Chapter 4

Detection, representation and classification of anomalies in grey scale images

4.1 Context and simplifications

In this chapter the restriction is made to *grey scale* images consisting of one *large homogeneous* object (the background) and a number of small objects which are anomalous.

It is assumed that the large object covers the majority of the image and is homogeneous with respect to local intensity and simple texture. However this background object may vary globally from the presence of a light source for example.

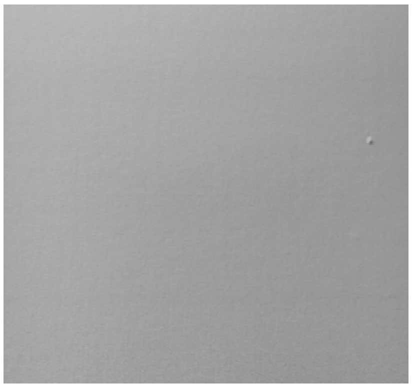
No specific assumptions are made about the number or nature of the anomalous object's other variation from the background region. There may be 0, 1 or perhaps more of these which are either localised or distributed across the image.

Although the restriction is made to grey scale images, colour images are presented. In these cases the images are treated and processed using only the intensity component as defined in Section 2.6.

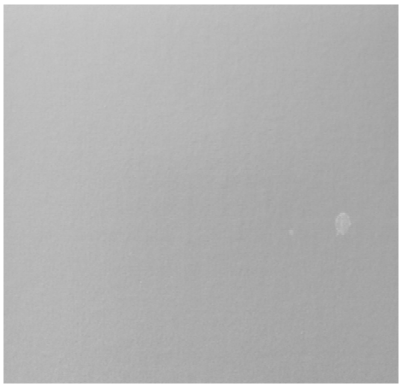
4.2 Example data and applications

Painted steel images[1] containing defects provide appropriate data and are used throughout. Figures 4.1 and 4.2 show a selection of these images in original and histogram equalised form. The *dirtspot* and *resin* defects are small and localised appearing to the far right of the respective images. The *dross* type defect is more subtle and distributed, observed as a ripple feature under the material's surface. The final two examples are more severe. The *frost* image contains a regular distributed feature which constitutes more of a global texture rather than an anomaly. Consequently this defect may be overlooked since it

[1]Images kindly supplied by BHP Research Laboratories Port Kembla.



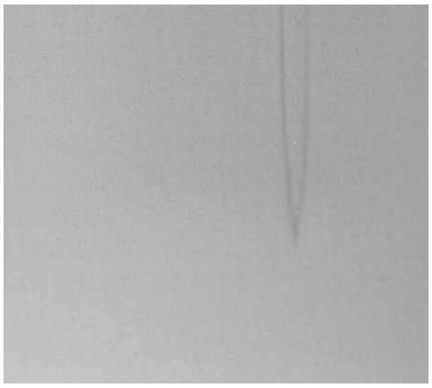
(a) *dirtspot* (2012)



(b) *resin* (2151)

(c) *dross* (2026)

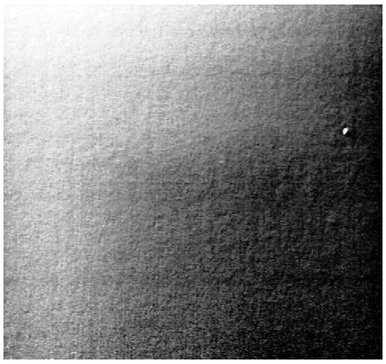
(d) two defects (2036)

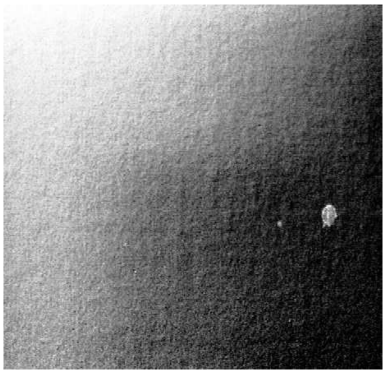


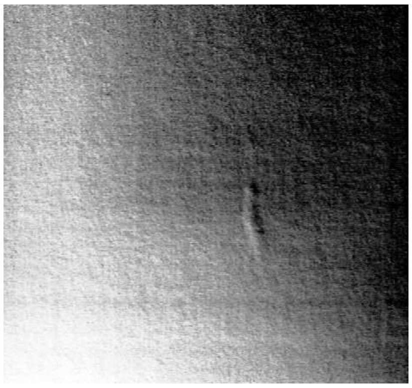
(e) *fingermark*

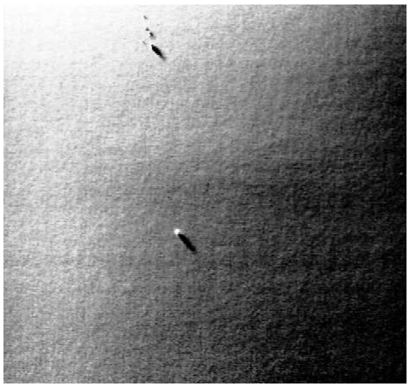
(f) *frost*

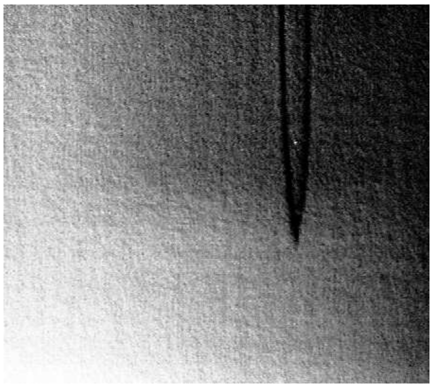
FIGURE 4.1: Painted steel image containing defects. (also see Figure 4.2)


(a) *dirtspot* (2012)


(b) *resin* (2151)


(c) *dross* (2026)


(d) two defects (2036)


(e) *fingermark*

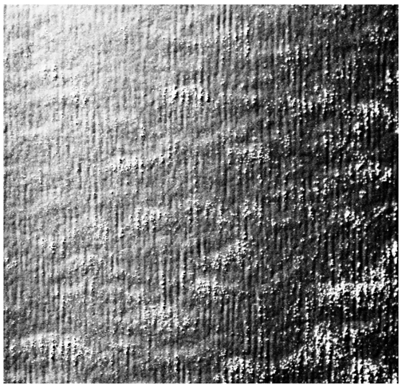
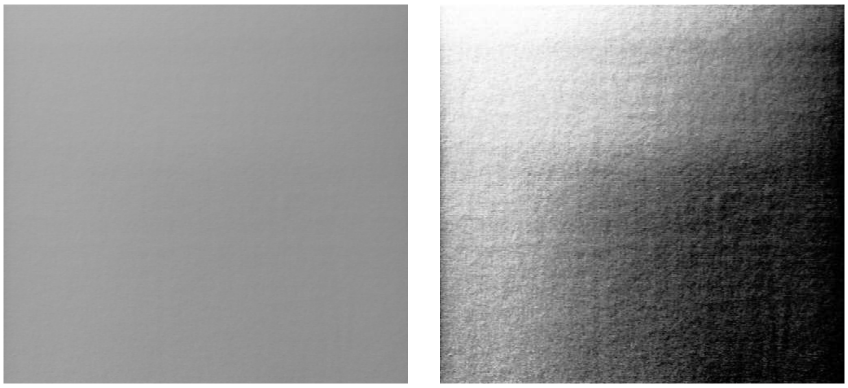

(f) *frost*

FIGURE 4.2: Images from Figure 4.1 after histogram equalisation.

violates the assumption that anomalies are small compared to the large homogeneous background object.

As is clear from the example images, the problem of locating and classifying is non-trivial and cannot be solved using simple image processing techniques. The project has since been abandoned by BHP researchers, partially due to lack of funding, and hence no additional data is available. Of the few hundred images, many are examples of infrequent severe defects while others do not contain visually observable defects (see Figure 4.3). Despite this, many images are appropriate for locating defects and there are sufficient examples of *dirtspot*, *resin* and *dross* type defects to perform classification.



(a) original (b) after histogram equalisation

FIGURE 4.3: A painted steel image supposedly containing a *dross* type defect.

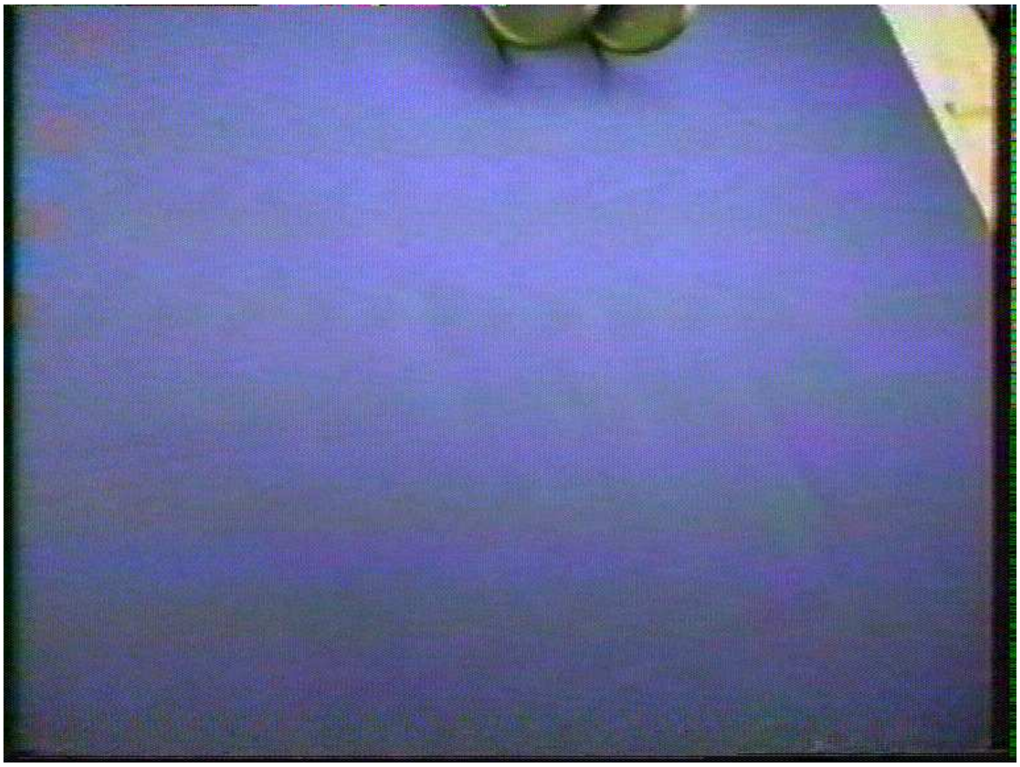
Other than painted steel, fabric images are considered. These images are from Natural Textiles Australia and have been captured from a video taken using a hand-held camera. The video shows production line fabric moving at high speeds over a conveyor belt while being visually inspected by low wage workers (Figure 4.4(a)). On detection of a defect in the material, the worker manually stops the conveyor and marks it. The digital images containing defects were captured from the moving fabric by using the frames prior to this stoppage.[2] To cater for the effects of motion blur, the odd and even interlaced frames are considered separately. Despite these considerations, the quality of most images is unacceptable for processing. An unusually good example is shown in Figure 4.5.

A few miscellaneous images are also used. These are presented alongside the results.

The practical applications for locating and classifying defects is clear as discussed in Section 1.5.1. Unfortunately, there are insufficient examples of defects in fabric images and furthermore the image quality is generally unsuitable. Since no contract has been made with either party, the practical focus is placed on deriving a generic algorithm to locate, represent and classify defects in the available painted steel images. A generic algorithm

[2] Custom written C programs using the *IRIS Video Library* on a Silicon Graphics workstation were used. These allow a set of consecutive frames to be captured without lossy compression, smoothing or post-processing incurred by the standard available video capturing tools.

(a) Site setup



(b) Raw video input image

FIGURE 4.4: The screening of production line fabric at National Textiles Australia.

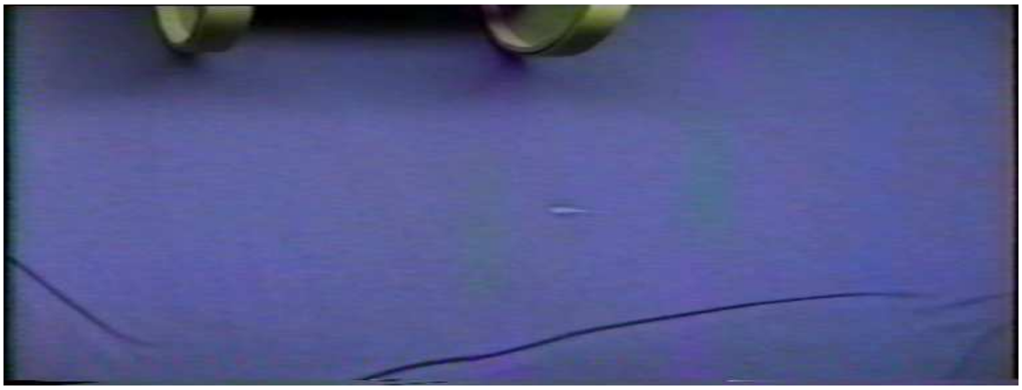


FIGURE 4.5: A good quality example of a fabric image containing a flaw.

should be applicable for industrial screening in a large number of applications. To date, no general reusable algorithms have been successful in this area as outlined in Section 2.2.

4.3 Local feature extraction

For anomaly detection in the case where the anomaly occupies a small area of the image, it is reasonable to choose local descriptions in such a way to detect texture like properties at some scale. Therefore a representation of a neighbourhood of points is sought, which makes similar neighbourhoods close and anomalous ones distant from the representation of normal data. This is achieved using a block of windows and representing this block as a point in a higher dimensional space. This provides information pertaining to autocorrelations in the image at a variety of scales.

In this method a mask, as depicted in Figure 4.6, is used. At the largest scale the mask consists of a square set of blocks of size *blocksize* $\mathbf{k}$. Each block is viewed as a window with an associated *windowsize* $\mathbf{n}$ measured in pixels. For the default case the blocks are placed adjacent to one another to form independent image regions. Hence the size of the entire mask is $\mathbf{nk}$ by $\mathbf{nk}$ pixels. The mask in Figure 4.6 is 21×21 pixels since $\mathbf{k} = 3$ and $\mathbf{n} = 7$.

This mask, which must be significantly smaller than the image, is initially placed in the top left corner of the image. Within each window a predefined *window measurement* $\mathbf{m}$ is taken. This measurement is used as the lowest level primitive and should be simple and general, yet effective. For anomaly detection the average pixel intensity is sufficient. Since there are $\mathbf{k}^2$ windows this forms a point in $\mathbb{R}^{\mathbf{k}^2}$ assuming the measurement forms a single real number. This point constitutes a single feature vector.

