution to their performance and accuracy.

Also, the use of PC or MNF transform allowed the elimination of redundant, irrelevant information and the reduction of the noise. Both methods, PC and MNF transforms, generated a reduced, uncorrelated set of data. Another advantage of using PC or MNF transform is due to their special properties for rotational corrections. Both transforms perform a rotational error correction on the original data, because they map a point from the original data space into a new feature space. PC transform and MNF transform are used to project the original data set into a new coordinate system, whose orthogonal eigenvectors basis form the new axes. For PC transform, the orthogonal eigenvectors are pointing in the directions of the largest and second largest signal variance as basis vectors, whilst for MNF transform, the orthogonal eigenvectors are pointing in the directions of the largest and second largest SNR values of the bands, as basis vectors. When images degraded by rotational errors were used, the special rotational properties exhibited by both transforms increased the algorithms' robustness.

Another appealing advantage of both algorithms, each of them presented in two versions, is the possibility of automatic truncation of the eigenvalues sequence. This method generated a substantial gain, in terms of performance and computational speed.

For PC transform, in the case of both algorithms, it is required to estimate the noise variance that dictates the number of bands selected for the correlation process, when the automatic truncation of eigenvalues series is implemented. Two independent methods used to estimate the noise variance, one using the Wiener filter and the other using wavelet transform, generated similar results. The effect was that a correct truncation of the eigenvalues series, in an optimum mean square error sense, was possible. The components that incorporated redundant or irrelevant information and were still affected by noise were discarded, without a loss, in terms of useful signal information. Regarding the estimation of the noise covariance matrix, which ensures the success of MNF transform, a few clarifications are in order. Two methods were attempted to try to estimate the noise covariance matrix. The method based on wavelets transform provided superior results, compared to near neighbour difference method. As a result, the SMBI algorithm generated similar results for the normalised correlation coefficients, using either the PC or the MNF transform. In the case of the MBIMOM algorithm, an inaccurate estimation of the noise covariance matrix did not generate better results for the correlation process, either using all bands or only a selected number of them. As a future research direction, the attention should be focused to find better estimation methods for the noise covariance matrix. The *multiresolution support*, proposed by Starck *et al.*, (1998) may be a solution to this problem.

Both algorithms proposed generated good results and the true match location, even in the case of images of poor quality, or templates whose contents were similar to other parts of the scene. The correlation peaks were narrow, distinctive and of remarkably high values.

For some images, the SMBI algorithm generated the maximum attainable value for the normalised correlation coefficient.

It can be concluded that a robust and accurate solution to the automated scene matching problem was provided. Further experiments must investigate the applicability of the proposed algorithms to images recorded by other types of imaging and non-imaging forming sensors.

The question to be answered through further research is if fewer invariant moments possess enough discriminatory power to distinguish and characterise accurately, objects of similar energy content within same scene.

Another research path is to investigate the use of independent information available from multisensors, instead of information generated by only one sensor. This is called a multisensor approach.

Another further research direction should be focused on using other feature descriptors, in parallel with those proposed during this dissertation.

The non-iterative pyramidal median transform could be a suitable solution to achieve a higher computational speed for both algorithms. It could, also, increase the performance.

Another gain, in terms of computational speed and performance should be investigated for the correlation process. The method of cross-correlating the wavelet transforms of two images, at a certain resolution level and normalising the cross-correlation, used by Myers (1996) in his work on fingerprint recognition system could be a good alternative to the correlation procedure adopted for this work, provided that the resolution levels are kept low.

Using a model-based approach, as suggested by Ratches *et al.*, (1997), may generate an improvement in speed, accuracy and performance for the scene matching problem. However, this approach cannot be used in the case of natural scenes, incorporating man-made objects due to the tremendous content variety of these types of images.

The algorithms proposed and verified through a wide spectrum of experiments are intended to offer a robust and accurate solution to the scene matching problem, especially when images that have been degraded by various perturbations are used as input images. Many more ideas and techniques may be focused in future on speed performance and on generating alternative solutions to those proposed by this dissertation. The approach that may be adopted is that of inherently parallel processing of initial data by using various algorithms. A decision for the match location based on identical response of parallel algorithms may be then made. This approach aims to eliminate any false fixes.

References

Akansu, A.N. and Haddad, R. A., *Multiresolution Signal Decomposition,* Academic Press, Inc., 1992.

Anton, H., *Elementary Linear Algebra,* John Wiley & Sons, 5th edition, 1987.

Banu, B. and Jones, T., Image understanding research for automatic target recognition, *IEEE AES Systems Magazine*, October 1993, pp. 15-23.

Barnea, D. I. and Silverman, H. F., A class of algorithms for fast digital image registration, *IEEE Transactions on Computers,* vol. C-21, February 1972, pp. 179-186.

Bezaie, B. and Srinath, M. D., Algorithms for fast image registration, *IEEE Transactions on Aerospace and Electronic Systems,* vol. AES-20, no. 6, November 1984, pp. 716-727.

Castleman, K. R., *Digital Image Processing,* Prentice-Hall, Inc., 1996.

Devijver, P. A. and Kittler, J., *Pattern Recognition – A Statistical Approach,* Prentice-Hall International, Inc., 1982.

Duda, R.O., and Hart, P.E., *Pattern Classification and Scene Analysis,* John Wiley & Sons, Inc., 1973.

Dudani, S. A., Breeding, K. J., McGhee, R. B., Aircraft identification by moment invariants, *IEEE Transactions on Computers,* vol. C-26, no.1, January 1977, pp. 39-45.

Fisher, R., Perkins, A., Walker, A., and Wol, E., *Hypermedia Image processing*, John Wiley & Sons, Ltd., 1996.

Frei, W., Image enhancement by histogram hyperbolisation, *Computer Graphics and Image Processing,* vol. 6, 1977, pp. 286-294.

Gerald, C. F., Wheatley, P. O., *Applied Numerical Analysis,* fifth edition, Addison-Wesley Publishing Company, 1994

Gonzales, R. C. and Wintz, P., *Digital Image processing,* Addison-Wesley Publishing Company, Inc., 1987.

Gonzales, R. C. and Woods, R. E., *Digital Image Processing,* Addison-Wesley Publishing Company, Inc., 1992.

Gordon, C., A generalisation of the maximum noise fraction transform, *IEEE Transactions on Geoscience and Remote Sensing,* vol. 38(1), January 2000.

