This is to certify that the
thesis entitled

TOWARD AN AUTOMATED RIVET INSPECTION SYSTEM FOR AGING AIRCRAFT

presented by

Unsang Park

has been accepted towards fulfillment
of the requirements for the

Master of Science	degree in	Department of Computer Science and Engineering

Lalita Udpa
Major Professor's Signature

May 6, 2004
Date

TOWARD AN AUTOMATED RIVET INSPECTION SYSTEM FOR AGING AIRCRAFT

By

Unsang Park

A THESIS

Submitted to
Michigan State University
In partial fulfillment of the requirements
For the degree of

MASTER OF SCIENCE

Department of Computer Science and Engineering

2004

ABSTRACT

TOWARD AN AUTOMATED RIVET INSPECTION SYSTEM FOR AGING AIRCRAFT

By

Unsang Park

This thesis describes work on an automated rivet inspection system for aging aircraft using magneto-optic imaging (MOI). MOI is a non-destructive evaluation technique that is being used increasingly in aircraft inspection. Even though MOI offers high efficiency in non-destructive inspection, the large area of material that needs periodic inspection has created a need for more efficient data interpretation methods: an automated inspection system. The proposed inspection algorithm focuses on rivets that are one of the common places where cracks originate.

Motion-Based Filtering (MBF) is developed as an effective filtering method for MOI images. MBF extracts only "moving objects" in a sequence of images and suppresses stationary background by using a multiple frame subtraction method. The filtered images are processed with rivet detection algorithms to properly locate rivets. Two rivet detection algorithms are developed based on Hough transformation and morphological operation. The detected rivets are classified by classification algorithms implemented by Hough transformation or Bayes decision rule.

The off-line test of the prototype automated rivet inspection system on 245 MOI rivet images showed up to 98% accuracy, but more data is needed for testing. Work is shown in speeding up the algorithms for possible real-time use. A proof-of-concept inspection system showed the capability of processing 3 to 5 images per second.

Dedicated to my sweet heart, Jung-Eun Lee,

my parents Se-Bong Park and Kyung-Shin Lee.

ACKNOWLEDGEMENTS

I would like to thank Dr. Lalita Udpa for giving me the research opportunity about aircraft inspection system. She always inspired me with new challenges and motivations during all my graduate school years.

I would like to thank Dr. George Stockman for his invaluable advice with many challenging problems. He always gave me warm and kind advice regarding my research and course work.

I would like to thank Dr. John Weng who taught me many useful techniques in computer vision class. I also appreciate his comments for my thesis.

I would like to thank Dr. Pradeep Ramuhalli with his advice and help in weekly meetings. I would like to thank Dr. Anil K. Jain for his advice about the sample correlation problem.

I would like to thank all MOI project members, Fan Yuan, Zhiwei Zeng, and Xin Liu, for their support on the project.

TABLE OF CONTENTS

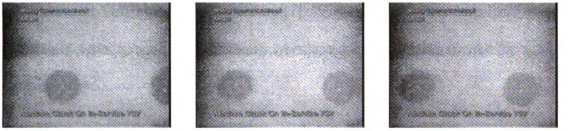
LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Nondestructive evaluation (NDE) is widely used in inspecting aircraft structures to detect surface and subsurface defects. The Magneto-Optic Imager (MOI) is a powerful device for the Nondestructive Inspection (NDI) of aging aircraft. MOI produces analog images of magnetic flux leakage associated with eddy current distribution around surface and subsurface structures. The main advantages of using MOI are its fast inspection speed and easy interpretation compared with conventional Eddy Current NDI instruments. However, due to the magnetic domain wall structures of the sensor, the MOI images are corrupted by serpentine pattern noise, which lowers the MOI inspection capabilities. The noise also poses difficulties in the quantitative analysis of MOI images, which is one of the crucial requirements for automatic inspection.

This thesis advances the state of rivet inspection from MOI along several lines. A proposed automated rivet inspection algorithm is composed of three major steps, Motion-based Filtering (MBF), rivet detection, and rivet classification. MBF is developed to preprocess the MOI images to remove the background noise using a multiple frame subtraction method. Objects, such as rivets and seams, persist from frame to frame and appear to move as sown in Figure 1.1. From the processed image, rivets are detected using Hough transformation or morphological operation. Finally, detected rivets are classified as normal or defective by additional Hough transformation method or Bayesian classifier. Algorithms shown to be successful are transformed in order to speed the processing for real-time application.



(a) (b) (c)

Figure 1.1: A sequence of sample MOI images[1]. Rivets and seam appear to move as the MOI scans over sample.

1.1 Nondestructive Evaluation Techniques

Nondestructive Evaluation (NDE) is the inspection of materials and components for determination of their integrity or detection of any abnormality and, without impairing their usefulness [1]. NDE is used for quality control of products in manufacturing and monitoring the product during operation to assess the remaining life. Based on the physical principles they rely on, NDE methods are categorized as Visual inspection, Ultrasonic inspection, Radiographic inspection, Eddy Current inspection, etc. A brief description of some of the widely used NDE techniques is provided in the following sections.

1.1.1 Visual Inspection

Visual inspection is the oldest and most widely used among all the NDE techniques. It is usually performed as a first step of the evaluation process because of its simplicity and low cost. The specimen is illuminated with light and inspected by the human eye. Using optical instruments such as microscope, telescope, and holography

[1] Text in the image came with the original data from Boeing and is not clearly legible. Throughout this thesis some original images have been enhanced for acceptable printing.

may increase the visual inspection capability [1]. Visual inspection can detect and analyze only surface abnormalities. More sophisticated techniques are needed for subsurface inspection.

1.1.2 Ultrasonic Inspection

Ultrasonic inspection is a versatile NDE technique applicable for most materials, metallic or non-metallic. It utilizes high frequency acoustic waves to detect surface and subsurface defects [1]. The ultrasonic energy travels through a test sample and reflects when it strikes a discontinuity. The reflected ultrasonic energy provides information related to subsurface defects. From the propagation distance of ultrasonic energy, derived from propagation velocity and time, the specimen thickness and location of defects are calculated [2]. An estimate of the shape of the defect is also obtained by moving the probe around the defect and measuring defect echo. Figure 1.1 shows the geometry of a specimen with an internal defect and its corresponding pulse-echo response.

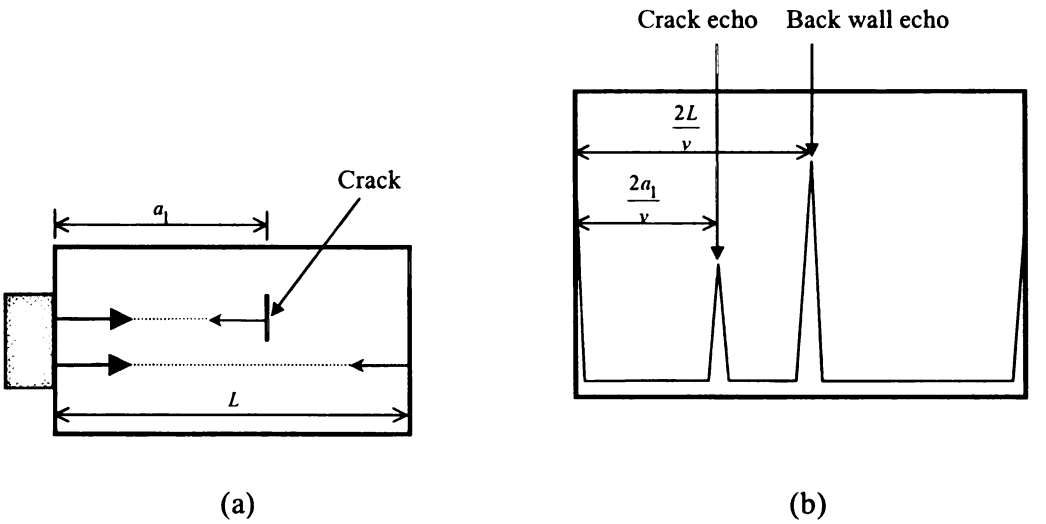


Figure 1.2: Schematic of (a) ultrasonic inspection (b) oscilloscope display of the reflected waves

If L is the thickness of the specimen and v is the corresponding velocity, echoes from the specimen back wall and defect appears at time $\frac{2L}{v}$ and $\frac{2a_1}{v}$ in the oscilloscope, respectively. Ultrasonic energy may propagate in any material which behaves in an elastic manner and is therefore applicable to many types of materials including metal, rubber, and plastics. Ultrasonic inspection has several advantages. First, it is portable and needs only one accessible surface of the specimen. Second, it can be used for inspecting a large (a few meters) thickness of sample. Third, it is relatively inexpensive and finally, it allows rapid and automated inspection. However, ultrasonic inspection has a couple of limitations. First, it needs a coupling medium to couple the ultrasonic energy into the specimen. The test specimen is either immersed in water or a coupling gel is applied. Second, with a coarse grained specimen, penetration depth may be reduced to as little as 50 or 100 mm at high frequencies and also reflections from grain boundaries result in significant noise in the signal. Third, highly skilled operators are required for data interpretation.

1.1.3 Radiography

Radiography is one of the most widely used NDE methods for the detection of internal defects such as porosity and voids [1]. Planar defects can also be detected with proper orientation. Radiography uses short wavelength electromagnetic radiation, such as X-rays or gamma rays that can penetrate materials. As the radiation enters the material, it is absorbed and the amount of absorption depends on the density and thickness of the material [3]. Internal defects can be detected by observing the variation of the amount of

absorption of the radiation. The degree of absorption is usually recorded on a film which is sensitive to the radiation source as in Figure 1.2. The location and shape of the defect is directly reflected on the film after irradiation.

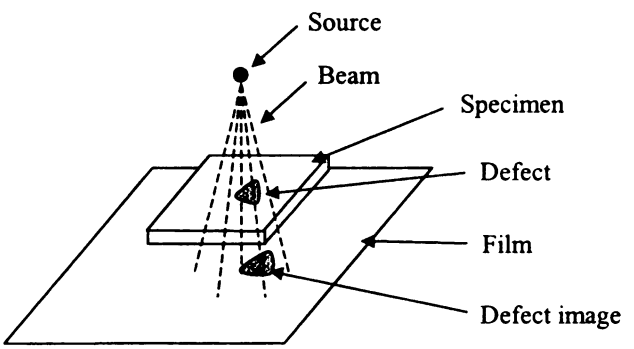


Figure 1.3: Schematic of radiographic inspection

Radiography can be used for most types of solid materials, both ferrous and nonferrous alloys, as well as nonmetallic materials and composites. However, radiographic inspection has several limitations. First, it is not portable and it is often difficult or impossible to position the film and source of radiation to obtain a radiograph of the desired area. Second, the depth of inspection is limited by the degree of penetration of the radiation in the specimen. The degree of penetration varies depending on the material. Third, the defects must be parallel to the radiation beam and must be at least 2% of the thickness of the specimen. Fourth, equipment including radiation source generator and safety facility is rather expensive.

1.1.4 Eddy Current Inspection

Eddy current inspection has been used for over four decades as a leading in-service NDE method. It can be applied to irregularly shaped conductive material for detecting surface and subsurface abnormalities such as cracks and corrosion. Eddy current testing uses alternating current flowing in a coil to produce an alternating magnetic field in accordance with Ampere's Law:

$$\nabla \times \vec{H} = \vec{J} \tag{1-1}$$

where $\vec{H}$ is the magnetic field in Ampere/meter and $\vec{J}$ is the current density in Amperes/meter^2. Faraday's Law of Induction provides the basis for electromagnetic induction and is given by the equation:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{1-2}$$

where $\vec{E}$ is the time-varying electric field intensity in Volt/meter and $\vec{B}$ is the time-varying magnetic flux density in Weber/meter^2. This equation illustrates that a time-varying magnetic field $\vec{B}$ will induce an emf in a conducting path. When the coil is brought close to an electrically conducting material, and eddy current is induced in the material due to the electromagnetic induction. The induced eddy current in the material generates a secondary magnetic field opposite in direction to the primary field. Figure 1.3 shows the schematic of electromagnetic induction in the specimen by the eddy current probe coil [2].

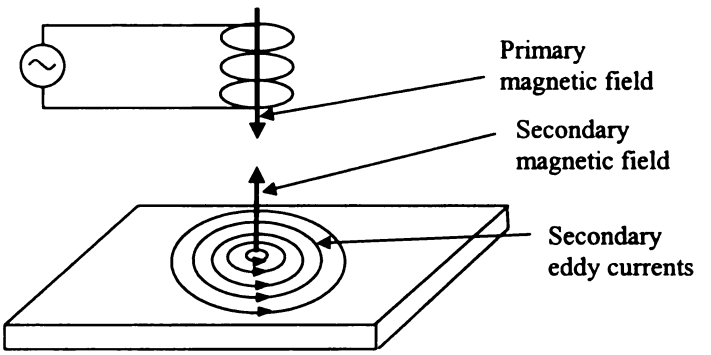


Figure 1.4: Eddy current induction in a specimen

The secondary magnetic field may be detected either as a voltage change across a second coil or by the perturbation of the impedance of the original coil. This impedance change is correlated with the surface and subsurface structure of the material, hence used for detecting defects in materials. The eddy currents in an infinitely large planar sample decay exponentially from the surface of the sample according to the equation [4]:

$$j = j_o \exp(-\frac{y}{\delta}) \tag{1-3}$$

where y is the distance below the surface of the sample, j_o is the current density at the surface of the sample, and δ is the depth of penetration. The depth of penetration is defined as [5]:

$$\delta = \frac{1}{\sqrt{\sigma \mu f \pi}} \tag{1-4}$$

where σ is the conductivity, μ is the relative permeability, and f is the inspection frequency. As shown in equation (1-3), the frequency should be carefully chosen to keep the depth of penetration large enough to reach the subsurface defects. Eddy current inspection has several advantages. First, it is portable and needs access to only one side of the specimen. Second, the equipment is cheap relative to Radiographic instruments. Third, it allows rapid and automated inspection. The major drawback of eddy current testing is that it can be used only for electrically conductive materials. Another drawback is that the depth of inspection is limited to a few millimeters [1].

1.2 Magneto-optic Imaging (MOI)

Magneto-optic Imaging (MOI) is a variant of the eddy current inspection method, and it is becoming increasingly practical for defect assessment in aircraft structures. This relatively new nondestructive testing method gives the inspector the ability to quickly generate real-time images of defects in large surface areas [6][7][8][9]. MOI is based on the use of a combination of principles of magneto-optic effects and eddy current induction. The theory and principles of an MOI system are presented in the following sections.

1.2.1 Eddy Current Excitation

MO Imaging differs in its operation from conventional eddy current testing in that MOI generates uniform sheet eddy currents in the specimen over the area of the magneto-optic sensor as shown in Figure 1.4 [8].