The mask is then shifted a distance δ_x in the $\mathbf{x}$ direction and the process is repeated to obtain another point in $\mathbb{R}^{\mathbf{k}^2}$. This process is repeated until the mask is no longer completely within the image. A similar shift (δ_y) is used to cover the image in the y direction. Having completed this process using average pixel intensity as the window

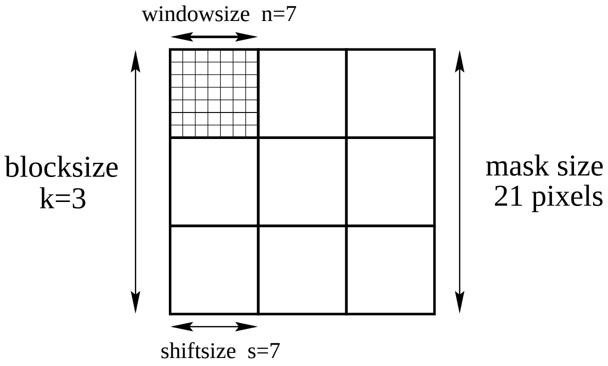


FIGURE 4.6: The structure of the mask used to obtain local image characteristics.

measurement, the original image is represented by numerous points in $\mathbb{R}^{\mathbf{k}^2}$. These contain a relationship between intensity gradients at a scale dependent on the window size used.

The value δ_x and δ_y should either be chosen as small as possible to obtain feature vectors at every possible offset, or set to the size of the entire mask to obtain independent estimates of image characteristics. Since the latter loses valuable information between mask placements, $\delta_x = \delta_y = 1$ is used. Hence the process fits the mask to every possible position within the image bounds, producing a data point for each location. This representation is however incomplete since it does not reveal the location of the flaw and hence has no way of interpreting the shape of the flaw, but simply its existence. To account for this a further two data points are added, these being the x and y offset of the mask. Hence the result gives a point in $\mathbb{R}^{k^2+2}$.

In summary, this process maps an image in $\mathbb{R}^3$ (intensity, x position, y position) into a series of points in $\mathbb{R}^{\mathbf{k}^2+2}$. For anomaly detection the window measurement $\mathbf{m}$ is simply the average pixel intensity.

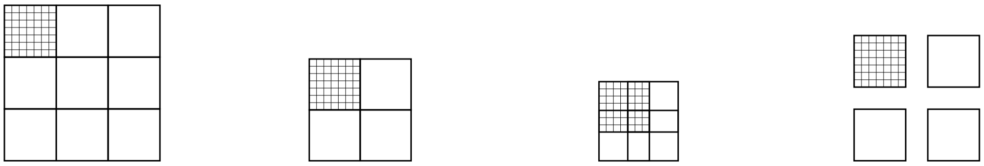
$$\mathbf{I}[\mathbf{i},\mathbf{x},\mathbf{y}] \quad \longmapsto \quad \mathbf{P}[\mathbf{m}_1,\mathbf{m}_2,\ldots,\mathbf{m}_{\mathbf{k}^2},\mathbf{x},\mathbf{y}]$$

4.3.1 Parameter choices, variations and generalisations

This feature extraction technique is not directly dependent on the image resolution at a pixel level. If an image acquisition system produces images with twice the resolution the process is easily adapted by simply doubling $\mathbf{n}$. The same cannot be said for many standard feature extraction or image processing techniques, such as the Sobel edge detector. A value between $\mathbf{n} = 4$ and $\mathbf{n} = 20$ is applicable for most images, providing a tradeoff between localisation and statistical significance or tolerance to noise.

The value of the blocksize $\mathbf{k}$ is perhaps more dependent on the application or type of image. It governs the order of the neighbourhood and the dimension of the corresponding feature space. The values $\mathbf{k} = 2$ and $\mathbf{k} = 3$ are usually suitable. Using $\mathbf{k} = 1$ obviously degenerates into a simple window measurement.

In some cases it is desirable to use a relatively large window and a 3 by 3 block neighbourhood, but still maintain a high degree of localisation. For example, using $\mathbf{n} = 20$ and $\mathbf{k} = 3$ the standard mask is 60 by 60 pixels which is perhaps undesirably large. Hence, a further parameter $\mathbf{s}$ (*shiftsize*) is used to allow overlap within windows of the same mask. Figure 4.7 shows the structure of masks with various parameter values.



(a) $\mathbf{k} = 3$, $\mathbf{n} = s = 7$ (b) $\mathbf{k} = 2$, $\mathbf{n} = s = 7$ (c) $\mathbf{k} = 2$, $\mathbf{n} = 7$, $\mathbf{s} = 4$ (d) $\mathbf{k} = 2$, $\mathbf{n} = 7$, $\mathbf{s} = 11$

FIGURE 4.7: Example masks with different parameters.

In general, the feature extraction requires one or more window measurements with the restriction that these are real numbers. It is suggested that these be simple measurements which assume little about the underlying structure of the data. While average intensity is sufficient for anomaly detection, other measurements have been implemented, namely the standard deviation of image intensity and local autocorrelations (used in [144] as defined on page 60). Possible alternatives include local frequency measurements and two dimensional moments. Histograms or probability distributions do not produce points in a Euclidean space and hence are inappropriate under the adopted framework.

Consequently, in a more general form, the process maps from $\mathbb{R}^3$, or $\mathbb{R}^5$ for colour images, into a set of points in $\mathbb{R}^{N\mathbf{k}^2+2}$ according to N real numbers corresponding to each window measurement $\mathbf{m}_i$.

$$\mathbf{I}[\mathbf{r}, \mathbf{g}, \mathbf{b}, \mathbf{x}, \mathbf{y}] \quad \longmapsto \quad \mathbf{P}[\mathbf{m}_{1,1}, \mathbf{m}_{1,2}, \ldots, \mathbf{m}_{1,N}, \mathbf{m}_{2,1}, \ldots, \mathbf{m}_{\mathbf{k}^2,N}, \mathbf{x}, \mathbf{y}]$$

4.3.2 Computational and memory considerations

The calculation of average local intensity within a window of defined size and shape can be implemented using the convolution theorem (see page 41) and fast Fourier transform. While this is more efficient than raw calculation for large windows, it is restricted to linear measurements[3], requires considerable temporary memory and cannot be further optimised. Through maintaining running sums, most windows can be processed by subtracting the pixels to the left and adding those to the right. This reduces the computation time by a factor of about $\mathbf{n}/2$, making direct processing much more efficient than applying the FFT. For example, using a 625 by 324 image and assuming the convolution mask can be designed in the frequency spectrum, an FFT, image multiplication and inverse FFT are required. These operations take approximately 8.8 seconds on a single R10000 processor.

[3]Obviously some nonlinear measurements can be achieved by multiple convolutions on processed image data. For example, standard deviation is obtained using results from convolving with the pixel intensity and squared pixel intensity.

Although independent of $\mathbf{n}$, this is far less efficient than the running sum method which takes around 0.3 second for a typical value ($\mathbf{k} = 7$) as indicated in Table 4.1.[4]

windowsize ($\mathbf{n}$)	local intensity without using running sums	local intensity using running sums	relative speed up	standard deviation with running sum techniques
4	0.467	0.187	2.5	0.440
7	1.13	0.261	4.3	0.656
10	2.09	0.339	6.2	0.874
15	4.42	0.465	9.5	1.22
20	7.23	0.587	12	1.58
30	15.8	0.816	19	2.21
50	38.8	1.24	31	3.37
100	108	2.10	51	5.64
300	68.3	3.03	23	6.99

TABLE 4.1: The time taken to calculate window measurements for a 625 by 324 image. Results are the average of performing the operation 10 times using an R10000 processor. All times are shown in seconds.

Memory usage is also a consideration. A 1024 by 1024 image produces approximately one million feature vectors (actually $[1024 - (\mathbf{ks} - 1)]^2$). Using $\mathbf{k} = 3$ and representing vector elements as *doubles* uses about 88 megabytes of memory. This representation (originally used) is impractical for large images. An alternative is to store the feature vectors symbolically by simply calculating the window measurements at each offset and storing these. Each feature vector is then obtained by collecting components from the appropriate offsets and attaching this offset to the vector. This incurs insignificant overheads and reduces memory requirements by a factor of $\mathbf{k}^2 + 2$, to 8 megabytes for the above example.

A further improvement is to use *short integers* to store sums and maintain the current windowsize for scaling. This restricts the extraction process to measurements of intensity with an assumed integer range. It is not general, and although implemented is not used or recommended unless memory usage is critical. For the above example it reduces the feature vectors to 2 megabytes of memory.

The painted steel images used for anomaly detection are about 480 by 450 pixels although vary in size. Using $\mathbf{k} = 3$ and $\mathbf{n} = 7$, feature vectors take 0.29 seconds to extract and occupies 1.7 megabytes of memory.

4.4 The feature space

Given that the assumptions made are correct, the feature space contains one dense cluster corresponding to the dominant homogeneous background and a few outliers which are considered anomalous. This is illustrated in Figure 4.8 using the feature space of the

[4]The number of feature vectors reduces as $\mathbf{n}$ increases, hence the time taken without using running sums reduces as $\mathbf{n}$ approaches the dimensions of the image.

dirtspot image (2012). Although not observable here the feature space consists of 201810 data points in $\mathbb{R}^{11}$ (since $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 7$, $\mathbf{X} = 485$ and $\mathbf{Y} = 454$). The first image shows the projection onto the plane defined by window measurements 2 and 4 (middle top and middle left). The distinction of individual outlying points indicates the relative sparseness of the feature space in regions corresponding to the anomaly. By projecting onto the variables corresponding to the masks' $(\mathbf{x}, \mathbf{y})$ offset the location of the anomaly is apparent. Figures 4.8(b) and (c) reveal the anomaly's image position as $\mathbf{x} \approx 430, \mathbf{y} \approx 280$. The significant variation in values along these these axes is caused by the light source at the top left of the image. The final projection, selected using the *xgobi's* [63] tour function, clearly indicates the dense cluster. It should be noted in all of these projections that a further nine dimensions are hidden from view.

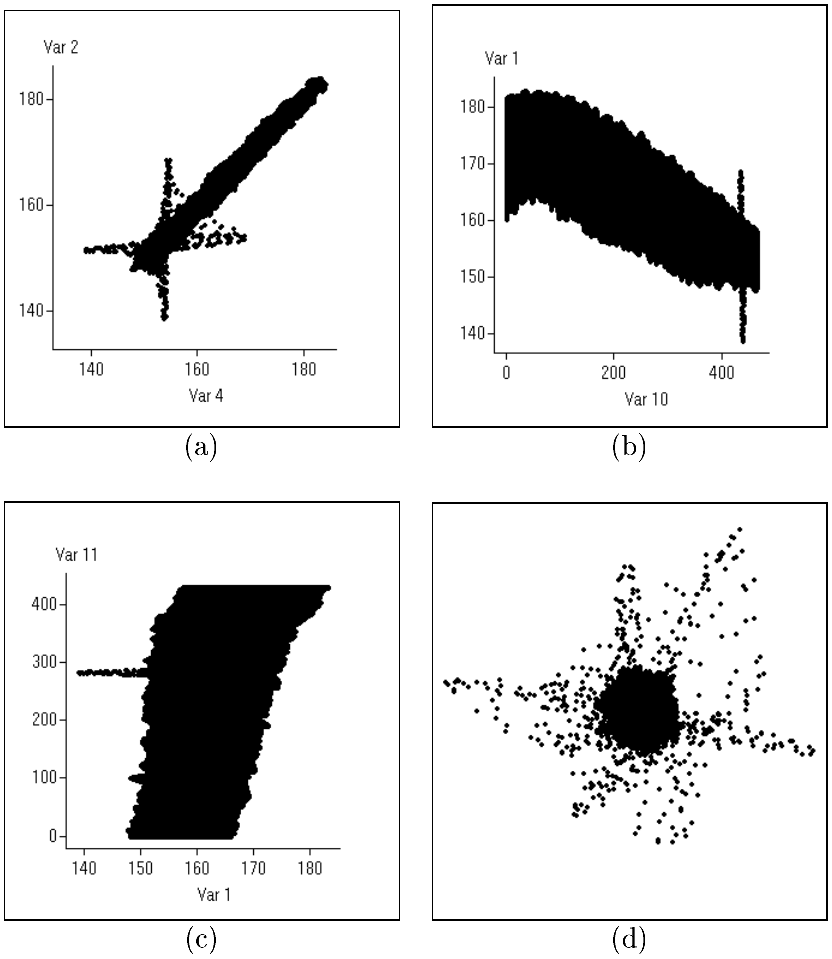
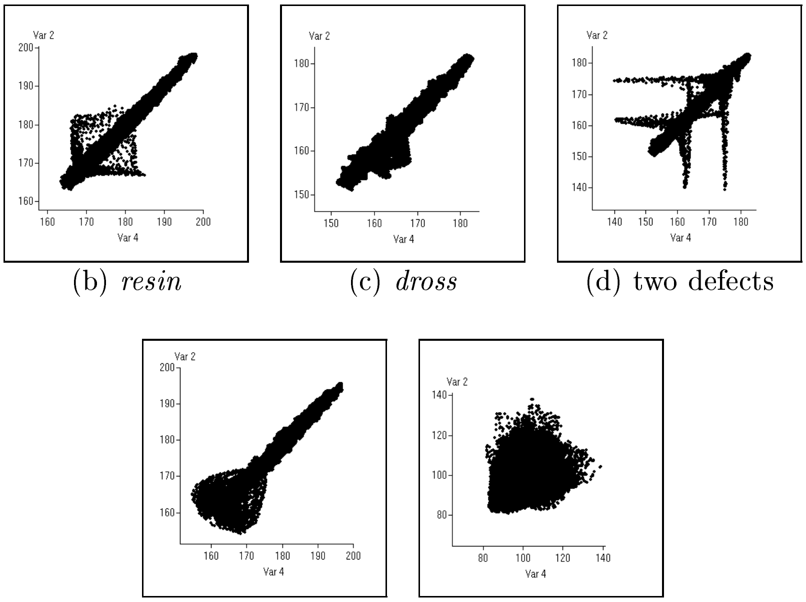


FIGURE 4.8: Projections of the feature space from an image containing a *dirtspot* defect. (Image *2012* from Figure 4.1(a))

The same simple projection (onto measurements 2 and 4) is shown for the other sample images in Figure 4.9. From these it is apparent that the majority of data points lie along the vector $[1, 1, 1, 1, 1, 1, 1, 1, 1]$ when the offset is discarded. In general form this is referred to as the *one vector* $\vec{\mathbf{1}}$ regardless of the dimension. Points along this vector correspond to the cases where measurements in each window are the same, which can be expected for this data. Although not clear from Figure 4.9(f) the feature space is also far more dense along $\vec{\mathbf{1}}$ for the *frost* defect.