Green, A., Berman, M., Switzer, P. and Craig, M., A transformation for ordering multispectral data in terms of image quality with implications for noise removal, *IEEE Transactions on Geoscience and Remote Sensing,* vol. 26, no.1, Jan. 1988, pp. 65-74.

Hall, E. L., Davies, D. L., Casey, M. E., The selection of critical subsets for signal, image and scene matching, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-2, no. 4, July 1980, pp. 313-321.

Hall, E., *Computer Image Processing and Recognition,* Academic Press, Inc., 1979.

Harris, D., Private communication

Hong, Z., Algebraic feature extraction of image for recognition, *Pattern Recognition,* vol. 24(3), 1991, pp. 211-219.

Hsia, T. C., A note on invariant moments in image processing, *IEEE Transactions on Systems, Man, and Cybernetics,* vol. SMC-11, no. 12, December 1981, pp. 831-834.

Hu, M. K., Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory,* February 1962, pp. 179-187.

Hummel, R., Image enhancement by histogram transformation, *Computer Graphics and Image Processing,* vol. 6, 1977, pp. 184-195.

Hussein, Z., *Digital Image Processing – Practical Applications of Parallel Processing Techniques,* Ellis Norwood, 1991.

Jain, A. K., *Fundamentals of Digital Image Processing,* Prentice-Hall International, Inc., 1996.

Jain, R., Kaoturi, R., and Schunck, B. G., *Machine Vision,* McGraw-Hill Book Company, 1972.

Jähne, B., *Digital Image Processing — Concepts, Algorithms, and Scientific Applications,* 4th edition, Springer- Verlag Berlin Heidelberg, 1997.

Kanatani, K., *Group- Theoretical Methods in Image Understanding,* Springer- Verlag, Berlin Heidelberg, 1990.

Keshavan, H. R. and Mandyam, D. S., Interpolative models in restoration and enhancement of noisy images, *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. ASSP-25, no. 6, December 1977, pp. 525-533.

Kuan, D. T., Sawchuk, A. A., Strand, T. C., and Chavel, P., Adaptive noise Smoothing filter for images with signal-dependent noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-7, no. 2, March 1985, pp. 165-177.

Lay, D. C., *Linear Algebra and Its Applications,* Addison-Wesley Publishing Company, 1994.

Lee, J., Woodyatt, S. and Berman, M., Enhancement of high spectral resolution remote-sensing data by a noise- adjusted principal components transforms, *IEEE Transactions on Geoscience and Remote Sensing,* vol. 28, no.3, May 1990,pp. 295-304.

Liu, Z. Q. and Caelli, T. M., Multiobject pattern recognition and detection in noisy background using a hierarchical approach, *Computer Vision, Graphics and Image Processing,* vol. 44, 1988, pp. 296-306.

Maitra, S., Moment invariants, *Proceedings IEEE,* vol. 67, April 1979, pp. 697-699.

Marion, A., *An Introduction to Image Processing,* 1st ed., Chapman and Hall, 1991, pp. 193-238.

Mastin, G., Adaptive filters for digital image smoothing: an evaluation, *Computer Graphics and Image Processing,* vol. 31, 1985, pp. 103-121.

Matlab^R, *Image Processing Toolbox, User's Guide,* version two, The Math Works Inc., May 1997.

Misiti, M., Misiti, Y., Oppenheim, G. and Poggi, J. M., *Wavelet Toolbox for Use with Matlab*^{R}, The Math Works, Inc., 1996.

Myers, L. J., *A Cost Effective Fingerprint Recognition System for Use with Low Quality Prints and Damaged Fingertips*, MSc. Dissertation, University of the Witwatersrand, 1996.

Nedeljkovič, V., Private communication

Papoulis, A., *Probability, Random Variables, And Stochastic processes,* third edition, McGraw- Hill, Inc., 1991

Pratt, W. K., Correlation techniques of image registration, *IEEE Transactions on Aerospace and Electronic Systems,* vol. AES-10, no. 3, May 1974, pp. 353-358.

Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T., *Numerical Recipes – The Art of Scientific Computing,* New York Press, Cambridge University, 1986.

Ratches, J. A., Walters, C. P., Buser, R. G. and Guenther, B. D., Aided and automatic target recognition based upon sensory inputs from image forming systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, no. 9, September 1997, pp. 1004-1019.

Reddi, S. S., Radial and angular moment invariants for image identification, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. PAMI-3, no. 2, March 1981, pp. 24-242.

Richards, J. A., *Remote Sensing Digital Image Analysis,* Springer-Verlag Berlin Heidelberg, 1986.

Roger, R. E., A faster way to compute the noise-adjusted principal components transform matrix, *IEEE Transactions on Geoscience and Remote Sensing,* vol. 32, no. 6, November 1994, pp. 1194-1196.

Russ, J., *The Image Processing Handbook,* CRC Press, Inc., 1992.

Shanmugan, K. S., and Breipohl, A. M., *Random Signals- Detection, Estimation and Data Analysis,* John Wiley & Sons, Inc., 1988, chapters two and three.

Snedecor, G. W. and Cochran, W. G., *Statistical Methods,* The Iowa State University Press, 7th edition, 1980.

Soland, D. E. and Narendra, P.M., Prototype automatic target screener, *SPIE*, vol. 178, Smart Sensors, 1979, pp. 175-184.

Starck, J.-L, and Murtagh, F., Image restoration with noise suppression using the wavelet transform, http://http.hq.eso.org/-fmurtagh/wavelets.html

Starck, J.-L., Murtagh, F.and Bijaoui, A., *Image Processing and Data Analysis-The Multiscale Approach,* Cambridge University Press, 1998

Switzer, P. and Green, A., Min/Max autocorrelation factors for multivariate spatial imagery, *Dept. of Statistics, Stanford University,* Tech. Rep. 6, 1984.

Weeks, A., Myler, H., and Wenaas, H., Computer-generated noise images for the evaluation of image processing algorithms, *Optical Engineering,* Vol. 32, No. 5, May 1993, pp. 982- 992.

Willis, A. J., Private communication.

Wong, R. Y. and Hall, E. L., Sequential Hierarchical Scene Matching, *IEEE Transactions on computers,* Vol. c-27, no.4, April 1978.