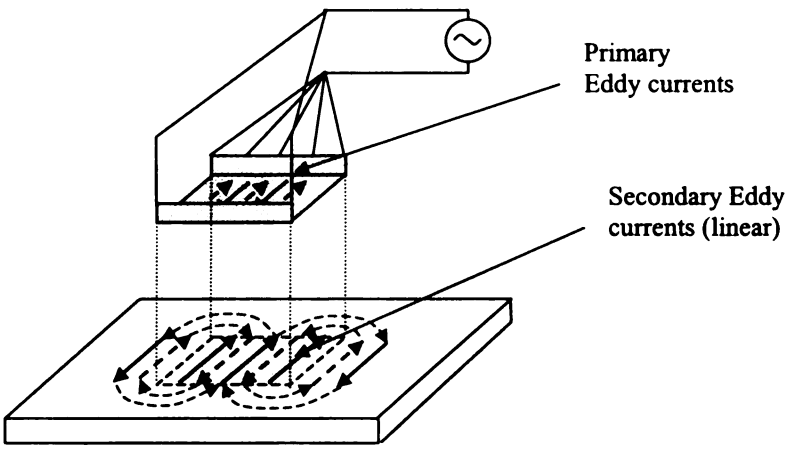


Figure 1.5: Uniform sheet-type eddy current excitation

The rectangular gray area on the specimen in Figure 1.4 is the active area of inspection. Alternatively, MOI uses rotating eddy current to eliminate the necessity of aligning induced current perpendicular to the long axis of cracks for successful detection. Rotating eddy current is provided using an alternate configuration with two separate primary coils connected to a single foil by two single-turn secondary windings. The two-source coil must produce currents perpendicular to each other and they must be out of phase by ninety degrees [7]. The current density in the specimen is represented by the equation:

$$J = J_o \sin(\omega t + \Phi) \tag{1-5}$$

The current density in the two primary coils is represented as:

$$J = J_o \sin(\omega t)\vec{i} + J_o \cos(\omega t)\vec{j} \tag{1-6}$$

where $\vec{i}$ and $\vec{j}$ are unit vectors along the x and y coordinate axes. The symbol J can be interpreted as a linear current density rotating with an angular frequency ω [7]. Rotating eddy current excitation eliminates the null zone observed in linear excitation in which a rivet is shown as a slotted screw. Figure 1.5 shows the effect of rotating linear current on the rivet images.

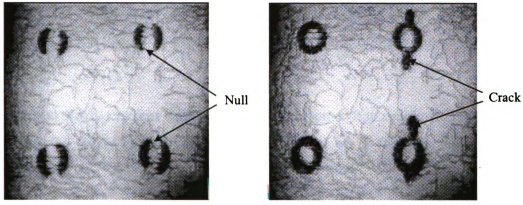


Figure 1.6: The effects of rotating sheet-type eddy current in MOI images of rivets. (a) Linear eddy current excitation (b) Rotating linear eddy current excitation [10].

1.2.2 Magneto-Optic Sensing

While the conventional eddy current method uses the impedance changes of a coil to detect abnormalities, MOI produces analog images that reflect abnormalities in the surface and subsurface of the specimen. The magneto-optic image is the cross-sectional view of the secondary magnetic field associated with eddy currents induced in the specimen. A flat specimen with no structure generates a secondary magnetic field tangential to the specimen. On the other hand, abnormalities in the specimen generate a magnetic field normal to the plane. This is shown in Figure 1.6 schematically.

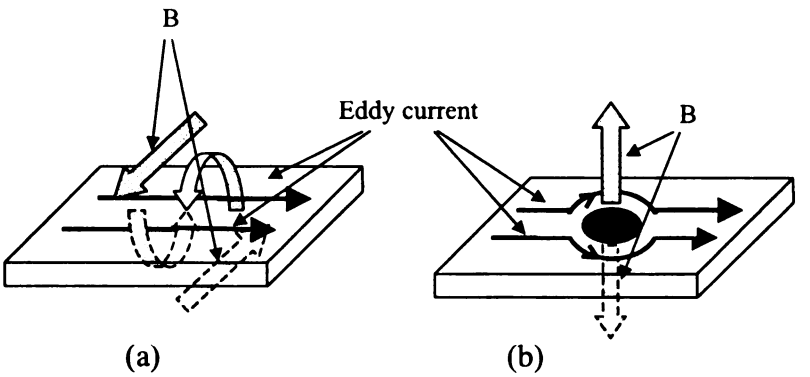


Figure 1.7: (a) Eddy current and induced B field on a flat specimen without any abnormality (b) Eddy current and induced B field on a flat specimen with an abnormality.

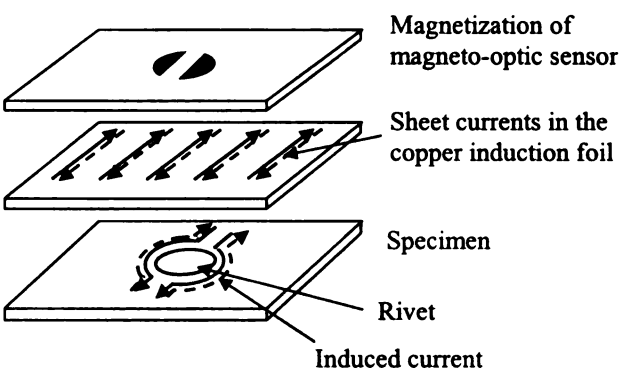


Figure 1.8: Magnetization of magneto-optic sensor [8].

The MOI sensor has an easy axis of magnetization, which is normal to the sample surface, and hence only the normal component of the secondary magnetic field is detected by the magneto-optic (MO) sensor. By placing the sensor parallel to the testing specimen, the

easy axis of the sensor is normal to the plane of the specimen. Thus, the secondary magnetic field produced by rivets, cracks, and corrosion is detected by the sensor, and the primary magnetic field parallel to the flat specimen is ignored. Figure 1.7 shows the magnetized area in the MO sensor by the magnetic field produced by a rivet.

1.2.3 Imaging

A magneto-optic image is produced by applying the principles of Faraday rotation. The Faraday magneto-optic rotation states that when polarized light travels through a magneto-optic material in a magnetic field, its plane of polarization is rotated. The amount of rotation is known as Faraday rotation and is calculated as [7][8][11]:

$$\theta \approx \theta_f \frac{(\vec{k} \cdot \vec{M})l}{|\vec{k}||\vec{M}|} \tag{1-7}$$

where $\vec{k}$ is the wave vector of the light, l is the thickness of the material, and $\vec{M}$ is the magnetic field. The MO sensor is characterized by a large specific Faraday rotation.

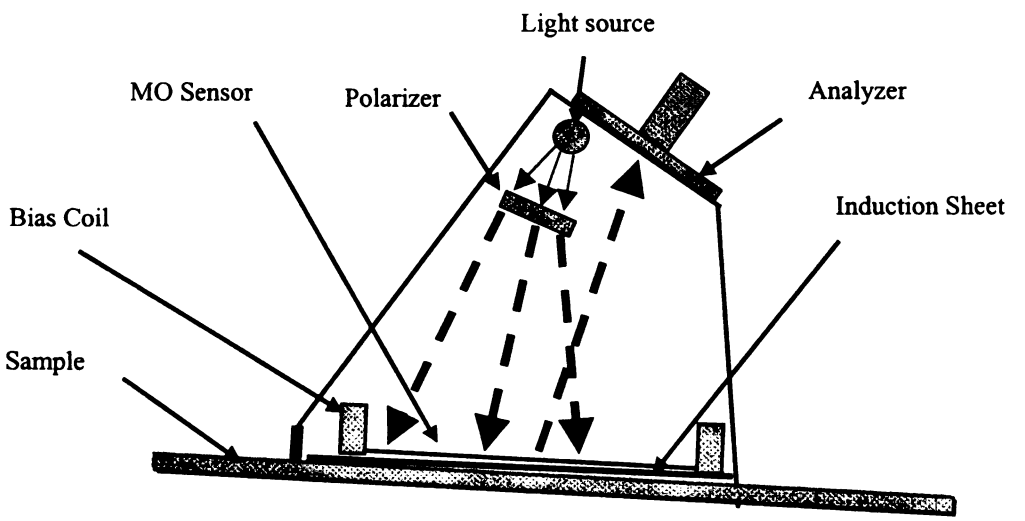


Figure 1.9: Schematic of the MOI instrument.

In MOI, light is transmitted through a polarizer into the MO sensor, reaches the back-reflecting surface under the sensor, and then returns to the analyzer to produce a magneto-optic image. The schematic of the MOI system with the paths of polarized lights is shown in Figure 1.8. A magneto-optic image is produced by a sequence of operations. The MO sensor is cleared by an erase pulse, and then magnetized by the normal magnetic field from abnormalities in the specimen. The magnetization of the MO sensor is retained until the next erase pulse, allowing the imaging operation to be performed. Images are generated and erased about 26 times a second [7]. A typical MOI system is shown in Figure 1.9.

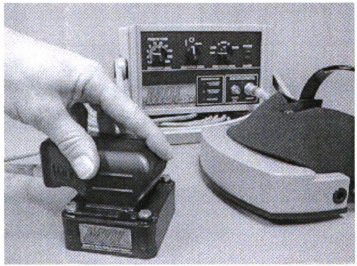


Figure 1.10: MOI system including handheld sensor (left), excitation unit (center), and wearable monitor (right).

1.3 Problem Statement

MOI generates real-time analog images that are the basis for defect detection. While the MOI image provides fast and easy inspection capability, it has several drawbacks. One of the most serious drawbacks is that the MOI image contains serpentine pattern noise from the domain structure of the MO sensor as seen in Figure 1.5. The noise hinders the probability of detection (POD) of second and third layer cracks and corrosion. Hence, the primary goal of this thesis is to develop a filtering algorithm to enhance the quality of MOI images and develop an algorithm for automated inspection based on a quantitative measure of MOI image features. In addition to accuracy of detection, the algorithms for filtering and automated inspection need to meet real-time processing conditions. Hence, problems of practical real-time implementation are also considered.

1.4 Thesis Organization

The thesis is organized into five chapters. Chapter 2 discusses the characteristics of MOI images and introduces a motion-based filtering (MBF) algorithm, which is an image processing algorithm for MOI images. Chapter 3 discusses the algorithm for automated rivet inspection based on the MBF results. Chapter 4 discusses the issues related to real-time implementation of the automated rivet inspection system. Chapter 5 gives a description of the results and discusses future work in this area.

Chapter 2: Image processing of Magneto-optic Images

2.1 Introduction

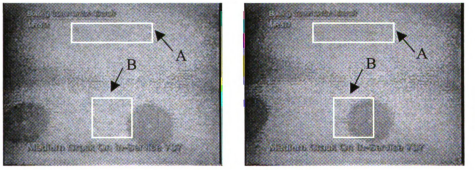
An MOI image contains serpentine pattern noise, which is due to the magnetic domain wall structures of the magneto-optic sensor. The noise lowers the probability of detection (POD) of defects and corrosion and makes data interpretation difficult [12]. In a sequence of MOI images, noise associated with the domain structures in the sensor is stationary as the sensor moves relative to test sample while images associated with structures (rivet) or corrosion in the test sample move from frame to frame due to relative motion of sensor and sample. Conventional image processing methods process each frame of MOI in a scan sequence [13][14]. In this chapter, we introduce a new image filtering technique based on the characteristics of MOI revealed in a sequence of images generated during scanning of the sample surface. The algorithm presented in this chapter is based on separating the moving parts from the stationary parts in the sequence of images. Noise is reduced by retaining and boosting only moving parts and suppressing the stationary part of images.

2.2 Analysis of MOI Images

The MOI inspection is generally conducted by a human operator by scanning the surface of a sample with the sensor. The MOI image data is interpreted by the operator in real-time or recorded as a video sequence for off-line interpretation. The method proposed in this thesis uses the information in a sequence of MOI images for enhancing the quality and hence the corresponding inspection capabilities of the MOI system. Let

$V=\{I_n(x,y), n=1,2,\ldots\}$ represent the video sequence, where $I_n(x,y)$ represents the n^{th} image in the video.

Figure 2.1 shows two consecutive frames in an MOI scanning video. The dark disks represent rivets of 1/8-inch diameter and the dark band running horizontally through the center of the image shows a subsurface seam in the airframe structure. The dark areas on the left and right sides are due to the leakage of uncompensated magnetic field on the edge of the inspecting area. Figure 2.1 shows that the background components are stationary from frame to frame while objects in the sample are moving. The rivets have moved to the left while the scan is directed to the right. The background noise is associated with magnetic domain walls in the sensor and is stationary relative to the sensor during the scan.



(a) (b)

Figure 2.1: Two consecutive MOI images in V; dark disks represent rivets, which are moving to the left.

MOI image data can be divided into two components, dynamic foreground and stationary background. Usually the foreground image is regarded as the signal and the background regarded as noise to be suppressed. Each image can be expressed as below.

$$I_n(x,y) = O_n(x,y) + B(x,y) \tag{2-1}$$

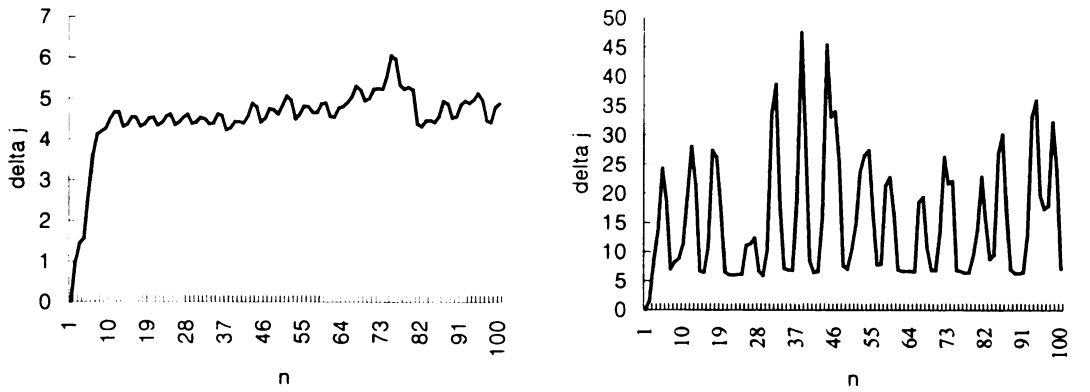
where $O_n(x,y)$ is the object component in the n^{th} frame and $B(x,y)$ is the serpentine noise pattern that is fixed for all n. In Figure 2.1, with the notations in equation (2-1), $I_n(x,y)$ is the entire image, $O_n(x,y)$ corresponds to the dark disks that represent the moving image of rivet and the seam above the rivets, and $B(x,y)$ is the background area.

Pixel intensity changes are observed in selected areas to show quantitatively the presence of an object feature in the MOI images. Two areas are chosen in a MOI inspection sequence as shown in Figure 2.1. Upper boxes A enclose only background images, while lower boxes B enclose both background and object images as the MOI scans the sample. Both boxes have the same area, 3600 pixels. The average pixel intensity changes are calculated in the selected boxes as pixel-by-pixel subtraction. Equation (2-2) presents the average intensity change for two boxes respectively.

$$\Delta j_n = \frac{1}{MN} \sum_{p=1}^{N} \sum_{q=1}^{M} (I_n(p,q) - I_1(p,q)) \tag{2-2}$$

where j represents the average pixel intensity, I_1 represents the first image, and M, N are the window dimensions. The values of $I_n(p,q)$-$I_1(p,q)$ in equation (2-2) are clipped to zero when they are negative.

Figure 2.2 shows the average pixel intensity changes, Δj_n, $1 \leq n \leq 100$, with respect to n in the selected boxed areas. The plot (a) corresponding to the region of box A is seen to stay fairly constant whereas plot (b) containing the object varies widely with time n, between background and object intensity levels.



(a) (b)

Figure 2.2: Average pixel intensity changes in a selected area that encloses (a) only background, and (b) both background and object.