(b) *resin* (c) *dross* (d) two defects

(e) *fingermark* (f) *frost*

FIGURE 4.9: The same projection as Figure 4.8(a) for feature spaces obtained from images in Figure 4.1(b)–(f).

In summary, for the majority of painted steel images, the feature space consists of a single dense cluster and a spare set of outlying points. In the few cases where this does not clearly occur, the image stretches or violates the assumptions enforced on this restricted decomposition problem (outlined in Section 4.1).

4.5 Identification of anomalies

4.5.1 Method

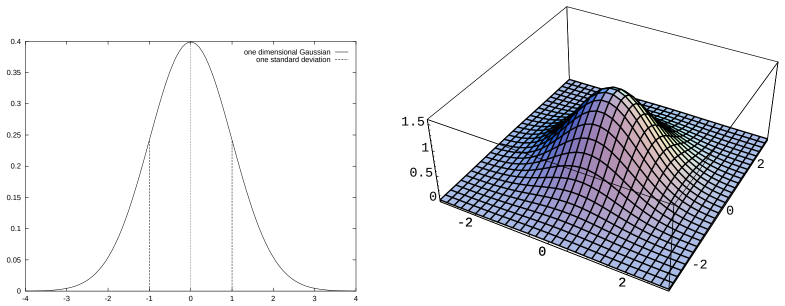
In general it is desirable to locate anomalous regions with respect to the context of the current image. Hence the location of anomalies is performed by examining a single image with no prior training or image knowledge. As outlined in the previous section, when the specified assumptions hold, the majority of points in the feature space form a single cluster. Furthermore feature vectors which are not *close* to this cluster arise from sections of the image which disagree with background homogeneity and are therefore considered anomalous. The centre of this cluster is determined by calculating the expected value μ of the points $\mathbf{P}$. Finding μ and determining those with a large distance from it should reveal image anomalies. This is a reasonable suggestion, but what is meant by *distance*? The

Euclidean distance is not necessarily appropriate in this new space, hence another metric is needed.[5]

This new metric is obtained by fitting a normal distribution to the entire feature space by calculating first and second order central moments which are most conveniently described by a mean μ and covariance matrix $\mathbf{C}$. The normalised eigenvectors of the covariance matrix provide a normalised and orthogonal set of basis vectors which give an appropriate metric. The distance from the centre with respect to this new basis, known as the *Mahalanobis distance*, is described by the following transformation which maps points from $\mathbb{R}^N$ to $\mathbb{R}$:

$$\begin{aligned} F : \mathbb{R}^N &\longrightarrow \mathbb{R} \\ \mathbf{x} &\longmapsto \sqrt{(\mathbf{x}-\mu)^T \mathbf{C}^{-1} (\mathbf{x}-\mu)} \end{aligned}$$

The Mahalanobis distance gives a measure of the number of standard deviations from the mean, thus giving a fair indication of image abnormality. It is important to note that the expected magnitude of a modeled distribution is proportional to the dimension due to the nature of the Gaussian distribution. For an ideal one dimensional Gaussian distribution about 68.27% lies within one standard deviation, i.e. between the vertical lines in Figure 4.10(a). In two dimensions this reduces to 39.35% corresponding to the volume under the surface of Figure 4.10(b) where $x^2 + y^2 \leq 1$ (a cylinder).



(a) $f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ (b) $f(x,y) = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}$

FIGURE 4.10: Standard sized one and two dimensional Gaussian distributions.

In just 6 dimensions less than 1.5% of points are within 1 standard deviation. Even given an ideal Gaussian distribution over 68 of the ≈200000 points are expected to have an Mahalanobis distance greater than 5 and hence possibly be mislabelled anomalous. This reduces quickly for 6 and 7 deviations as outlined in Table 4.2.

The feature space does not form an ideal Gaussian model. For a start the ($\mathbf{x}$,$\mathbf{y}$) offsets form a uniform plane which is well defined. Furthermore the density of points along the vector $\vec{\mathbf{1}}$ is reasonably constant corresponding to the proportion of the image with relatively dark, medium and light intensities. Consequently more non-anomalous points

[5] A metric is defined on page 32.

dimen- sion	number of standard deviations						
	1	2	3	4	5	6	7
1	0.32	0.046	2.7×10^{-3}	6.3×10^{-5}	5.7×10^{-7}	2.0×10^{-9}	2.6×10^{-12}
2	0.61	0.14	0.011	3.4×10^{-4}	3.7×10^{-6}	1.5×10^{-8}	2.3×10^{-11}
3	0.80	0.26	0.029	1.1×10^{-3}	1.5×10^{-6}	7.6×10^{-8}	-
4	0.91	0.41	0.061	3.0×10^{-3}	5.0×10^{-5}	2.9×10^{-7}	-
5	0.96	0.55	0.11	6.8×10^{-3}	1.4×10^{-4}	9.5×10^{-7}	-
6	0.99	0.68	0.17	0.014	3.4×10^{-4}	2.8×10^{-6}	-
7	0.99	0.78	0.25	-	-	-	-

TABLE 4.2: The proportion of points outside N standard deviations (i.e. with a Mahalanobis distance of greater than N) of the center of a Gaussian distribution for various dimensions.

After solving the first two dimensions manually, Mathematica was used to solve the problem using numerical integration. The data is restricted by two factors. Firstly the computation time increases exponentially with dimension and secondly the accuracy suffers when the function approaches zero (for a high number of standard deviations). The second factor can be realised by the significant variation in values returned using Mathematica's 'NIntegrate' and 'Integrate' functions with 7 standard deviations for the 3 dimensional case. Consequently extreme cases have been omitted. In general these numbers are intended for illustration purposes only.

are expected at a distance of 4 or 5 standard deviations than is suggested in Table 4.2.

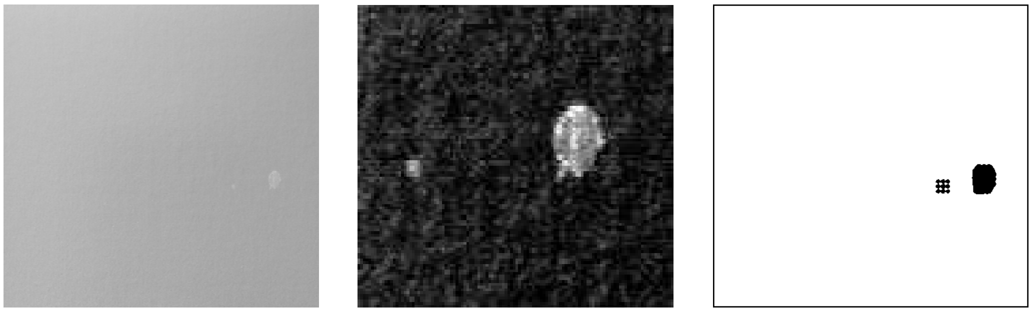
Given the structure of the feature space for painted steel images, an alternative is to fit a cylinder along $\vec{1}$ with a radial Gaussian component. While this is attractive for this image class it unnecessarily assumes a constant local intensity. Hence it is less general and consequently less appropriate and hence not considered here.

4.5.2 Results

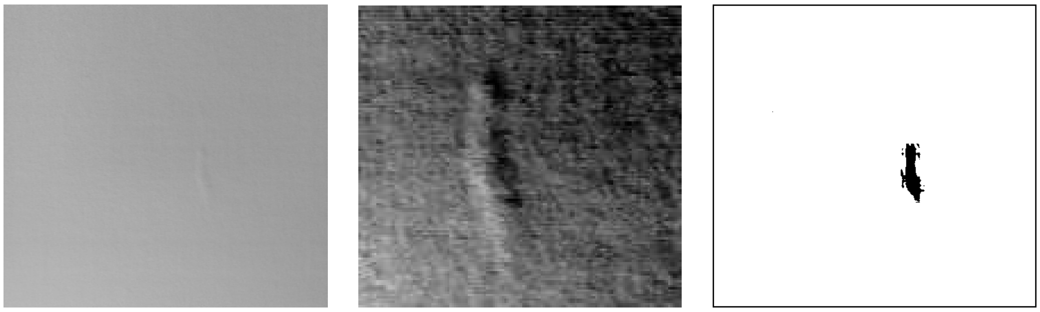
The results using a Mahalanobis threshold of 6.5 are shown in Figures 4.11 to 4.14. In each case the feature space of the image is modelled and hence anomalies are located with respect to image characteristics. For a real-time defect screening system, it is more reasonable to model the feature space of an ideal image, identifying defects by points which disagree from this ideal model. This would allow defects in images without large homogeneous backgrounds to be identified, but require strictly controlled lighting conditions between images. This is not examined further since the intensity and lighting direction varies considerably across images.

Figure 4.11 concisely illustrates the success of this method. The small defect in the *resin* image contains four small holes. These occur from mask placements where the defect is evenly separated into four different windows causing it to be diluted. Results for other painted steel images are shown in Figure 4.12. As previously outlined the *frost* defect is so distributed that it constitutes the homogeneous background model. Hence only severe local anomalies are identified.

Figure 4.13 outlines anomalies corresponding to shadows, creases and defects. In this



(a) original *resin* image (b) equalised section (c) anomalous region

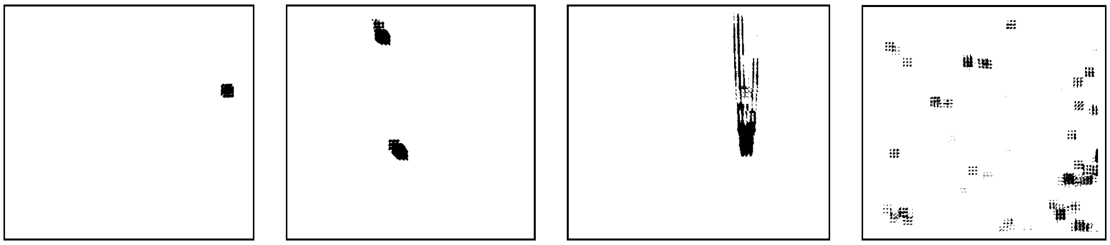


(d) original *dross* image (e) equalised section (f) anomalous region

FIGURE 4.11: Detection of *resin* and *dross* type flaws from images 2151 and 2026. Obtained using a Mahalanobis threshold of 6.5.

case a simple threshold of image intensity locates only the defect, but this uses unjustified knowledge and is not general. Furthermore, the distinction between creases and potential tears is difficult under this experimental setup. A real-time implementation requires an abundance of training data and would benefit from colour processing with a yellow background conveyor to distinguish creases from tears. In general however, this example does highlight the areas which do not conform with the background characteristics.

Finally, Figure 4.14 illustrates anomaly detection for a less uniform image. In this case the regular vertical texture is described by sufficient local descriptors, or feature vectors, that it is, again, learnt. Regions of text and irregular edges are identified as anomalous. In



(a) *dirtspot* (b) two defects (c) *fingermark* (d) *frost*

FIGURE 4.12: The detection of defects from the remaining images in Figure 4.1 also obtained with a threshold of 6.5.

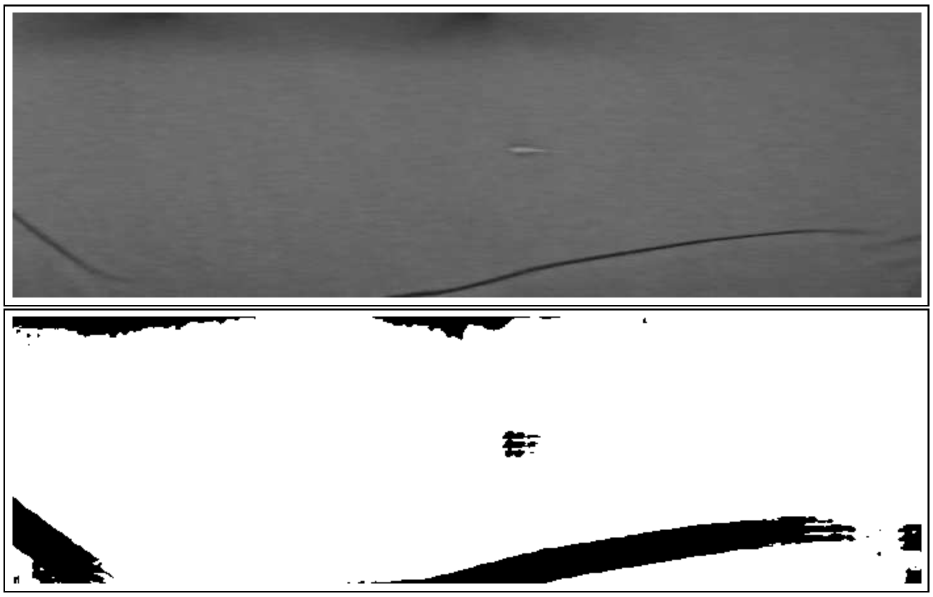


FIGURE 4.13: The detection of anomalies in a section of fabric.

this example some of the text coincides with the uniform vertical texture and is hence not detected, which should be expected given no prior contextual information or knowledge. Again, in both images the ares which stand out above the homogeneous background(s) are successfully located. Although this image has a high resolution (1024 by 859) the same parameters have been used, i.e. $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 7$ and a threshold of 6.5.