Appendix A

Principal Components Analysis (PCA)

Let $[X_1, \ldots, X_N]$ be a $p \times N$ matrix of pixel vectors. A random pixel vector X is selected and the value of the (i,j)th pixel vector is denoted by $X(i,j)$. The mean vector of X can be estimated from a sample of N such vectors by:

$$m_X = E\{X\} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{A.1}$$

where m_X is the mean value of the pixels and E is the expectation operator (Hall, 1979). The covariance matrix can be expressed as:

$$C_X = E\{(X - m_X)(X - m_X)^T\} \approx \frac{1}{N}\sum_{i=1}^{N} X_i X_i^T - m_X m_X^T \tag{A.2}$$

An unbiased estimate of the covariance matrix is given by:

$$C_X = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - m)(X_i - m)^T \tag{A.3}$$

A normal procedure is to first calculate a matrix whose columns are in the form $\hat{X} = X_i - m_X$ and to compute the covariance matrix afterwards. Therefore, the columns of the $p \times N$ matrix B will have a zero sample mean, and is said to be in *mean deviation form:*

$$B = [\hat{X}_1 \hat{X}_2 \ldots \hat{X}_N] \tag{A.4}$$

Using equation (A.4), the *(sample) covariance matrix*, C_X, can be rewritten as:

$$C_X = \frac{1}{N-1} BB^T \tag{A.5}$$

Since any matrix of the form BB^T is positive semidefinite, so is C_X. The covariance matrix is of size $p \times p$, with real entries and symmetric ($C_x = C_x^T$). The operation of subtracting the mean vector from each data point will result in translating the coordinates from the initial origin to the centroid of the data. In the new coordinates, the mean vector is zero and the covariance matrix becomes equal to the *scatter matrix* (*autocorrelation matrix*, that gives essentially the same information about the dispersion of the distribution, but is not normalised). The diagonal elements, C_{ii}, are the variances of the individual random variables, while the off-diagonal elements, C_{ij}, (for $i \neq j$, with $j = 1,\ldots,p$) are their covariances. If C_{ij} entries are equal to zero, this means that the random variables are uncorrelated. It was proved that the analysis of multivariate data $X_1, \ldots\ldots, X_N$ is greatly simplified when most or all of the variables $X_1, \ldots, X_p$ are uncorrelated. This means that the covariance matrix C_x is diagonal or nearly diagonal (Lay, 1994).

Therefore, the covariance matrix C_x must be expressed in a diagonal form. The fact that the covariance matrix C_X is symmetric makes it very appealing. The spectral theorem from linear algebra states that a symmetric matrix, C_X in this case, has the following properties (Lay, 1994):

a) C_X has p real eigenvalues, counting multiplicities.
b) If λ is an eigenvalue of C_X with multiplicity k, then the eigenspace for λ is k dimensional.
c) The eigenspaces are mutually orthogonal, in the sense that eigenvectors corresponding to different eigenvalues are orthogonal.
d) C_X is orthogonally diagonalisable.

Therefor, the covariance matrix may be expressed in a diagonal form, using the unique set of *eigenvalues* and *eigenvectors* of the matrix, generated by its characteristic equation, to construct the diagonalisation of the covariance matrix. A matrix C_X is said to be *orthogonally diagonalisable* if there is an orthogonal matrix, U, (square invertible matrix, such that $\mathbf{U}^{-1} = \mathbf{U}^T$) and a diagonal matrix, Δ, such that:

$$C_X = U\Delta U^{-1} = U\Delta U^T \tag{A.6}$$

It is simple to prove that such a matrix, C_X , is a symmetric one. The way to build the factorisation of the covariance matrix, in order to diagonalise it, is as follows: let Δ be the diagonal matrix, with the eigenvalues of C_X arranged so that $\lambda_1 \geq \lambda_2 \geq ...\lambda_p \geq 0$ (there are no restrictions in which order to enter the eigenvalues on the main diagonal of Δ, as long as the columns of U are the matching eigenvectors) and U is an orthogonal matrix whose columns are the corresponding unit eigenvectors $u_1, u_2, ..., u_p$ (for ease of calculation, the eigenvectors of the covariance matrix are normalised, and, still, matrix U will be orthogonal).

The next step is to define a linear transformation, (the standard matrix of the transformation is chosen to be an orthogonal p-square matrix, denoted by A), that generates a new vector Y, from any vector X, such that :

$$Y = A(X - m_X) \tag{A.7}$$

This equation is known as the *Karhunen-Loeve transform*.

It can be shown that the mean of the new random vector Y, m_Y is equal to zero. The idea behind the whole procedure was to find a way such that the covariance matrix of Y would be diagonal. The covariance matrix of the new vector Y is equal to:

$$C_Y = E\{(Y - m_Y)(Y - m_Y)^T\} = E\{YY^T\} \tag{A.8}$$

since vector Y is characterised by zero mean. Now the effort will be concentrated on computing $E\{YY^T\}$. Using equations (A.6) and (A.7), it can be proved that:

$$\begin{aligned} C_Y &= E\{YY^T\} = E\{A(X - m_X)[A(X - m_X)]^T\} = AE\{(X - m_x)(X - m_X)\}A^T = \\ &= AC_XA^T = AU\Delta U^TA^T \end{aligned} \tag{A.9}$$

If the transformation matrix $A = U^T$, in other words, this matrix is equal to the transposed matrix of the eigenvectors of C_X, this will result in the following statement (using the fact that one theorem from linear algebra states that a matrix A has orthonormal columns if, and only if, $A^T A = I$ (I being the identity matrix) $U^T U = I$ and $(U^T)^T = U$), using equation (A.9):

$$C_Y = U^T U \Delta U^T \left(U^T\right)^T = \Delta \qquad \text{(A10)}$$

Therefore, the covariance matrix, C_Y, is equal to the diagonal matrix of eigenvalues of the covariance matrix of vector X, C_X. The off-diagonal elements of the new covariance matrix are 0, which means that the elements of new vector Y are uncorrelated. More than this, the two covariance matrices have the same eigenvalues. The effect of applying equation (A.7) is to transform a vector X, from the p-dimensional data space into a corresponding vector Y in the new feature space. The eigenvectors (the columns of U) are a new basis for the transformation. Therefore, the new pixel vector $Y(i,j)$, can be expressed as :

$$Y(i,j) = U^T (X(i,j) - m_X) \qquad \text{(A11)}$$

Appendix B

Maximum Noise Fraction (MNF)

The steps taken to implement the NAPC are explained below.

1) Using Σ_N, the orthonormalised eigenvector matrix, E, and the diagonal matrix of its eigenvalues, Δ_N, can be computed. Then, $E^T \Sigma_N E = \Delta_N$.

