Image Understanding Algorithms for Remote Visual Inspection of Aircraft Surfaces.

Priyan Gunatilake[*], M.W. Siegel[**], A.G. Jordan[*,**], G. Podnar[**]

[*]Department of Electrical and Computer Engineering
[**]Robotics Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, PA. 15215

ABSTRACT

Visual inspection is, by far, the most widely used method in aircraft surface inspection. We are currently developing a prototype remote visual inspection system, designed to facilitate testing the hypothesized feasibility and advantages of remote visual inspection of aircraft surfaces. In this paper, we describe several experiments with image understanding algorithms that were developed to aid remote visual inspection, in enhancing and recognizing surface cracks and corrosion from the live imagery of an aircraft surface. Also described in this paper are the supporting mobile robot platform that delivers the live imagery, and the inspection console through which the inspector accesses the imagery for remote inspection. We discuss preliminary results of the image understanding algorithms and speculate on their future use in aircraft surface inspection.

Keywords: ANDI, CIMP, visual inspection, image understanding, stereo vision, multiresolution analysis, wavelets, surface defect detection

1. INTRODUCTION

Visual inspection of aircraft is the most widely used method employed for ensuring the structural integrity of an aircraft skin and its substructure. For example, a typical heavy inspection carried out on a commercial aircraft after every 12,000 flying hours, is about 90% visual and 10% non-destructive inspection(NDI).[1] Visual inspection involves putting a human inspector on the body of the aircraft to visually examine its surface for defects such as cracks, corrosion, damaged rivets, lightning strikes etc. This practice raises safety issues for the inspector, is time consuming, and suffers at times from being ineffective due to inspector fatigue or boredom.[2]

An attractive alternative to the current inspection practice is remote visual inspection. In remote visual inspection, the inspector examines, at an inspection console, high-quality imagery of the inspection surface that is captured and delivered to the console by a remote mobile robot on the body of the aircraft. The robot may be teleoperated via low level controls, it may navigate autonomously under computer control, or typically something in between with high level commands issued by the inspector and low level details decided and executed by the computer. This method, while inherently safe (since the inspector is on ground), allows for direct human observation of the remote aircraft surface. It also provides for computer processing of the delivered imagery for image processing, enhancing and understanding. Image processing involves adjusting contrast or range of the imagery dynamically, for improved visualization. Image enhancement amplifies high spatial frequencies of the imagery to highlight features suggestive of surface defects which are typically of high frequency nature[3]. Image understanding via characterization and recognition of surface defects, allows for automated defect detection and classification of the surface imagery. With the aid of these facilities, an inspector can safely, quickly and accurately perform the necessary visual inspection from the inspection console.

In Section 2 of this paper, we describe a prototype mobile robot called the Crown Inspection Mobile Platform (CIMP) designed to test and demonstrate the hypothesized feasibility and advantages of the remote visual inspection of an aircraft surface. Also included in section 2 is a brief description of the predecessor to CIMP, the Automated NonDestructive Inspection (ANDI) robot.[4,5] Section 3 discusses the inspection console that displays the remote imagery and a graphical user interface (GUI) that provides the inspector with access to image processing, enhancing and understanding algorithms. Section 4. contains a brief discussion of image understanding for surface defect detection and a description of two common aircraft surface defects. Section 5. describes a surface crack detection algorithm. Section 6. describes a surface corrosion detection algorithm. Section 7 contains a discussion of the surface crack detection and corrosion detection algorithms and description of our future work. Section 8. contains the conclusion.

2. CIMP

The first aircraft-capable mobile robot developed at CMU was ANDI (the Automated NonDestructive Inspector of aging aircraft). ANDI successful1y demonstrated mobility, manipulation and navigational capabilities on an aircraft surface. However, due to the ANDI project* emphasis on mobility and navigational issues, the delivery of high quality visual imagery useful for remote visual inspection was not addressed at length. After the initial demonstrations of ANDI, the second author launched another research effort* with the twin objectives of designing a high quality remote imaging system that delivers useful inspection data and developing an inspection console consisting of a graphical user interface (GUI) and a library of image enhancement and understanding algorithms, through which an inspector could access, enhance and recognize surface defects from the live imagery.

CIMP was developed as a part of this second research effort. CIMP is a wireless remote-controlled mobile vehicle that carries a sensor package designed to deliver high quality, live imagery of the aircraft crown on which it travels. The sensor package of CIMP contains a stereoscopic pair of inspection cameras, a dynamic lighting array consisting of two fixed flood lights and a rotatable directional light source, and a stereoscopic pair of proprioceptive navigational cameras. The inspection cameras were developed in our laboratory, and are constructed in a geometrically correct imaging configuration that provides 3.5x magnified, natural, easy to view, high quality stereoscopic imagery of the aircraft surface.[6,7] The navigational and proprioceptive cameras provide a wide-angle stereoscopic view of CIMP with respect to the aircraft body that is used by the inspector to control and navigate CIMP. Left and right frames of the inspection or navigational camera pairs are interleaved at 120 Hz on a monitor in the inspection console, and viewed stereoscopically through active eyeware. Figures 1 and 2 show the ANDI and CIMP robots.

