

## Remote Enhanced Visual Inspection of Aircraft by a Mobile Robot

Mel Siegel and Priyan Gunatilake

The Robotics Institute / School of Computer Science / Carnegie Mellon University  
Pittsburgh PA 15213 USA

phone: +412 268 8802, fax: +412 268 5569, email: mws+@cmu.edu

*Aircraft skin inspection methodology, and the technical and economic advantages that can be expected from doing it with the aid of robotics, automation, and computer vision technologies, are reviewed. Robots for deployment of inspection systems are described briefly. Computer vision methods and algorithms for detection of cracks, surface corrosion, and subsurface corrosion are then discussed in detail. Cracks and surface corrosion are both detected by algorithms that pipeline preprocessing/enhancement, multiresolution/wavelet based feature extraction, and neural net based feature vector classification; subsurface corrosion is detected by a structured laser light surface profiling technique for the pillowing that subsurface corrosion causes. Functionality is illustrated by application to current data.*

### I. Introduction

Aircraft skins are regularly inspected both for known classes of problems (primarily cracking and corrosion) in specific locations (known via prior experience and theoretical stress modeling), and for random damage, as from lightning strikes, bumps by ground support vehicles, etc. Approximately 90% of this inspection is visual; the remaining 10% is instrumented, primarily using eddy current sensors, with ultrasonic and other instruments employed occasionally.

Robotics and automation technology has potentially large contributions to make in this field: the advantages of "getting the man off the airplane", the tireless concentration of a robot, the guarantee of proper and consistent instrument deployment protocol, the inherent navigational and record-keeping capabilities of a mobile robot under computer control, etc., are all clear. Furthermore, in an environment where the records that are currently preserved usually indicate only pass/fail, there are enormous opportunities for predictive maintenance that are enabled by bringing robotics, automation, data storage, and statistics into the picture.<sup>1,2</sup>

We have built two mobile robots for this environment: ANDI, the Automated NonDestructive Inspector<sup>3</sup>, and CIMP, the Crown Inspection Mobile Platform. ANDI was designed to deploy eddy current sensors; however inasmuch as ANDI employed suction cups to give it access to a large fraction of the aircraft skin surface, the lion's share of this project's effort went into mobility-related issues. To minimize the distraction of building a machine with a comprehensive mobility capability, CIMP's motion was restricted to the aircraft crown; this allowed us to focus successfully on measurement issues. Because 90% of aircraft inspection in the commercial sector is visual, the economic viability of a robotic solution is dictated by its ability to enable remote inspection. An additional edge may be provided by enhanced visual inspection capabilities, e.g., image processing to aid the inspector in finding anomalies, and automated image understanding to bring anomalies to the inspector's attention. We will begin this paper by summarizing the above background and context; these topics were presented in detail at ETVSIM'97<sup>1</sup>. We will then present the results of our field trials of CIMP, providing aircraft inspectors with an interface to teleoperate it and to observe remote stereoscopic video with remote control of camera viewpoint and skin illumination angle and direction.

In the main body of this paper we will show current laboratory progress toward image enhancement and automated image understanding, including algorithms that automatically mark cracks around rivet heads, patches of corroded skin surface, and skin surface deformations indicative of underlying subsurface corrosion.

## II. Background: skin defect detection and classification

To our surprise and delight, aircraft inspectors have been spontaneous and enthusiastic advocates for using computer image enhancement and automated image understanding for flaw detection; they are, however, skeptical about the likelihood that we will succeed at the latter.

The goal of an image understanding algorithm for aircraft inspection is to recognize and classify certain surface flaws that might appear in the live imagery. The recognition capability of an algorithm is achieved by correlating features of the imagery with prior or learned knowledge of the surface flaw types. However, developing a successful image understanding algorithm remains a non-trivial challenge, due primarily to the difficulty of generalizing and encoding in an algorithm the notions that humans use to discriminate normal from defective, the limited resolution and dynamic range of practical imaging systems, and the confounding effects of environment factors such as illumination.

Given these limitations, an attractive 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 as a coarse prefilter for the inspector who does the fine screening. Another scenario is the interactive use of these algorithms by an inspector to obtain a second opinion about a particular suspicious flaw. This possibility is most attractive when the real-time inspector is relatively inexperienced, in general or with respect to a specific problem, compared to the inspectors whose expertise has been incorporated (explicitly or implicitly) in the algorithm; in this scenario the computer fulfills a training role in addition to its direct inspection role.

In the spirit of these anticipated operating scenarios, we have developed prototype algorithms that detect surface cracks, surface corrosion, and subsurface corrosion evidenced by surface pillowing. Our crack detection algorithm is discussed in Section III and our surface and subsurface corrosion detection algorithms are discussed in Section IV.

## III. Crack detection algorithm

The crack detection algorithm that we have developed is modeled closely on the inspectors's practice of using grazing angle directional lighting to enhance crack visibility. We emulate the directional lighting produced by an inspector's flashlight by employing a remotely controlled rotatable directional light source on CIMP. The remote inspector can 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. A flow chart for this algorithm is shown in Figure 1.