2) Construct a renormalisation matrix $F = E\Delta_N^{-1/2}$, for which the following relation
holds:
$F^T \Sigma_N F = I$ and $F^T F = \Delta_N^{-1}$. Notice that Δ_N must have an inverse, which means that it is necessary that Σ_N be nonsingular, and, therefore, positive definite.

3) Transform the data covariance matrix, Σ_D, by F to obtain the noise-adjusted data covariance matrix, Σ_{adj}, such that:

$$\Sigma_{adj} = F^T \Sigma_D F \tag{B.1}$$

4) From Σ_{adj}, its orthogonal eigenvector matrix, G, can be computed:

$$G^T \Sigma_{adj} G = \Delta_{adj} \quad \text{with} \quad G^T G = I \tag{B.2}$$

5) Finally, the NAPC transformed image is the set of pixel vectors, $Y(i,j)$, produced from the original pixel vectors, $X(i,j)$, by the transform:

$$Y(i,j) = H^T\left(X(i,j) - m_X\right) = (FG)^T\left(X(i,j) - m_X\right) = G^T F^T\left(X(i,j) - m_X\right) = \tag{B.3}$$
$$= G^T \Delta_N^{-1/2} E^T\left(X(i,j) - m_X\right)$$

with $H = FG$, H representing the NAPC transform matrix, and using the definition of the renormalisation matrix $F = E\Delta_N^{-1/2}$.

Appendix C

This appendix contains relevant codes written for use in this dissertation. The codes are written for use with MATLAB^R version 5.2b. Files that are part of Matlab or its toolboxes are not included, but many of these files were used. Codes that were written to try different ideas that for a reason or another were not selected in the final version of the dissertation are not included, also. This appendix displays only those codes that were written for use in the final ideas. The aim of each code is explained under the comments of the code.

Histogram equalisation

```
function [ImHistEq]=eqhist(I)
% This function displays bmp images and performs histogram equalisation
% I is the original image to be processed.
% ImHistEq is the final image that was histogram equalised.
% The program displays the original image and its histogram, histogram equalised,
% image and its histogram.
%
[I,map]=imread('C:\Mirella\Thesis_Images\UKFmat.bmp');
imhist(ind2gray(I,map),256);
figure;
imshow(I,map);
newmap=histeq(I,map);
figure;
imhist(ind2gray(I,newmap),256);
figure;
imshow(I,newmap);
ImHistEq=I;


 for i = 1:levels,
    diff = 9999;
```

```
    pos = 0;
    for j = 1:levels,
      if (abs((i-1) - temp(1,j))) < diff
        pos = j;
        diff = abs((i-1) - temp(1,j));
      end
      g(1,i) = pos;
   end
end
% lastly, calculate output image
for i = 1: xdim,
  for j = 1:ydim,
    im3(i,j) = g( f(round(255*im1(i,j)) + 1) + 1);
  end
end
else
    error ('Dimension mismatch');
end
```

Filtering and correlation

```
function C=corrim(OI,DI,n)
% Filters an image and correlates with the original one
% OI is the original image.
% DI is the image degraded by noise.
x=[3:n];
C=[];
for i=3:n
        H=fspecial('type of filter',i,5);
        Y=filter2(H,DI);
        corr=corr2(Y,OI);
        C=[C;corr];
end;
plot(x,C,'k+:')
xlabel('Size of filter mask');
```

```
ylabel('Correlation coefficient');
```

Filtering using median filter (mask of odd size)

```
function C=corrmedodd(OI,DI,n)
% Filters an image,using median filter,only odd mask size, and correlates with
% original
% OI is the original image.
% DI is the image degraded by different types of noise
x=[3:2:n];
C=[];
for i=3:2:n
        B=medfilt2(DI,[i i]);
        corr=corr2(OI,B);
        C=[C;corr];
end;
plot(x,C,'line type color')
xlabel('Size of filter mask');
ylabel('Correlation coefficient');
```

Intensity changes in an image

```
function [ichb,ichd]=intchange(in,n1,n2)
% changes the intensity grey values of an image
% in is the intensity original image,in double format
% ichb is the intensity changed image(brighter)
% ichd is the intensity changed image(darker)
% n1 is a scalar added to the original image in
% n2 is a scalar subtracted from the original image in
% inD=double(in)/255;
sizein=size(in);
nplus=n1;
nminus=n2;
intchplus=[];
for i=1:1:sizein(1)
        for j=1:1:sizein(2)
                intchplus(i,j)=in(i,j)+nplus;
```

```
            end
end
intchminus=[];
for i=1:1:sizein(1)
            for j=1:1:sizein(2)
                        intchminus(i,j)=in(i,j)-nminus;
            end
end
% allocate intermediate matrices to discard unwanted values
% new intensity changed matrix image(brighter)
intp=intchplus;
sizep=size(intp);
for i=1:1:sizep(1);
     for j=1:1:sizep(2);
           if intp(i,j)>1
           intp(i,j)=1;
           else
           intp(i,j)==intchplus(i,j);
        end
    end
end
% new intensity changed matrix image(darker)
intm=intchminus;
sizem=size(intm);
for i=1:1:sizem(1);
     for j=1:1:sizem(2);
          if intm(i,j)<=0
          intm(i,j)=eps;
          else
          intm(i,j)==intchminus(i,j);
          end
     end
end
ichb=intp;
ichd=intm;
```

```
figure
imshow(ichb);
figure
imshow(ichd);
```

Wavelet de-noising

```
function Iden = wvthrsh(I,sorh,level,typ,k)
% implements wavelet thresholding for denoising
% estimates noise characteristics from variance at first level
% function Iden=wvthrsh(I,'sorh',numlevel,'type',k)
% sorh='s' for soft thresholding
% sorh='h' for hard thresholding
% numlevel=number of wavelet decomposition levels
% 'type'=wavelet type
% k=number of standard deviations used for the threshold
if nargin==1
  typ='sym3';
 level=3;
 sorh='s';
 k=1.2;
elseif nargin==2
        typ='sym3';
        sorh='s';
        k=1.2;
end;
%modelling noise standard deviation
[Cn,Sn]=wavedec2(randn([50,50]),level+1,typ);
devvect=[];
for i=1:level+1;
      h=detcoef2('h',Cn,Sn,i);
      v=detcoef2('v',Cn,Sn,i);
      d=detcoef2('d',Cn,Sn,i);
      devvect=[devvect,[std2(h);std2(d);std2(v)]];
end;
%fitting curves to standard deviation data
```

```
ph=polyfit(1:level+1,devvect(1,:),2);
pv=polyfit(1:level+1,devvect(2,:),2);
pd=polyfit(1:level+1,devvect(3,:),2);
% wavelet decomposition
[Cscn,Sscn]=wavedec2(I,level,typ);
% calculate threshold
hvar=std2(detcoef2('h',Cscn,Sscn,1));
vvar=std2(detcoef2('v',Cscn,Sscn,1));
dvar=std2(detcoef2('d',Cscn,Sscn,1));
i=1:level;
thr=[hvar.*polyval(ph,i);dvar.*polyval(pd,i);vvar.*polyval(pv,i)];
thr=k*thr;
% threshold coefficients and reconstruct the image
[Iden,Cscnd,Sscnd]=wdencmp('lvd',Cscn,Sscn,typ,level,thr,sorh);
```