4.5.3 Implementation details

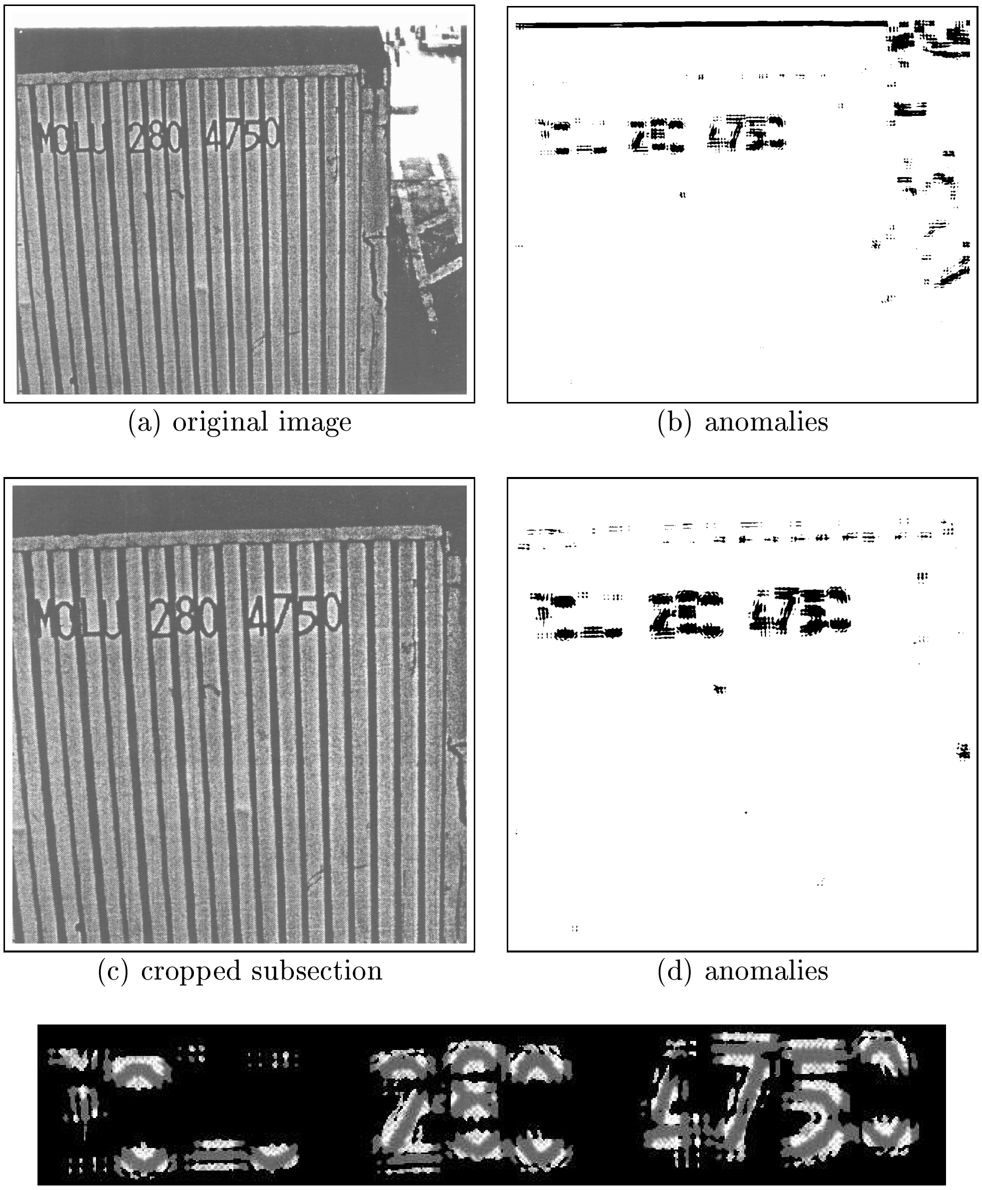
In summary, implementation requires the following steps.

- For a grey scale image $\mathbf{I}[\mathbf{i}, \mathbf{x}, \mathbf{y}] \subset \mathbb{R}^3$ extract local feature vectors as outlined in Section 4.3 using an average pixel intensity measurement, $\mathbf{k} = 3$, $\mathbf{n} = \mathbf{s} = 7$ and $\delta_x = \delta_y = 1$. This forms the feature set $\mathbf{P} \subseteq \mathbb{R}^{11}$.

- Model all points $\mathbf{p} \in \mathbf{P}$ using a Gaussian distribution by taking up to second order geometric moments. This model is characterised by the number of data points n, the mean μ and a covariance matrix $\mathbf{C}$, calculated according to the equations

$$\mu_i = \left(\sum_{\mathbf{p} \in \mathbf{P}} \mathbf{p}_i \right) / n$$

$$\begin{aligned} \mathbf{C}_{ij} &= E(\mathbf{p}_i - \mu_i) E(p_j - \mu_j) \\ &= \left(\sum_{\mathbf{p} \in \mathbf{P}} \mathbf{p}_i \mathbf{p}_j - \mu_j \sum_{\mathbf{p} \in \mathbf{P}} \mathbf{p}_i - \mu_i \sum_{\mathbf{p} \in \mathbf{P}} \mathbf{p}_j + n \mu_i \mu_j \right) / (n - 1.0) \end{aligned}$$



(a) original image

(b) anomalies

(c) cropped subsection

(d) anomalies

(e) Regions of original corresponding to anomalies.

FIGURE 4.14: Location of anomalous sections in a miscellaneous image with homogeneous background characteristics.

- Calculate the inverse of $\mathbf{C}$. The matrix will have positive semi-definite form and hence the inverse is most easily calculated by determining the eigensystem of $\mathbf{C}$. The eigensystem also comes in useful for determining and plotting the principal axes. This is achieved by decomposing $\mathbf{C}$ into its eigenvectors $\mathbf{E}$ and a diagonal matrix $\mathbf{D}$ whose non-zero elements correspond to the square of the eigenvalues λ_i^2. This decomposition is performed by a tridiagonalisation function which uses a series of transformations to sequentially tridiagonalise rows and columns into sub-matrices which are orthogonal and symmetric. This decomposition also allows the *square root* of $\mathbf{C}$, i.e. $\mathbf{B}$ such that $\mathbf{C} = \mathbf{B}\mathbf{B}^T$, to be determined[6].

$$\mathbf{C} = \mathbf{E}\mathbf{D}\mathbf{E}^T = \mathbf{E}\mathbf{Q}\mathbf{Q}\mathbf{E}^T = (\mathbf{E}\mathbf{Q})(\mathbf{E}^T\mathbf{Q}^T) = (\mathbf{E}\mathbf{Q})(\mathbf{E}\mathbf{Q})^T = \mathbf{B}\mathbf{B}^T$$

  Since the matrix $\mathbf{E}$ is orthonormal and $\mathbf{D}$ simply scales each dimension, the inverse is

$$\mathbf{C}^{-1} = \mathbf{E}^T\mathbf{D}^{-1}\mathbf{E}$$

  Note also that $\mathbf{C} = \mathbf{C}^T$ and $\mathbf{B} = \mathbf{B}^T$.

- Calculate the Mahalanobis distance of each point $\mathbf{p}$ with respect to this model as defined on page 144.

- Label all points with a distance greater than 6.5 as anomalous.

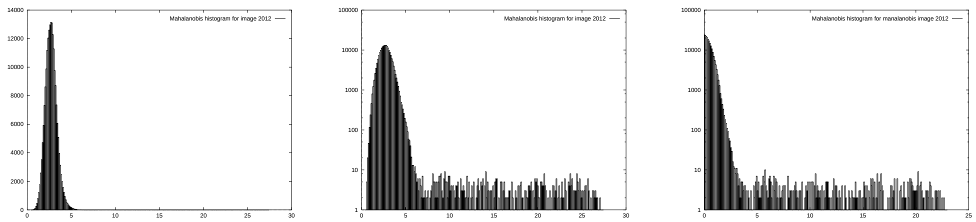
The above steps take about 7.1 seconds on the painted steel images using an R10000 processor. Modelling the points including the calculation of eigenvector, eigenvalues and the inverse covariance matrix takes 3.2 seconds. The calculation of the Mahalanobis distance for each data point takes 3.9 seconds and the thresholding less than 0.03 seconds. Obviously the evaluation of Mahalanobis distances can be performed in parallel.

4.5.4 Consideration of Mahalanobis distance

While using a static Mahalanobis threshold of 6.5 works very well across numerous images, its success relies on the assumptions made in Section 4.1. Examining the distribution of the Mahalanobis distance within the feature space provides an insight into how well these assumptions hold. This distribution is referred to as the *Mahalanobis histogram*.

Figure 4.15(a) and (b) show the Mahalanobis histogram for the default *dirtspot* image on linear and logarithmic scales. It is clear that the majority of points lie close to the model

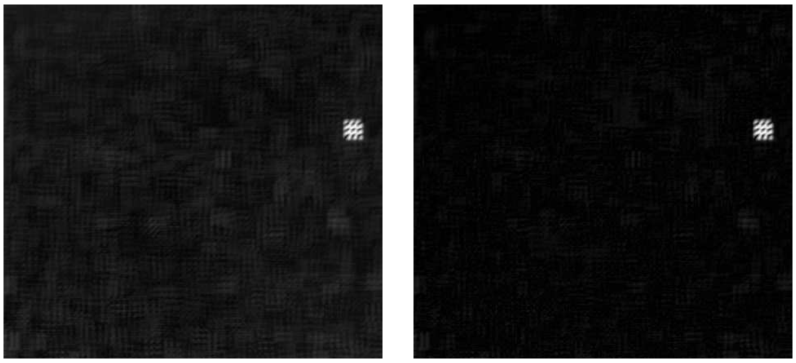
[6] This matrix provides a representation of the elements of $\mathbf{C}$ which is linear and may be more desirable for comparing matrices for recognition or clustering of objects.



(a) linear scale (b) logarithmic scale (c) remodelled histogram

FIGURE 4.15: Mahalanobis histograms for *dirtspot* image 2012.

of the feature space. Furthermore, as a result of the image's well defined homogeneous background, the histogram is unimodal and appears similar in form to a one dimensional Gaussian. By modelling this as a Gaussian and taking the Mahalanobis distance of the points in histogram (a) with respect to this new model a new Mahalanobis histogram is formed. Shown on a logarithmic scale in Figure 4.15(c), this provides a greater distinction between the background and the anomaly as illustrated in Figure 4.16. Repeating this procedure produces an increasingly narrower cluster at a distance of zero, causing points to either converge or diverge. This provides a novel automatic thresholding procedure specific to unimodal distributions with anomalous points.

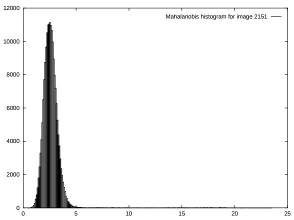


(a) from Figure 4.15(a) (b) from Figure 4.15(c)

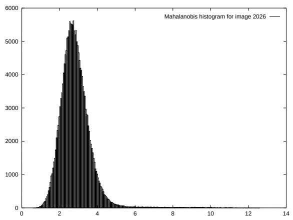
FIGURE 4.16: Mahalanobis distance from original and *refined* background model.

Mahalanobis histograms for other painted steel images result in similar distributions. As expected the *fingermark* image contains more anomalous points and the *frost* image a more gradual transition between the background and anomalous points. Figure 4.17(f) shows the effect of repeated modelling on the *frost* feature space.

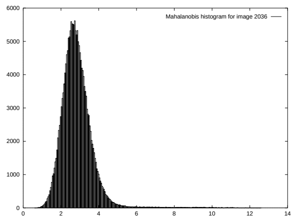
The crate images from Figure 4.14 form more interesting Mahalanobis histograms. For the first image three histogram features correspond to non-anomalous feature vectors. Thresholding confirms that these correspond to dark regions, vertical texture and white regions respectively (from left to right). By cropping the image, the first feature is weakened and merges with the second, and the third feature is eliminated. This structure is



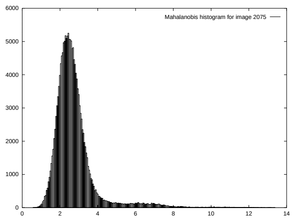
(a) *resin*



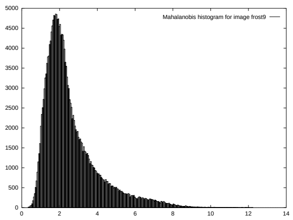
(b) *dross*



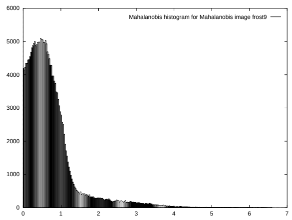
(c) two defects



(d) *fingermark*



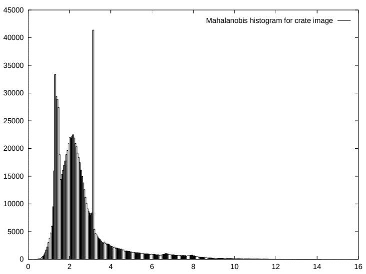
(e) *frost*



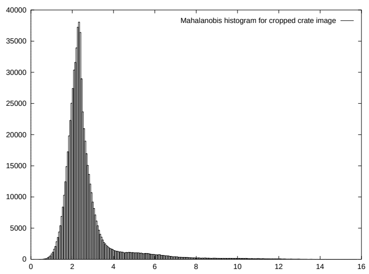
(f) *frost* (*refined*)

FIGURE 4.17: Mahalanobis histograms from original and *refined* background model.

intuitively correct. In the first image three distinct types of homogeneous background are clear, whereas in the second only one type is dominant. Regardless of this, in both cases the anomalous data points incur large Mahalanobis distances.



(a) from original



(b) from cropped image

FIGURE 4.18: Mahalanobis histograms for the crate images in Figures 4.14(a) and (c).

The final example in Figure 4.19 contains a relatively small although significant sized whale (with calf) on an ocean background[7]. The whale has regular local characteristics and anomalies result from incidental features such as splashes. The strength of the Mahalanobis distances and the histogram indicate the presents of these objects. Although the default threshold does outline anomalies, other values produce more meaningful results.

4.5.5 Isolating anomalies

In general it may not be sensible to isolate anomalies since one type of anomaly may be distributed across the image. However, for the majority of painted steel images it is

[7]Again note that only the intensity of the image is used.

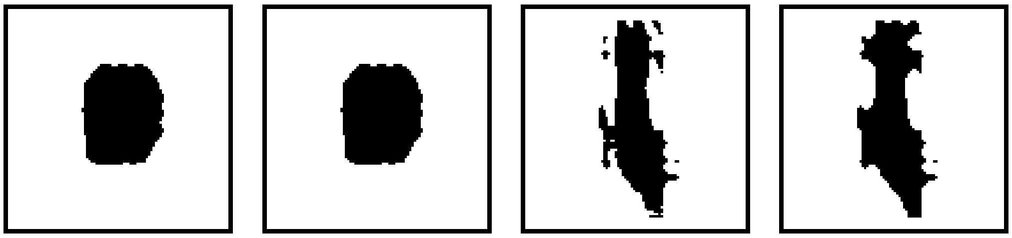
(a) original (b) Mahalanobis distance (c) Mahalanobis histogram

(d) default threshold (e) threshold = 5 (f) threshold = 2

FIGURE 4.19: The modelling of a image containing whales in the ocean.

sensible to examine spatially disjoint defects.