This result is utilized for eliminating the stationary background from the sequence of frames and the issue of motion analysis is involved here. Various motion analysis techniques are discussed in the next section for this purpose.

2.3 Motion segmentation

Detecting and estimating motion in a video sequence is increasingly gaining interest in diverse areas such as video surveillance, object tracking, video compression, and etc. [15][16] Frame subtraction techniques usually segment the motion first, and then further analyzes the motion. On the contrary, the optical flow technique starts by analyzing the motion which is then segmented based on the analysis if necessary. In MOI images, motion analysis is mostly restricted to motion segmentation. Therefore, some

motion analysis techniques focusing on motion segmentation are discussed in following sections.

2.3.1 Background subtraction

The simplest method of motion extraction is background subtraction [17]. Each image is compared with the reference background image and the difference between the two images is extracted. This method is used when the background is static over a relatively long period of time, which is often the case in video surveillance. In video surveillance, the reference background is periodically updated. The background subtraction is very simple to implement and computational cost is very low, which makes it optimal for real-time processing. However, keeping the reference image static is not trivial in many situations. The reference is easily corrupted by a small camera oscillation. In addition, even when the background is stationary with respect to noisy motion, it is usually not static with respect to illumination. The illumination change is detected in the background subtraction and estimated as a motion.

2.3.2 Frame subtraction

Frame subtraction against past images is used for motion detection when the reference image is not obtainable. Most frame subtraction technique use two or three consecutive frames [18] [19] [20]. The background image is assumed not to change much across the frames. The detected change in two or three frames is used to retrieve the outline of the moving object. The moving object is segmented from the outline or further processing is performed to refine the segmentation. The frame subtraction technique can

be used in more general cases than background subtraction because it does not need the reference image. However, the way of segmenting the motion in frame subtraction depends on the velocity of the motion in the image. If the motion is too slow, it cannot be detected, while the segmentation is overestimated if it moves too fast.

2.3.3 Optical flow

Horn and Schunck used a motion constraint equation to analyze motion in a sequence of images [21]. By solving the equation, each pixel in the image is evaluated for its magnitude and direction of motion. A set of pixels that are correlated with the optical flow is segmented as one region. The set of pixels with large magnitude of motion is corresponding to the moving object in given image. Optical flow provides more information about the motion in an image by estimating direction of motion, which allows more complicated analysis than background subtraction or frame subtraction. However, the motion between consecutive frames is assumed to be small in optical flow. The greatest disadvantage of optical flow is its high computational cost. This lowers the applicability of optical flow in real-time systems and for analyzing MOI images. Another disadvantage of optical flow is the noisy result that causes difficulty in the motion segmentation.

2.4 Motion-based Filtering (MBF)

In MOI images, a reference background image is hard to obtain because frames may contain moving objects most of the times. The background can be obtained by getting off the MOI instrument from a sample, but the background changes substantially

when the instrument is later put on the sample surface. However, the background is stationary while MOI scans over a small region of a sample. In addition, the velocity of a moving object is controllable because it is the scanning rate of the human operator or robot system. Thus, frame subtraction across a certain number of frames is a suitable motion analysis method for MOI images for the purpose of the motion segmentation. The model for frame subtraction in MOI images is discussed in next section.

2.4.1 Frame subtraction with MOI images

Assume each image in the sequence has both object and background regions in it and object pixels $O(x,y)$ have lower (darker) intensity than that of the bright background $B(x,y)$. Let D_i be the difference image obtained by subtracting current image $I_n(x,y)$ from the past image $I_{n-i}(x,y)$.

$$D_i(x,y) = I_{n-i}(x,y) - I_n(x,y) \tag{2-3}$$

Then, D_i can be divided into 4 segments D_i^1, D_i^2, D_i^3, and D_i^4 as described below.

$$\begin{aligned} D_i^1 &= \{(x,y) |\ (x,y) \in O_{n-i} \cap O_n \} \\ D_i^2 &= \{(x,y) |\ (x,y) \in B_{n-i} \cap B_n \} \\ D_i^3 &= \{(x,y) |\ (x,y) \in O_{n-i} \cap B_n \} \\ D_i^4 &= \{(x,y) |\ (x,y) \in B_{n-i} \cap O_n \} \end{aligned} \tag{2-4}$$

where O_n represents the moving objects, and B_n represents the stationary background at time n. It can further be noted that under the assumption $O_n(x,y) < B(x,y)$, difference images in the 4 regions have zero, negative, or positive intensities. The expected intensities in the different segments of D_i can be identified as below.

$$
\begin{aligned}
D_i^1 &= \{(x,y) |\ D_i(x,y) = 0\} : \textit{zero intensity} \\
D_i^2 &= \{(x,y) |\ D_i(x,y) = 0\} : \textit{zero intensity} \\
D_i^3 &= \{(x,y) |\ D_i(x,y) < 0\} : \textit{negative intensity} \\
D_i^4 &= \{(x,y) |\ D_i(x,y) > 0\} : \textit{positive intensity}
\end{aligned}
\tag{2-5}
$$

To simplify the problem, we replace the pixel intensities for D_i^3 by zero. As a result, the difference image pixels in D_i^1, D_i^2, and D_i^3 have zero intensities and only region D_i^4 has positive intensity values. Therefore the resulting difference image can be divided into two distinct regions with zero and non-zero intensities and the non-zero intensity area, D_i^4, belongs to the object, O_n, in $I_n(x,y)$ as illustrated in Figure 2.3.

2.4.2 Additive Frame Subtraction (AFS)

In Figure 2.3, part of the object is also lost while the background noise is substantially removed. The object that is lost by subtraction belongs to D_i^1.

Figure 2.3: A binary difference image from two timely consecutive images.

The entire extent of moving objects, including D_i^1, can be recovered using multiple difference images. This is done by subtracting the current image from several past images, and combining the difference images using an OR operation to get the binary image, $S_{n,OR}$, or a MAX operation to get the grayscale image, $S_{n,MAX}$, as described below.

$$S_{n,OR}(x,y) = \underset{1 \le i \le w}{OR} \{D_i(x,y)\} \tag{2-6}$$

$$S_{n,MAX}(x,y) = \underset{1 \le i \le w}{MAX} \{D_i(x,y)\} \tag{2-7}$$

where $S_{n,OR}$ or $S_{n,MAX}$ is the integrated difference image with OR or MAX operation and w represents the number of difference images. In the OR operation, any pixel value $S_{n,OR}(x,y)$ is the result of OR operation on $\{D_1(x,y), \ldots, D_w(x,y)\}$, which is zero if all $D_i(x,y)$ are zeroes, and 1 otherwise. On the other hand, in MAX operation each pixel

value $S_{n,MAX}(x,y)$ is the result of MAX operation on $\{D_1(x,y), \ldots, D_W(x,y)\}$, which is the maximum value of $D_i(x,y)$, $1 \leq i \leq w$. The method described above is called Additive Frame Subtraction. In practice, $S_{n,OR}$ or $S_{n,MAX}$ can be computed recursively as described below.

$$S_{k,OR}(x,y) = OR\{S_{k-1,OR}(x,y), D_k(x,y)\},\ S_1 = D_1 \quad 2 \leq k \leq w \tag{2-8}$$

$$S_{k,MAX}(x,y) = MAX\{S_{k-1,MAX}(x,y), D_k(x,y)\},\ S_1 = D_1 \quad 2 \leq k \leq w \tag{2-9}$$

where $S_{k,OR}$ is $S_{n,OR}$ and $S_{k,MAX}$ is $S_{n,MAX}$, when $k=w$.

Selecting the optimal value of w is an issue in Additive Frame Subtraction. Two possible techniques for choosing w are presented below.

2.4.3 Choice of *w* in AFS

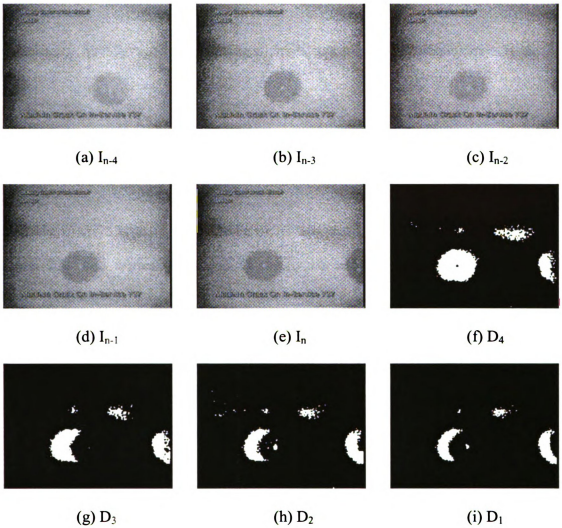
The value w, which is the number of difference images, needs to be chosen with care. In general, w is a function of the object size and scanning velocity. If w is too small, the algorithm will fail to extract the entire object from the MOI sequence and if w is too large, it will result in a large computation time. The optimum value w can be obtained in two ways.

(i) Static method: Let v be the scanning velocity (inches/sec), s the object size along the scanning direction(inches), and f the frame rate(frames/sec). The relationship between w, v, s, and f is expressed as

$$w \geq \frac{s \times f}{v} \tag{2-10}$$

In equation (2-10), we can predefine w only if we set the scanning velocity to a constant because s and f are constant.

(ii) Dynamic method: An alternative approach is to calculate the area of the region with non-zero intensity, the number of pixels in D_i^4 in the difference image $D_i(x,y)$. As illustrated in Figure 2.4 and 2.5, typically the area of $D_i(x,y)$ increases with i and reaches a maximum when the object in I_n and I_{n-4} are completely separated. An optimal estimate of w can be chosen as the corresponding i for which the area reaches its maximum.



(a) I_{n-4} (b) I_{n-3} (c) I_{n-2}

(d) I_{n-1} (e) I_n (f) D_4

(g) D_3 (h) D_2 (i) D_1

Figure 2.4: The typical behavior of the white area of $D_i(x,y)$ in a sequence of difference images. The area of D_i^4 reaches its maximum at $i=4$ in this case.

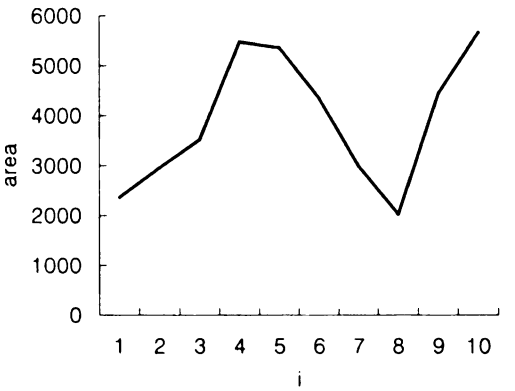


Figure 2.5: The typical behavior of the white area of $D_i(x,y)$ in a sequence of difference images.

2.4.4 Post-processing of MBF

Post processing is performed to enhance the image contrast and remove arbitrary isolated noise pixels. As discussed in Section 2.3.3, the output of Additive Frame Subtraction can be binary or grayscale. The grayscale filtered images are processed in a few post processing steps, while the binary filtered images are directly used as initial data for further analysis such as automatic rivet classification.

The first post processing performed on both binary and grayscale output images is median filtering. The median filter is defined as:

$$J(x,y) = median\{I(x,y) \mid x - w \le x \le x + w,\ y - w \le y \le y + w\} \tag{2-11}$$

$$w = \left\lfloor \frac{W}{2} \right\rfloor - 1, \quad W \text{ is an odd number}$$

where $J(x,y)$ is the filtered image, $I(x,y)$ is the original image, and W is the square kernel window size. The next operation is contrast enhancement. The contrast enhancement increases the contrast of the objects in the combined difference image, so that they become more visible. A grayscale stretching is first applied to map the intensity distribution of combined difference images to the range from 0 to 255. The mapping function is described as:

$$S_n'(x,y) = \frac{255 * S_n(x,y)}{(\max(S_n(x,y)) - \min(S_n(x,y))} \tag{2-12}$$

A threshold operation is performed to remove small arbitrary noise pixels in the stretched MAX image. The threshold operation is expressed as:

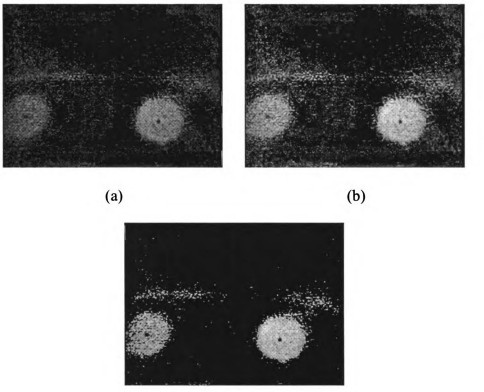
$$S_{n.MAX}'(x,y) = \begin{cases} S_{n.MAX}(x,y), \text{ if } S_{n.MAX}(x,y) \geq threshold \\ 0, \text{ otherwise} \end{cases} \tag{2-13}$$

The threshold value can be chosen by Otsu's method [22] which finds a value that separates the pixel intensity distribution of two classes. The threshold value minimizes within-class variance in equation (2-14) and at the same time maximizes between-class variance.

$$\sigma_w^2(t) = q_1(t)\,\sigma_1^2(t) + q_2(t)\,\sigma_2^2(t) \tag{2-14}$$

where $\sigma_w(t)$ is the within-class variance, $\sigma_1(t)$ and $\sigma_2(t)$ are variances of each class, $q_1(t)$ is the probability that a pixel has an intensity value less than or equal to t, and $q_2(t)$ is the probability that a pixel has an intensity value greater than t. The value t can be found by a sequential search through all possible values or through a recursive relationship described in [20]. The Otsu's threshold values for typical combined difference MOI images was seen to lie between 50 and 80.

The resultant images after each post-processing step are shown in Figure 2.6. The overall motion-based filtering algorithm summarized in the flowchart in Figure 2.7 is applied to the sequence of images to generate the sequence of filtered images. These results are presented next.



(a) (b)

(c)

Figure 2.6: (a) Combined difference image (b) after stretching (c) after thresholding.

2.4.5 Overall Algorithm of MBF

The overall procedure for motion-based filtering is shown below and in Figure 2.7.