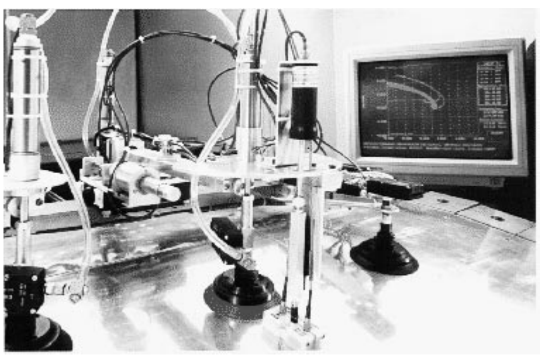

Fig 1. Automated NonDestructive Inspection (ANDI) robot

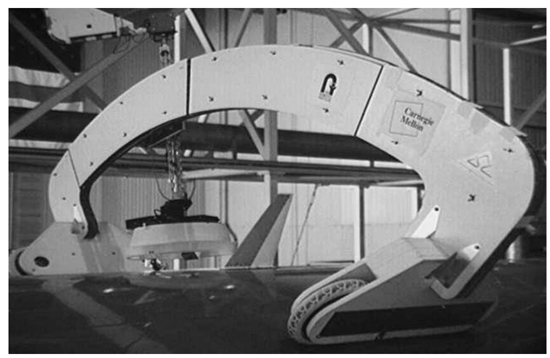

Fig 2. Crown Inspection Mobile Platform (CIMP) robot

3. INSPECTION CONSOLE

Fig 3a. live video station

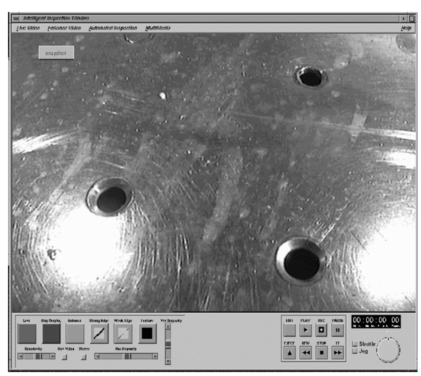

Fig 3b. Intelligent Inspection Window(IIW)

The inspection console, through the display of stereoscopic imagery delivered by the inspection cameras and the proprioceptive navigational cameras, provides for remote visual inspection of aircraft surface and remote control and navigation of CIMP on the aircraft body. The current inspection console consists of two primary displays and their supporting equipment and a radio transmitter (of the type used to control model vehicles) that controls forward and backward motion, left

*See acknowledgments for additional contributions right steering, camera position and orientation, and lighting selection and orientation. The first display is a monitor that provides live, flicker-free, full spatial and temporal resolution per eye, stereoscopic imagery of either the inspection or navigational camera pair. The second is a Silicon Graphics Indy Workstation with a GUI that we call the Intelligent Inspection Window(IIW). The IIW performs a variety of tasks; it displays live monoscopic or still stereoscopic imagery on a part of its canvass called the display screen; it acts as the operational interface and output display unit to the image enhancement and understanding algorithms that are tied to the menus and buttons of the IIW; and in the future, it will contain facilities for creating, editing, storing and retrieving multimedia records of surface defects. Figure 3a. displays the live video station. Figure 3b. displays the IIW of the image processing station.

4. IMAGE UNDERSTANDING AND SURFACE DEFECTS - BACKGROUND

The goal of an image understanding algorithm is to recognize and classify certain surface flaws from the live imagery. The recognition capability of this algorithm is achieved by correlating features of the live imagery with prior or learned knowledge of the surface flaw type. A high correlation of a feature in the live imagery with a flaw type will result in the feature being classified as a flaw of the correlated type. However, developing a successful image understanding algorithm remains a non-trivial challenge due to its dependency on factors such as normal and defect feature characterization, imaging resolution and environment to name a few.

One possible scenario for application of image understanding algorithms in remote visual inspection is screening large volumes of image data. The image understanding algorithm can conservatively label all plausible defects, so that the inspector separates a larger fraction of actual defects from normal features in a smaller volume of raw data. Another scenario is the interactive use of these algorithms by inspectors to obtain a second opinion about a particular suspicious flaw. The latter possibility is most attractive when the real-time inspector is relatively inexperienced, in general or with respect to a specific problem, compared to the inspector or inspectors whose expertise has been incorporated (explicitly or implicitly) in the algorithm; in this case the computer fulfills a training role in addition to its direct inspection role.

We have at present developed image understanding algorithms designed to detect surface cracks and corrosion. Surface cracks and corrosion are two of the more important and common defects that are aggressively inspected for by visual inspectors.

4.1. Surface cracks

Pressurization and de-pressurization of the aircraft during every flying cycle causes its body to expand and contract in a manner similar to inflating and deflating of a balloon. This expansion and contraction induces stress fatigue at rivets (which hold the aircraft surface skin to its frame), resulting in the growth of cracks outward from the rivets.

The growth of a surface crack is essentially exponential in nature. There are many reliable models[8] which predict crack growth quite accurately as a function of the number of pressurization and depressurization cycles. The goal of visual inspection is to detect cracks that are above a minimum threshold length. This threshold length provides a safety margin that allows a crack to be missed in two or three consecutive inspections before it is big enough to endanger the structure of the aircraft.

One of the main methods inspectors use to find cracks is to observe the reflection of directional lighting incident on a rivet location, using a flashlight held at a low angle to the surface[9]. Absence of reflecting light from an edge (line on the surface) emanating from the rivet suggests the possibility of a crack; on the other hand, reflection of light indicates a scratch, which if small is harmless. Therefore the task for an inspector is to first detect edges emanating outwards from the rivets and then discriminate the cracks from scratches and other edges from that edge pool. Unfortunately, due to the presence of tens of thousands of rivets on the aircraft body, inspection for cracks is a demanding and tiring task for the inspector.

4.2. Surface corrosion

Corrosion is common due to the frequent exposure of the aircraft body to environments such as aircraft operating fluids, liquids spilled in the galleys, lavatory liquids, moisture of sea air etc. Corrosion can appear as subsurface or surface corrosion. Surface corrosion is recognized by the appearance of corrosion texture. Subsurface corrosion is recognized by the bulging of the affected surface region, referred to as "pillowing" by the inspectors. Since corrosion results in a loss of structural material of the affected area, early detection is crucial. Corrosion is also known to induce cracking.

5. SURFACE CRACK DETECTION ALGORITHM

The crack detection algorithm that we have developed is modeled closely on the widely practiced test for detection of cracks using directional lighting. We simulate the directional lighting produced by the inspectors flashlight with a remotely controlled rotatable directional light source on CIMP. The inspector can remotely rotate the light source around a rivet location and examine the resulting live monoscopic or stereoscopic imagery of the rivet and its neighborhood for cracks. In addition, the inspector can run the crack detection algorithm on these images for detection or verification of cracks in the live imagery. The stereoscopic imagery can also be recorded (at slightly reduced resolution) on a standard VHS recorder for future examination or computer processing.