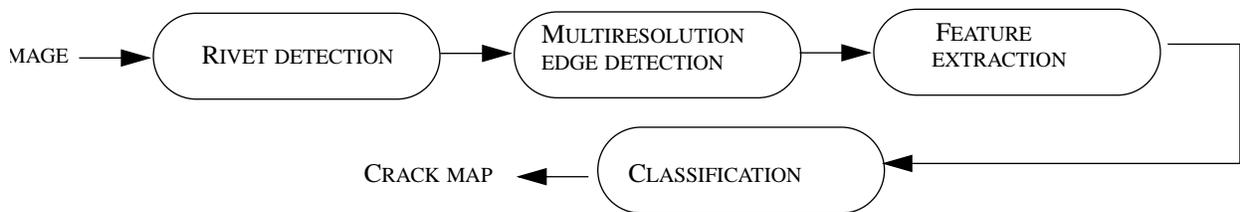
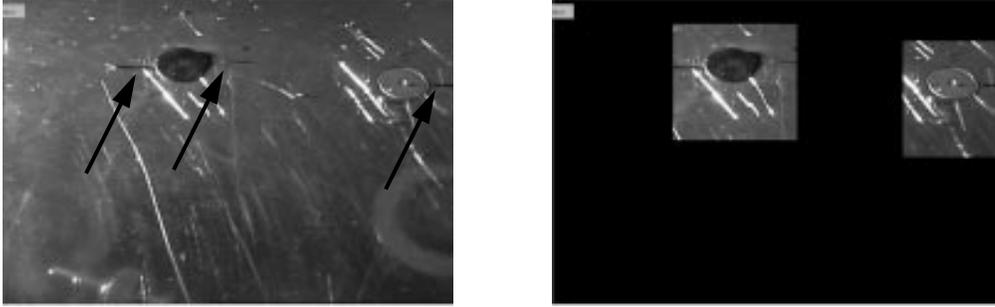


Figure 1: Flow chart of the surface crack detection algorithm

### III.1. RIVET DETECTION MODULE

The first module of our crack detection pipeline finds rivets in the image presented to it. Since cracks appear in the neighborhood of rivets, finding rivets in the first stage permits subsequent stages to focus on the areas ("regions of interest", ROIs) that are most likely to contain cracks. Rivets are identified by detecting the circular arc edges made by their heads; a region of interest is then defined as a square centered on a rivet. Figure 2 shows a section of an aircraft surface with three simulated cracks appearing as dark lines emanating horizontally from the two rivets, and the two ROIs found by the algorithm.

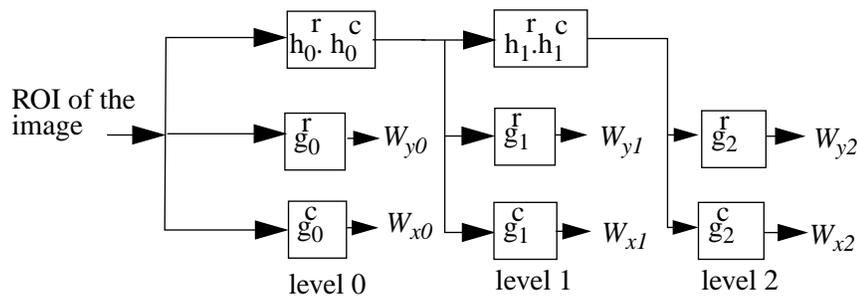


**Figure 2: Rivet “region of interest” (ROI) finding algorithm. (left) Aircraft skin showing two rivets. Arrows point to three short black horizontal lines (hairs from a fine brush) that simulate narrow cracks. (right) ROIs found by the algorithm. Notice reasonable behavior even for a rivet near the edge of the frame.**

### III.2. MULTIREOLUTION EDGE DETECTION MODULE

A crack is typically very small compared to most visible features of the aircraft surface; this motivates us to develop, for the second stage, a multiresolution edge detection algorithm that generates edges at different resolutions. For example, cracks and other very small edges appear only at the high resolutions while relatively large edges belonging to rivets, scratches, dirt marks, lap joints, metal repair plates etc. continue to appear at much coarser resolutions. In addition, Mallat<sup>4</sup> has shown that the local shape of an edge can be characterized within the multiresolution framework by analyzing the evolution of local maxima generated by wavelet- based filters across resolutions. This is important for crack characterization and detection, since cracks are typically of the step edge type. This has motivated us to use wavelet filters in our multiresolution edge detection algorithm. In summary, we have chosen a wavelet filter based multiresolution edge detection framework because it allows us to estimate the size of the edge and also to describe the type of the edge. Both are important features in discriminating cracks from non-cracks.

Multiresolution edge detection is a four-step process wherein each ROI is first decomposed into different resolutions by successive smoothing, followed by edge detection at each resolution. Listed below are the steps taken to generate the multiresolution edges within each ROI in the crack detection algorithm.