Feature extraction, moment normalisation, noise estimation for MBIMOM

```
function [mom_x,cov_n]=featmom(x,momwin)
% Finds seven invariant moments of x and normalises them
% Estimates noise covariance matrix using wavelets
% function [mom_x,cov_n]=featmom(x,momwin)
% momwin=size of sliding window
% Columns of mom_x are bands
% Calls: momovl.m, wvnoise.m
% Called by none
if nargin==1
  momwin=[9 9];
end;
% Invariant moments, organised in seven band image
s=size(x);
momss=[(s(1)-momwin(1)+1),(s(2)-momwin(2)+1)];  %size of one band
mom_x=momovl(x,momwin);
% Noise Estimation
mom_n=[];
for i=1:7
    band=reshape(mom_x(:,i),momss);
```

```
    jmpx=ceil(momss(1)/50);
    jmpy=ceil(momss(2)/50);
% Noise estimation using Wavelets
    n=wvnoise(band(1:jmpx:momss(1),1:jmpy:momss(2)));
    mom_n=[mom_n, n(:)];
end;
%Normalisation of seven  band invariant moments image
jump=ceil(prod(momss)/2000);
nsize=size(mom_n);
mean_r=mean(mom_x(1:jump:prod(momss),:))
std_r=std(mom_x(1:jump:prod(momss),:))
mom_x=(mom_x-ones(prod(momss),1)*mean_r)./(ones(prod(momss),1)*std_r);
mom_n=(mom_n-ones(nsize(1),1)*mean_r)./(ones(nsize(1),1)*std_r);
% Noise Covariance matrix
cov_n=cov(mom_n)
```

Wavelet noise estimation

```
function In = wvnoise(I,level,typ)
% Implements inverse hard wavelet thresholding to estimate noise
% threshold is calculated in same way as for wvthrsh.m
% function In=wvnoise(I,numlevel,'type')
% I=the image from which to remove the noise
% numlevel=the number of wavelet decomposition levels
% 'type'=the wavelet type
% In=the noise image
% Calls: thrnoise.m
% Called by: featmom.m
if nargin==1
  typ='sym3';
  level=2;
elseif nargin==2
      typ='sym3';
end;
% Noise modelling
[Cn,Sn]=wavedec2(randn([50,50]),level+1,typ);
```

```
devvect=[];
for i=1:level+1;
        h=detcoef2('h',Cn,Sn,i);
        v=detcoef2('v',Cn,Sn,i);
        d=detcoef2('d',Cn,Sn,i);
        devvect=[devvect,[std2(h);std2(d);std2(v)]];
end;
% Finding standard deviation curves
ph=polyfit(1:level+1,devvect(1,:),2);
pv=polyfit(1:level+1,devvect(2,:),2);
pd=polyfit(1:level+1,devvect(3,:),2);
% Wavelet decomposition
[Cscn,Sscn]=wavedec2(I,level,typ);
% Threshold Calculation
k=2.5;  %number of standard deviations for threshold
hvar=k*std2(detcoef2('h',Cscn,Sscn,1));
vvar=k*std2(detcoef2('v',Cscn,Sscn,1));
dvar=k*std2(detcoef2('d',Cscn,Sscn,1));
% Thresholding and Reconstruction
a=appcoef2(Cscn,Sscn,typ,,level);
a=zeros(size(a));
for i=level:-1:1
        hthr=hvar.*polyval(ph,i);
        h=thrnoise(detcoef2('h',Cscn,Sscn,i),hthr);
        vthr=vvar.*polyval(pv,i);
        v=thrnoise(detcoef2('v',Cscn,Sscn,i),vthr);
        dthr=dvar.*polyval(pd,i);
        d=thrnoise(detcoef2('d',Cscn,Sscn,i),dthr);
        a=idwt2(a,h,v,d,typ,Sscn(level-i+3,:));
end;
In=a;
```

Hard threshold for wavelet noise estimation

```
function xthr=thrnoise(x,thr)
% Performs inverse hard thresholding on x
```

```
% function xthr=thrnoise(x,thr)
% Calls: none
% Called by wvnoise.m
% thr is the threshold
xthr=x.*(abs(x)<thr);
```

Multiband (seven bands) image using moment feature extraction

```
function outimage=momovl(inimage,wsize);
% Program for calculating 7-band image of moments for  input image inimage
% Output is 7-band image called outimage,
% of size(size(inimage(1)-size(wsize(1)+1))* size(inimage(2)-size(wsize(2)+1));
% Each col of outimage is one band (column  by column, not row by row);
% wsize is the sliding window size (preferably square and odd size)
% Calls: mominv7.m
% Called by: featmom.m
% Vladimir Nedeljkovic, April 1997
%
if nargin==1
   wsize=[9 9];
end;
imsize=size(inimage);
outimage=zeros(imsize(1)-wsize(1)+1,7*(imsize(2)-wsize(2)+1));
window=zeros(9,9);
f = waitbar(0,'Calculating moments, please wait...');
for i=1:imsize(1)-wsize(1)+1,
        waitbar(i/(imsize(1)-wsize(1)+1))
        for j=1:imsize(2)-wsize(2)+1,
                window=inimage(i:i+wsize(1)-1,j:j+wsize(2)-1);
                mom=mominv7(window);
                outind=j:(imsize(2)-(wsize(2)-1)):6*(imsize(2)-(wsize(2)-1))+j;
                outimage(i,outind)=mom';
        end;
end;
```

```
close(f);
momsiz=(imsize(1)-wsize(1)+1)*(imsize(2)-wsize(2)+1);   %size of one band
outimage=reshape(outimage,momsiz,7);
```