As seen in Figure 4.11(f) feature vectors corresponding to the same anomaly may not correspond to spatially connected image pixels. To cater for this a morphological *closing* operation is applied. This is implemented as a dilation followed by erosion with a circular mask. The effect of applying this operation on results from Figure 4.11 appear in Figure 4.20.



(a) *resin* defect (b) after *closing* (c) *dross* defect (d) after *closing*

FIGURE 4.20: Effect of performing morphological *closing* operation.

Each spatially disjoint anomaly of *adequate size* is located and labelled. The illustrated defects contain 2136 and 2404 points. The small defect in the left of Figure 4.11(b) results in 377 feature vectors, hence regions with fewer than 50 points are considered noise and disregarded.

4.6 Object representation

For this restricted problem there are only two objects. The object corresponding to the background is simply modelled by the number of points, mean and covariance of the entire feature space. An alternative is to remodel the background excluding the points deemed anomalous. This makes little difference given the assumed relative sizes and either way it is modelled by n, μ and $\mathbf{C}$. Note that the number of data points along with the mean ($\mathbf{x}$,$\mathbf{y}$) offset provide information about the size of the original image.

The anomalies are initially represented by removing the background points from the feature space, therefore leaving a set of anomalous (or defective) feature vectors $\mathbf{D} \subset \mathbf{P}$. In this form the set $\mathbf{D}$ is considered for further analysis.

4.7 Representation of anomalies and higher level processing

This section deals with the problem of representing a set of anomalous points $\mathbf{D}$ (possibly defective) for further processing and subsequent classification. It is suggested that $\mathbf{D}$ comprises of a set of lower dimensional manifolds embedded in an $\mathbf{N}$ dimensional space. Furthermore sections of these manifolds correspond to specific low level image features such as defect edge characteristics. The ability to extract such topological structure, which is highly non-trivial in the continuous case, is made more difficult due to the relative sparseness of discrete points in the feature space. However, through viewing projections of $\mathbf{D}$ with *xgobi*[8] [63] well defined structure is visually apparent.

Given that a point $\mathbf{d} \in \mathbf{D}$ is part of an embedded manifold of dimension $\mathbf{n}$, the neighbourhood surrounding this point will lie on an $\mathbf{n}$ dimensional hyperplane. A local Karhunen-Loeve Transformation is used to determine this local dimension $\mathbf{n}$ associated with the point $\mathbf{d}$ as well as the affine section corresponding to this hyperplane. The algorithm for this transformation follows.

4.7.1 Local Karhunen-Loeve transformation (KLT) algorithm

- Take a point $\mathbf{d} \in \mathbf{D}$ of dimension $\mathbf{k^2}$ for *blocksize* $\mathbf{k}$. i.e. the point with the $\mathbf{x}$ and $\mathbf{y}$ offset omitted. Omitting the $\mathbf{x}$ and $\mathbf{y}$ offsets simplifies the problem by limiting all elements of $\mathbf{d}$ to the same units.

- Model the neighbourhood of $\mathbf{d}$ by calculating the mean μ and covariance matrix $\mathbf{C}$ of all points *close* to $\mathbf{d}$.

  - Since all feature vector elements have equal units, points are considered to be close if they are within a certain Euclidean distance from the current point. This distance is referred to as the *resolution radius* (it is a generalisation of that discussed in Section 2.8.3).

  - Alternatively this neighbourhood can be thought of as the region defined by centring a *ball* (i.e. a solid "sphere" in $\mathbb{R}^9$) over $\mathbf{d}$.

- Assign the *eigenvalues*, λ_i of the covariance matrix.

[8] This interactive program provides time varying projections of $\mathbf{N}$ dimensional datasets to a 2 dimensional screen.

- Sort the $\sqrt{\lambda_i}$ in descending order.

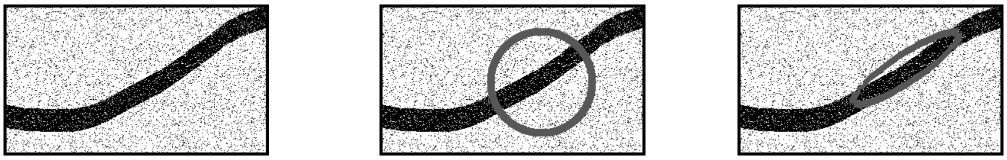
- Determine the dimension of the local neighbourhood by determining the number of *principal* eigenvalues. For example if $\sqrt{\lambda_i} = [6.1,\ 5.8,\ 3.6,\ 2.4 \times 10^{-3}, \ldots]$, the dimension would be 3. Obviously the number of *principal* eigenvalues may not be well defined.

- Store the *number of data points*, *mean*, *covariance matrix*, *eigenvalues* and *eigenvectors* for the local model of the current point. These quantities more than adequately define the neighbourhood.

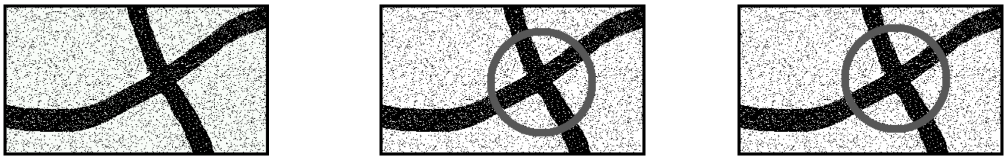
- Repeat for all points in $\mathbf{D}$.

4.7.2 An illustrative example in a low dimension

The local Karhunen-Loeve Transformation can be illustrated more simply by the two dimensional case depicted by Figure 4.21.



(a) an image containing a line (b) local neighbourhood (c) fitted ellipse



(d) an image containing an intersection (e) local neighbourhood (f) fitted ellipse

FIGURE 4.21: A simple illustration of the local Karhunen-Loeve Transformation.

Any line with a sufficiently bounded curvature, such as Figure 4.21(a), can be modelled locally by a short line segment. In this case the feature space $\mathbf{D}$ consists of two dimensional points $(\mathbf{x}, \mathbf{y})$ corresponding to black pixels. The circle in Figure 4.21(b) outlines the neighbourhood of a point $\mathbf{d} \in \mathbf{D}$. The mean μ and covariance matrix $\mathbf{C}$ calculated within this neighbourhood are represented by a *positive semi-definite form* shown as an ellipse in Figure 4.21(c).[9] As suggested by the ratio between the length and width of this ellipse, $\mathbf{C}$ has one principal eigenvalue, hence this local neighbourhood is modelled as one dimensional. The location and orientation of this line segment is also well described. These descriptions provide the information necessary for automatic stroke detection with a high

[9]Note that the *points* forming the ellipse's boundary have a *Mahalanobis distance* of 1 with respect to the positive semi-definite form defined by μ and $\mathbf{C}$ for this local neighbourhood.

degree of certainty. Intersections between strokes can be identified since these are located at the end of *simple strokes* and are locally two dimensional as shown in Figures 4.21(d)-(f). This method produces impressive results in locating and representing the strokes of Kanji characters [300]. It should be noted that although similar to the *Spot Algorithm* [33] described in Section 2.8.3, the neighbourhood centred on each pixel is modelled rather than independent neighbourhoods.

4.7.3 Choosing the resolution radius

Applying a general $\mathbf{N}$ dimensional local Karhunen-Loeve Transformation requires the choice of an appropriate *resolution radius* and a method of determining the number of principal eigenvalues.

The resolution radius should be large enough to ensure that the corresponding neighbourhood contains enough points to form a non-degenerate model. In 2 dimensions at least 3 points are required to produce a non-degenerate model and several more (perhaps 5 or 6) are needed to make the model meaningful. In $\mathbf{N}$ dimensions at least $\mathbf{N}+1$ points are required and as a rule of thumb $2\mathbf{N}$ to $3\mathbf{N}$ points for a reasonable model. On the other hand choosing a large resolution radius forms a neighbourhood less local to $\mathbf{d}$ which may contain points belonging to nearby structures. This is analogous to the example of two parallel lines which at the appropriate scale appear as one line. If a large resolution radius is used these lines cannot be distinguished.

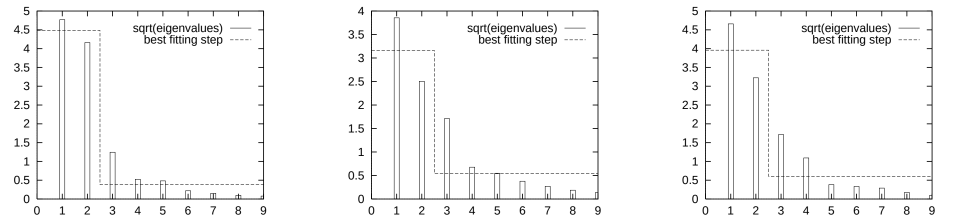
One potential solution to tackling degeneracy without compromising locality is, for each point $\mathbf{d}$, to dynamically change the size of the resolution radius. Beginning with a small resolution radius the algorithm counts the number of points in the corresponding neighbourhood, increasing the resolution radius until it encompasses a *sufficient* number of points. While this method is attractive it not only produces less uniform models across the dataset $\mathbf{D}$, but often results in inferring structure at a smaller than desired scale. For example, using this method on the data in Figure 4.21(a) would result in modelling a neighbourhood within the width of the line as two dimensional (i.e. as a solid *blob*) rather than one dimensional (i.e. part of a line). In this work a constant resolution radius has been chosen through examination of various anomalous datasets using *xgobi*.

4.7.4 Determining dominant eigenvectors based on eigenvalues

The method of determining the number of principal eigenvectors is now examined. The values of λ_i provide a measurement of length corresponding to the eigenbasis of $\mathbf{C}$. Given an ellipse of length 5 and width 1 it is fair to infer a local dimension of 1 and hence model it as a line segment. However for a length of 5 and a width of 2 it is not so clear cut. For this simple two dimensional case the desired tolerance can be converted into a simple threshold, however the general extension into $\mathbf{N}$ dimensions is non-trivial.

One method is to simply look at the ratio between each eigenvalue[10] and the largest

[10] The square root of the eigenvalue is used. The word *eigenvalue* is used instead of *the square root of*



(a) 2 dimensional (b) 3 dimensional (c) 2 or 4 dimensional

FIGURE 4.22: Determining the number of principal eigenvalues.

eigenvalue, i.e.

$$dimension = \max(i) \mid \left(\frac{\lambda_i}{\lambda_1} > threshold\right)$$

however this produces unsatisfactory results for examples such as $\lambda = [10\ \ 3.1\ \ 3.0]$ $\lambda = [10\ \ 3.1\ \ 2.9]$ $\lambda = [10\ \ 2.9\ \ 2.8]$ which given a threshold of 30 percent are deemed to be of dimension 3,2 and 1 respectively.

A second method is to locate a sudden change in eigenvalue magnitude by considering the ratio between successive eigenvalues, i.e.

$$dimension = \max(i) \mid \left(\frac{\lambda_i}{\lambda_{i-1}} > threshold\right)$$

This is also unattractive since it takes no account of the relative magnitude of the current eigenvalue. In the extreme case $\lambda = [1\ \ \frac{1}{2}\ \ \frac{1}{4}\ \ \frac{1}{8}\ \ \ldots]$ would either be considered 1 dimensional or **N** dimensional depending on the threshold chosen.

The method adopted is a combination of the two above, taking into account both the change between successive eigenvalue and the *distance* from the largest eigenvalue.

$$dimension = 1 + \max(i) \mid \left(\frac{\lambda_1}{\lambda_i} \times \frac{\lambda_{i-1}}{\lambda_i} < threshold\right)$$

Through examining descending eigenvalues produced from different types of data, this method produces intuitively superior results and is robust with respect to a choice of threshold. For example, the number of principal eigenvalues in Figure 4.22(c) is unclear. This method results in 2 for a low threshold (5 to 25), 4 for a larger threshold (anything up to 1000) and 7,8 or 9 for extreme values.

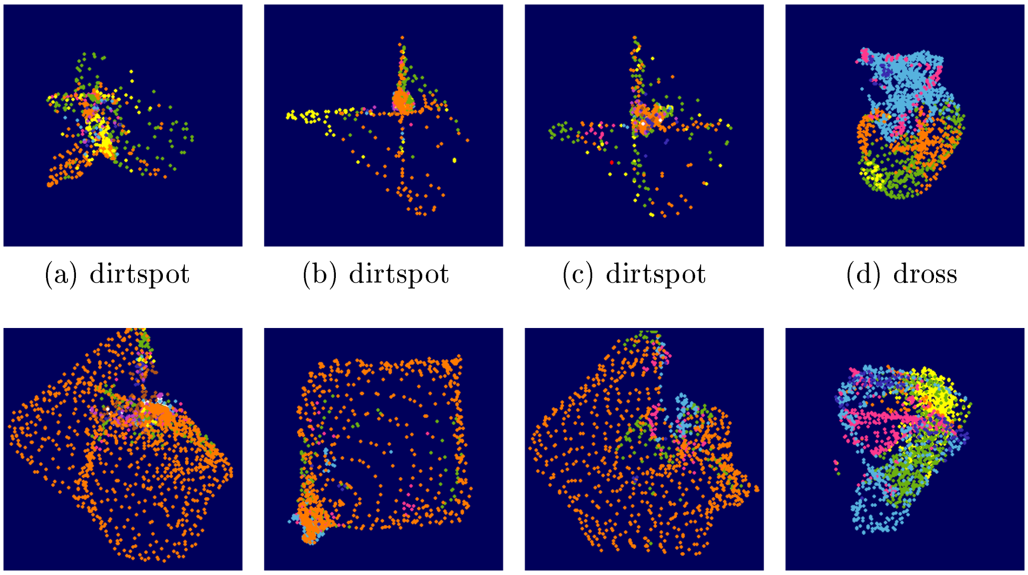
A more common mathematical technique for distinguishing between two types of objects (i.e. principal and insignificant eigenvalues) is to fit a step function with the minimum least squared error. Although this seems reasonable the results are unimpressive as illustrated by Figure 4.22. In Figure 4.22(b) the third eigenvalue is almost half the magnitude of the principal eigenvalue, however the *optimal* step function gives two principal eigen-

the eigenvalue throughout this section to make the descriptions more concise.

values. A further concern with this method is the inability to identify when **N** principal eigenvalues exist. In this case fitting a step function produces somewhat arbitrary results.

4.7.5 Application to anomalous feature vectors

The results of performing a local Karhunen-Loeve Transformation on several different defective painted steel images are illustrated in Figure 4.23. Each image represents a projection of all anomalous points in the 9 dimensional feature space onto a flat screen. It is very important to note that all images represent the *same projection* obtained using the inter-process communications *link* option in *xgobi*. Each point has been assigned a colour according to the local dimension of its neighbourhood, as defined in Table 4.3.