Figure 4a. shows a section of an aircraft surface containing two natural cracks and several scratches appearing in the neighborhood of a rivet hole. The first crack emanating from the rivet hole is 1/2 inch of length while the second crack which is partly masked by the scratch beside it is 1/3 inch long. The output of the surface crack detection algorithm is shown in figure 4b. Edges that are marked in black indicate suspected cracks. Edges marked in grey indicate edges that the algorithm detects but classifies as “non cracks” The algorithm detects the two known cracks which are marked in black in the output image. It also correctly classify’s the edges of the rivet hole and scratches as “non crack” edges. The other edges that are marked in black are false alarms for cracks. Figure 5 displays a block diagram of the crack detection algorithm.

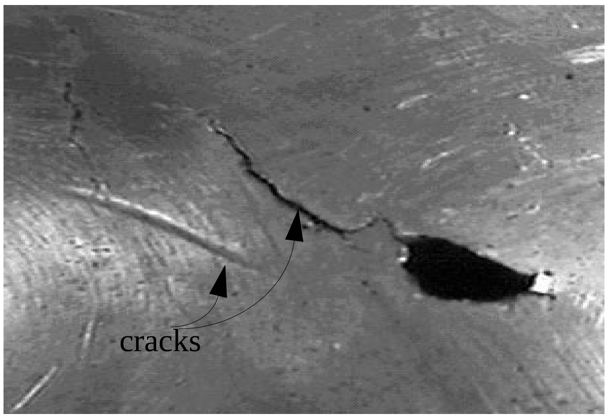

Fig 4a. Metal surface with two cracks and other crack-like features

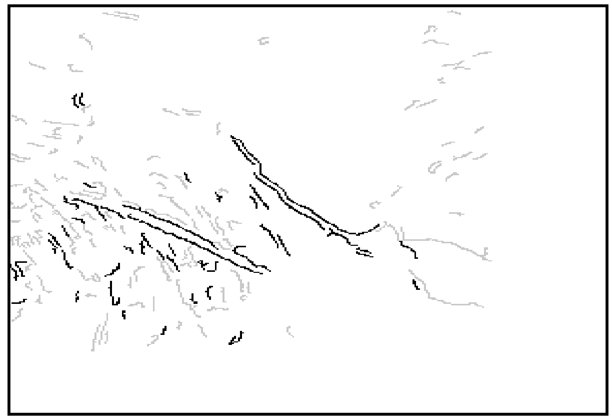

Fig 4b. Output of crack detection algorithm
Cracks in black and non-cracks in grey

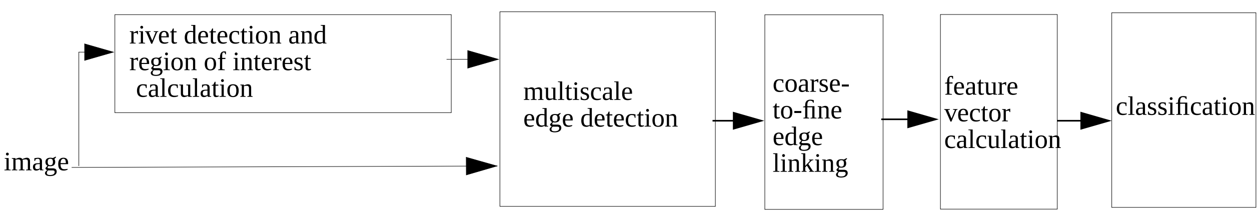

Figure 5. Block diagram of the surface crack detection algorithm

5.1. Rivet detection and region of interest calculation

The first step of our crack detection algorithm is to find rivet locations in the image. Since cracks appear in the neighborhood of rivets, finding rivet locations enables the algorithm to focus on areas that are most likely to contain cracks. The neighborhood surrounding the rivet that is examined for cracks is defined as the Region of Interest (ROI). The algorithm defines for each detected rivet, a ROI. By focussing on ROIs, the algorithm avoids unnecessary processing of features outside the ROIs they are not likely to be cracks. Rivets are identified by detecting the circular arc edges made by their heads. Rivet detection and ROI determination consists of the following steps.

1. Smooth the image with a gaussian filter for noise reduction. Convolve the smoothed image with a finite-difference approximation for the partial derivatives. Calculate the gradient magnitude image.
2. Calculate the histogram of the gradient magnitude image. Define the low threshold to be the value such that the interval, low threshold to the maximum value in the histogram, contains 20% of the total pixels in the image. Similarly, define the high threshold as the value that contains 10% of the pixels (This process is known as double thresholding). Threshold the gradient magnitude image at the high threshold and mark the pixels above the threshold as edge points. Link the edge points based on 8-neighbors to create edges.
3. Threshold the gradient magnitude image at low threshold. Grow edges found in 2) with new edge pixels above the low threshold. Discard the edges that are less than a minimum length (we use 10 pixels).

4. Fit a circular arc to each edge. Calculate the fitting error and discard all edges above a minimum fit error. The remaining edges are from the rivet heads which we assume to be the only objects in the surface image with circular edges.
5. Merge the edges that are close to each other to form groups of circular edges, each belonging to a separate rivet present in the image. Calculate the centroid (i_0, j_0) of each group of circular edges. The ROI for each rivet is defined as $ROI = \{(i, j) \| i_0 - os \le i \le i_0 + os, j_0 - os \le j \le j_0 + os\}$ where *os* is an offset. The pixel range is bound by the image size (0: M-1, 0: N-1).

Figure 6a. displays a section of an aircraft surface with three simulated cracks appearing as dark lines from the two rivets. This image is processed by the crack detection algorithm. Figure 6b. displays the two ROIs found by the algorithm