Invariant moments calculation for a single window (image)

```
function out = mominv7(in);
% This script file calculates Hu's seven invariant moments for a single %window(image)
% in is a matrix image of any size
% out is a column vector of size 7
% Calls: none
% Called by: momovl.m
% V. Nedeljkovic, April 1997
sizein = size(in);
x = [1:1:sizein(2)];
x = (ones(size(x))'*x)';
y = [1:1:sizein(1)];
% first compute moments of order p+q
m00 = sum(sum(in));
m10 = sum(in*x);
m10 = m10(1);
m20 = sum(in*x.^2);
m20 = m20(1);
m30 = sum(in*x.^3);
m30 = m30(1);
m01 = y*sum(in')';
m02 = (y.^2)*sum(in')';
m03 = (y.^3)*sum(in')';
m11 = y*(in*x(:,1));
m12 = y.^2*(in*x(:,1));
x2 = x.^2;
m21 = y*(in*x2(:,1));
m22 = (y.^2)*(in*x2(:,1));
% compute central moments
xbar = m10/m00;
```

```
ybar = m01/m00;
mu00 = m00;
mu10 = 0;
mu01 = 0;
mu20 = m20 - xbar*m10;
mu02 = m02 - ybar*m01;
mu11 = m11 - ybar*m10;
mu30 = m30 - 3*xbar*m20 + 2*m10*xbar^2;
mu12 = m12 - 2*ybar*m11 - xbar*m02 + 2*ybar^2*m10;
mu21 = m21 - 2*xbar*m11 - ybar*m20 + 2*xbar^2*m01;
mu03 = m03 - 3*ybar*m02 + 2*ybar^2*m01;
% compute normalized central moments
eta10=  mu10/mu00^(.5*(1 + 0) +1);
eta01=  mu01/mu00^(.5*(0 + 1) +1);
eta02=  mu02/mu00^(.5*(0 + 2) +1);
eta11 = mu11/mu00^(.5*(1 + 1) +1);
eta12 = mu12/mu00^(.5*(1 + 2) +1);
eta21 = mu21/mu00^(.5*(2 + 1) +1);
eta20 = mu20/mu00^(.5*(2 + 0) +1);
eta30 = mu30/mu00^(.5*(3 + 0) +1);
eta03=  mu03/mu00^(.5*(0 + 3) +1);
% compute seven invariant moments
p1 = eta20 + eta02;
p2 = (eta20 - eta02)^2 + 4*eta11^2;
p3 = (eta30 - 3*eta12)^2 + (3*eta21 - eta03)^2;
p4 = (eta30 + eta12)^2 + (eta21 + eta03)^2;
p5 = (eta30 - 3*eta12)*(eta30 + eta12)*((eta30 +eta12)^2 - 3*(eta21 + ...
     eta03)^2) + (3*eta21 - eta03)*(eta21 +...
      eta03)*(3*(eta30 + eta12)^2 - (eta21 + eta03)^2);
p6 = (eta20 - eta02)*((eta30 + eta12)^2 - (eta21 + eta03)^2) + ...
     4*eta11*(eta30 + eta12)*(eta21 + eta03);
p7 = (3*eta21 - eta03)*(eta30 + eta12)*((eta30 + eta12)^2 - 3*(eta21 + ...
     eta03)^2) + (3*eta12 - eta30)*(eta21 + eta03)*(3*(eta30 + eta12)^2 ...
     - (eta21 + eta03)^2);
out = [p1 p2 p3 p4 p5 p6 p7]';
```

Principal Component Transform for use with MBIMOM

```
function [pc_x,pc_tem,u,d,nvar,ind]=pcXfmom1TR(feat_x,feat_tem,nob)
% function [pc_x,pc_temp]=pcXfmom1TR(feat_x,feat_temp,nob)
% Performs PC transform on feat_x and feat_tem
% Transformation matrix is calculated with feat_x and used on both feat_x and feat_mom

% feat_x=real image features (generated with featmom)
% feat_tem=template image features (generated with featmom)
% nob=the number of bands of PC data used
% d represents the eigenvalues organised in descending order
% u represents the matching eigenvectors
% Calls:none, but uses the output from featmom
% called by: none, but it is used by distim2

tic
if nargin==2
    nob=7;
end;
%Data Covariance Matrix
s=size(feat_x);
jump=ceil(s(1)/2000);
cov_r=cov(feat_x(1:jump:s(1),:));

%Principal component X form
[u1,d1]=eig(cov_r);
d1d=diag(d1);
[d1ds,I]=sort(d1d);
d1ds=rot90(d1ds,2);
d=d1ds;
Ids=rot90(I,2);
i=1:1:nob;
u=[u1(:,Ids(i,1))];
```

```
% Find the eigenvalues that lie above the noise variance for the truncation in optimum
% mean square error sense of eigenvalues series
% noise variance of the covariance matrix cov_r estimated with Wiener filter
[x,noise]=wiener2(cov_r);
nvar=noise
clear x;
ind=1;
while sum(d(nob-ind:nob))<=nob*nvar
    ind=ind+1;
end;
ind=nob-ind
% a=u;
a=(u(:,1:ind));
pc_x=feat_x*a;
pc_tem=feat_tem*a;

% plot of eigenvalues of covariance matrix cov_r
numbands=[1 2 3 4 5 6 7];
plot(numbands,d,'b--',numbands,d,'bo')
hold on
xlabel('Number of bands');
ylabel('Eigenvalues');
grid on;
```

MNF Transform for use with MBIMOM

```
function [mnf_x,mnf_tem,u,d,ind]=mnfXfmom1TR(feat_x,feat_tem,cov_n,nob)
```

```
% Performs MNF transform on feat_x and feat_tem(used with MBIMOM)
% Uses same transformation matrix to transform feat_tem
% feat_x=real image features (generated with featmom)
% feat_tem=template image features(generated with featmom)
% cov_n=the noise covariance matrix of the noise in feat_x
% nob=the number of bands used in the transformation
% Data Covariance Matrix
% Vladimir Nedeljkovic,April, 1997
tic
s=size(feat_x);
jmp=ceil(s(1)/2000);
cov_r=cov(feat_x(1:jmp:s(1),:));

%Maximum Noise Fraction X form
[u1,d1]=eig(cov_r,cov_n);
d1d=diag(d1);
[d1ds,I]=sort(d1d);
d1dr=flipud(d1ds);
d=d1dr
Ids=flipud(I);
I=Ids;
i=1:1:nob;
u=[u1(:,I(i,1))];

% Eigenvalues truncation in an optimum mean square error sense
ind=1;
while sum(d(nob-ind:nob))<=nob
  ind=ind+1;
end;
ind=nob-ind
% a=u;
a=(u(:,1:ind));

% MNF transform form
mnf_x=feat_x*a;
```

```
mnf_tem=feat_tem*a;
toc

% plot of eigenvalues generated by solving the generalised symmetric
% eigenvalue equation for data covariance matrix cov_r, and noise covariance matrix
% covariance matrix,cov_n (noise in feat_x)
nobands=[1 2 3 4 5 6 7];
plot(nobands,d,'b--',nobands,d,'bo')
hold on
xlabel('Number of bands');
ylabel('Eigenvalues');
grid on;
```