(a) dirtspot (b) dirtspot (c) dirtspot (d) dross

(e) resin (f) resin (g) resin (h) dross

FIGURE 4.23: Representations of anomalies based on the Karhunen-Loeve Transformation.

colour	dimension	colour	dimension	colour	dimension
yellow	1	hot pink	4	orchid	7
orange	2	sky blue	5	peru	8
yellow green	3	slate blue	6	white	9

TABLE 4.3: Dimension key for results shown in Figures 4.23 and 4.24.

Through a real-time examination of these datasets many similarities are noticeable between defects of the same class.[11] Images with a *resin* type defect consist mainly of a two dimensional surfaces with a large surface area, *dirtspot* defects consist of two and three dimensional data and *dross* defects are distinguished by neighbourhoods of a higher

[11]These similarities cannot be fully realised from the projections in Figure 4.23 since 7 of the 9 dimensions are *hidden* behind each point.

dimension. At first the distinction between *resin* and *dirtspot* defects is less clear. By representing these coloured points with respect to spatial location more distinguishing characteristics become clearer, as seen in Figure 4.24. The top row of images represent *dirtspot* defects and bottom row *resin* defects. Distinguishing features allow image defects to be classified as presented in Section 4.8.

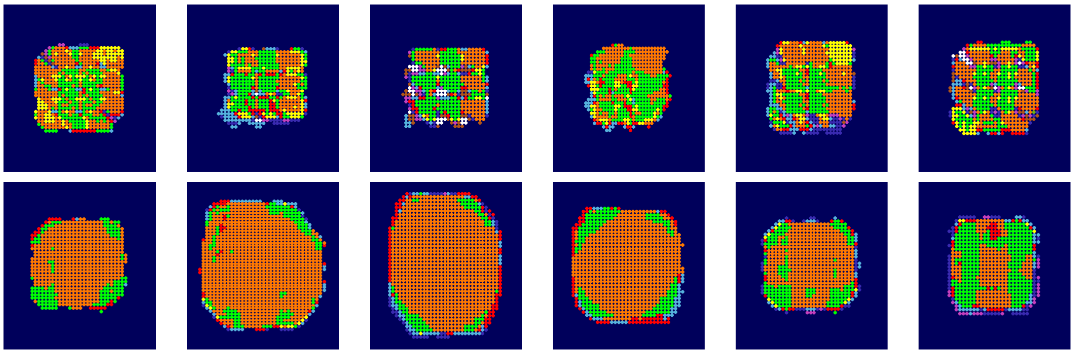


FIGURE 4.24: Spatial projections of the feature space corresponding to images with *dirtspot* (top) and *resin* (bottom) type defects.

Ideally it is desirable to combine or amalgamate neighbourhoods of the same dimension and *appropriate* alignment into manifolds. This is extremely complicated in the **N** dimensional case, made more difficult by the sparseness of points in the discrete space. The two and three dimensional cases are studied by Cooper [300].

4.8 Classification of painted steel defects

Figures 4.23 and 4.24 begin to illustrate the structure of the anomalous feature vectors, suggesting sufficient information to successfully classify different types of defects. Since the dataset used contains just 13, 22 and 23 examples of *dross*, *dirtspot* and *resin* type defects, a small number of measurements is necessary[12]. There are many possible measurements such as those outlined below.

- Since the dimension of the three types of defects seems to vary, the average local dimension of the feature vectors is a possible measurement. Based on illustrations, this number is expected to be low for *resin* which contain mostly two dimensional data points and higher for the other two defect types.

- The standard deviation of the local dimension. Examination shows *dirtspot* defects to have a higher value than the other two.

- Either the number or percentage of points with a local dimension of 1, 2, et cetera. Although this is a good set of measurements it cannot be used since it involves

[12]Classification using a large number of measurements would lack the necessary training data and lead to degeneracy or increased likelihood of gross misclassification.

too many numbers. Neither one is particularly better than the others and using all of them leads to degeneracy during later modelling due to the sparseness of points. Figure 4.25 shows a projection of such measurements. Obvious clusters are not apparent, although defects of the same type tend to be close to one another. It should be noted that this measurement is sensitive to changes in the threshold used to determine local dimension. It also lacks consideration for neighbouring dimensions. For example sets of feature vectors of dimension 2 would be equidistant from a set of dimension 3 and 9.

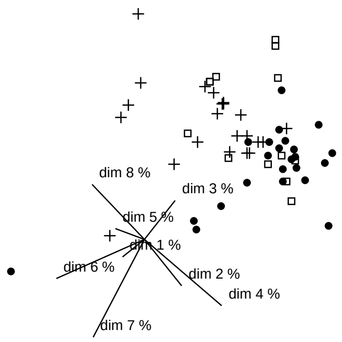
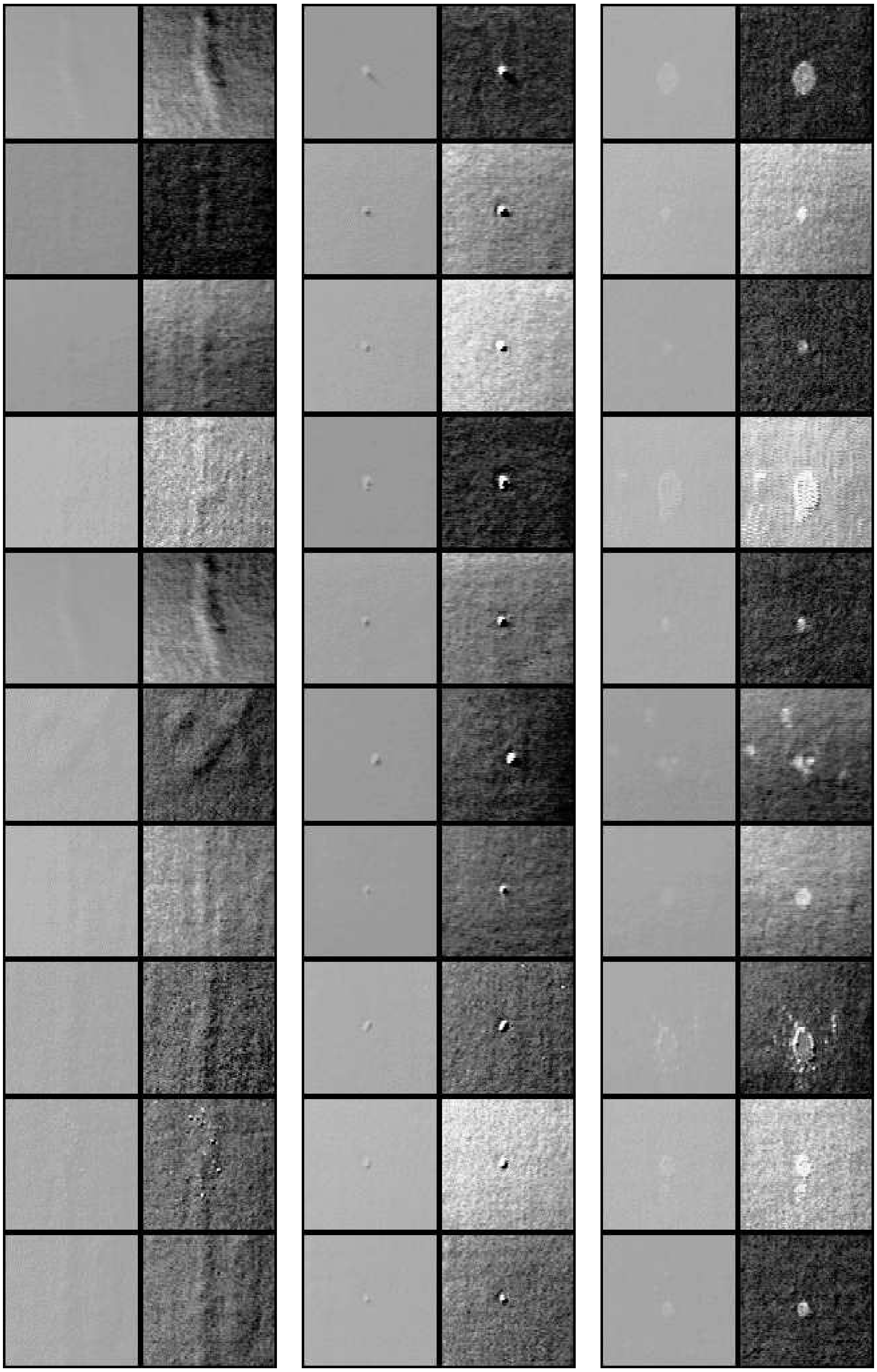


FIGURE 4.25: The percentage of points of each local dimension for *dross* (square), *resin* (ball) and *dirtspot* (plus/cross).

- The size of the defect. This is a very good measurement for distinguishing *dross* from *dirtspot* and *resin* since it occupies more of the image. However size is not used for classification due to its crudeness. It is more desirable to extract features from the structure of the anomalous feature vectors rather than the fact that one type is bigger than another.

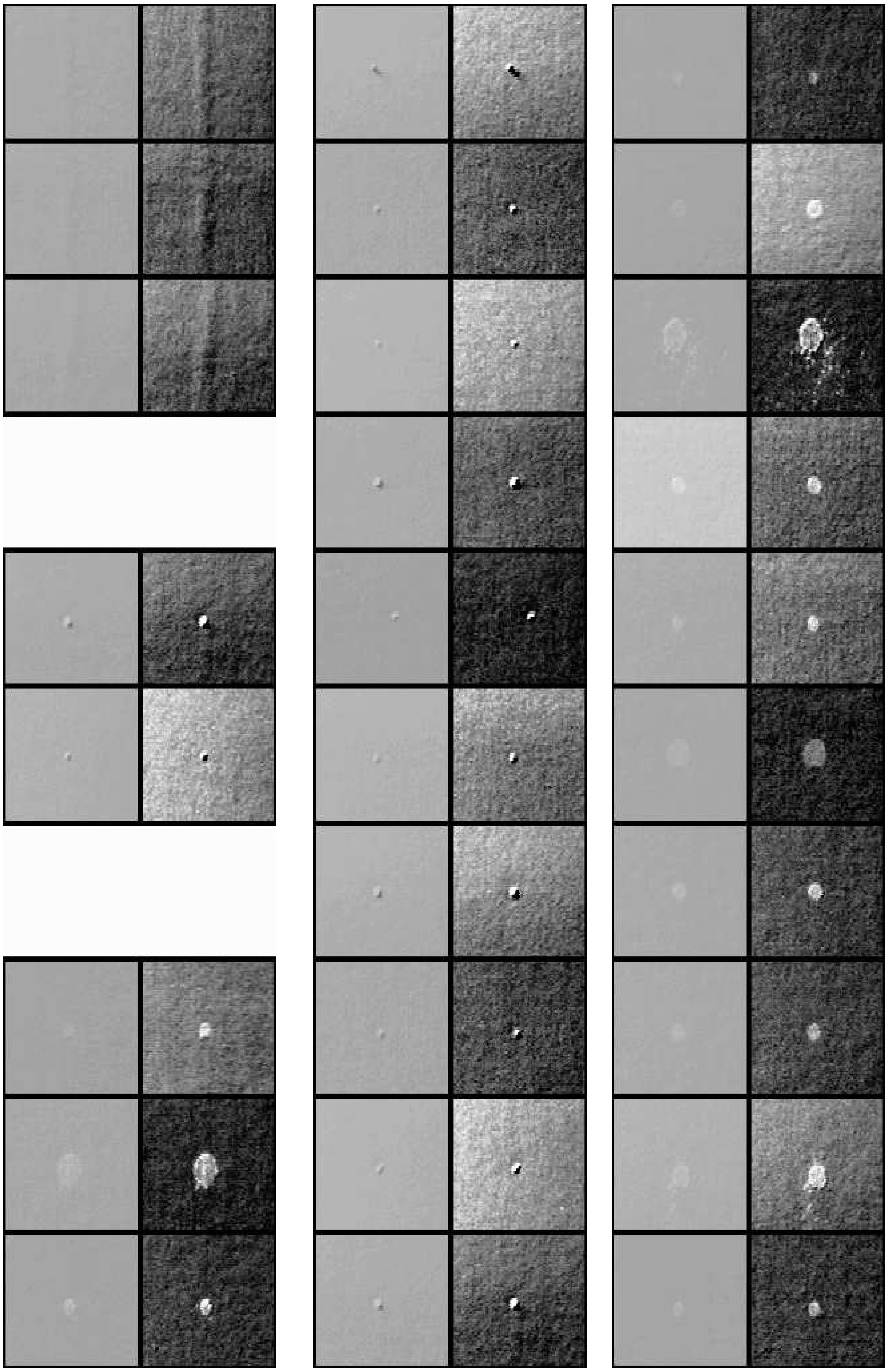
- The percentage or number of points of a certain local dimension which lie close to the centre of the defect in the original image. As seen in Figure 4.24 *resin* defects are typified by two dimensional data close to the defect's centre. The distance from the edge of the defect can be determined by using the Euclidean distance measure for binary images. By counting the number of two and three dimensional feature vectors within the central 60 or 80 percent of the centre the *dirtspot* and *resin* type defects shown in Figure 4.24 are easily distinguishable.

- The average local dimension of defective feature vectors close to the defect's centre. Again the Euclidean distance measure is suggested, with measurements examined for various proportions of the defect. It is unclear that any single one of these measurements is particularly useful in distinguishing between classes and is hence not useful for such a limited dataset.



(a) *dross* (b) *dirtspot* (c) *resin*

FIGURE 4.26: [part 1] Images used for classification.



(a) *dross*, *dirtspot* and *resin* (b) *dirtspot* (c) *resin*

FIGURE 4.26: [part 2] Images used for classification.

- The spread of the defective feature vectors. A simple measure for the spread of the feature vectors is given by the *maximum defectiveness*. As outlined in Section 4.5 each feature vector can be assigned a measure of deviation according to its Mahalanobis distance with respect to the overall model of the image. The highest value of this distance provides a measure of maximum defectiveness.

Analysis and experimentation suggests that four measurements are useful for classifying the restricted available dataset. These are the average local dimension, the size of the defect, the percentage of two dimensional points close to its centre and the maximum defectiveness. However, as outlined above, the size does not require the use of the local Karhunen-Loeve transformation or possible manifold modelling. Since it does not provide an insight into the correct classification based on structure between the relatively large or distributed *dross* type defect and the other defects it is not used. Furthermore, the percentage of two dimensional points close to the defect's centre is based on characteristics specific to the chosen dataset. These characteristics have not been automatically determined, but rather deemed important through human judgement. The measurement is not considered general and is also excluded.