1. Let $D_1 = I_{n-1}$-I_n, $S_1 = D_1$, $i = 2$.

2. Calculate $D_i = I_{n-i}$-I_n

3. Calculate S_i with OR operation, if binary, or MAX operation, if grayscale, on S_{i-1} and D_i.

4. For dynamic selection of w, calculate A_i, which is the area of D_i,

5. If $i<w$ and $i<n-1$, or $A_i > A_{i-1}$, then $i = i+1$ and go to step 2.

6. Post-processing

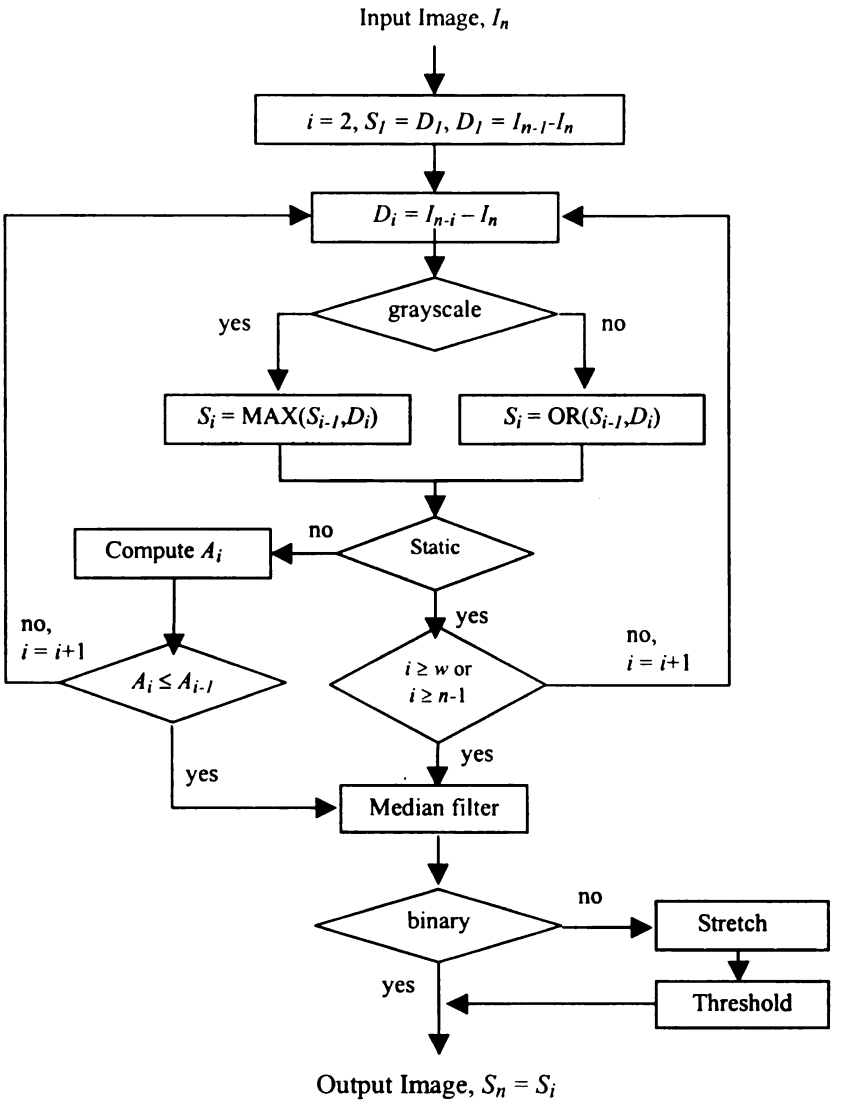
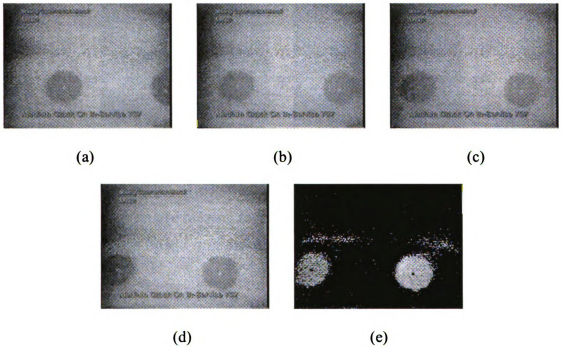


Figure 2.7: Algorithm for motion-based filtering. *w* denotes number of difference images.

2.5 Results of MBF

Figure 2.8(e) shows the resulting gray scale image obtained by applying the motion-based filtering algorithm to MOI images in Figure 2.8(a)-(d). Three difference images are obtained by subtracting the image in Figure 2.8(d) from those in Figure 2.8(a)-(c), and then MAX operation is performed on the difference images followed by post-processing. As seen in Figure 2.8, background noise is reduced to nearly zero while moving objects are retained in their original shape.



(a) (b) (c)

(d) (e)

Figure 2.8: (a),(b),(c),(d) MOI images in time sequence. Rivets are moving to the left while sensor is moving to the right; (e) Motion-based filtered image of the original image (d).

The seam, across the width of the image, is almost lost in the filtered image in Figure 2.8(e). There are two reasons for this. First, the original image of the seam does

not have enough contrast to be separated from the background. Second, if an object extends across the image and parallel to the scan direction, it appears stationary in the video sequence and hence is lost in the filtered image as illustrated in Figure 2.9.

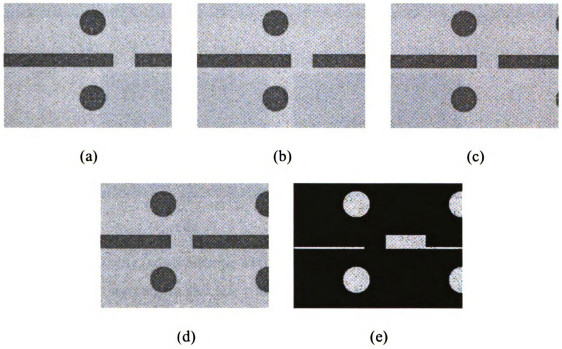


Figure 2.9: (a),(b),(c),(d) Synthetic MOI images in time sequence with uni-directional scan. Rivets are moving to the left while sensor is moving to the right; (e) Motion-based filtered image with uni-directional scan.

One possible way to overcome this problem is to perform a scan orthogonal to the seam or a multidirectional scan so that the seam does not appear stationary. Figure 2.10 shows a synthetic video sequence, obtained using a scan path that is sinusoidal about the seam direction.

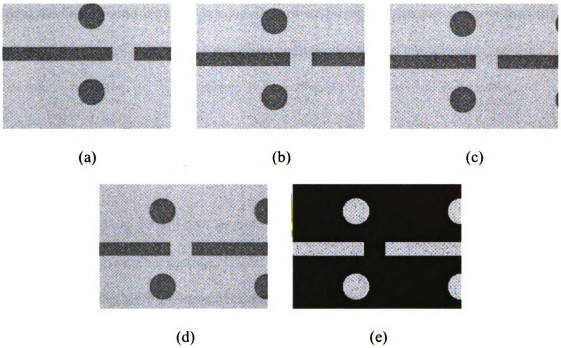


Figure 2.10: (a),(b),(c),(d) Synthetic MOI images in time sequence, multi-directional scan. Rivets are moving to the left while sensor is moving to the right with small sinusoidal up down motion; (e) Motion-based filtered image with multi-directional scan.

The effectiveness of the performance of the motion-based filtering was measured quantitatively with the raw and filtered gray scale images using image contrast as a criterion. Contrast is one of measures of object visibility [23]. Image contrast is defined as:

$$C = \frac{c_o - c_b}{c_b} \tag{2-15}$$

where c_o and c_b are the average intensity of object and background, respectively determined from a sample window of each region. However equation (2-15) cannot be

used if the background intensity is equal to zero or the object intensity is lower than the background intensity. Therefore a modified contrast function is defined as:

$$C' = \frac{|c_o - c_b|}{255} \tag{2-16}$$

The contrast of an object will be 1 in an ideal case when the difference of intensities between the object and background is 255. Three images are selected to measure the effectiveness of the MBF in terms of contrast as in Figure 2.11.

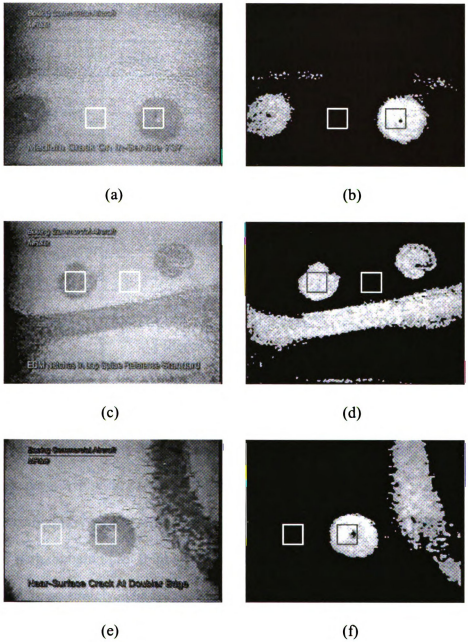


Figure 2.11: Results of MBF. Rectangular box shows selected areas for measuring contrast. (a)(b) Image 1, (c)(d) Image 2, (e)(f) Image 3.

A rectangular box of size 30×30 pixels is selected in background and object for each image to measure the contrast. The measured contrasts of objects in MOI images in Figure 2.11, before and after filtering, are shown in Table 2.1.

Table 2-1: Contrast of objects in MOI images before and after filtering.

Images		Contrast
Image1	Original	0.14
	Filtered	0.71
Image2	Original	0.17
	Filtered	0.62
Image3	Original	0.19
	Filtered	0.49
Average	Original	0.17
	Filtered	0.61

2.6 Conclusion

In this chapter, a new filtering method for enhanced MOI images was presented. The MBF separates moving parts from stationary parts in a sequence of images. The proposed method is shown to remove effectively the serpentine background noise associated with domain in MOI images. This can help the human operator interpret the MOI data more accurately. The result of MBF also provides a binary image output for automated rivet inspection. The following chapter studies automated aircraft inspection methods based on the MB-filtered MOI images.

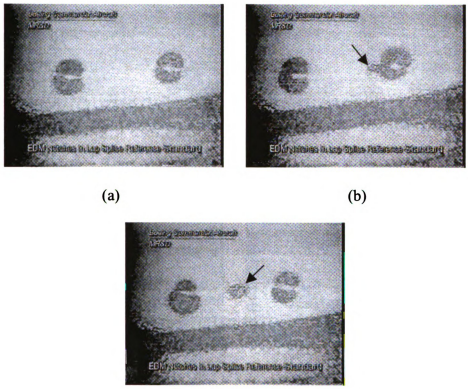
Chapter 3: Algorithms for An Automated Rivet Inspection System With MOI

3.1 Introduction

This chapter introduces signal processing algorithms for automated rivet inspection in airframe structures. Among the various types of problems in aircraft inspection, the detection of cracks under rivets is one of the major challenges facing the aviation industry. In the inspection of the aircraft skin, the large number of rivets makes the manual inspection time consuming and laborious. The human operator needs to scan a large area around every rivet and analyze the acquired images. In practice, the majority of the rivets are good and only a few rivets are defective. This causes the human operator to become accustomed to normal rivets and to tend to ignore defective rivets. An automated rivet inspection system should increase speed, accuracy, consistency and hence reliability of the inspection. The rivet inspection algorithm can be used in manual scanning to assist the human operator or in a fully automated robot inspection system. The rivet inspection algorithm developed in this thesis comprises three major steps, namely, preprocessing, rivet detection, and rivet classification. This chapter explains the implementation details in each stage of the inspection algorithm and presents the inspection results on MOI data.

3.2 MOI Images

Some examples of MOI images of cracks and corrosion are shown in this section. Distinctive features of the defect images will be used as the basis for classification. Sample MOI images of cracks around rivets are shown in Figure 3.1.

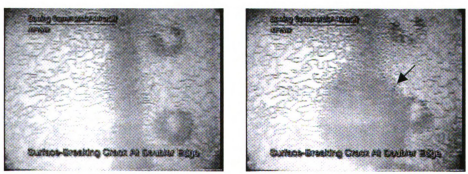


(a) (b)

(c)

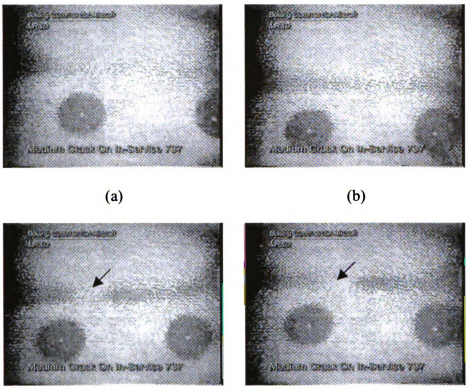
Figure 3.1: (a) Two normal rivets (b) Right rivet has a radial crack (c) A crack between two rivets.

Figure 3.2 and 3.3 show an example of an MOI image of a crack around a seam.



(a) (b)

Figure 3.2: (a) Seam and two normal rivets (b) A crack along the seam



(a) (b)

(c) (d)

Figure 3.3: (a) (b) Normal seam and two rivets (c) (d) Crack along the seam.

In Figure 3.2, the crack is seen as a large object (indicated by an arrow), whereas in Figure 3.3, the crack is seen as a breakage in the seam. This difference is due to the geometrical configurations of the crack and seam. Figure 3.4 presents sample MOI images of a corrosion dome.

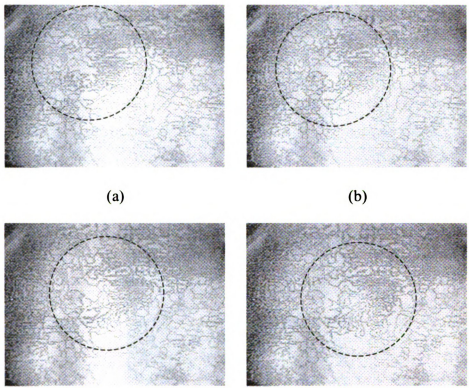
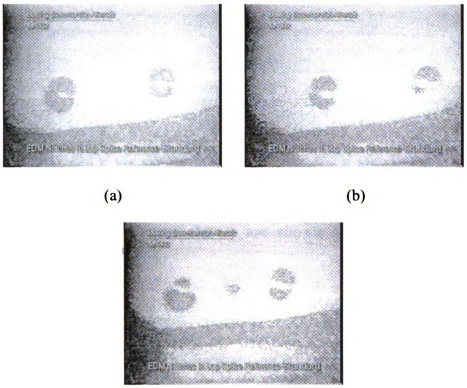


Figure 3.4: Sequence of images with corrosion dome. Corrosion is moving from up left to down right, as the MOI sensor is moving.

The serpentine pattern noise is particularly dominant in the corrosion image, and severely degrades the inspection capability of MOI. As observed in the previous example, MOI images of defects do not represent the exact shape of defects, and the defect images appear differently depending on the airframe structure around them. The defect images are also validated by the finite element modeling technique [24]. Another challenge for the MOI is that the images are affected by variations in the scanning procedure as shown in Figure 3.5.

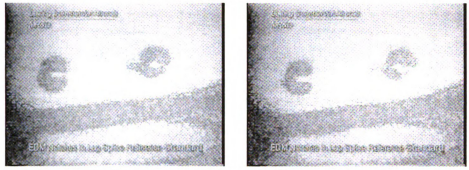


(a) (b)

(c)

Figure 3.5: Images of normal objects are shown differently because of the wobbling of the MOI instrument.

The sensor wobble causes a variation in the distance between test sample and MOI, which in turn results in the induced magnetic fields and hence the MOI images from same objects. Such variation in the data from the same object presents a significant challenge to the development of automated rivet recognition and classification. The variation of MOI images with defective rivets is shown in Figure 3.6.



(a) (b)

Figure 3.6: Images of defective objects are shown differently because of the wobbling of the MOI instrument.

Assuming some prior knowledge about the test geometry and defects images, algorithms for automated rivet inspection will be discussed in the next section.

3.3 Overall Approach of Automated Rivet Inspection

With the wide variation in shapes of defects in MOI images, different inspection algorithms need to be developed for each type of defect. Among the various problems, rivet inspection is one of the most important because rivets are common sites where cracks can develop. The cracks around a rivet are classified into two categories, radial and circumferential, according to its configuration relative to the rivet as described in Figure 3.7.

(a) (b)

(c) (d)

Figure 3.7: (a) A schematic rivet with a radial crack (b) A schematic MOI image of the rivet with a radial crack (c) A schematic rivet with a circumferential crack (b) A schematic MOI image of the rivet with a circumferential crack.