Multiband Image Correlation

```
function outim=distim2(inim,tem,imsize,tsize,wsize);
% function outim=distim2(inim,tem,imsize,tsize,wsize)
% The function performs multiband image correlation
% inim=real image features
% tem=template image features
% imsize is the original single band image size
% tsize is the original single band template image size
% wsize the sliding window size used to calculate the features
% for inim and template
% outim is the correlation matrix and is the same size as
% the original image (i.e. imsize)
% Vladimir Nedeljkovic, 1997
% Calls: none
% Called by: none
tic
sinim=size(inim);
sin=imsize-wsize+[1 1];
inim=reshape(inim,sin(1),sin(2)*sinim(2));
stem=size(tem);
```

```
swin=tsize-wsize+[1 1];
tem=tem(:);
lt=length(tem);
sout=sin-swin;
outim=zeros(sout);
nmwind=sqrt(sum(tem.*tem));
f=waitbar(0,'Correlation performed,please be patient...');
for i=1:sout(1),
    for j=1:sout(2),
        jind=[];
        for tmp=1:sinim(2)
            jind=[jind,j+(tmp-1)*sin(2):j+swin(2)-1+(tmp-1)*sin(2)];
        end;
          window=reshape(inim(i:i+swin(1)-1,jind),lt,1);
          cc=(sum(window.*tem))/(nmwind*sqrt(sum(window.*window)));
          outim(i,j)=cc;
    end;
     waitbar(i/sout(1))
end;
close(f);
outim=[zeros(sout(1),floor((imsize(2)-sout(2))/2)),outim,zeros(sout(1),ceil((imsize(2)-
sout(2))/2))];
outim=[zeros(floor((imsize(1)-sout(1))/2),imsize(2));outim;zeros(ceil((imsize(1)-
sout(1))/2),imsize(2))];
max(max(outim))
[i,j]=find(outim==max(max(outim)));
mc=([i,j])
toc
% Plot maximum values of correlation matrix across columns
figure;
plot(max(outim))
hold on
xlabel('Correlation matrix columns');
ylabel('Normal correlation coefficient');
```

```
% Displays the image of the correlation matrix
figure
imagesc(outim)
hold on
colorbar('horiz')
xlabel('Columns correlation matrix');
ylabel('Rows correlation matrix');
% Plot the values of the correlation matrix
figure
mesh(abs(outim))
hold on
xlabel('Columns correlation matrix')
ylabel('Rows correlation matrix')
zlabel('Normal correlation coefficient')
```

Translation of an image

```
function [SI,SIcr]=shift(IO,n1,n2)
% translates the original image by indicated number of rows  and columns
% SI is the shifted image on right direction and down direction
% IO is the original image to be shifted
% n1 is the number of rows the original image is shifted by, it
% must be a positive integer
% n2 is the number of columns the original image is shifted by,
% it must be a positive integer
% SIcr is the shifted image, whose dimensions are equal to the  original image IO
% Calls:none
% Called by:none
sizeIO=size(IO);
Int=IO;
SI=ones(sizeIO(1)+n1-1,sizeIO(2)+n2-1);
m=sizeIO(1)+n1-1;
n=sizeIO(2)+n2-1;
SI(n1:m,n2:n)=Int;
figure
```

```
imagesc(SI)
colormap(gray)
SIcr=imcrop(SI,[SI(1,1) SI(1,1) sizeIO(2)-1 sizeIO(1)-1]);
figure
imagesc(SIcr)
colormap(gray)
```

Spatial multiband image algorithm with PC transform

```
function[xcol,wincol,dxdes,V,ind]=smbfwPC1(x,win,subsize)
% function [xcol,wincol]=smbfwPC1(x,win,subsize)
% PC transform for SMBI algorithm
% Extracts neighbourhood vectors from x(real image,i.e.,scene image
% and win(template image)
% Uses wavelets to estimate noise variance (for truncation) or  Wiener filter
% Finds PC transformation matrix based on neighbourhood vectors of x
% Finds the eigenvectors that lie above the noise
% Transforms x and win with these eiegenvectors
% x=sensor image
% win=template(reference) image
% subsize=size of the sliding neighbourhood window
% Calls: none
% Called by: none, but the output used by distim2
% Noise variance estimation
tic
sx=size(x);
jmpx=round(sx(1)/50);
jmpy=round(sx(2)/50);
n=wvnoise(x(1:jmpx:sx(1),1:jmpy:sx(2)));
nvar=std2(n)^2
% [w,noise]=wiener2(x);
% nvar=noise;
clear w;
% Neighbourhood Vector Extraction
xcol=im2col(x,subsize,'sliding');
wincol=im2col(win,subsize,'sliding');
```

```
clear x win;
xcol=xcol';
wincol=wincol';

% Find PC Xform Matrix
jmpcov=round(prod(sx-subsize)/2500);
sxcol=size(xcol);
cx=cov(xcol(1:jmpcov:prod(sx-subsize),:));
psub=prod(subsize);
[vx dx]=eig(cx);
clear cx;
dxd=diag(dx);
[dxs,I]=sort(dxd);
dxdes=flipud(dxs);
Ides=flipud(I);
i=1:1:psub;
V=[vx(:,Ides(i,1))];
h=GCF
figure(h)
r=1:psub;
plot(r,(dxdes(1:psub)),'r-',r,(dxdes(1:psub)),'ro')
hold on
xlabel('Number of pixels within sliding neighbourhood window')
ylabel('Eigenvalues')

% Find eigenvectors above the noise
ind=1;
while sum(dxdes(psub-ind:psub))<=psub*nvar
    ind=ind+1;
end;
ind=psub-ind;
a=(V(:,1:ind));
%a=V;
xcol=(xcol*a);
wincol=(wincol*a);
```