**Figure 3: Multiresolution Edge detection of each ROI at three levels**

1. Filter each ROI with the filter bank shown in Figure 3. This will result in a 3-level (resolution) decomposition of the ROI. Filters  $h_0$  and  $g_0$  denote the filters corresponding to low-pass and high-pass wavelet filters at level 0, the highest resolution. Filters  $h_n$  and  $g_n$ , used at coarser resolutions, are derived from  $h_0$  and  $g_0$ .  $h_n$  and  $g_n$  are formed by including  $(2n-1)$  zeros between the coefficients of  $h_0$  and  $g_0$ , where  $h_0 = \{0.125, 0.375, 0.375, 0.125\}$ , and  $g_0 = \{-2, 2\}$ .  $W_{yn}$  and  $W_{xn}$  denote the wavelet transform images at level  $n$  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

$$M_i = \sqrt{W_{xi}^2 + W_{yi}^2} \quad \text{for } i=0,1,2 \quad \text{and} \quad A_i = \text{atan}(W_{yi}/W_{xi}) \quad \text{for } i=0,1,2$$

3. Threshold each magnitude image  $M_i$  for  $i=0,1,2$  using a dynamic threshold calculated using its histogram. Pixels above the threshold are marked as edge points.
4. 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.

Edges of the same object or feature are usually present in more than one scale. We use a coarse-to-fine edge linking algorithm to attach a propagation depth to each edge. Note that small edges such as cracks have a smaller propagation depth than larger edges. The propagation depth and several additional features of the edge are taken to comprise a feature vector that can be used to classify the edge to be a crack or non-crack.

### III.3. FEATURE EXTRACTION MODULE

We are now in a position to assign a feature vector to each edge at level 0 (the highest resolution). We have selected the following attributes of an edge found by the multiscale edge detection process:

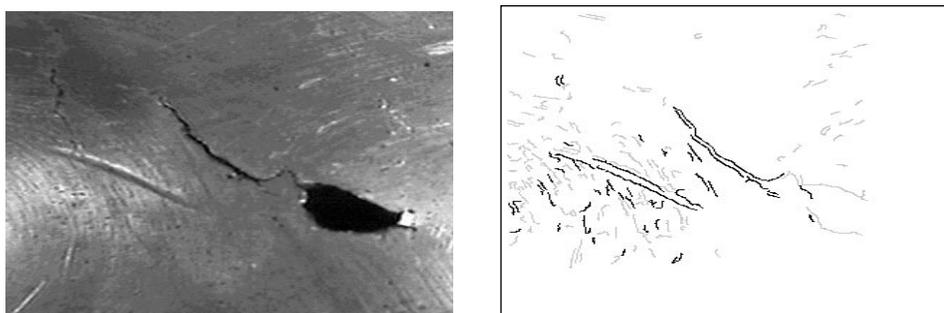
1. Average wavelet magnitude of active pixels (pixels belonging to an edge) at level 0.
2. Average wavelet magnitudes of any linked edges in levels 1 and 2.
3. Signs of  $\text{sum}(W_x)$  and  $\text{sum}(W_y)$ , where  $W_x$ ,  $W_y$  are the wavelet coefficients in the x and y directions.
4. Propagation depth number.
5. Directions of a active pixel at scale 1.
6. Number of active pixels.

### III.4. CLASSIFICATION MODULE

The feature vectors are classified into one of two classes: cracks or non-cracks. To classify the vectors we use a neural network with six inputs, one hidden layer with four elements, and one output, trained under back propagation with momentum. We generated 14 feature vectors of simulated cracks (black fibers of approximately 10  $\mu\text{m}$  diameter) and 30 feature vectors of non-cracks corresponding to rivet edges and scratches. A training set of 7 simulated cracks and 15 non-cracks was used to train the network. After 1000 training cycles, the network was approximately 71.5% accurate in predicting cracks and had a 27% false alarm rate for the test set edges. Application to a badly damaged rivet hole is illustrated in Figure 4.

## IV. Corrosion detection algorithm

A comprehensive corrosion detection algorithm needs to detect both surface and subsurface corrosion. Surface corrosion is detected by color and texture visually suggestive of corrosion, whereas subsurface corrosion is detected by distortion (“pillowing”) of the surface. Thus a comprehensive algorithm requires



**Figure 4: (left) Photograph of a badly damaged rivet hole with a large crack and a deep, crack-like dent. (right) Output of the crack detection algorithm, with crack-like edge features shown in dark gray and non-crack-line edge features shown in light gray. [This is a very severe example; really interesting cracks are so small, in length or width or both, that they would be invisible when printed at this resolution.]**

an image and a shape profile of the inspection surface to detect both types of corrosion. In this section, we describe the algorithm we have developed to detect surface corrosion; in the next section we describe our subsurface corrosion detection apparatus and algorithm.