A rivet inspection algorithm for detecting radial cracks around a rivet is discussed in the remainder of this chapter. A typical rivet has roughly a circular or oval shape, while it has an additional protruding blob when there is a radial crack as shown in Figure 3.8.

(a) (b) (c) (d)

Figure 3.8: Typical rivet images in MOI inspection after MBF filtering: (a)(b) normal rivets (c)(d) defective rivets.

The overall approach of automated rivet inspection is a three-step procedure as depicted in Figure 3.9.

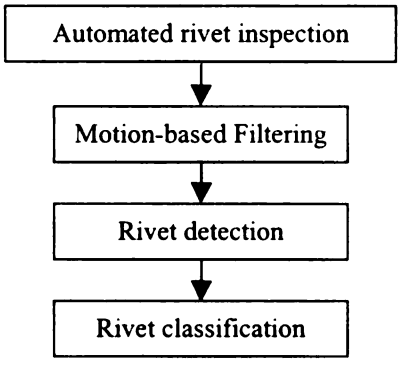


Figure 3.9: Overall approach of automated rivet inspection algorithm.

The raw image obtained by MOI is fed to the MBF module described in Chapter 2 to generate a binary image. Ideally, the binary image is devoid of background noise and contains all information about objects in the raw image such as rivets, defects, and corrosion. The binary image result is used in the subsequent rivet detection and rivet classification modules. MBF was discussed in Chapter 2, so that the following sections discuss issues related to rivet detection and rivet classification.

3.3.1 Automated Rivet Detection

We introduce two different approaches for rivet detection. The first approach is based on circular Hough transformation and the second approach is based on a morphological image processing method. Both rivet detection methods identify all circular objects in the images. Both algorithms detects a circle that encloses the rivet but not the blob associated with the defect.

3.3.2 Hough Transformation Technique

The Hough transformation method was originally proposed to detect straight lines in a given image. Later, the circular Hough transformation was proposed as an efficient method for detecting circles in an image [25] [26]. The rivet detection based on Hough transformation is composed of four steps as follows.

i) The MOI images processed with MBF has zero values for background and non-zero values for object. Therefore, the edge detection is performed by detecting object-background discontinuities as:

$$I'(x,y)=1, \quad \textit{if } I(x,y)\neq 0 \textit{ and } \{I(x-1,y)=0 \textit{ or } I(x+1,y)=0 \textit{ or } I(x,y-1)=0 \textit{ or } I(x,y+1)=0\} \\ 0, \quad \textit{otherwise} \tag{3.1}$$

where, $I(x,y)$ is the input image and $I'(x,y)$ is the edge image.

ii) Gradient and magnitude of gradient for each edge pixel is computed using the Sobel operator for the edge pixels detected in step i):

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3.2}$$

where, $Sobel_x$ and $Sobel_y$ are the Sobel operators for x-axis and y-axis respectively.

$$g_x = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{x,i,j} \times I_{r+i-2,c+j-2}$$

$$g_y = \sum_{i=1}^{i=3}\sum_{j=1}^{j=3} Sobel_{y,i,j} \times I_{r+i-2,c+j-2} \qquad (3.3)$$

$$g = \sqrt{g_x^2 + g_y^2}$$

g_x and g_y are the gradients with respect to x-axis and y-axis respectively, g is the gradient magnitude, and '×' represents convolution operation.

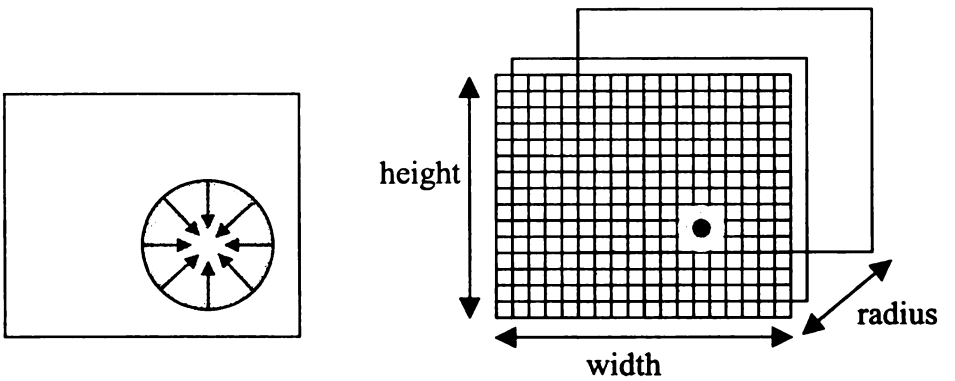
iii) Transform the coordinates of the edge pixel to the three-dimensional (x, y, radius) accumulator. The possible center coordinate is calculated for predefined range of radius r as:

$$x_c = x - r\cos\theta \quad (\cos\theta = g_x / g)$$
$$y_c = y - r\sin\theta \quad (\sin\theta = g_y / g) \qquad (3.4)$$

where, x_c and y_c are the possible center coordinates and r is the given radius. Based on equation 3.4, the accumulator collects the votes for candidate circles as:

$$A(x_c, y_c, r) = A(x_c, y_c, r) + 1, \quad for \quad x_c = x - r\cos\theta \quad (\cos\theta = g_x / g)$$
$$y_c = y - r\sin\theta \quad (\sin\theta = g_y / g) \qquad (3.5)$$

where, $A(x_c, y_c, r)$ is the 3-dimensional accumulator, x and y are the coordinates of edge pixels in the input image. The schematic of the voting and accumulator is shown in Figure 3.10.



(a) (b)

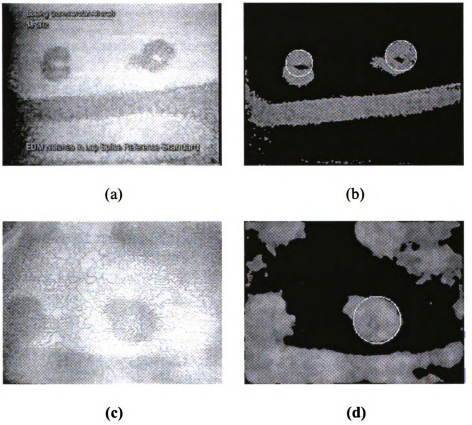
Figure 3.10: Schematic of voting in 3-dimensional accumulator in Hough transformation. (a) Each edge votes for its center coordinate (b) Accumulator collects the votes.

iv) Determine the Hough circle of the rivet by choosing peak values in the accumulator as:

$$\begin{aligned} &\text{if } A(x_c, y_c, r_c) \geq T, \text{ then } I(x_c, y_c) \text{ is center of a circle with radius } r_c \\ &\text{else, skip } A(x_c, y_c, r_c) \end{aligned} \tag{3.6}$$

where, T is the threshold value. If no accumulator element has a value larger than of equal to the threshold T, then the given image is decided not to have a circular object.

For correct results of Hough transformation, some parameters need to be predefined, such as range of r ($r_a \leq r \leq r_b$) and threshold value of the votes T. In our rivet detection algorithm, the range of radius is between 15 and 45 pixels and the threshold value is chosen as 150 experimentally. Typical results of the Hough transformation-based rivet detection are shown in Figure 3.11.



(a) (b)

(c) (d)

Figure 3.11: (a) and (c) Original MOI images (b) and (d) Rivet detection by Hough transformation.

3.3.3 Morphological Operation Technique

In morphological operation-based rivet detection, the MBF filtered image is first segmented using connected component analysis to determine the bounding rectangle of

each object. Object broken by a few pixels' distance need to be considered as a connected object, so that a closing operation is performed. A closing operation is a dilation operation followed by erosion operation. The dilation of binary image B by structuring element S, denoted by B ⊕ S, is defined by [20]:

$$B \oplus S = \bigcup_{b \in B} S_b \tag{3.5}$$

where, S_b represents a translation of set of pixels S by a position vector b. The erosion, denoted by B ⊖ S is defined by [20]:

$$B \ominus S = \{b \mid b + s \in B \quad \forall s \in S\} \tag{3.6}$$

Connected component analysis finds objects that are surrounded by the background pixels. Two connected components and their bounding rectangles are shown in Figure 3.12.

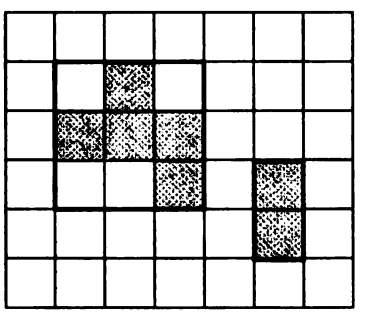


Figure 3.12: Schematic of connected components and bounding rectangles in an image.

An examination of the width, height, and aspect ratio of the bounding rectangle can quickly determine potential segments of a rivet. The initial center and radius c and r are calculated from the bounding rectangle by:

$$\begin{aligned} c &= I(c_x, c_y) \\ c_x &= (x_1 + x_2)/2 \\ c_y &= (y_1 + y_2)/2 \\ r &= \min(c_x - x_1, x_2 - c_x, y_1 - c_y, c_y - y_2) \end{aligned} \tag{3.7}$$

Then, morphological erosion operation is performed. The erosion operation is performed in each bounding rectangle with a structuring element of size W. The size W is chosen as the largest value for which the eroded image is not empty. The structuring element is described in Figure 3.13.

(a) (b)

Figure 3.13: Morphology of structuring element. (a) Square shape (b) Circular shape.

Either square or circular structuring elements can be used for rivet detection. The square-shaped structuring element is easier to implement than circular structuring elements. The unbiased center, c' and radius, r' are calculated after the erosion according to:

$$
\begin{aligned}
c' &= I(c_x', c_y') \\
c_x' &= (x_1' + x_2')/2 \\
c_y' &= (y_1' + y_2')/2 \\
r' &= \min(c_x' - x_1, x_2 - c_x', y_1 - c_y', c_y' - y_2)
\end{aligned}
\tag{3.6}
$$

The intermediate results of this procedure are shown in Figure 3.14.

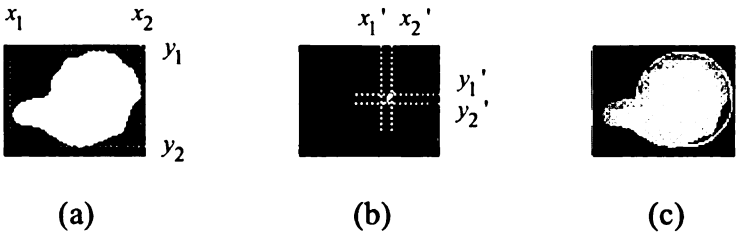


Figure 3.14: Example results of intermediate steps of morphological operation for rivet detection with intermediate variables: (a) initial rivet image (b) after iterative erosion operation (c) after rivet detection.

The procedure for rivet detection using morphological operation is summarized in Figure 3.15.

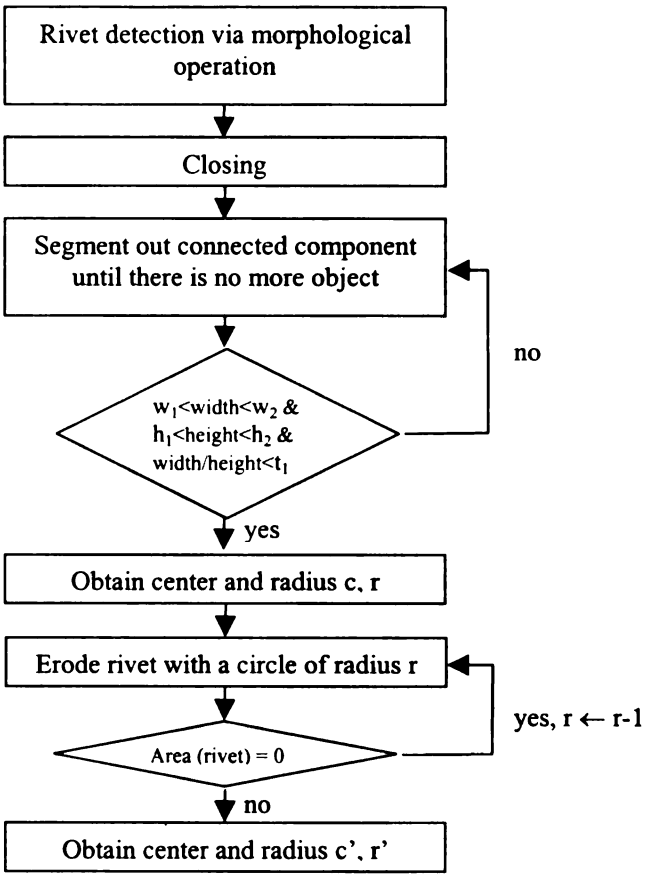


Figure 3.15: Rivet detection algorithm using morphological operation.

The variables w_1, w_2, h_1, h_2, t_1 in the rivet detection algorithm are chosen from values of typical rivet dimensions. Typical results of the morphological operation-based rivet detection are shown in Figure 3.16.

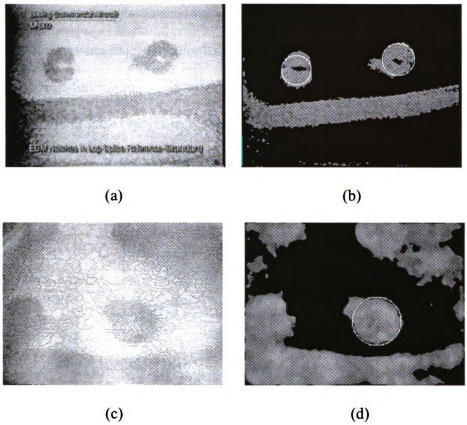


Figure 3.16: **(a) and (c) Raw images. (b) and (d) Rivet detection by morphological operation.**

3.4 Automated Rivet Classification

The final step in this system is the automated classification of a rivet as 'good' or 'bad' (with radial crack). Two approaches for rivet classification are developed. The first method uses a two-pass Hough transformation while the second approach is based on a Bayesian classifier with appropriate feature.

3.4.1 Two-pass Hough Transformation Classifier

In this approach, a second Hough transformation is applied to the set of edges that do not belong to the circles detected in the first Hough transformation. In the case when a

rivet has a crack, the edges associated with the crack will be detected as a small circle whose center lies outside the rivet boundary. Consequently, the detection of a circle in the second Hough transformation implies that the rivet is defective. For the second Hough transformation, the range of radius is chosen between 3 and 30 and the threshold value is chosen as 50. The flow chart of the Two-pass Hough transformation algorithm is shown in Figure 3.17.