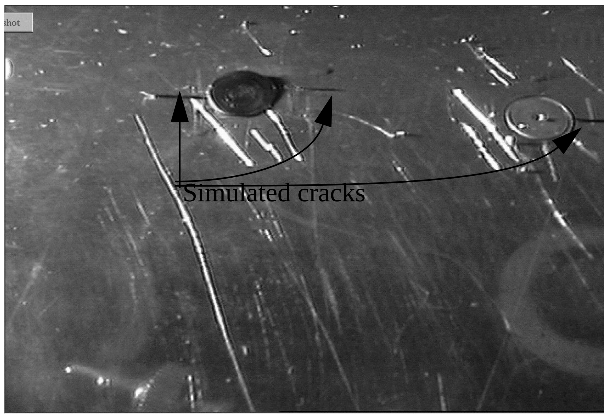

Fig 6a. Metal surface with three synthetic cracks and other crack like features

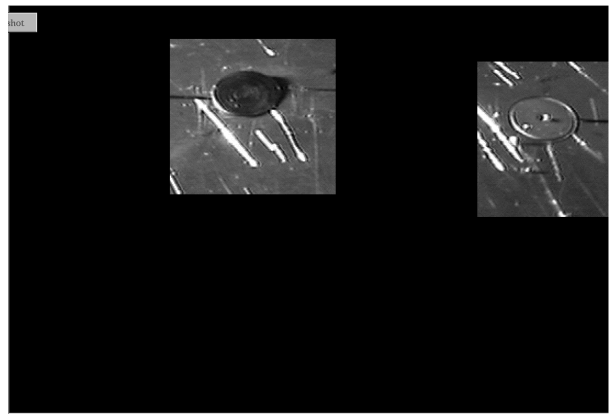

Figure 6b. ROI’s determined for live image in fig 6a

5.2. Multiscale edge detection

A ROI in the live image contains a large number of edges, most of which are caused by rivet edges, scratches, dirt marks, lap joints of metal repair plates on the surface and occasionally real cracks. Therefore, we need an analysis framework which lends itself to the discrimination of the small fraction of cracks which are edges of interest to us, from those edges that are not of interest.

A crack is typically very small compared to other objects present on the aircraft surface such as rivets, scratches etc. This motivated us to select a multiscale edge detection framework for the detection and analysis of edges in the ROIs. Multiscale edge detection is defined as detection of edges at multiple scales or, equivalently multiple resolutions. Here, scale implies the size of the neighborhood in which intensity changes are detected for edge determination. In multiscale edge detection, edges belonging to small objects appear at low scales or high resolutions while edges of large objects appear at higher scales or coarse resolutions. Therefore, performing multiscale edge detection and analysis on the detected edges in the ROIs will allow us to characterize each edge by assigning a relative size corresponding to the object that created the edge. This is an important feature useful in the discrimination of cracks from non-cracks due to the relative small size of a typical crack in comparison to other objects appearing on the aircraft surface.

Multiscale edge detection is a two step process where the ROI is first decomposed into different resolutions usually by successive smoothing, followed by edge detection at each resolution. We have selected wavelet based filters for the projection of the ROI to different resolutions and estimation of intensity variation in them for the use for multiscale edge determination. Wavelets are basis functions with good spatial and frequency localization that is controlled by a scaling parameter attached to the wavelets. Hence, they are a natural choice for multiresolution analysis due to the ease of defining the resolution of interest through the use of the scaling parameter of the wavelet.

We have chosen the cubic spline and its derivative the quadratic spline, described by Mallat[10], as our scaling and wavelet functions. The frequency responses of these functions are shown in figure 7. Note that scaling and wavelet functions are low pass and high pass in nature. The wavelet transform of a ROI at scale s is equal to the convolution of the ROI with a filter derived from the wavelet of scale s. Since the wavelet we chose is the derivative of a smoothing function, the wavelet transform is equivalent to first smoothing the ROI to a scale s by a smoothing filter, and then taking its first derivative. This is identical to the sequence of operations undertaken in classical edge detection. Note that the edge points of the ROI at a particular scale corresponds to the extrema of the wavelet transform of that scale. By varying the scaling parameter of the wavelet on a scale 2^i

(dyadic scale), we generate edges of the ROI at multiple scales.

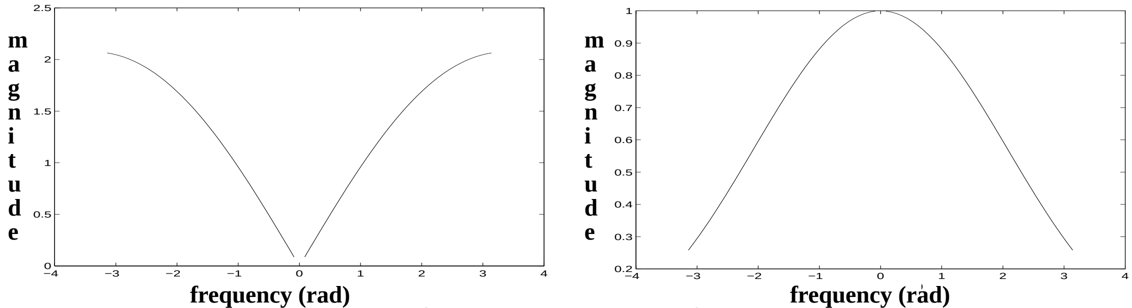

Fig 7. Frequency response of the wavelet and scaling functions

To summarize, listed below are steps taken to generate multiscale edges within each ROI in our crack detection algorithm

1. Filter each ROI with the filter bank shown in figure 8. This will result in a dyadic scale (scale = 1, 2,4) decomposition of the ROI. Filters h_0 and g_0 denote the filters corresponding to the scaling and wavelet functions at scale 1. Filters h_n and g_n* used at dyadic scale 2^n (n>0) are derived from h_0 and g_0. W_{yn} and W_{xn} denote the wavelet transform images at scale 2^n in the y (row) and x (column) directions of the ROIs while r and c denote row and column filtering.
2. Calculate the magnitude M_i and angle A_i images for each scale of the wavelet transform images W_x and W_y using
3. $$M_i = \sqrt{W_{xi}^2 + W_{yi}^2} \quad \text{for i=0,1,2}$$
4. $$A_i = \operatorname{atan}(W_{yi}/W_{xi}) \quad \text{for i=0,1,2}$$
5. Threshold each magnitude image $M_{i=0,i,2}$ using a dynamic threshold calculated using its histogram. Pixels above the threshold are as marked edge points.
6. Link edge points based on 8-neighbors if their corresponding angles differ less than a maximum angle. This produces edges that are smoothly varying in direction which are characteristic of natural edges such as cracks.