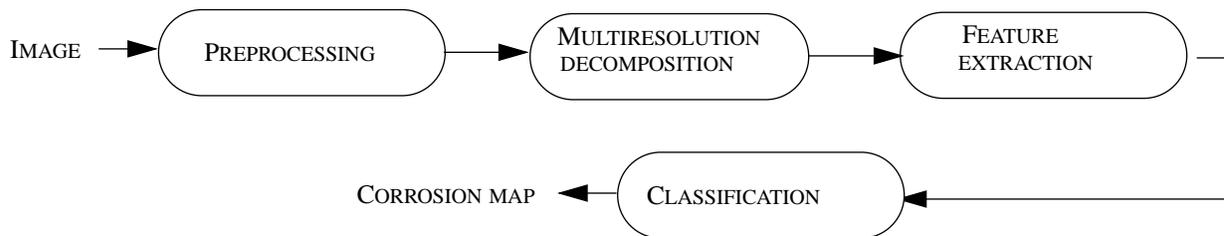
#### IV.1. SURFACE CORROSION DETECTION ALGORITHM

The surface corrosion detection algorithm we have developed segments an image into three regions; corrosion with high confidence, corrosion with low confidence, and non-corrosion. The algorithm detects surface corrosion by performing a texture discrimination operation to recognize corrosion texture.

Texture discrimination is an active research field in computer vision. Despite the presence of a number of successful texture discrimination algorithms in the literature, our experiments show that they are not suitable for our application for the following reasons:

1. Surface corrosion cannot be described by a regular (repeated) texture pattern. Many texture discrimination algorithms in the literature perform discrimination of regular textured patterns such as those featured in the Brodatz album<sup>5</sup>.
2. The algorithm must perform corrosion detection of images obtained under very different lighting conditions. The images captured by CIMP have different lighting due to inspector control of the lighting array, changes in hangar overhead lighting, shadows etc.
3. The algorithm should be able to integrate information obtained from multiple images (e.g., under different lighting) of a scene to create a single corrosion map.

We are not aware of an existing texture discrimination algorithm that satisfies these three requirements. The surface corrosion detection algorithm we developed was designed to address the above issues. While sharing some common texture discrimination operations with other algorithms, our algorithm differs from them in a few important ways. Figure 5 displays a flow chart of the surface corrosion detection algorithm.



*Figure 5: Flow chart of the surface corrosion (texture) detection algorithm*

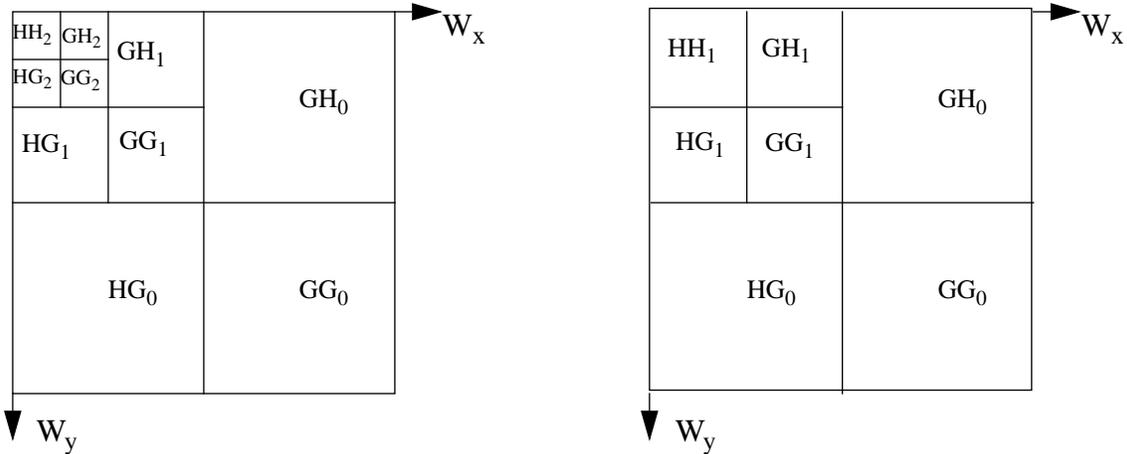
##### IV.1.1. PREPROCESSING MODULE

The stereoscopic cameras mounted on CIMP’s sensor-pod capture live composite video of the aircraft surface. The composite video is transmitted to a remote host computer which digitizes each frame into a raw 24-bit RGB image. The preprocessing module converts the RGB image into a YIQ image. R,G and B components are very highly correlated; converting to less correlated color space components, such as Y (luminance) and I, Q (chrominance) allow more compact and efficient representation of the input image.

Expressed implicitly in these color-space transformations is the importance we place on the color information. Color is an important attribute of texture. In our application of surface corrosion detection, it is a vital attribute, especially in the absence of repeated spatial patterns, in describing corrosion texture.