The classification of all available *dross*, *dirtspot* and *resin* images is now examined. The defective sections of the original images are shown in Figure 4.26. The images to the left represent the original intensity, with those to the right equalised to enhance contrast.

Figure 4.27 shows the measurements of average local dimension and maximum defectiveness for these images. It is clear that defects occupy different regions, although there is some interaction between the classes. This interaction can be expected due to the visual similarity between very small *resin* defects and *dirtspot* defects observable from the original images in Figure 4.26.

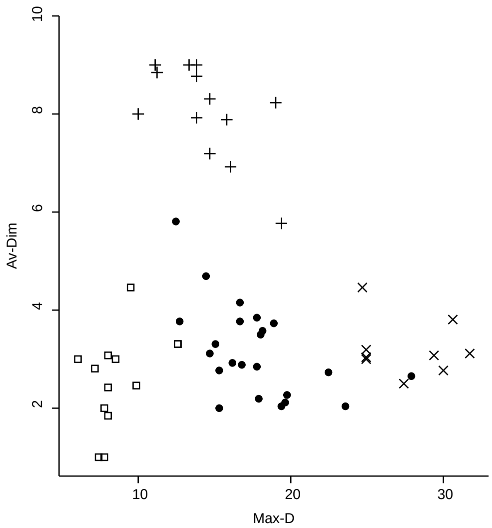


FIGURE 4.27: Measurements from *dross* (square) , *resin* (ball) and *dirtspot* (plus/cross) images.

Classification is achieved through the use of statistical methods equivalent to the Quadratic Neural Network [301] [302]. Due to the sparseness and arrangement of the measurements for this dataset 2 classes are used to represent *dirtspot* defects, giving a total of 4 classes. Training is performed by modelling 57 of the 58 images, with the final image classified according to this model. This process is repeated until all images have been classified. Note that each classification is independent making use of separate training and testing sets. The data is separated in this way to ensure a sufficiently high number of measurements for each class. This is necessary to obtain a better class model and avoid degeneracy.

The results are shown in Table 4.4. The initial results classify each defective image by determining the closest class by comparing the Mahalanobis distances between classes. This method is often referred to as the *Maximum Likelihood*.[13] As seen in Table 4.4(a), 51 of the 58 images are classified correctly. This classification does not take into account points which are a long way from all models or points which lie close to the boundary between classes. The final classification, shown in Tables 4.4(a) and 4.4(b), assigns a high certainty to images which agree well with a class model and are closer to this model than any other by at least 1 standard deviation. Using this method 49 images are classified correctly, 8 are uncertain and 1 is misclassified. In all cases the low certainty results from images being close to class boundaries with alternative classifications shown in Table 4.4(c).

Processing time

Although successful and mathematically elegant the implementation of these techniques takes a significant amount of computation. The time taken with consideration for a real-time system is outlined below.

- The feature extraction takes 0.29 second.
- The modelling and anomaly identification takes about 7.1 seconds. However for a practical system a model can be pre-calculated and possibly refined off-line and hence is not necessarily time-critical. Given this model, the calculation of the Mahalanobis distance is independent for each data point and can therefore be performed in parallel. In a practical system a number of computers each with a digital camera capturing a subsection of the material is appropriate.
- The morphological operations and determination of the largest defect takes about 2 seconds although has been implemented very generally without consideration for performance. In practice it may be possible to perform the analysis of a detected defect off-line, depending on the physical setup of the operation.

[13] Strictly speaking it is not actually the maximum likelihood by mathematical definition. The maximum likelihood should provide the likelihood of a model fitting predefined data and not how well an additional data point agrees with the model of other points.

	Maximum Likelihood		
Defect type	dross	dirtspot	resin
dross	10	0	3
dirtspot	0	20	2
resin	1	1	21

(a) Maximum likelihood classification.

	Classification					
	dross		dirtspot		resin	
	Certainty					
Defect type	high	low	high	low	high	low
dross	10	0	0	0	0	3
dirtspot	0	0	20	0	0	2
resin	0	0	1	0	18	3

(b) Classification considering class boundaries.

	Classification								
	dross			dirtspot			resin		
	Alternative classification								
Defect type	none	dirtspot	resin	none	dross	resin	none	dross	dirtspot
dross	10	0	0	0	0	0	0	0	3
dirtspot	0	0	0	20	0	0	0	0	2
resin	0	X[a]	X[a]	1	0	0	18	1	2

(c) Possible alternative classifications for results with a low certainty.

TABLE 4.4: Classification results for painted steel images shown in Figure 4.26.

[a] This images corresponds to the point at coordinates 12,0.8 in Figure 4.27. When removed from the training data this point lies approximately the same distance from all classes (distances of 3.37, 3.59 and 3.63 standard deviations). Consequently the 'X' cannot satisfactorily be assigned to any of these classes.

- Performing the local KLT takes up to the order of 100 seconds and is dependent on the number of data points in the anomaly. The *dirtspot* defect (image 2012) which has 668 anomalous points takes 20 seconds whereas for the *dross* defect (image 2026) with 5330 anomalous points 96 seconds are required. Again no optimisation techniques are used, but it is unclear how to efficiently determine the set of points within a certain resolution radius in $\mathbb{R}^9$. Again this can probably be performed off-line.
- Determining the local dimension for all of these models takes less than 0.01 seconds.
- Once the defect has been modelled, the final measurements and classification time are insignificant.

In summary the real-time detection of anomalies in painted steel images appears to be feasible given a sensible implementation design. However, the classification methods proposed here require off-line processing or massive optimisations.

4.9 Anomaly removal: An illustration of a partial Down-Write

This section examines the problem of image restoration via anomaly removal. It obviously raises the question, *why would anyone want to restore a defective image?* In the case of the painted steel images it at first seems pointless to attempt to remove the defect from the image. However it is desirable for the following reasons.

- Restoring an image may be aesthetically pleasing. This is not applicable for painted steel images. However it is applicable to Chapter 6.
- The ability to recreate an instance of the modelled object provides insight into the information captured by the model. Hence it outlines the power and completeness of the model. Earlier in this chapter it was suggested that the multivariate Gaussian model describes the general global characteristics of the image, thus determining what is normal and what is defective. This model makes the assumption that the feature space consists of one large Gaussian cluster and a few anomalous points. Assuming this to be correct and assuming the model is effective, an algorithm should be able to use it to restore anomalous sections of the image according to *what the model expects to see*. Hence the successful restoration of an image's defective sections confirms (at least intuitively) the validity of the model.
- Similarly the reconstruction outlines or at least suggests the limitations of the model. This is the information which is lost by amalgamating feature vectors into a more concise higher level description. As discussed in Section 3.2 an unacceptable reconstruction does not necessarily prove the inadequacies of the model.

4.9.1 The method

The main principle behind the method is the property of the multivariate Gaussian distribution which when restricted to, or cut by, a hyperplane forms another multivariate Gaussian in a lower dimension. For example, cutting the two dimensional Gaussian shown in Figure 4.10(b) with a line ($y = mx + c$) forms a function which has the shape of a one dimensional Gaussian. This new function is a *scaled* normal distribution. It is scaled in the sense that the integral, or area under the curve, is not necessarily equal to 1.

Using this principle, the image model which is a multivariate Gaussian in 11 dimensions is restricted to a specific $\mathbf{x}$ offset using the appropriate hyperplane. This is in turn restricted to a specific $\mathbf{y}$ position, thus giving a model appropriate for a single image position. The parameters of this new model provide the expected feature vector values and the expected variation of vector elements correspond to the specified offset in the image, thus allowing an artificial data point to be constructed.

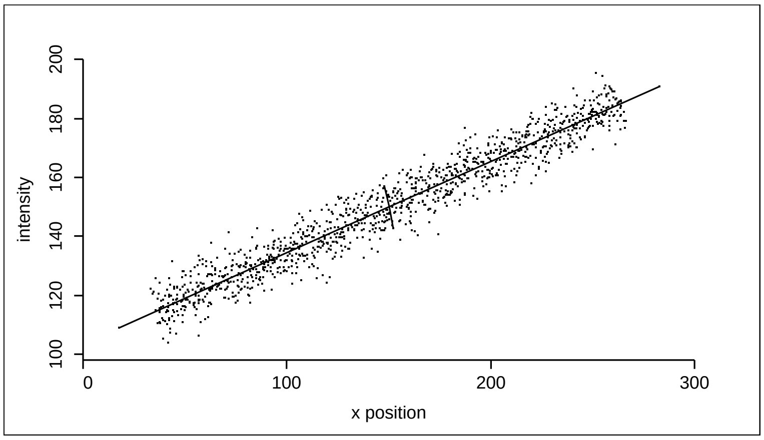


FIGURE 4.28: Two dimensional data modelled by a Gaussian.

To begin with, consider a distribution over two variables corresponding to position and intensity. This is modelled with a two dimensional Gaussian as shown in Figure 4.28. The lines correspond to orthogonal[14] eigenvectors of length $2\sqrt{\lambda}$ for the associated eigenvalue λ (according to the model's covariance matrix $\mathbf{C}$). The basis defined by $\mathbf{C}$ gives rise to a quadratic metric defined for these two variables. Hence the distance between two points can be calculated in units of standard deviations with respect to the basis of $\mathbf{C}$ regardless of position or intensity.[15] Consequently the restriction to a specific position (or variable) gives rise to a model with equivalent standard deviation or covariance terms in the remaining variables.

The expected value of the distribution at a given spatial position is non-trivial. In Figure 4.28 the expected intensity at $\mathbf{x} = 50$ is about 118, whereas at $\mathbf{x} = 250$ it is

[14] These do not appear orthogonal since the scales on each axes differ.

[15] The Mahalanobis distance is a special case of this quadratic metric from the point to a fictitious point at the mean.

about 180, varying linearly between the two. The problem of determining this equates to maximising the result of the Gaussian function with the associated restrictions. This is now addressed.

Consider an N dimensional Gaussian model restricted to j dimensions according to $N-j$ restrictions. For this application $N = 11$, $j = 9$ and the restrictions are to the $\mathbf{x}$ and $\mathbf{y}$ offsets.

$$f(\mathbf{p}) = \frac{1}{\sqrt{2\pi}^{\frac{1}{N}}|\mathbf{C}|} e^{-\frac{1}{2}[(\mathbf{p}-\mu)^T\mathbf{C}^{-1}(\mathbf{p}-\mu)]}$$

Without loss of generality the restrictions are made to the last $N-j$ variables. Hence the problem involves determining the maximum of the composite $f \circ \mathbf{i}_a$ where

$$\begin{aligned} \mathbf{i}_a : \mathbb{R}^j &\rightarrow \mathbb{R}^N \\ \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_j \end{bmatrix} &\rightarrow \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_j \\ a_{j+1} \\ \vdots \\ a_N \end{bmatrix} \qquad a \in \mathbb{R}^{N-j}, \quad a = \begin{bmatrix} a_{j+1} \\ \vdots \\ a_N \end{bmatrix} \end{aligned}$$

For a predefined covariance matrix $\mathbf{C}$, the leading term of $f(\mathbf{p})$ is constant, hence the problem is simplified to finding the restricted maximum of

$$\max\left\{e^{-\frac{1}{2}[(\mathbf{p}-\mu)^T\mathbf{C}^{-1}(\mathbf{p}-\mu)]}\right\}$$

The maximisation is equivalent to the minimisation of the quadratic form, i.e.

$$\min\left\{(\mathbf{p}-\mu)^T\mathbf{C}^{-1}(\mathbf{p}-\mu)\right\}$$

The problem is further simplified by centring the model around the origin, hence treating each point with respect to μ. This reduces the problem to finding

$$\min\left\{(\mathbf{p}^T\mathbf{C}^{-1}\mathbf{p}) \circ \mathbf{i}_b\right\}$$

As discussed above the resulting function is Gaussian hence this minimisation acts upon a function which is symmetric and monotonically increasing from a single global minimum (in the same way the Mahalanobis distance increases monotonically from the mean). Hence the problem is solvable by differentiation.

By differentiating $\mathbf{p}^T\mathbf{C}^{-1}\mathbf{p} : \mathbb{R}^N \to \mathbb{R}$ and recalling that $\mathbf{C}$ is symmetric,

$$Df = \left(\frac{\partial f}{\partial \mathbf{p}_1}, \frac{\partial f}{\partial \mathbf{p}_2}, \dots \frac{\partial f}{\partial \mathbf{p}_N},\right) = \mathbf{p}^T\mathbf{C} + \mathbf{p}^T\mathbf{C}^T = 2\mathbf{p}^T\mathbf{C}$$

Since the last $N-j$ terms are constant, differentiation with the composite $\mathbf{i}_b$ yields

$$D((\mathbf{p}^T\mathbf{C}^{-1}\mathbf{p}) \circ \mathbf{i}_b) = \left(\tfrac{\partial f}{\partial \mathbf{p}_1}, \tfrac{\partial f}{\partial \mathbf{p}_2}, \dots \tfrac{\partial f}{\partial \mathbf{p}_j},\right) = 2\mathbf{p}^T\hat{\mathbf{C}}$$

$$\text{where } \mathbf{i}_b = \left[\begin{array}{c} \overbrace{\begin{array}{ccc} & & \end{array}}^{j} \quad \overbrace{\begin{array}{ccc} & & \end{array}}^{N-j} \\ \begin{array}{cccc|ccc} 1 & 0 & & 0 & 0 & \dots & 0 \\ 0 & 1 & & 0 & 0 & \dots & 0 \\ & & \ddots & 0 & & \vdots & \\ 0 & 0 & & 1 & 0 & \dots & 0 \\ 0 & 0 & & 0 & 0 & & 0 \\ & & \ddots & & & \ddots & \\ 0 & 0 & & 0 & 0 & & 0 \end{array} \end{array}\right]$$

where $\hat{\mathbf{C}}$ is the inverse covariance matrix with the last $N-j$ columns removed.