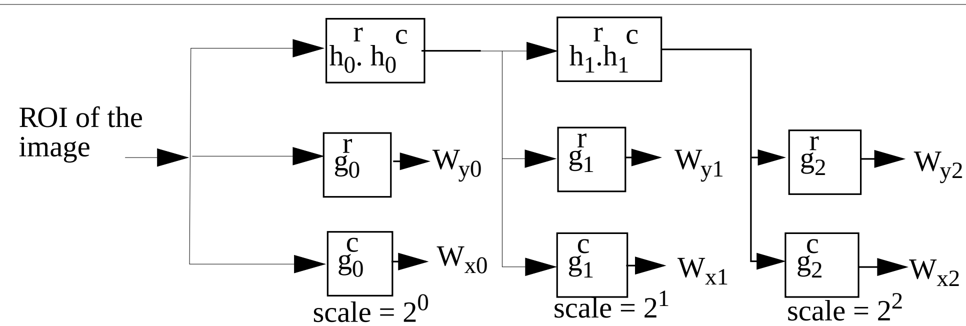

Fig 8. Multiresolution decomposition of the live image

The above process will generate a list of edges in the ROIs at each scale. Figure 9 displays these edges of the ROIs in Fig 6b.

5.3. Coarse-to-fine edge linking

Multiscale edge detection described in the last section generates edges at a set of scales, for each ROI in the image. It can be observed from figure 9 that edges of the same object is present in more than one scale. For example, parts of the rivet head edges appear in all three scales while the simulated cracks shown in figure 6a appear only at the first two scales.The next step of this process is to assign to each edge, a feature value that will provide information about the size of the object that produced the edge. This size information is useful in the discrimination of edges of cracks from edges of non cracks appearing in each ROI.

* h_n (g_n) is formed by including 2^n-1 zeros between the coefficients of h_0(g_0). h_0 = {0.125, 0.375, 0.375, 0.125}, g_0 = {-2, 2}

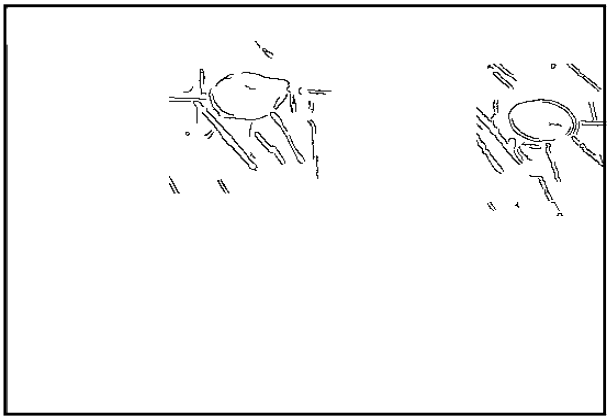

Fig 9a. Edge image at scale 1

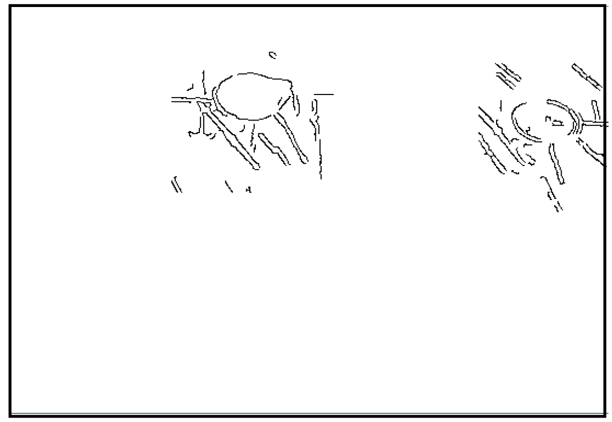

Fig 9b. Edge image at scale 2

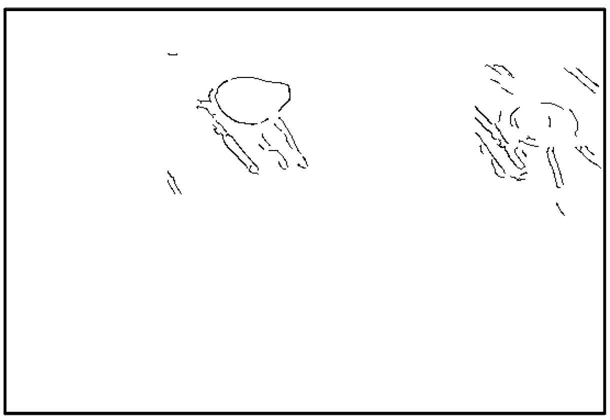

Fig 9c. Edge image at scale 4

Fig 9. Edge images extracted at multiple scales

We model an edge belonging to an object appearing at multiple scales, as the propagation of that edge along scale (or resolution) space. We define propagation depth as the number of scales in which the edge appears. We then assign a propagation depth value to each edge at scale 1. The propagation depth captures the size information of the object revealed by multiscale edges. For example, edges of objects that are small will have a lower propagation depth than edges of objects that are large. This explains why edges corresponding to the simulated cracks which are fine black fiber of approximately 10 μm , appear only in two scales whereas edges of the rivets and scratches appear in all three scales in figure 9.

We use a coarse-to-fine edge linking process to find the propagation depth of all edges appearing at scale 1. The coarse-to-fine edge linking process attempts to trace an edge from a coarse resolution (high scale) to a fine resolution (low scale). We define active pixels as those pixels that belong to the edge of reference. Given below are the steps of the coarse-to-fine edge linking process

1. Assign to each edge, in each scale (scale = 1, 2,4), a feature vector with the following components.
   a. Centroid of the active pixels
   b. Average wavelet magnitude of the active pixels
   c. Number of active pixels that constitute the edge
2. For each edge **E** in scale 4, define a window centered on its centroid in scale 2. Find all unlinked edges {e} in scale 2 that are within the window. Find the edge e_i of {e} that produces the minimum weighted square difference between itself and **E**. Link **E** to e_i.
3. Do 2). for each edge in scale 2 with edges in scale 1
4. Now, for each edge in scale 1, count the number of links (how deep its connected). This will be the propagation depth of that edge.

Figure 9 illustrates the coarse-to-fine edge linking process. Note that edges A and B have propagation depths of 1 and 2

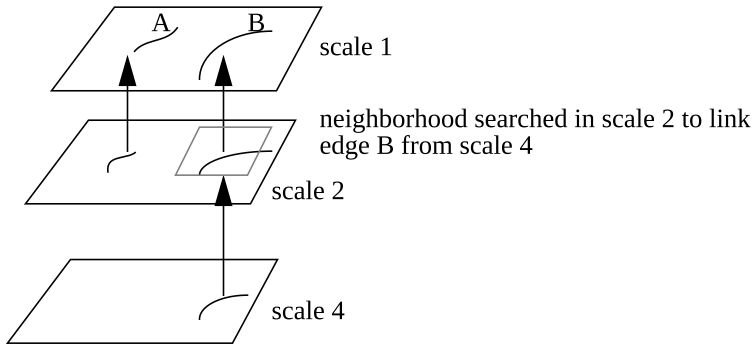

Fig 9. Coarse-to-fine edge linking of edges A and B

respectively.