##### IV.1.2. MULTIRESOLUTION DECOMPOSITION MODULE

We use a multiresolution framework to analyze the input image and describe texture characteristics. We perform a three-level (resolution) discrete wavelet frame transform (DWFT) on the Y-channel. The DWFT is simply a discrete wavelet transform (DWT) without the subsampling operation. The DWFT leads to an overcomplete (redundant) representation of Y in terms of the 10 subbands shown in Figure 6 (left). Note that all subbands are therefore of the same size as the input image. Each subband corresponds to a particular scale and orientation. The redundancy has the advantage that the representation of Y in terms of



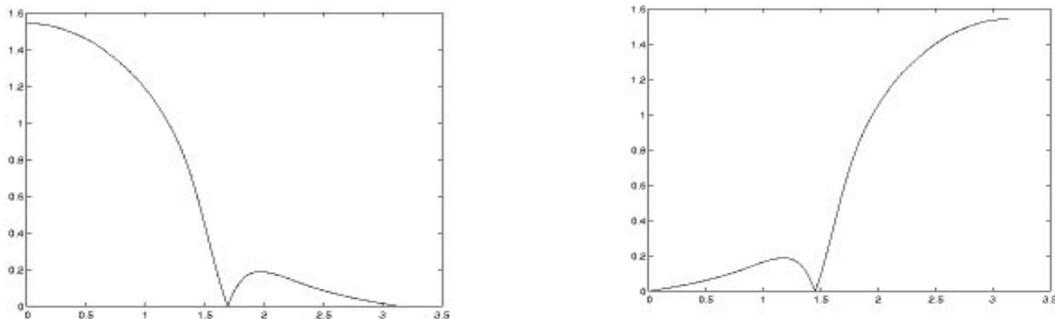
**Figure 6: Subband decomposition of images based on texture scale and orientation. *H* and *G* denote low and high pass regions in the *x* and *y* directions. The subscript denotes the level or resolution. Smaller squares correspond to lower spatial frequencies; the square adjacent to the origin is the DC component. Different squares of the same size correspond to different orientations at the same scale. (left) *Y*-channel contains the luminance information; 3-level DWFT gives 10 subbands. (right) *I*- and *Q*-channels contain the chrominance information; 2-level DWT gives 7 subbands**

the subbands is translationally invariant. This is a very desirable property for texture recognition. The *I*- and *Q*-channels are decomposed into two resolutions using the standard DWT. Note that the DWT includes a vertical and horizontal subsampling operation by 2 at each level. The resulting seven subbands created from the chrominance channels are shown in Figure 6 (right).

The wavelet filter we use to perform the DWFT and the DWT is the Battle-Lemaire (BL) 32-tap FIR filter<sup>6</sup>. The frequency responses of the base low-pass and high-pass BL filters are shown in Figure 7. The subsequent filters for the higher levels of the DWFT are derived from the base low-pass and high-pass FIR filters by adding  $(2^i - 1)$  zeros between each pair of taps at level *i*. Hence, the length of the BL filters for levels 0, 1 and 2 are 32, 63, 125 respectively. The base low-pass and high-pass BL filters are used in all levels of the DWT for *I*- and *Q*-channel decomposition.

#### IV.1.3. FEATURE EXTRACTION MODULE

In the feature extraction module, we divide the *Y* image into 32x32 pixel non-overlapping blocks. Each block is assigned a feature vector comprised of 10 features extracted from the *Y*-channel subbands and 4 features extracted from the *I*- and *Q*- channel subbands. We chose a block as a unit region for classification because block-based feature vector elements have a high signal-to-noise (SNR) ratio due to weighted averaging of pixels in the block. In addition, since corrosion is typically spread over a region,



**Figure 7: Frequency response of the low-pass (left) and high-pass(right) Battle-Lemaire 32-tap FIR filters.**

block-based feature vectors lead to a robust description of corrosion texture, resulting in fewer false alarms and misses in the classifier relative to other units of classification, e.g., pixels. A block-based approach also leads to a reduction of computational and storage requirements.

The feature elements that we compute from the Y-, I- and Q- channel subbands represent various ratios of energies. Energy at a pixel in a subband is defined as the square of that pixel's value. Ten feature vector elements are calculated for each 32x32 pixel block of Y from its subbands:

$$Y_p^{SB}(x, y) = \frac{\sum_{i, j = -(b\text{size})/2}^{(b\text{size})/2} w(i-x, j-y) (SB_p(i-x, j-y))^2}{\sum_{i, j = -(b\text{size})/2}^{(b\text{size})/2} w(i-x, j-y) \left( (HG_p(i-x, j-y))^2 + (GG_p(i-x, j-y))^2 + (GH_p(i-x, j-y))^2 \right)}$$

(1)