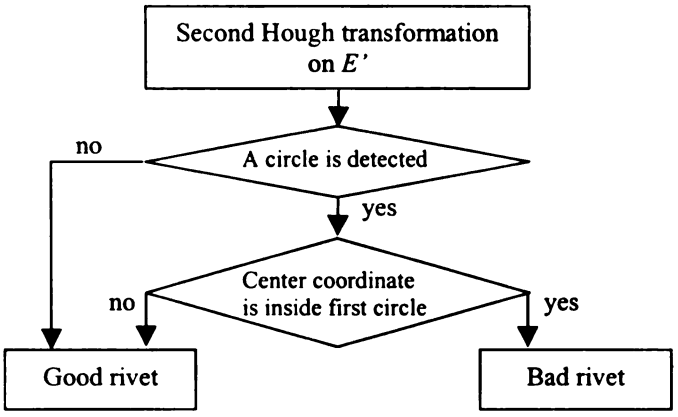
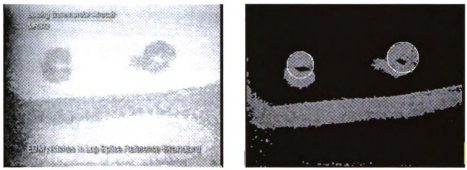
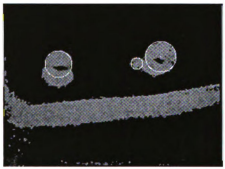


Figure 3.17: Second pass Hough transformation for rivet classification. *E'* represents the image of rivet edge that is outside of the circle detected in the first Hough transformation.

A typical result of implementing the two-pass Hough transformation classifier is presented in Figure 3.18 and 3.19.

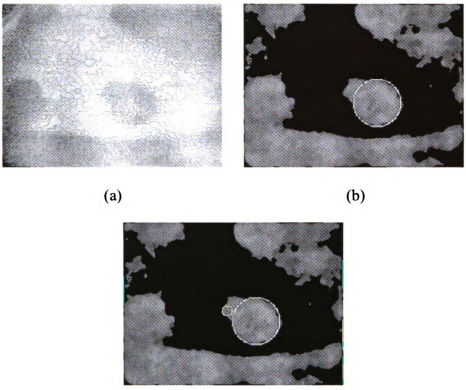


(a) (b)



(c)

Figure 3.18: Two-pass Hough transformation method: (a) raw image (b) after first Hough transformation (c) after second Hough transformation. Right rivet is classified as defective.



(a) (b)

(c)

Figure 3.19: Two-pass Hough transformation method: (a) raw image (b) after first Hough transformation (c) after second Hough transformation. Right detected is classified as defective.

The small circle close to the large circle indicates that the rivet is defective in Figure 3.18 (c) and Figure 3.19 (c).

3.4.2 Bayesian Classifier

Let ω_1 represent the class of 'good' rivets and ω_2 represents the class of 'defective' rivets. The Bayesian decision rule [27] makes a decision based on the statistical model according to the following rule.

$$rivet \in \omega_1 \ \ if \ \ P(\omega_1 \mid x) > P(\omega_2 \mid x);\ otherwise \ \ decide \ \ \omega_2 \tag{3.7}$$

where, $P(\omega_1|x)$ and $P(\omega_2|x)$ represents posterior probability, and x represents the MOI rivet data. The posterior probability $P(\omega_1|x)$ is the probability that the data x belongs to class ω_1 given data x. From Bayes rule we have:

$$P(\omega_i \mid x) = \frac{p(x \mid \omega_i)P(\omega_i)}{p(x)} \tag{3.8}$$

where, $P(x|\omega_i)$ is the conditional probability density function of class ω_i, $P(\omega_i)$ is the prior probability, and p(x) is normalization factor to make the sum of posterior probability equal to 1. Using Bayes rule, the decision rule with probability density functions and prior probabilities is obtained as:

$$rivet \in \omega_1 \ \ if \ \ p(x \mid \omega_1)P(\omega_1) > p(x \mid \omega_2)P(\omega_2);\ otherwise \ \ decide \ \ \omega_2 \tag{3.9}$$

where, $p(\cdot)$ represents a probability density function. Assuming equal prior probability, a decision rule using the conditional probability density functions is obtained as:

$$rivet \in \omega_1 \ \ if \ \ p(x \mid \omega_1) > p(x \mid \omega_2);\ otherwise \ \ decide \ \ \omega_2 \tag{3.10}$$

The probability density function $p(x \mid \omega_i)$, assumed to be Gaussian, is expressed as in Eq. (3.11) for the univariate case with the estimated mean μ_i and variance σ_i^2.

$$p(x \mid \omega_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i} \right)^2 \right] \tag{3.11}$$

The multivariate probability density function is obtained with estimated mean vector $\vec{\mu}_i$ and covariance matrix $\vec{\Sigma}_i$ when the dimension of the feature vector is *d* can be expressed as:

$$p(\vec{x} \mid \omega_i) = \frac{1}{\left(\sqrt{2\pi}\right)^{d/2} |\vec{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2}(\vec{x}-\vec{\mu}_i)^t \vec{\Sigma}_i^{-1}(\vec{x}-\vec{\mu}_i) \right] \tag{3.12}$$

3.4.3 Feature Selection

For the Bayesian rivet classifier, a variable *d* is defined as:

$$d = \sqrt{(x-c_x)^2 + (y-c_y)^2} - r \tag{3.13}$$

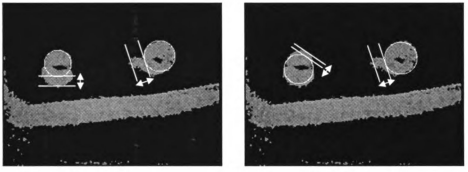
where, *(x,y)* is the coordinate of a point on the edge outside of the circle obtained by the rivet detection algorithm. A set of *d* values are obtained for each edge pixel *(x,y)* value as:

$$D = \{ d_i \mid i = 1,\ldots,n_e \},\ n_e \text{ is the number of edge pixels} \tag{3.14}$$

The negative values of d are discarded from consideration because they do not give information about the defect. A feature *f* is then computed as:

$$f = n^{th} \text{ largest element in } D$$
$$n = \lfloor t \times n_e \rfloor \tag{3.14}$$

where, $0 < t < 1$, n_e is the total number of elements in D. Values larger than n are trimmed out from D to reduce the effect of noise. The value of t is chosen as 0.1 in our experiment. Feature f represents a quantitative measure of the degree of circularity of the rivet. The graphical representation of the feature f is shown in Figure 3.20.



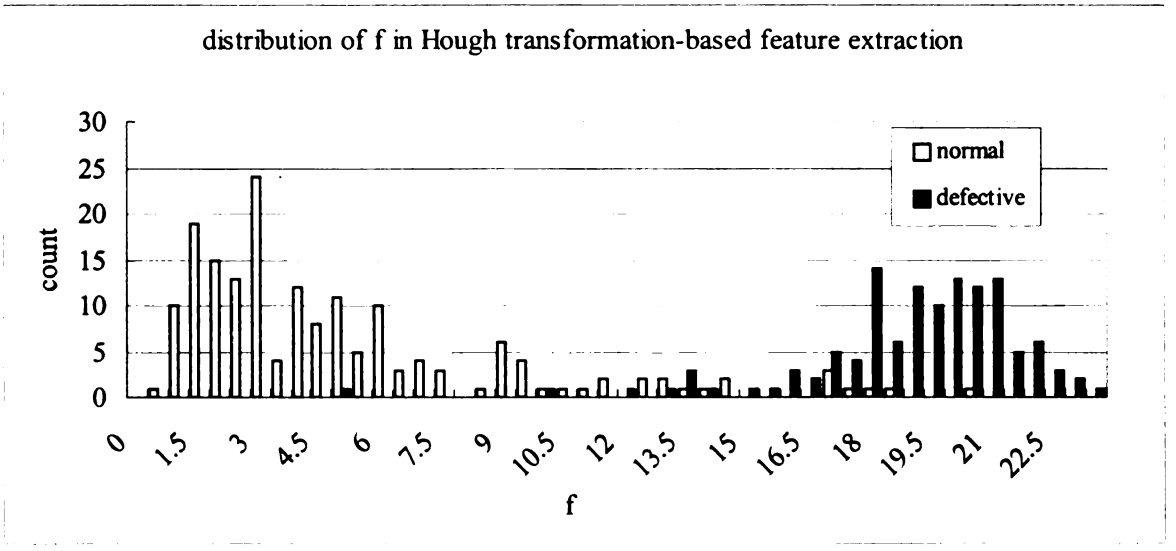
(a) (b)

Figure 3.20: Graphical representation of the feature f for Bayesian classifier (a) Hough Transformation-based rivet detection (b) Morphological operation-based rivet detection.

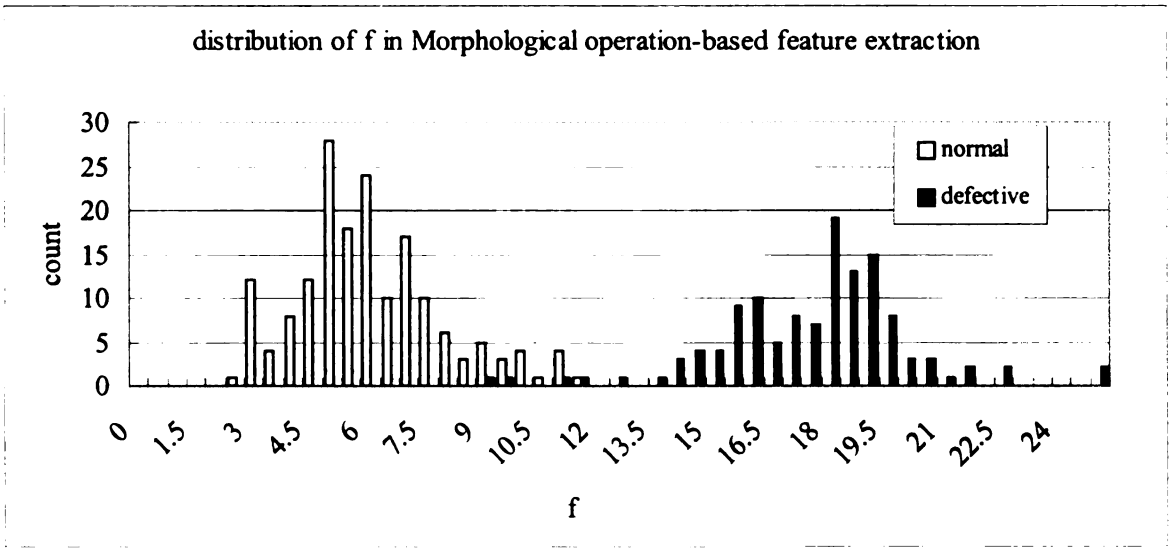
3.4.4 Distributions of the selected feature

Our sample MOI images for the classification test contains 124 defective rivet images and 169 normal rivet images. However defective rivet images are obtained from only two defective rivets and normal images are obtained from 6 different rivets. Therefore, inspecting the correlation of the feature f from those images is a crucial part

for the validity of the rivet classification test. The variations of MOI images from the same object are shown graphically in Section 3.2. The variations of MOI images are also shown by the distribution of feature *f* in Figure 3.21. The feature *f* from normal and defective rivets shows unimodal distributions in Figure 3.21. The distributions of feature *f* allow the rivet classification test to be valid with the MOI rivet images, which are captured multiple times from the same object.



(a)



(b)

Figure 3.21: The distributions of feature *f* from (a) Hough transformation-based rivet detection (b) Morphological operation-based rivet detection.

3.5 Results of Automated Rivet Inspection

A sample sequence of MOI images recorded during aircraft inspection is used to test the rivet inspection algorithm. The MOI data contain a total of 293 rivet images with 124 defective rivet images and 169 normal rivet images. Both Hough transformation-based method and Morphological operation-based method show excellent results for automated rivet detection on the given data set and these results are summarized in Table 3.1.

Table 3-1: Accuracy of two rivet detection algorithms.

Detection algorithm	Rivet detection
Hough Transformation	99% (291/293)
Morphological Operation	99% (291/293)

The automated rivet inspection is tested on a set of MOI images which contain 124 defective and 169 normal rivet images in total. Training data were selected with 30 defective and 30 normal rivet images. The remaining data containg 94 defective rivets and 139 normal rivets were used for testing the performance. For the two-pass Hough transformation classifier, entire rivet images are used for the heuristic search for optimal parameters and test with same set of images. For Bayesian classifier, 100 training data sets were selected randomly and the result presented in Table 3.2 is the average of 100 test results. The performance of the automated rivet inspection algorithm is shown in Table3.2.

Table 3-2: Inspection accuracies of three different inspection algorithms.

Inspection algorithm	Accuracy
Two-pass Hough	52 %
Hough-Bayes	95 %
Morph-Bayes	98 %

From the results shown in Table 3.2, the Bayesian classifier is seen to outperform the Hough classifier. Between the two different rivet detection methods, morphological operation-based rivet detection enables the following Bayesian classifier to perform better. The superiority of the morphological operation-based rivet detection is in the higher inter-class variance for the feature f between a defective rivet image and normal rivet image with oval shape as illustrated in Figure 3.19. Even though the overall inter-class variance is larger with Hough transformation-based rivet detection method as shown in Figure 3.20, the classification between circular normal rivet image and defective rivet image is a trivial problem in many cases.

The detected circles enclosing the rivet using the two rivet detection algorithms differ when the rivet has an oval shape. The Hough transformation-based method finds a circle that fits the edge of the rivet image as much as possible, while the morphological operation-based method finds a circle that is aligned at the true center of the rivet image.

3.6 Conclusion

This chapter addressed the problem of automated processing of MOI data obtained in aircraft skin inspection. Two rivet detection algorithms and two rivet classification algorithms were introduced. The combinations of rivet detection and rivet classification algorithms have been evaluated with the given MOI data. The off-line test results show that the Bayesian classifier is superior to the two-pass Hough classifier. The morphological operation-based rivet detection method is also shown to be beneficial for the higher accuracy of the subsequent Bayesian classifier. These results have to be regarded with caution: more defective rivets need to be imaged for more comprehensive testing. The real-time implementation issues of the automated rivet inspection are discussed in the following chapter.

Chapter 4: Real-Time Implementation Of Automated Rivet Inspection System

4.1 Introduction

So far, we have discussed various algorithms for automated image enhancement via Motion-based Filtering, rivet detection, and classification. In practice when the aircraft skin is inspected, the data interpretation and rivet classification need to be performed in real-time. The MOI instrument generates about 26 images per second [7], so that the image processing system should be able to process a maximum of 26 images per second for real-time inspection capability. The initial inspection algorithm was first implemented and tested off-line on a general-purpose computer using MATLAB. Two approaches were considered for enabling real-time operation of the system. The first approach for making the software more efficient and faster was based on improvements to the algorithms. The second approach relies on the use of a Digital Signal Processing (DSP) board. The algorithm was translated into a C++ program. Further optimizations of each component of the automated rivet inspection system for real-time processing capability are discussed in the following sections.

4.2 Optimization of Motion-Based Filtering

The MBF algorithm, tested on frames of video clips recorded during the MOI inspection, was optimized earlier to provide the best quality of processed images. However the efficiency of the algorithm in terms of time and memory was not considered in its early development stage. The original algorithm for MBF was not fast enough for

real-time inspection. MOI instrument generates 26 images per second, which allows about 38 milliseconds of processing time per image if all 26 images are to be processed. However, in practice the data is over sampled, and hence the number of images per second can be lower for real-time image processing. On reviewing the movie clips of MOI images and subsampling, it was seen that three images per second was the minimum processing speed required for real-time inspection. Therefore the lower limit of the processing speed is based on three images per second, which translates to 333 milliseconds of processing time per image. Table 1 shows the initial processing time of each step of MBF in Matlab and C++ programs.

Table 4-1: Processing time for filtering one image with MBF algorithm using ten past images.