5.4. Feature vector calculation

We are now in a position to assign a feature vector for each edge in scale 1. The feature vector assigned to the edge will characterize its properties so that edges of cracks can be discriminated from edges of non-cracks based on classification of the feature vectors. We have selected the following attributes of an edge to be included in a feature vector.

1. Average wavelet magnitude of active pixels
2. Propagation depth number
3. Average wavelet magnitudes of any linked edges in scale 2 and scale 4
4. Signs of sum (W_x) and sum (W_y) where W_x, W_y are the wavelet coefficients in the x and y directions of a active pixel at scale 1
5. Number of active pixels

The wavelet magnitudes at each scale of an edge that propagates down multiple scales was included because it has been shown that under certain conditions, these values characterize the shape of an edge (ex. step or a ramp edge).[10]

5.5. Feature classification

The feature vectors are classified into one of two classes; cracks or non-cracks. We use a six input, 1 hidden layer with four elements and one output neural network trained under backpropagation with momentum to classify the feature vectors. We generated 14 feature vectors of simulated cracks and 30 feature vectors of non-cracks corresponding to rivet edges and scratches. A training set of 7 simulated cracks and 15 non-cracks were used to train the network. After 1000 training cycles, the network was approximately 71.5% accurate in predicting cracks and a 27% false alarm rate for the test set edges.

Figure 6a is repeated in figure 10a. The output image of the crack detection algorithm is shown in figure 10b. As before, marked in black and grey are edges classified as cracks and non-cracks by the algorithm.

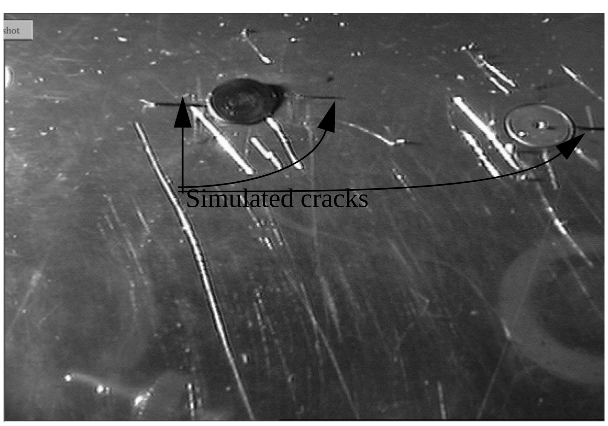

Fig 10a. Metal surface with three synthetic cracks and other crack like features

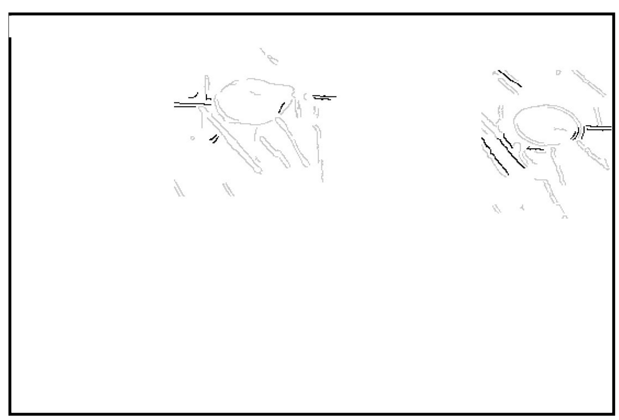

Fig 10b. Output of the crack detection algorithm.

6. SURFACE CORROSION DETECTION ALGORITHM

A comprehensive corrosion detection algorithm needs to detect both surface and subsurface corrosion. Surface corrosion is detected by texture suggestive of corrosion while subsurface corrosion is detected by bumps on the surface. Therefore, such an algorithm requires an image and a shape profile of the inspection surface to detect both types of corrosion. In this section, we describe an algorithm developed to detect surface corrosion· This algorithm will be a part of a comprehensive corrosion detection algorithm[*].

We detect surface corrosion by segmenting the image into regions of texture suggestive of corrosion and corrosion free areas. Texture can be well described by scale and orientation. This has resulted in the development of many methods based on multiresolution, multiorientation based approaches that allow scale and orientation based analysis of textures[11,12]. Our corrosion detection algorithm is based on a similar method.

Figure 11a displays an image of a corroded section of an aircraft surface. Figure 11b. shows the output of the surface corrosion detection algorithm. Bright and dark areas respectively indicate corrosion and corrosion free areas in the image. Figure 12 shows a block diagram of the algorithm