Here  $Y_p^{SB}(x, y)$  is the feature vector element derived from a block with mid-point  $(x, y)$  in Y-channel subband  $SB = \{sb \mid sb \subset HG_p, GG_p, GH_p; p = 0, 1, 2\}$ ,  $w$  is a weighted gaussian mask, and the block size is denoted by  $b\text{size}$ . Equation (1) expresses the ratio between the energy of a block in subband  $SB$  and its total energy at level (resolution)  $p$ . Using  $SB = \{HG_p, GG_p, GH_p\}$  and  $p=0,1,2$  generates nine feature elements for each block. The tenth feature element derived from Y-channel subbands is:

$$Y_2^{HH}(x, y) = \frac{\sum_{p=0}^2 \sum_{i, j = -(b\text{size})/2}^{(b\text{size})/2} w(i-x, j-y) \left( (HG_p(i-x, j-y))^2 + (GG_p(i-x, j-y))^2 + (GH_p(i-x, j-y))^2 \right)}{\sum_{i, j = -(b\text{size})/2}^{(b\text{size})/2} w(i-x, j-y) (HH_2(i-x, j-y))^2}$$

(2)

This is the ratio between the ac subband energy of a block and its dc subband energy.

The I- and Q- channel subbands add four feature elements to the feature vector. Given a Y- channel of dimension  $(M \times N)$ , the I- and Q- channel subbands are of dimension  $(M/2 \times N/2)$  and  $(M/4 \times N/4)$  at levels 0 and 1 of the DWT decomposition. The blocks in I- and Q- channel subbands corresponding to the non-overlapping 32x32 pixel blocks in Y-channel are found by the following two steps:

1. Project the mid-points  $(x, y)$  of blocks in Y to corresponding mid-points of blocks in I- and Q- channel subbands at levels (resolution)  $p=0$  or 1 using:

$$(x_p^{I(Q)}, y_p^{I(Q)}) = \left( \frac{x}{2^{p+1}}, \frac{y}{2^{p+1}} \right)$$

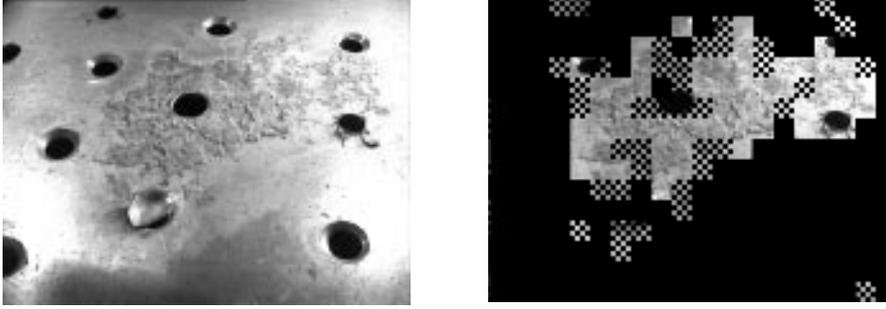
(3)

where  $(x_p^{I(Q)}, y_p^{I(Q)})$  are the mid-points of a block at level  $p$  in I- and Q- channel subbands corresponding to a block in Y-channel subband with mid-point  $(x, y)$ .

2. Construct a block centered on  $(x_p^{I(Q)}, y_p^{I(Q)})$  of size 32x32 pixels.

This process creates overlapped blocks in I- and Q- channel subbands. The overlap increases as the level increases (resolution decreases). This provides better estimation of the low-frequency signals in the chrominance subbands at coarser resolutions. For each block centered at  $(x, y)$  in Y, four chrominance features are calculated using corresponding blocks centered at  $(x_p^{I(Q)}, y_p^{I(Q)})$  in I- and Q- channel subbands.

The two chrominance features computed from the blocks centered at  $(x_p^I, y_p^I)$  in I- channel subbands are given by Equation (4) for  $p=0$  and 1.



**Figure 8: (left) corrosion around a rivet hole of an aircraft skin panel. (right) output of algorithm.**

$$I_p(x_p^I, y_p^I) = \frac{\sum_{i, j = -(bsize)/2}^{(bsize)/2} w(i-x_p^I, j-y_p^I) \left( \left( HG_p^I(i-x_p^I, j-y_p^I) \right)^2 + \left( GG_p^I(i-x_p^I, j-y_p^I) \right)^2 + \left( GH_p^I(i-x_p^I, j-y_p^I) \right)^2 \right)}{\sum_{i, j = -(bsize)/2}^{(bsize)/2} w(i-x_1^I, j-y_1^I) \cdot HH_1^I(i-x_1^I, j-y_1^I)}$$

(4)

Here  $w(i, j)$  and  $bsize$  are the gaussian weight mask and the block size respectively. Equation (4) describes the ratio between the sum of the subband energies at level  $p$  for a block centered at  $(x_p^I, y_p^I)$  in  $I$  and its dc subband energy. The two features from  $Q$  are extracted in an analogous manner.

#### IV.1.4. FEATURE CLASSIFICATION MODULE

In the last module, we use a three-layer feed-forward neural network (14 inputs, 40 hidden layer neurons, 2 outputs) to classify the above described 14-element feature vector into one of three classes: corrosion with high confidence, corrosion with low confidence, and non-corrosion. The neural network was trained using a set of 3200 hand-classified corrosion and non-corrosion feature vectors. The network was trained for 2000 epochs and at the completion of training had a probability of detection of 94% on a hand-classified test feature vector set of 800.