	MATLAB	C++
Capture	-	-
RGB to gray	20 ms (x11)	20 ms (x11)
subtraction	20 ms (x10)	1 ms (x10)
Max	1 ms (x10)	1 ms (x10)
Threshold	10 ms	1 ms
Median Filter (5 x 5)	50 ms	220 ms
Gray level stretch	15 ms	1 ms
Total	505 ms	462 ms

* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

The C++ version outperforms the Matlab version in frame subtraction, threshold, and stretching operations. However Matlab is seen to outperform the C++ version in the median filter operation, indicating the Matlab algorithm for median filter operation is more efficient. This indicates the need for optimization in the C++ version. A systematic optimization of each step of the algorithm is explained in the following sections.

4.2.1 Frame Grabbing

Frame grabbing is the first step of the image processing in MOI inspection. In our configuration, a PCI slot based frame grabber, which gets input signal from a VCR, or a USB port based web camera are used to capture frames. The time taken for frame capture depends on the clock speed of the computer system and the overall architecture of the hardware system. As tested with the 2 GHz CPU PC with 512 Mbytes RAM, the average capture time varies from 15 to 60 ms. This operation mostly depends on the performance of the hardware, with no scope for software optimization.

4.2.2 RGB to Grayscale conversion

Almost all frame capture devices use a color image format of 16-bit or 24-bit. A camera device which provide gray scale tends to be more expensive than the one with color feature. The color image needs much more time for processing than a grayscale image, because each of the red, green, and blue values need to be read and processed separately. For the purpose of defect detection using MOI, a grayscale image with 0~255 intensity values is sufficient. Therefore converting the color image to gray scale before performing any image processing is needed to reduce processing time. Our experiment

revealed that processing 8-bit gray scale image is about 7 times faster than processing 16-bit color with respect to a simple 5 by 5 average filtering. However, converting a color image to gray scale is also a time intensive operation, because the color to gray conversion is performed on every pixel, which amounts to 76800 operations per image of size 320 by 240 pixels. The color conversion formula is described as:

$$Gray = \mathrm{Re}d \times 0.299 + Green \times 0.587 + Blue \times 0.114 \tag{4-1}$$

An optimized color image conversion method was designed by building a color conversion table so that color conversion is performed by a simple table lookup operation. Given blue (B), green (G), and red (R) values for a pixel, the key for the table is determined by:

$$R << 16 \,||\, G << 8 \,||\, B \tag{4-2}$$

where, << represents a bit wise shifting operation and || represents a bit wise OR operation. For a 16-bit color image, there are 2^{16} different key values and each entry has a value of size 8 for the gray scale, which makes a total 2^{19} bits (64 Kbytes). Using table lookup for color conversion is very fast and the memory requirement of 64Kbytes is not high. The measured time for the image conversion from 16-bit to 8-bit is now only 2ms, which is only 1/10 of the previous conversion method as described below.

Table 4-2: Time for converting color image to gray scale.

	Normal Conversion	Using Conversion Table	Table build time (off-line)	Space required
Converting 16-bit color image to 8-bit gray scale	20~23 ms	1~2 ms	5 ms	64 Kbytes

* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

4.2.3 Frame subtraction and Combining

In the original algorithm of MBF [28], the combined difference image is obtained by subtracting the current image from w previous images and performing an OR operation on difference images as:

$$D_i(x,y) = I_{n-i}(x,y) - I_n(x,y)$$
$$S_n(x,y) = \underset{1 \le i \le w}{OR} \{D_i(x,y)\} \tag{4-3}$$

where, $D_i(x,y)$ is the ith difference image, $I_n(x,y)$ is the nth image, and $S_n(x,y)$ is the nth combined difference image. In this case there are w subtractions and w-1 OR operations, or $2\times(w\text{-}1)$ operations total. However, the number of operations to get a combined difference image can be reduced from $2\times(w\text{-}1)$ to w, if the OR operation is performed first with w previous images and the current image is subtracted from the combined image as:

$$s_n(x,y) = \underset{1 \le i \le w}{OR} \{I_{n-i}(x,y)\}$$

$$S_n(x,y) = s_n(x,y) - I_n(x,y) \tag{4-4}$$

This method is about twice as fast as the previous method.

4.2.4 Thresholding

The threshold operation can be integrated into the subtraction operation. When subtraction is performed between two corresponding pixels, a zero value is assigned to the destination pixel if the difference is less than the threshold as:

$$J(x,y) = \begin{cases} I_1(x,y) - I_2(x,y) & \text{if } I_1(x,y) - I_2(x,y) \ge threshold \\ 0 & otherwise \end{cases} \tag{4-5}$$

4.2.5 Median Filtering

Three median filtering algorithms, quick median search, moving median with sorting, and moving median with histogram [29], were tested and moving median with histogram was proven to be the fastest. The median filtering operation obtains the median value from the set of elements in each kernel window. Let the set $S=\{i_1,i_2, \ldots, i_n\}$ be the elements in a $w \times w$ window. Then the median value is defined as:

$$Median = k^{th} \text{ largest element in } S \tag{4-6}$$

$$k = \frac{n-1}{2}, \quad n \text{ is an odd number}$$

In quick sort, a pivot element is arbitrarily chosen from S and partitioning is performed to satisfy the following conditions:

$$\forall i \text{ such that } i \in P_1,\ i < pivot$$
$$\forall i \text{ such that } i \in P_2,\ i > pivot \tag{4-7}$$

where, P_1 and P_2 are partitions. The partitioning operation is performed on P_1 and P_2 recursively until the number of elements of all partitions are one. All partitions are combined at the last step to obtain the sorted list of S. In modified quick sort, each partitioning is performed on only one of two partitions, which contains the k^{th} largest element. The operation stops when the pivot is k^{th} largest, in which case the pivot is the median value. The modified quick sort-based median search is faster because it minimizes the sorting operation: $\Theta(n)$ whereas quicksort is $\Theta(n \log n)$.

In moving median with sort, the median value is obtained at the first kernel window by sorting S. After moving the window to the next position, w elements are removed and w elements are added to S, thus the sorting operation becomes very fast by removing the overlap of pixels.

In moving median with histogram, a histogram is built and operated during the median filter. The histogram holds the counts of values in current window, so that the median value is found by counting the histograms from 0 to 255: the bin with the

accumulated counts larger than or equal to k becomes the median. This median bin is remembered at each move of window, and as adding and removing elements from the histogram, the previous median bin is adjusted and the median is quickly searched. More formal explanation of the moving median with histogram is described below. A histogram of size 256 is defined as:

$$H(i) = 0,\ 0 \le i \le 255 \tag{4-8}$$

The moving median with histogram initializes variables as:

$$\begin{aligned} &H\left(I\left(x+i,y+i\right)\right) = H\left(I\left(x+i,y+i\right)\right)+1, \quad -w<i<w, \quad -w<j<w \\ &M_{current} = \arg\min_{j} \sum_{i=0}^{j} H(i) \ge k \\ &N_{less} = \sum_{i=0}^{Mcurrent} H(i) \end{aligned} \tag{4-9}$$

where, $I(x,y)$ is the original image, $M_{current}$ is the median value in current window and k is the variable defined in equation (4-5). As the window is moved, w elements are removed from H and the variables are updated as:

$$\begin{aligned} &If\ S(i) < M_{current},\ then\ N_{less} = N_{less} - 1,\ H(i) = H(i) - 1 \\ &Else\ H(i) = H(i) - 1 \end{aligned} \tag{4-10}$$

where, S(i) represents an element in the given window. The variables are updated similarly as w elements are added as:

$$If\ S(i) < M_{current},\ then\ N_{less} = N_{less} + 1,\ H(i) = H(i) + 1$$
$$Else\ H(i) = H(i) + 1 \tag{4-11}$$

After updating the variables, the median value is obtained quickly by the following expression.

$$If\ N_{less} \geq k,\ then\ M_{current} = \arg\max_{j} \left\{ (N_{less} - \sum_{i=j}^{Mcurrent} H(i)) \geq k \right\}, j \leq M_{current}$$
$$Else\ M_{current} = \arg\min_{j} \left\{ (N_{less} + \sum_{i=Mcurrent}^{j} H(i)) \geq k \right\}, M_{current} \leq j \tag{4-12}$$

The measured time for the three median filtering algorithms are shown in Table 4.3 and the moving median with histogram method is the fastest.

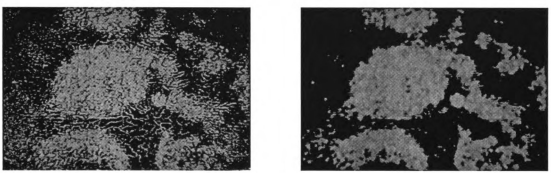
Table 4-3: Processing time of median filtering with various algorithms.

Algorithm \ Kernel size	5 by 5	7 by 7
Quick Median search	200 ~ 250 ms	400~450 ms
Moving Median with Sort	100~150 ms	150~200 ms
Moving Median with Histogram	20~25 ms	20~40 ms

* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

Figure 4.1 shows the effect of median filtering on a combined difference image with severe noise. Median filtering is more important for improving the image quality as the number of difference images increases.



(a) (b)

Figure 4.1: Effect of median filtering in MBF. (a) Before median filtering. (b) After Median filtering with 5 by 5 kernel. Bright objects represent rivets and corrosion.

4.2.6 Contrast Stretching

Stretching is performed to enhance the contrast of objects in the filtered image. The transformed image *J(x,y)* after stretching operation is denoted by:

$$J(x,y) = 255 \times \frac{I(x,y)}{Max(I(x,y))} \tag{4-13}$$

For better contrast, the highest 1% of intensity values is trimmed in getting the maximal intensity value. Rather than doing the full search, a histogram method can be used. The modified stretching operation combined with histogram searching is expressed as:

$$J(x,y) = 255 \times \frac{I(x,y)}{N}$$

$$\arg\max_{N}\left(\sum_{i=N}^{255} H(i) \geq 768\right) \tag{4-14}$$

where H represents the histogram.

4.3 Results of the optimized MBF

The resulting execution time of the MBF algorithm after the optimization described above is shown in Table 4.4.

Table 4-4: Measured time for each step of MBF before and after optimization.

	Before optimization Intel 2GHz, C++		After optimization Intel 2GHz, C++
Capture	20 ms	Capture	20 ms
RGB to gray	20 ms (x11)	RGB to gray	1 ms (x11)
Subtraction	1 ms (x10)	Max	1 ms (x10)
Max	1 ms (x10)	Subtraction Threshold	1 ms
Threshold	1 ms		
Median Filter (5x5)	220 ms	Median Filter (5x5)	25 ms
Gray level stretch	1 ms	Gray level stretch	1 ms
Total	482 ms	Total	68 ms

* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

The total execution of the optimized MBF algorithm is about 70 ms with subtraction performed on 10 past images. Therefore, the optimized MBF algorithm is capable of processing about fourteen images per second.

4.4 Optimizing Rivet Detection and Classification

The Hough transformation takes a few hundred milliseconds to execute, which makes it unsuitable for real-time processing. On the contrary, the morphological operation-based rivet detection takes only a few tens of milliseconds to execute. The execution times of the two different rivet detection algorithms are shown below.

Table 4-5: Execution times of Hough transformation-based and morphological operation-based rivet detection.

	Hough transformation	Morphological operation
Rivet detection	400 ms	30 ms

* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

Due to the fast execution time, morphological operation-based rivet detection is suited for the real-time rivet inspection algorithm. Therefore, morphological rivet detection is chosen as a suitable algorithm to be implemented in the real time rivet inspection system. The performances of the two different rivet detection algorithm with respect to accuracy were shown to be the same in Table 3.1 and 3.2. For rivet classification, the Bayesian classifier is superior to the Hough transformation-based classifier with respect both to execution time and accuracy. Thus the combination of morphological operation-based

rivet detection and Bayesian classifier is chosen in the real-time automated rivet inspection system. Testing with more data is necessary to justify this analysis.

4.5 Personal Computer-Based Proof-of Concept System

The automated rivet inspection system was implemented on a PC in the C++ programming language to test its real-time capability. The experimental setup of the test is shown in Figure 4.2.

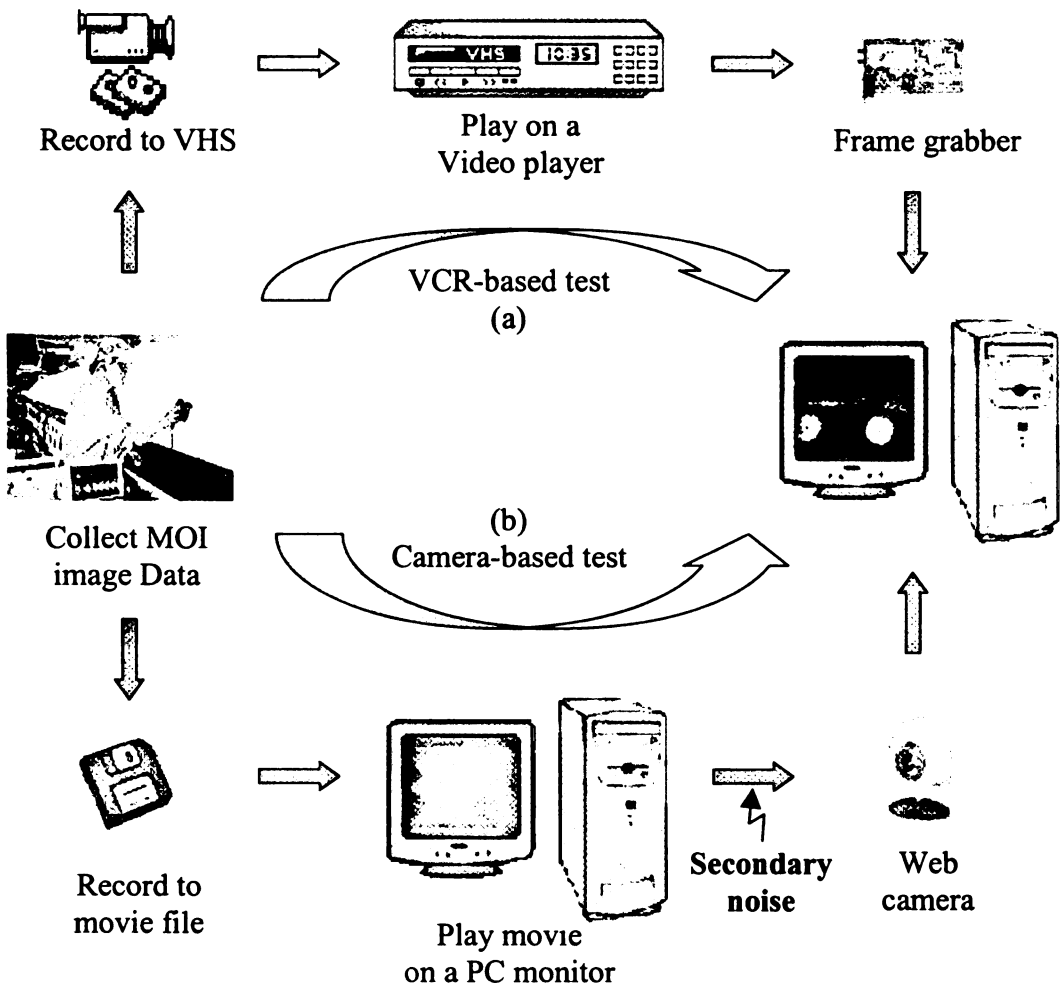


Figure 4.2: A proof-of-concept system for real-time automated rivet inspection.