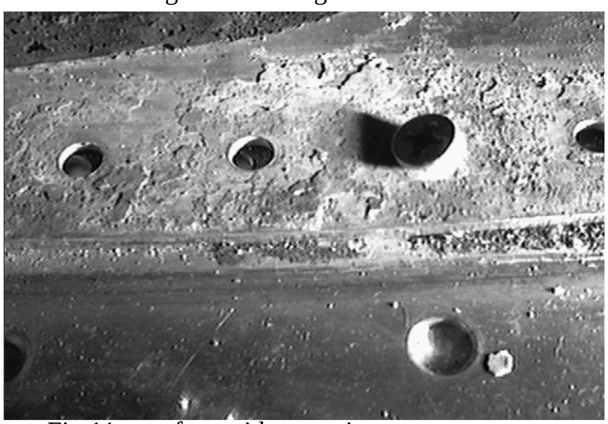

Fig 11a. surface with corrosion

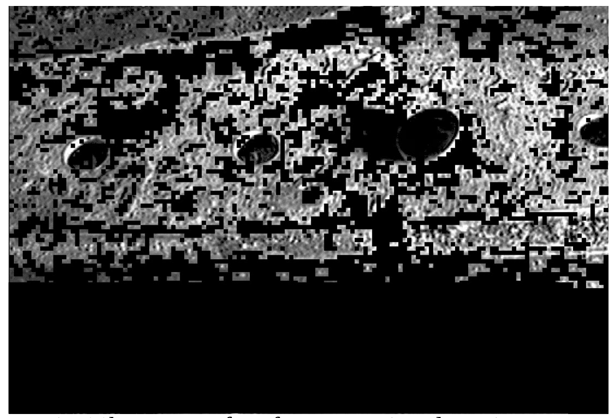

Fig 11b. output of surface corrosion detection algorithm

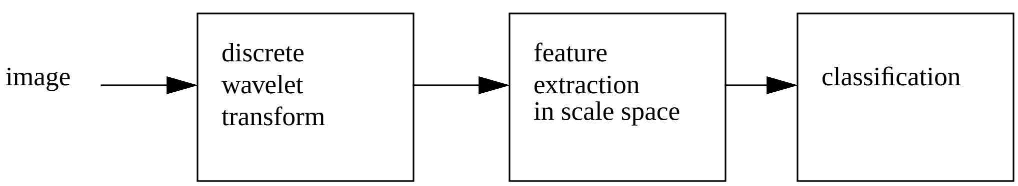

Fig 12. surface corrosion detection algorithm

6.1. Discrete wavelet transform (DWT)

We perform a multiresolution, multiorientation decomposition of the image using the DWT. The discrete wavelet transform can be thought of as filtering of the image into sub-bands by an array of scale and orientation specific filters. Since wavelets have good spatial and frequency localization, the wavelet coefficients provides a good characterization of the texture.

We have selected Daubechies D6 orthogonal wavelet[**] for the DWT. The orthorgonal wavelet prevents correlation between scales in the decomposition of the image by DWT. We perform a three-level wavelet decomposition of an image which results in 10 sub-bands.

*We are also working in parallel, on an algorithm to detect subsurface corrosion by examining a surface shape map acquired by an active vision range sensor on CIMP. Once developed, this algorithm will be integrated with the surface corrosion detection algorithm to form a comprehensive aircraft corrosion detection algorithm

** D6 filter coefficients are = {0.3327, 0.8069, 0.4599, -0.1350, -0.0854, 0.0352}

Figure 13 displays the corresponding sub-bands created by the transform.

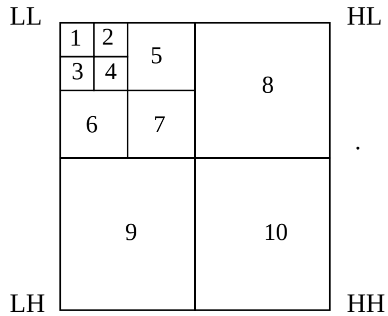

Fig 13. Three-level wavelet decomposition resulting in sub-bands 1 to 10

6.2. Feature extraction

The image is divided into non-overlapping blocks of size 8x 8 pixels. For each block, a 10 dimensional feature vector is assigned whose components represent energy within the block in each wavelet transform frames $W_{j=1:10}$. Let the energy of block B(i) in W_j be given by $E_j(i)$. Then

$$E_j(i) = \sum_{(k,l) \in B(i)} w_j(k,l)^2$$

where w_j(k,l) is the wavelet coefficient at (k,l) in the wavelet transform frame W_j. The feature vector is normalized by the total energy of the block.

6.3. Feature classification

We generated a training and test set of 2400 vectors (1200 each corrosion and corrosion free vectors) each, using a set of images of corrosion and corrosion free surfaces. A clustering algorithm was applied on the training sample vectors to find three prototype vectors representing clusters of the corrosion vectors and five prototype vectors representing clusters of corrosion free vectors in the training set. The algorithm uses a 1 nearest neighbor method to classify a new feature vector into corrosion or corrosion free classes based on its distance to the prototype vectors. The algorithm was able to detect 95% of the corrosion vectors of the test set.

Figure 14a displays an image of a corroded section of an aircraft surface. Figure 14b shows the output of the surface corrosion detection algorithm. Bright and dark areas indicate corrosion and corrosion free areas in the image.

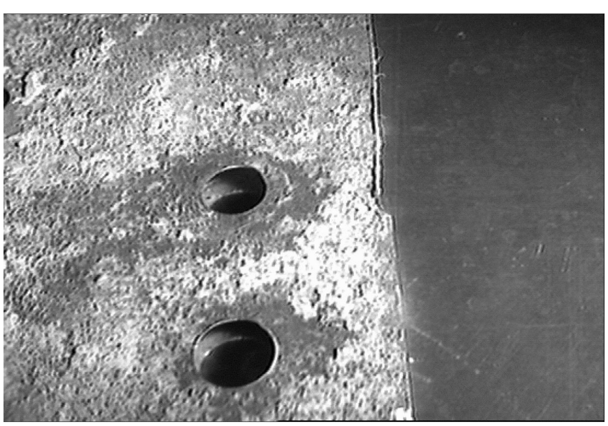

Fig 14a. surface with corrosion

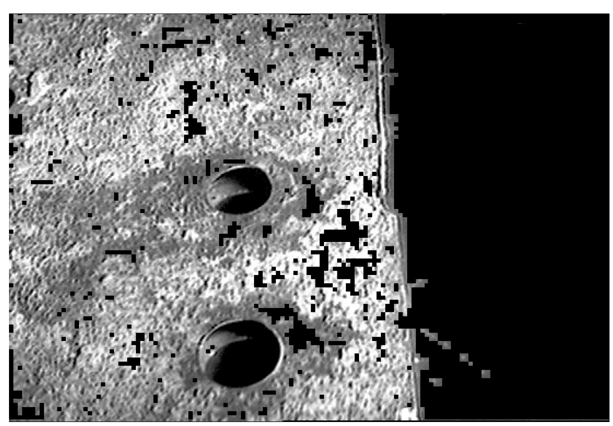

Fig 14b. output from corrosion detection algorithm

7. DISCUSSION AND FUTURE WORK

7.1. CIMP and remote visual inspection

We have successfully demonstrated CIMP's remote control and imaging capability to Northwest Airlines at their Minneapolis 747 maintenance and inspection facility and to USAir at their Pittsburgh maintenance and inspection facility*. Our demonstration proved that state-of-the-art 3D stereoscopic video technology implemented by us and operated by inspectors not specifically trained in its use, delivers imagery of sufficiently high visual quality that aircraft inspectors and NDI supervisors were willing to accept it (and sometimes prefer it) as an alternative to direct visual inspection.