Although training was performed assuming two classes (corrosion and non-corrosion), we extended the number of classes to three in our implementation. This is accomplished by defining a classification “confidence” measure given by Equation (5)

$$\text{confidence} = \|\text{output}[1] - \text{output}[2]\| \quad (5)$$

where  $\text{output}[1]$  and  $\text{output}[2]$  are the outputs of the neural network. The classes are defined by rules:

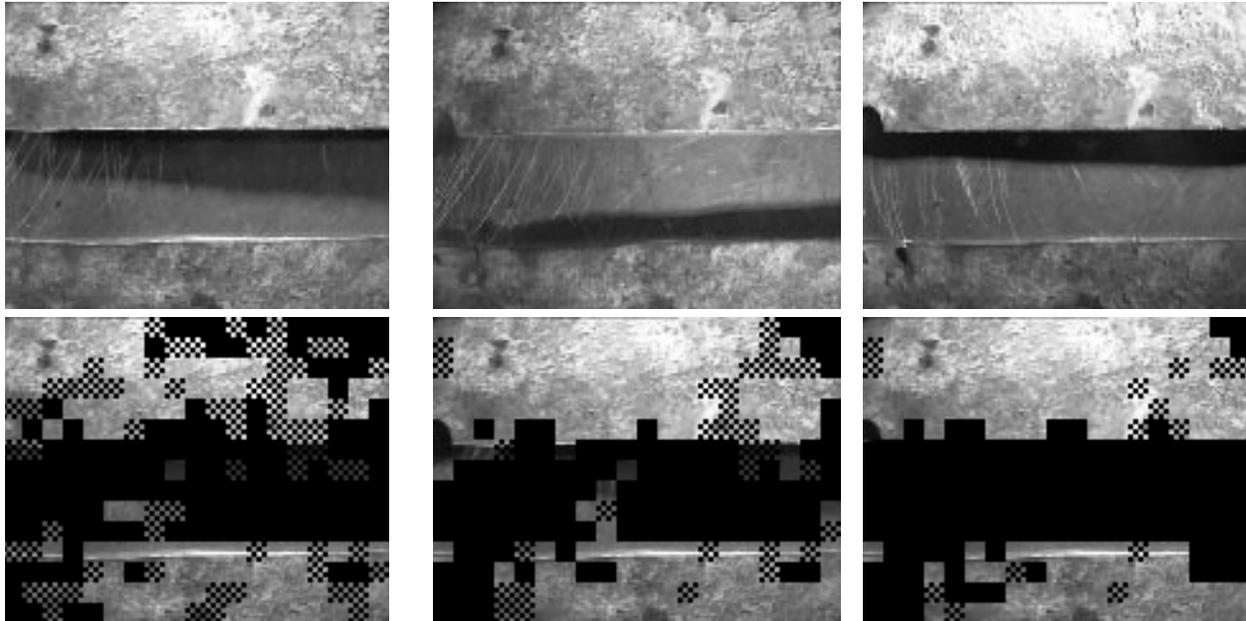
$(\text{output}[1] > \text{output}[2])$  and  $\text{Confidence} \geq T$  is interpreted as corrosion with high confidence

$(\text{output}[1] > \text{output}[2])$  and  $\text{Confidence} < T$  is interpreted as corrosion with low confidence

$\text{output}[1] \leq \text{output}[2]$  is interpreted as non-corrosion

where  $T$  is a user-specified threshold. We find  $T=0.65$  gives good results.

Figure 8 displays, on the left, an image that includes both corroded and uncorroded aircraft skin (removed from an airplane as part of a maintenance program), and on the right, the output of our surface corrosion detection algorithm. Gray scaled, checkered and black areas indicate our algorithm’s identification of the corrosion with high confidence, corrosion with low confidence and non-corrosion areas respectively in this image.



*Figure 9: (top) Three images obtained under different lighting conditions. The image includes two regions of corrosion (top and bottom) and a region of non-corrosion (mid). Notice the changes in shadow and intensity among the three images. (bottom) The successively classified images. Grey-scale, checkered and black blocks indicate corrosion with high-confidence, corrosion with low-confidence and non-corrosion regions. The bottom right image is the final classified image.*

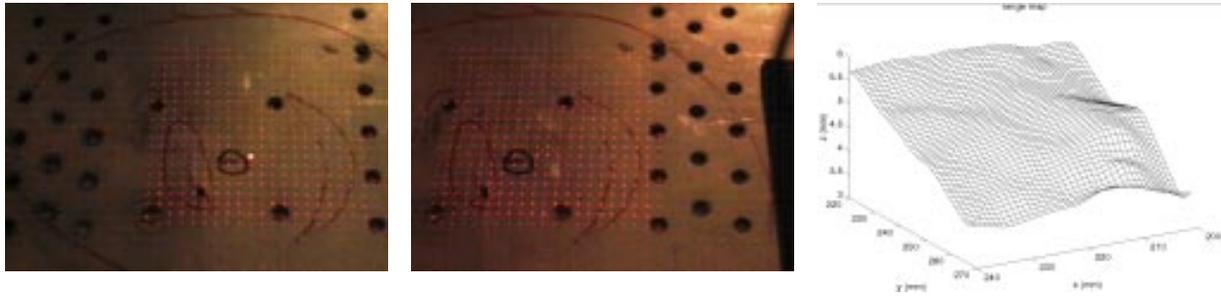
#### IV.1.5. MULTIPLE IMAGE PROCESSING AND INFORMATION FUSION