As shown in Figure 4.2, two different proof-of concept systems were developed. The major difference is the method of input of the image data: Figure 4.2 (a) uses a VCR and Figure 4.2 (b) uses a camera. The VCR-based method is better than the camera-based method because it does not involve a source of secondary noise.

4.6 Digital Signal Processor (DSP)-Based Prototype System

The automated rivet inspection algorithm is being implemented on a DSP board platform. The program is developed in the framework of Code Composer Studio (CCS). The CCS translates a C++-like program into an assembly program. The prototype system is depicted in Figure 4.3.

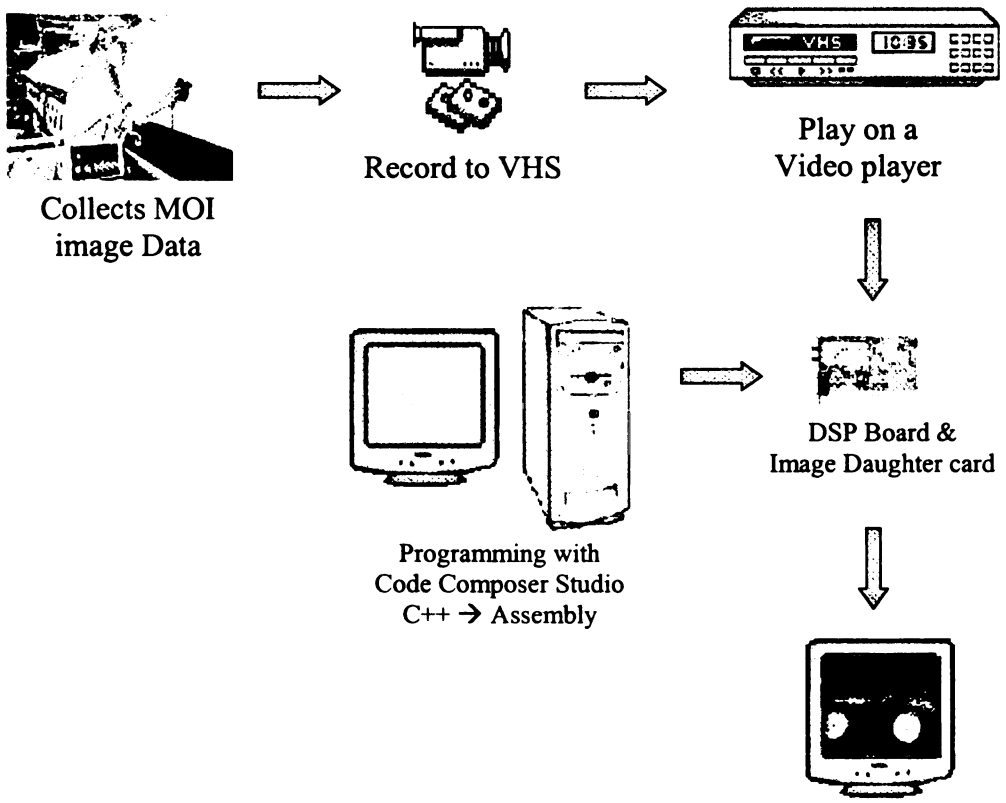


Figure 4.3: A prototype system for the real-time automated rivet inspection system.

4.7 Results of Real-Time Automated Rivet Inspection System

The execution times of the automated rivet inspection processes are shown in Table 4.6.

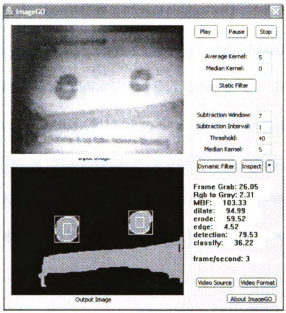
Table 4-6: Execution time of the automated rivet inspection

	Execution time
MBF	68 ms
Rivet detection (including preprocess)	150 ms
Rivet classification	30 ms
Total	249 ms

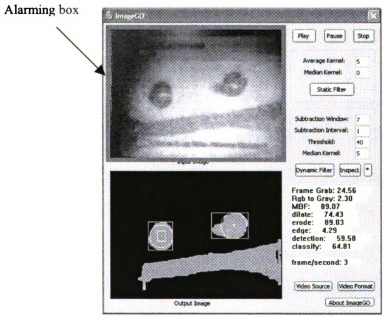
* Measured with 2GHz CPU, 512 RAM PC

* Image size: 240 x 320

The results of the C++ program for automated rivet inspection is shown in Figure 4.4. The program is capable of processing 3 ~ 5 images per second and shows a red box to indicate that there is a defective rivet in current image.



(a)



(b)

Figure 4.4: Results of the C++ proof-of-concept system. (a) Raw and processed MOI image without defect. (b) Raw and processed MOI image with defective rivet, the raw image is boxed to highlight the defective rivet.

4.8 Conclusion

The full rivet inspection algorithm, including motion-based filtering, rivet detection and classification, was implemented and tested on a 2 GHz, 512 RAM PC. The test showed capability of the system for processing 3 to 5 images per second. The rivet inspection algorithm is being implemented on a DSP board, which is the target platform. The implementation has to be adapted for the small memory of the DSP board. The algorithm will also be optimized to fully utilize the fast image processing capabilities of the DSP board.

Chapter 5: CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Magneto-Optic Imager (MOI), while providing fast and easy nondestructive inspection capabilities, provides two major challenges. First, the magneto-optic image is severely contaminated with the serpentine pattern noise. Second, the large inspection area of an aircraft body with thousands of rivets makes the manual inspection of MOI tedious and prone to operator fatigue.

The serpentine pattern noise not only lowers the probability of detection (POD) but also makes the inspection subjective. By analyzing the characteristics of magneto-optic images, the Motion-Based Filtering (MBF) algorithm was developed, where only moving objects in a sequence of images are extracted and stationary background is suppressed by using a multiple frame subtraction method. The serpentine pattern noise is removed by MBF due to its stationary nature in a sequence of images. The MBF greatly improves the POD by removing background noise in the magneto-optic images. The contrasts of MOI images are improved from .17 to .61 on average by MBF operation. MBF also affects the potential to build an automated inspection system based on MOI because it is simpler and faster to process binary output images of MBF than the original color images. With the binary output images of MBF processing, an automated inspection algorithm was developed.

The automated inspection algorithm characterizes certain distinct features from defect images. The wide variance of features among various defects makes it necessary for the inspection algorithm to be adaptive. The first automated inspection algorithm was

developed to inspect rivet images. Rivets are the key location from which a crack originates. The inspection algorithm first detects a rivet in a magneto-optic image, obtains features of the rivet, and then makes a decision based on the features. Rivet detection algorithms, based on Hough transformation and morphological operations were developed and evaluated. Both algorithms for rivet detection show promising results on real magneto-optic image data, however, more images of defective rivets are needed for convincing results.

Two rivet classification algorithms were also developed for the purpose of identifying defective rivets, namely i) two-pass Hough transformation classifier and ii) Bayesian classifier. The two-pass Hough transformation classifier applies Hough transformation on the input image twice, first to detect the rivet and second to detect blob like objects associated with a radial crack. The Bayesian classifier uses a feature value in the image to differentiate defective rivets from normal rivets. The feature represents the length of a blob attached to a rivet image when there is a radial crack. These features were collected from training data of defective and normal rivet images. The Bayesian classifier obtains the feature value from the test image and decides if it belongs to the distribution of defective or normal rivet images, then classifies the rivet accordingly. The Bayesian classifier outperforms the two-pass Hough classifier with respect to classification accuracy. Between the two rivet detection algorithms, morphological operation-based rivet detection is shown to provide higher accuracy for the following Bayesian classifier. This result needs to be verified by testing with more images of defective rivets.

A proof-of-concept system was designed to test the automated rivet inspection algorithm with respect to its real-time capability. Each step of the automated rivet inspection algorithm was optimized for real-time capability and higher inspection accuracy. The final algorithm for automated rivet detection was composed of MBF, morphological operation-based rivet detection, and Bayesian classifier. The proof-of-concept system shows that the automated inspection algorithm is capable of processing 3 to 5 images per second, which can be considered near real time. This execution time is expected to be much faster when the algorithm is implemented on a DSP board.

5.2 Future Work

The MBF algorithm is based on the assumption that the background noise of MOI is static. While this is true in most MOI images, some MOI images show abrupt background intensity changes, especially around a large object. This is believed to be due to the large magnetic field around a large anomaly. Improving the MBF algorithm to detect this static background change from a moving object in the image would make the algorithm more robust. The background intensity change occurs in a fixed area and does not move its centroid. Therefore, one of the possible approaches is to track the motion of the an object in the image and differentiate it from a static intensity change in the background.

The automated inspection algorithm needs to be developed for types of defects other than rivets, such as cracks around a seam and corrosion. After all algorithms are developed for all types of defects, they should be combined for the automated aircraft inspection system that is capable of inspecting various types of defects. In addition, the

inspection algorithm can use the temporal information of decisions. In MOI inspection, an object is observed in more than one image, which generates a sequence of decisions for an object. By observing the sequence of decisions, an erroneous decision due to a perturbation of signal can be suppressed.

Finally, the automated rivet inspection needs to be implemented on the DSP system. With the aid of Code Composure Studio (CCS) most of the C++ version program is expected to be reusable except for video capture and memory management routines. The original algorithm needs to be optimized for the small memory of the DSP board. The algorithm is also being optimized to fully utilize the specialized computing power of the DSP board. The DSP version of the inspection system would be more convincing evidence of the effectiveness of automated rivet inspection. The DSP board must be customized to make the system smaller and integrated with the commercial MOI system either as a portable inspection instrument or a component of a self-guiding robot which represents the fully automated rivet inspection system.

BIBLIOGRAPHY

[1] B. Raj, T. Jayakumar, and M. Thavasimuthu, "Pratical Non-Destructive Testing," 2nd Ed. 2002

[2] K. G. Boving, Non-Destructive Evaluation Handbook. London: Butterworths, 1989.

[3] B. D. Cullity, "Elements of X-Ray Difraction," 2nd Ed., Addison-Wesley Publishing Company, Inc.

[4] G. L. Fitzpatrick, D. K. Thome, R. L. Skaugest, E. Y. C. Shih, and W. C. L. Shih, "Novel Eddy Current Field Modulation of Magneto-Optic Garnet Films for Real-Time Imaging of Fatigue Cracks and Hidden Corrosion, " Physical Research Inc., Torrance, California, 1993

[5] D. E. Bray and R. K. Stanley, Nondestructive Evaluation, New York, McGraw Hill 1989.

[6] "Development of an Improved Magneto-Optic/Eddy-Current Imager," Office of Aviation Research, DOC/FAA/AR-97/37, Washington D.C.

[7] Fitzpatrick G. L. et al. "Magneto-optic/Eddy Current Imaging of Aging Aircraft: a New NDI Technique", Material evaluation, pp. 1402-1407, Dec. 1993

[8] W. C. L. Shih, G. L. Fitzpatrick, D. K. Thome, R. L. Skaugset, and E. Y. C. Shih, "Aircraft Inspection with the Magneto-optic/Eddy Current Imager," CSNDT Journal, Canada's National NDT Magazine, March/April, 1994, Vol. 15, No. 2, pp. 13-30.

[9] G. L. Fitzpatirck, D. K. Thome, R. L. Skaugset, E. Y. C. Shih, and W. C. L. Shih, "Magneto-Optic/Eddy Current Imaging of Aging Aircraft: A New NDI Technique," Materials Evaluation, pp. 1402-1407, Dec. 1993

[10] Official Website of PRI Research & Development Corp, http://www.prirnd.com/prod02.htm,

[11] G.L. Fitzpatrick, et al. "Magneto-Optic/Eddy Current Imaging of Subsurface Corrosion and Fatigue Crack in Aging Aircraft", Review of Progress in Quantative Nondestructive Evaluation, V15, 1996

[12] V. J. Brechling and F. W. Spencer, "The Validation Process ads Applied to the Magneto-Optic/Eddy Current Imager (MOI)," Materials Evaluation, pp. 815-818, July 1995.

[13] Jaein L. and Udpa S.S., "Defect extraction in MOI image based on wavelet packet transform and morphological technique", in Review of Progress in Quantitative

Nondestructive Evaluation, eds. D.O. Thompson and D.E cChimenti, Vol. 20, 2001, pp. 611-618

[14] Lemistre M. and Decitre J.M., "Corrosion and cracks detection in metallic structures by magneto-optic image processing", in Proceedings of International Conference on Quality Control by Artificial Vision, editing by Cepadues, Le Creusot (France), 2001, pp. 65-70.

[15] H. H. Nagel, "Overview On Image Sequence Analysis," Image Sequence Processing and Dynamic Scene Analysis, pp 2-39, Springer Verlag, 1983

[16] H. H. Nagel, "Image Sequence – Ten (Octal) Years – From Phenomenology Towards a Theoretical Foundation," Proc. International Conference On Pattern Recognition," pp. 1174-1185, Paris, France, 1986

[17] A. Shio and J. Sklansky, "Segmentation of people in motion," Proc. Of the IEEE Workshop on Visual Motion, pp. 325-332, Princeton, NJ, 1991

[18] C. Anderson, P. Burt, G. van der Wal, "Change detection and tracking using pyramid transformation techniques," SPIE Intelligent Robots and Computer Vision, V. 579, pp. 72-78, 1985

[19] Y. Z. Hsu, H. H. Nagel, and G. Rekers, "New likelihood test methods for change detection in image sequences," Computer Vision, Graphics and Image Processing, 24, pp. 73-106, 1984

[20] L. G. Shapiro, G. C. Stockman, "Computer Vision". New Jersey: Prentice Hall, 2001

[21] B.K.P Horn, B.G. Schunck, "Determining Optical Flow", Artificial Imtelligence, 17, 1981, pp. 185-203

[22] Otsu, N., "A threshold selection method from gray-level histograms", IEEE Trans. Syst., SMC-9, 1979, 62-66

[23] D. S. Bright, D. E. Newbury, E. B. Steel, "Visibility of objects in computer simulations of noisy micrographs", J. of Microscopy, 189(8), Jan. 1998, 25-42

[24] P. Ramuhalli, J. Slade, U. Park, L. Xuan, L. Udpa, "Modeling and Signal Processing of Magneto Optic Images for Aviation Applications," Presented at the International Conference on Smart Materials Structures and Systems, Bangalore, India, Dec12-14, 2002

[25] Duda, R., and Hart, P., "Use of the Hough Transform to Detect Lines and Curves in Pictures", Communications of the ACM 15, pp: 11–15, 1975.

[26] Tina Yu Tian, Mubarak Shah, "Recovering 3D Motion of Multiple Objects Using Adaptive Hough Transform," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 10, OCTOBER 1997

[27] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification," John Wiley & Sons, Inc. 2001, pp. 20-45

[28] Unsang Park, Lalita Udpa, George. C. Stockman, Bill Shih, Jerry Fitzpatrick, "Real-time Implementation of Motion-based Filtering in Magneto-Optic Imager," Presented at 30th Annual Review of Progress in Quantitative Nondestructive Evaluation, Green bay, Wisconsin, Jul. 29-Aug. 1, 2003

[29]. T. S. Huang, G. J. Yang, G. Y. Tang, "A Fast Two-Dimensional Median Filtering Algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, V. ASSP-27, No. 1, Feb, 1979