Hence, the global minimum is present when $\mathbf{p}^T\hat{\mathbf{C}} = 0$ (j equations) and $\mathbf{p}_{j+1} \dots \mathbf{p}_N$ are the defined restrictions. This leads to the matrix $\mathbf{A}$

$$\mathbf{A} = \left[\begin{array}{cc} \overbrace{\phantom{\hat{\mathbf{C}}\qquad\qquad}}^{j} & \overbrace{}^{N-j} \\ & \begin{array}{cccc} 0 & 0 & \dots & 0 \\ & \vdots & \vdots & \\ 0 & 0 & \dots & 0 \end{array} \\ \hat{\mathbf{C}} & \\ & \begin{array}{cccc} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ddots & \\ 0 & 0 & & 1 \end{array} \end{array}\right]$$

$$\text{and} \qquad \mathbf{p}^T\mathbf{A} = [\ \overbrace{0\ 0\ \dots\ 0}^{j}\ \overbrace{\mathbf{p}_{j+1}\ \dots\ \mathbf{p}_N}^{N-j}\ ]$$

Rearranging this, the relationship is minimised when

$$\mathbf{p} = \Psi \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{p}_{j+1} \\ \vdots \\ \mathbf{p}_N \end{bmatrix} \quad \textit{where} \quad \Psi = \left[\mathbf{A}^T\right]^{-1}$$

The matrix Ψ is referred to as the *prediction matrix*. It allows the expected value of the feature vector to be predicted given the spatial restriction on the last $N-j$ dimensions.

Given these properties the following steps perform image restoration.

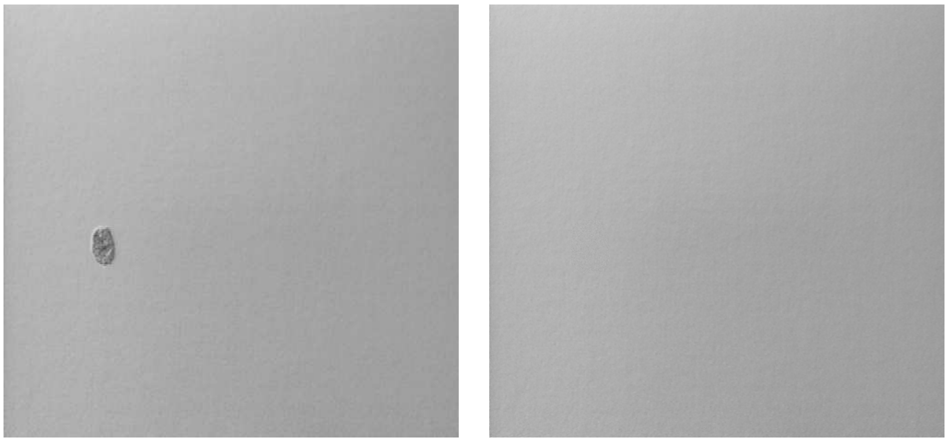
- Calculate the prediction matrix Ψ of the Gaussian model by restricting the last two of the $\mathbf{k}^2+2$ values. Note that this only needs to be done once.
- Take an anomalous point $\mathbf{d} \in \mathbf{D}$ with defined $(\mathbf{x}, \mathbf{y})$ coordinates.
- Calculate the expected value of this point based on the model restricted to this specific image location. This is performed by multiplying Ψ with the vector obtained by setting the first $\mathbf{k}^2$ elements to 0 and the last two to the $(\mathbf{x}, \mathbf{y})$ offset relative to the original model's mean μ. This expected value corresponds to the mean $\hat{\mu}$ of the restricted model.
- Use $\hat{\mu}$ and $\mathbf{C}$ to recreate an artificial data point. Since it is desirable to obtain a pixel value only one of the elements is considered. Gaussian random noise with the appropriate standard deviation is added to the corresponding component of $\hat{\mu}$ to form the artificial pixel value.
- Repeat this process for all anomalous data points, thus removing the anomaly from the original image.
- Use the pixel values from the original image elsewhere.

It is important to note that this procedure performs image restoration by providing artificial pixel values to patch the defect from the original image and is *not strictly a DownWrite*. Applying this prediction technique to every image offset rather than only those corresponding to points in $\mathbf{D}$ constitutes the complete DownWrite of the background object.

4.9.2 Results of image restoration

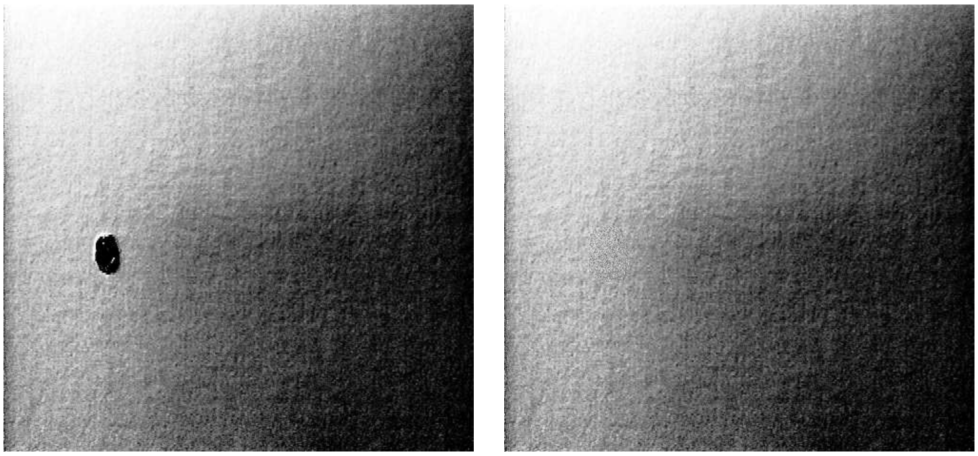
The restoration of an image containing a *dirtspot* or other subtle defect fails to provide insight into the process due to lack of size and obvious features. Therefore a more severe defect is used, as illustrated in Figures 4.29 to 4.31.

As seen in Figure 4.29 the algorithm successfully removes the defect, replacing it with pixels which are not noticeable over the homogeneous background. In fact the reconstruction is so good that the restored area is not immediately obvious even after histogram equalisation. The region of interest in Figure 4.30(b) is debatably only apparent due to its known existence and location, as can be judged by covering Figure 4.30(a). It is interesting to obtain a non-biased opinion by asking someone to *locate the defective region* only revealing Figure 4.30(b).



(a) original (b) restoration

FIGURE 4.29: Restoration of painted steel image number 2127 using the global image model to remove the anomaly.

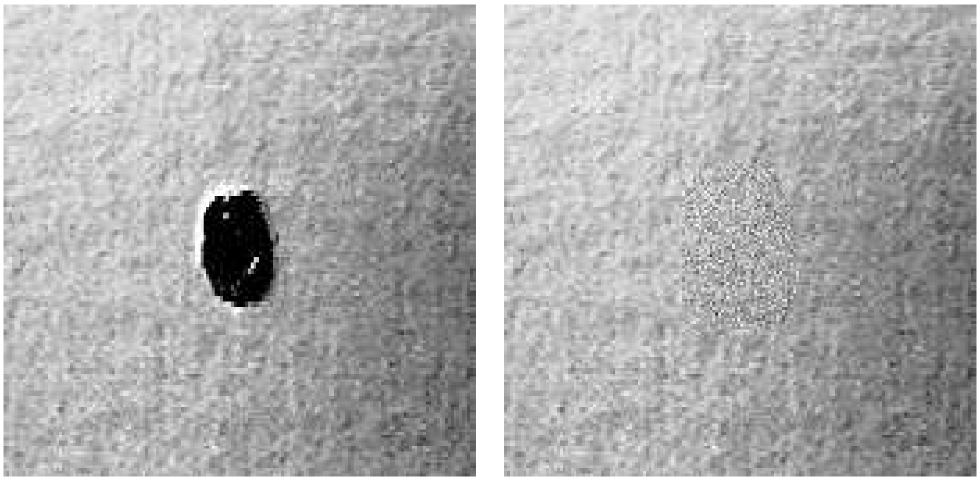


(a) original (b) restoration

FIGURE 4.30: Restoration from last Figure with histogram equalisation.

The final representation is perhaps the most important. Here the *interesting* section of the last image is enlarged to show the fine local texture of the restored image. This section of the image illustrates the *limitations of the model.* The current image model only allows the prediction of a local mean and variance and does not describe the small scale local textural characteristics of the image. In general this shows the limitations of the Gaussian model of the feature space and *not* the information captured by individual feature vectors.

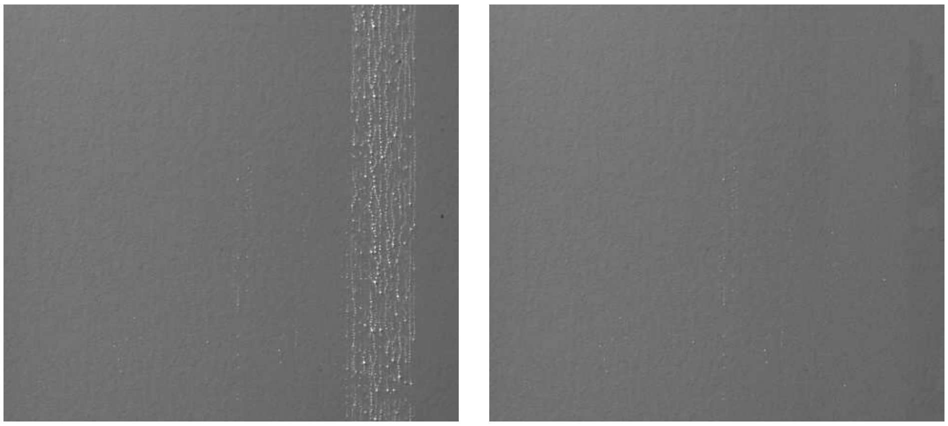
The second example uses an image which stretches the assumption that the defect is relatively small. In this case approximately a quarter of the feature vectors are deemed anomalous. Consequently these points bias the global model causing the reconstructed area to appear marginally brighter that is desirable. Such images outline the need to



(a) original (b) restoration

FIGURE 4.31: A section from Figure 4.30 outlining the limitations of the Gaussian model.

remodel the feature space without including the anomalous feature vectors as previously suggested.



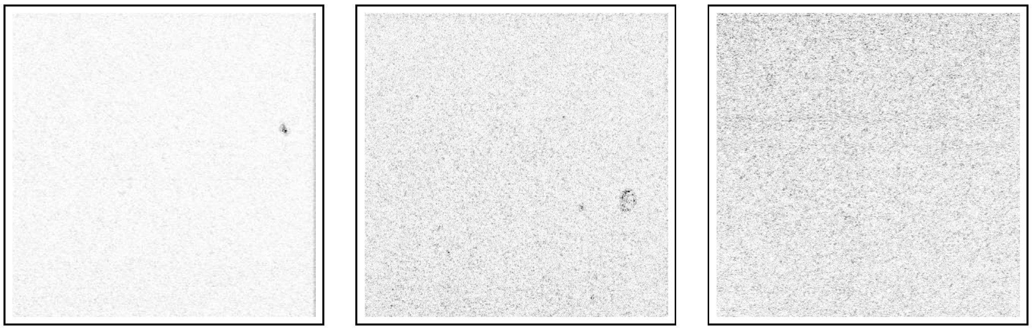
(a) original (b) restoration

FIGURE 4.32: Restoration of a larger defect.

4.10 Summary and concluding remarks

This chapter has introduced a novel feature extraction method which is effective and makes few assumptions about the image. The low level extraction process is adaptable to changes in image resolution, desired neighbourhood complexity and basic measurement changes. This is used with a multidimensional Gaussian to model the homogeneous background characteristics of an image which is assumed to occupy the majority of the image. The model allows anomalous regions of the image to be identified as illustrated on several different images.

With the emphasis placed on locating defects in painted steel images, it is difficult to compare this technique with other feature extraction methods. The automatic defect finding systems presented in the literature make very specific assumptions based on expected image content and are difficult to compare to such a general approach. However, local histogram techniques, template matching, the frequency spectrum and Gabor filters have been considered. Local histograms are successful in locating severe isolated defects such as *resin* or *dirtspots* but struggle on more distributed ones such as *dross*. Template matching via cross-correlation is not particularly effective since there is no real standard example of a particular defect and samples vary too much. Frequency techniques provide reasonable results, but again no obvious common structure exists in the frequency spectrum for subtle defects such as *dross*. The *dirtspot* and *resin* defects tend to have high frequency components as seen in Figure 4.33. In general all of these methods enforce undesirable assumptions. They do not fit well into the adopted framework and provide no clearer methods for higher level representation or classification.



(a) *dirtspot*, [0..24] (b) *resin*, [0..16] (c) *dross*, [0..9]

FIGURE 4.33: Applying a high pass filter to images from Figure 4.1. All images are displayed between 0 (white) and the maximum value (black) as indicated.

As well as identifying anomalies the feature extraction method provides a description from which higher level structure is observed. This structure is examined using the local Karhunen-Loeve transformation (KLT) to model the local neighbourhood associated with each point. Ideally these points should be amalgamated into manifolds and represented appropriately for classification. This is an extremely difficult problem due to the limited number of points in the high dimensional feature space. Furthermore, once manifolds in the anomalous feature space have been located an appropriate method of representation is unclear. It must be noted that an appropriate representation would be complex, involving many numbers and hence require a vast amount of training and testing images.[16]

Classification of isolated painted steel defects is presented using simple measurements from these modelled feature vectors. Considering the limited structure used from the Karhunen-Loeve Transformation the classification results are extremely good. Out of the

[16]Since fewer than 30 suitable examples of each defect are currently available this is not seen as useful for the classification of these painted steel images.

58 images, 49 are classified correctly, 9 are uncertain and 1 is incorrectly classified.

Unfortunately due to the limited size of the dataset and the unavailability of new data the work here is inconclusive and has not been implemented into a real-time screen system. Preliminary analysis suggests such a system is feasible, although it would require parallel processing and off-line classification. In summary, simple optimisations reduce the feature extraction time to about 0.3 seconds for the entire image and the non-optimised identification about 3 seconds. These can be performed in parallel and other modelling performed off-line.

Finally, the information captured by the background model is examined through image restoration and illustrated by the removal of anomalies. Although quite simple, the procedure outlines the effectiveness of the model and its obvious limitation. The method of restricting the Gaussian model to a specific location provides a useful tool for reconstructing image region characteristics, taking care of the effects of simple lighting variations.

Overall the methods fit under the framework of the UpWrite. Low level primitives are combined into feature vectors and in turn structure is examined to identify and represent higher level entities. The identification and representation of objects constitutes the first real UpWrite, in which feature vectors are amalgamated according to the global background model into two objects, namely the background and the anomalies. The background is represented as a set of real numbers, i.e. a point in a Euclidean space $\mathbb{R}^{\mathbf{N}}$. This abstract representation may be used to reconstruct an instance of the object via the appropriate DownWrite as presented. The anomalies are amalgamated into spatially disjoint entities, further modelled and represented by two measurements for classification. A DownWrite from this representation is not considered.