The surface corrosion detection algorithm is capable of processing multiple images, and of fusing information extracted from individual images to generate a single classified image. This capability is important in handling images obtained under varying lighting conditions. Lighting conditions may vary due to inspector specified changes of CIMP's lighting array, changes in hanger overhead lighting and shadows.

Given  $n$  images, the algorithm processes the first image and creates a table containing the classification label and the associated confidence measure calculated using Equation (5). Subsequent images are processed, and each classified block is compared with the corresponding block entry in the table. The table entry is updated if the confidence measure of a block in the current image is greater than the confidence entry in the table. Thus, the final table will contain block classifications with the highest confidence measures. Figure 9 (top) shows three images obtained under varying lighting conditions. Figure 9 (bottom) illustrates the successively refined classifications. The image on the bottom right is the final classified image.

#### IV.2. SUBSURFACE CORROSION DETECTION SYSTEM

Subsurface corrosion may be visible externally because of the surface "pillowing" it induces. Pillowing is a change in skin surface shape rather than surface texture; it is detectable as an increase in skin surface height toward the center of each rivet row-and-column-bounded rectangle over a region suffering from subsurface corrosion. Stereoscopic cameras are well suited to creating surface altitude maps, but the low density of high contrast features on aircraft sheet metal expanses makes the critical step of identifying corresponding points in left and right images very difficult. We circumvent this difficulty by illuminating the surface with a laser that projects a square grid of  $17 \times 17$  spots; by concentrating on these spots rather than on natural textural features of the surface, the correspondence problem is easily solved. Figure 10



**Figure 10: Method of detecting subsurface corrosion by elevation mapping of the visible surface. Three frames are left and right perspectives of the projected laser grid illuminating a sloping aircraft sheet metal surface, and the corresponding depth map. Elevation resolution is about 0.5 mm.**

shows the result of applying this method to a slightly wrinkled piece of aircraft belly skin that is sloping downward from back to front.

## V. Conclusion

At the previous conference in this series we discussed the aircraft skin inspection problem in detail<sup>1</sup>, and we broadly summarized our initial efforts toward building algorithms to aid (and perhaps eventually to automate) the detection of the two main classes of problems, cracks and corrosion. In this paper we have discussed in detail the results of our ongoing work. In the case of both cracks and surface corrosion, we have shown reasonable success using wavelet-based image filtering and neural net-based feature classification methods; in the case of subsurface corrosion, we have shown the value of high resolution elevation mapping for detection of the resulting surface pillowing. By “reasonable success” we mean that to date we have demonstrated systems with reasonable capabilities for filtering large volumes of visual inspection data that allow the human inspectors to concentrate on smaller volumes of data that have larger densities of flaws. This is consistent with our current goal of providing aids for the human inspectors. True automation, however, will require a very much lower false alarm rate with no loss in sensitivity to real flaws.

## Acknowledgments

We thank Northwest Airlines and US Airways for generous access to their facilities and personnel, and for providing samples. We thank Alan Guisewite for invaluable assistance in all laboratory phases of this research, and Gregg Podnar for the same, and for the lion’s share of the design and construction of the CIMP robot and its sensor package.

## References

1. Mel Siegel and Priyan Gunatilake, “Remote Inspection Technologies for Aircraft Skin Inspection”, Proceedings of the 1997 IEEE Workshop on Emergent Technologies and Virtual Systems for Instrumentation and Measurement (Niagara Falls CA), May 1997, pp 69-78.
2. Chris Seher, Mel Siegel, and William M. Kaufman, “Automation Tools for NonDestructive Inspection of Aircraft: Promise of Technology Transfer from the Civilian to the Military Sector”, Fourth Annual IEEE Dual-Use Technologies and Applications Conference, Griffis AFB NY, May 1994.
3. M. W. Siegel, W. M. Kaufman, C. J. Alberts, “Mobile Robots for Difficult Measurements in Difficult Environments: Application to Aging Aircraft Inspection”, Robotics and Autonomous Systems, Elsevier, Vol 11, 1993, pp 187-194.
4. S. Mallat and S. Zhong, “Characterization of signals from multiscale edges”, IEEE Trans. on Patt. Anal. and Mach. Intell., vol 12, pp 629 - 639, July 1990.
5. P. Brodatz, *Textures, A photographic album for artists and designers*, Dover Publications, New York, 1966.
6. G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley MA, 1996, p 254.