7.2. Surface crack detection algorithm

Based on our experience with the algorithm and insights gained through limited testing, we are convinced that the multiscale edge analysis framework on which the algorithm is based, is the appropriate framework for extraction and analysis for aircraft surface cracks. We are encouraged by its performance in detecting simulated cracks though it was trained and tested using only a small sample of simulated cracks. We have the following plans for further development of this algorithm.

1. Inclusion of new features in the feature vectors that describe edges - For example, pairing each edge with its complement edge (rising and falling edges form a pair). This is a promising feature since the left to right (top to bottom for horizontal edges) ordering of the signs of the paired edges provide information regarding intensity change that created the edge (intensity rise or intensity fall). Intensity rises typically due to scratches and intensity falls due to cracks.
2. Linking neighboring edges with similar feature vectors to each other since longer edges better define the object that produced these edges.
3. Enrichment of the data to the detection algorithm by providing multiple images of the same surface obtained under different lighting conditions. Cracks that are invisible in one image may become visible in another due to the change in lighting. This can be achieved by controlling the dynamic lighting array of CIMP from the inspection console remotely.
4. Obtaining a larger library of natural surface cracks for algorithm training and testing

7.3. Surface corrosion detection algorithm

The current surface corrosion detection algorithm is successful in detecting surface corrosion as indicated by the performance on the test images. We plan on training and testing the algorithm with a wider array of corrosion test samples. We are also currently developing a subsurface corrosion detection algorithm to detect surface bumps from a surface profile map of the inspection surface obtained via an active vision sensor developed in our laboratory. We will in the future form a comprehensive aircraft corrosion detection algorithm by integrating these two separate algorithms.

8. CONCLUSION

Our research efforts are directed at testing the hypothesized feasibility and advantages of remote visual inspection. To test this premise, we have built CIMP, a prototype mobile robot that carries a remote imaging system and an inspection console that allows the inspector to view monoscopic or stereoscopic imagery of the remote inspection surface. In addition, the inspection console provides the inspector with a library of image enhancements and understanding algorithms that can be used to process, enhance and understand the remote imagery to aid the detection of surface defects. Through field testing, we have demonstrated successfully that our remote imaging system delivers imagery of sufficient high visual quality that aircraft inspectors are willing to accept it as an alternative to direct visual inspection. In this paper, we have described two image understanding algorithms for surface crack and corrosion detection and have reported preliminary test results which are promising. We believe that further development of these algorithms based on their adaptation to real world environments through extensive testing will significantly increase their probability of flaw detection and make them successful and productive tools for remote visual inspection.

ACKNOWLEDGMENTS

ANDI was a joint project of the Carnegie Mellon Research Institute(CMRI) and the Robotics Institute(RI), with W.M Kaufman and M.W. Siegel, co-PI's, funded by the FAA Aging Aircraft Research Program. CIMP was a project of the Robotics Institute and Aircraft Diagnostic Corporation, M.W. Siegel PI, funded partially by the Ben Franklin Technology Center of Western Pennsylvania (grant RR 10032) with in-kind assistance from USAir and valuable coorporation from Northwest Airlines. The authors wish to thank Alan Guisewite for his technical support and encouragement.

* The IIW and the library of image enhancement and understanding algorithms were not part of these demonstrations since they are still under development.

BIBLIOGRAPHY

1. C. Seher, “The national aging aircraft nondestructive inspection research and development plan”, Proceedings of the International Workshop on Inspection and Evaluation of Aging Aircraft, May 21, 1992.

2. W. T. Shepherd, “Human factors in aircraft maintenance and inspection”, Conference on aging aircraft and structural airworthiness,” NASA conference Publication 3160, pp 301-304, 1991.

3. P. D. Gunatilake, M. W. Siegel, A.G. Jordan and G. W. Podnar, “Image Enhancement and Understanding for Remote Visual Inspection of Aircraft Surface”, Proceedings of the SPIE Conf. of Nondestructive Evaluation of Aging Aircraft, Airports and Aerospace Hardware, Vol 2945, pp 416-427, December 1996.

4. M. W. Siegel, “Automation for Nondestructive Inspection of Aircraft “, Conference on Intelligent Robots in Field, Factory, Service and Space (CIRFFSS’94), Paul J Weitz (NASA/JSC), ed., AIAA, AIAA/NASA, pp. 367 - 377, Houston TX, March 1994.

5. M. W. Siegel, W. M. Kaufman and C. J. Alberts, "Mobile Robots for Difficult Measurements in Difficult Environments: Application to Aging Aircraft Inspection", Robotics and Autonomous Systems, Vol. 11, pp 187 - 194, July 1993.

6. S. Grinberg, G. W. Podnar, and M. W. Siegel, “Geometry of Binocular Imaging”, Proceedings of the SPIE/IST Conference (San Jose), ed., SPIE/IST, SPIE/IST, February 1994.

7. V. S. Grinberg, G. W. Podnar, and M. W. Siegel, “Geometry of Binocular Imaging ll: The Augmented Eye”, Proceedings of the SPIE/IST Conference (San Jose), ed., SPIE/IST, SPIE/IST, February 1995.

8. P Tong, K Arin, D.Y. Jeong, R. Grief, J.C Brewer and S.N. Bobo, “Current DOT research on the effect of multiple site damage on structural integrity”,NASA conference Publication 3160, pp 111-157, 1991.

9. S.N. Bobo,” Visual Nondestructive Inspection Advisory Circular”, 1993.

10. S. Mallat and S. Zhong, “Characterization of signals from multiscale edges”, IEEE Trans Patt. Anal. and Mach. Intell., vol 12, pp 629-639, July 1990.

11. A.C. Bovik and M. Clark, “Multichannel texture analysis using localized spatial filters”, IEEE Trans. Signal Processing, vol. 39, pp 2025-2043, Sept 1991.

12. M. Unser and M. Eden, “Multiresolution feature extraction and selection for texture segmentation”, IEEE Trans. on Patt. Anal. and Mach. Intell., vol. 11, No. 7, pp: 717-728, 1989.