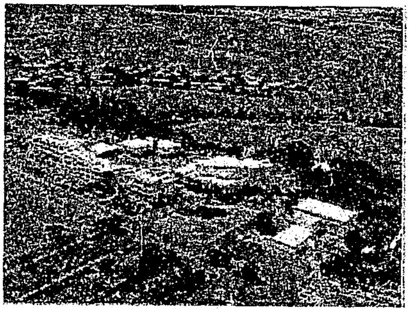
```
toc
```

Spatial multiband algorithm with MNF transform

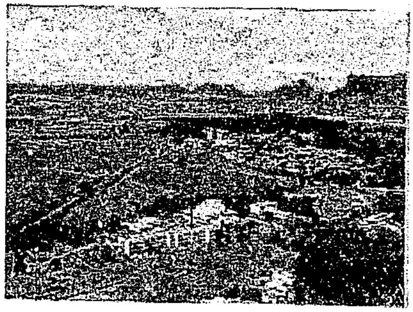
```
function [xcol,wincol,dxdes,V,ind]=smbfwMNF1(x,win,subsize)
% function [xcol,wincol]=sbmfvMNF1(x,win,subsize)
% MNF transform for SMB
% Extracts neighbourhood vectors from x and win
% Uses wavelets to estimate noise covariance
% Finds MNF transformation matrix based on neighbourhood vectors of x
% Finds the eigenvectors that lie above the noise
% Transforms x and win with these eigenvectors give
% x=real(sensed) image
% win=template(reference,synthetic) image
% subsize=size of the sliding neighbourhood window
% A. Willis (1997)
% Calls: none
% Called by: none, but the output used by distim2
if nargin==1
    subsize=[4 4];
end;

% Noise covariance estimation
tic
sx=size(x);
jmpx=round(sx(1)/50);
jmpy=round(sx(2)/50);
n=wvnoise(x(1:jmpx:sx(1),1:jmpy:sx(2)));     %wavelets
%n=x(:,1:sx(2)-1)-x(:,2:sx(2));  %near neighbour differences
ncol=im2col(n,subsize,'sliding');
ncol=ncol';
cn=cov(ncol);
clear ncol;

% Find neighbourhood vectors
xcol=im2col(x,subsize,'sliding');
```

```
wincol=im2col(win,subsize,'sliding');
clear x win;
xcol=xcol';
wincol=wincol';

% Find MNF Xform Matrix
jmpcov=round(prod(sx-subsize)/2500);
sxcol=size(xcol);
cx=cov(xcol(1:jmpcov:prod(sx-subsize),:));
psub=prod(subsize);
[vx dx]=eig(cx,cn);
clear cx,cn;
dxd=diag(dx);
[dxs,I]=sort(dxd);
dxdes=flipud(dxs);
Ides=flipud(I);
i=1:1:psub;
V=[vx(:,Ides(i,1))];
h=GCF
figure(h)
r=1:psub;
plot(r,(dxdes(1:psub)),'b-',r,(dxdes(1:psub)),'b*')
hold on
xlabel('Number of pixels within sliding neighbourhood window')
ylabel('Eigenvalues')
% Noise Truncation and MNF Xform
ind=1;
while sum(dxdes(psub-ind:psub))<=psub
    ind=ind+1;
end;
ind=psub-ind
a=(V(:,1:ind));
xcol=(xcol*a);
wincol=(wincol*a);
toc
```

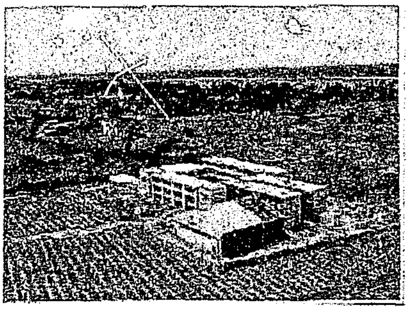
Appendix D



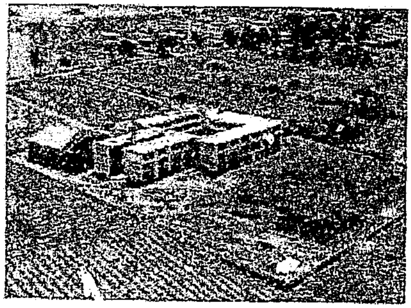
B 1



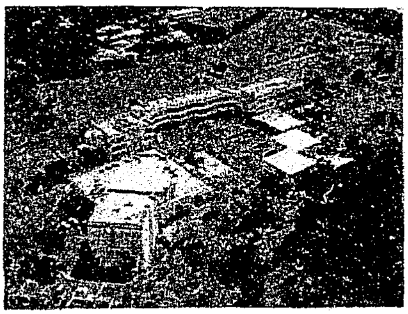
B 2



B 3

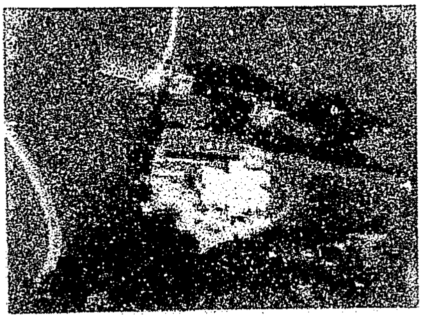


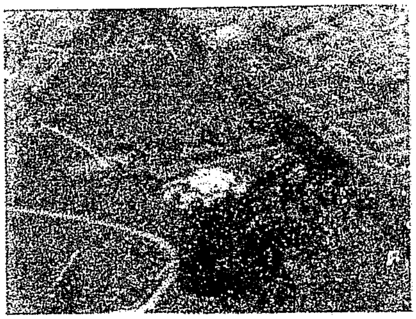
B 4



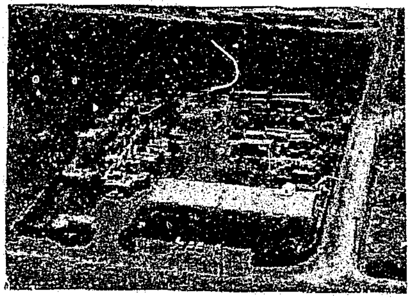
B 5

B 6





B 7 **B 8**



B 9

The location of the images is as follows:

B1 Mahatma Gandhi Institute, Moka, Mauritius

B2 B1, taken from a different altitude and view angle

B3 State Secondary School, Surinam, Mauritius

B4 B3, taken from a different altitude and view angle

B5 B1, taken from a different altitude and view angle

B6 Image downloaded from Internet, unknown location

B7 Beechdown Squash Club, Basingstoke, Hampshire, England

B8 B7, taken from a different altitude and view angle

B9 "Building Construction Engineering", headquarters, Bambous, Mauritius