Chapter 4

Edge detection

Low level image analysis aims to extract reliable, local geometric features from an image, in order to: compare one image to another, retrieve shape information, or obtain information for some type of higher level vision [43, 69]. Object boundaries produce step discontinuities in the image intensity, while generally only small oscillations occur between the boundaries. One of the primary goals of low level image analysis is to determine the location of the discontinuities within the image, a process called edge detection. Unfortunately, early vision research determined that edges could be found everywhere in a digital image, due to the oscillations remaining in the image after the digitisation process [69]. Thus in image analysis, smoothing is required to separate the true edges from the inherent noise. This led to the notion of scale-space, where instead of simply examining a feature at a particular location, the feature is examined at a particular location and scale [69].

This chapter introduces the concept of scale-space and multi-scale analysis. Image smoothing is proposed as the mechanism for producing images at various scales, and different smoothing schemes are examined. The chosen scheme called the affine and morphological scale space (AMSS) is presented, along with an explanation of the numerical implementation used, as well as the type of results it produces when applied to long-bone images. The problem of scale selection is also examined, as is the chosen solution to this problem.

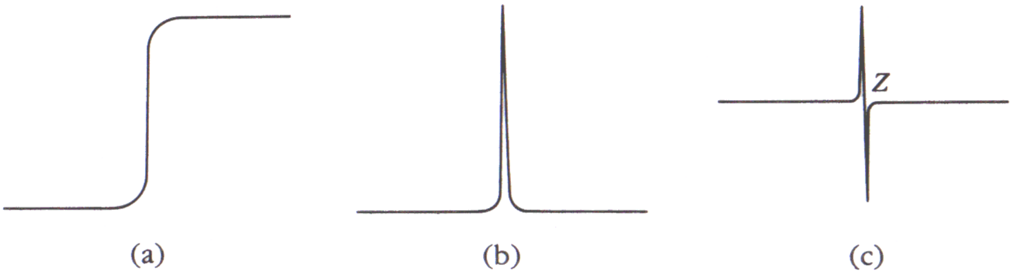


Figure 4.1: An intensity change in (a) produces a peak in (b) its first derivative, and a zero-crossing Z in (c) its second derivative. Replicated from Marr [69].

4.1 Edge detection review

Within an image, an edge can most simply be defined as a change of intensity that takes place over a number of pixels. To detect these edges, algorithms generally calculate a derivative of the intensity change. Most of these algorithms utilise the first derivative of the image intensity, otherwise known as the intensity gradient of the input image, in which peaks correspond to the edges to be detected. For a one-dimensional signal $I(x)$, the first derivative $I'(x)$ is often calculated using:

$$I'(x) = -\frac{1}{2} \cdot I(x-1) + 0 \cdot I(x) + \frac{1}{2} \cdot I(x+1) \tag{4.1}$$

Some other edge detection algorithms use the second derivative of the image intensity, which is a measure of the rate of change of the image intensity gradient. Detection of zero-crossings in the second derivative determines the location of local maxima in the gradient [69]. Again, for a one-dimensional signal $I(x)$, the second derivative $I''(x)$ is often calculated using:

$$I''(x) = 1 \cdot I(x-1) - 2 \cdot I(x) + 1 \cdot I(x+1) \tag{4.2}$$

A one-dimensional example showing the first and second derivatives of an edge is shown in Figure 4.1.

Traditionally edge detection in a two-dimensional image is performed using a first derivative operator. Examples of these are the simple Roberts Cross operator, and the Prewitt [84] and Sobel [98] operators. These work to some extent for simple images with sharp edges and minimal amounts of noise, but for noisy and complicated images they generally produce false and incomplete edges. An alternative is to use a second derivative operator such as the Laplacian, or the Marr-Hildreth edge algorithm [68]. This algorithm searches for zero crossings in an image that is convolved with a Laplacian of Gaussian $\nabla^2 G$ function, otherwise known as the Mexican-hat operator:

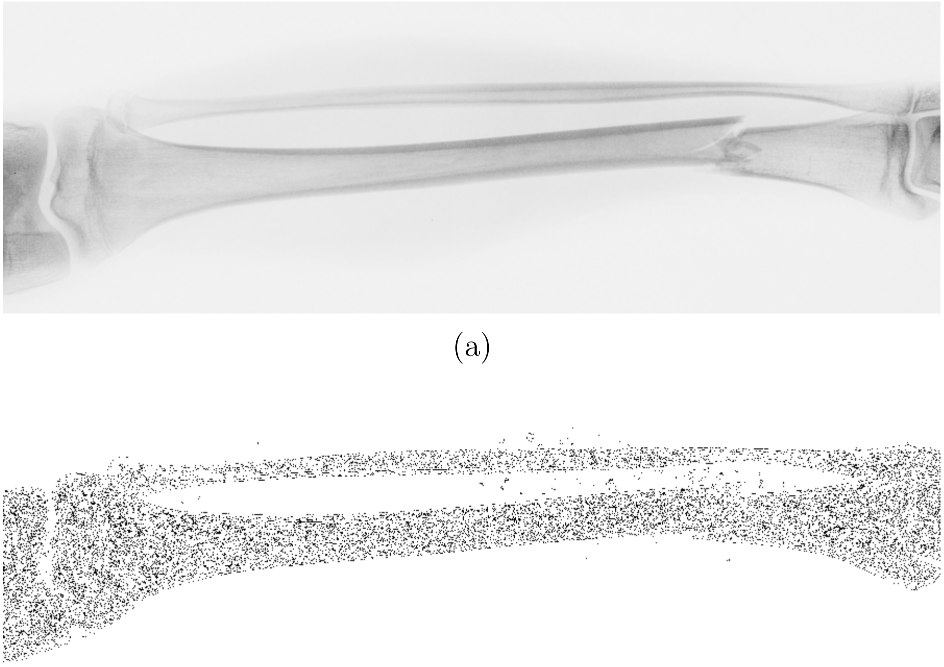
$$\nabla^2 G(r) = -\frac{1}{\pi\sigma^4}\left(1 - \frac{r^2}{2\sigma^2}\right)e^{-\frac{r^2}{2\sigma^2}} \tag{4.3}$$

The Marr-Hildreth algorithm was suggested because:

- The operator is symmetric (unlike the first derivative operators) so edges are found in all orientations
- The zero-crossings of the second derivative are easier to locate than the maxima of the first derivative and always form closed contours
- There are receptor fields in the eyes of animals that behave like this operator [69]

Unfortunately, the influence of noise is considerable due to the second derivative, and the operator marks many locations that are not actually edges.

Better edge detection results are obtained by using the Canny edge detector [25]. The first step of this four step procedure involves finding the maxima of the partial derivative of the image function I in the direction orthogonal to the edges, and smoothing the signal along the edge direction. Then non-maximal suppression is performed, connected sets of edge points are formed into lists, and a threshold is applied to the resulting image to eliminate insignificant edges using threshold hysteresis. An example of the edges detected with the Canny edge detector is shown in Figure 4.2. This image was produced using the standard Matlab [72] edge detection commands, and shows that the edges found by the algorithm are not necessarily a good representation of the objects within the image, because much of what is detected is the inherent noise present in the image. The aim of this chapter is to produce a better representation of the edges within a long-bone x-ray image than those shown in Figure 4.2b using the Canny edge detector.



(a)

(b)

Figure 4.2: Example results produced by the Canny edge detection algorithm. (a) The original image, and (b) the black lines are the edges detected by the algorithm.

4.2 Scale space

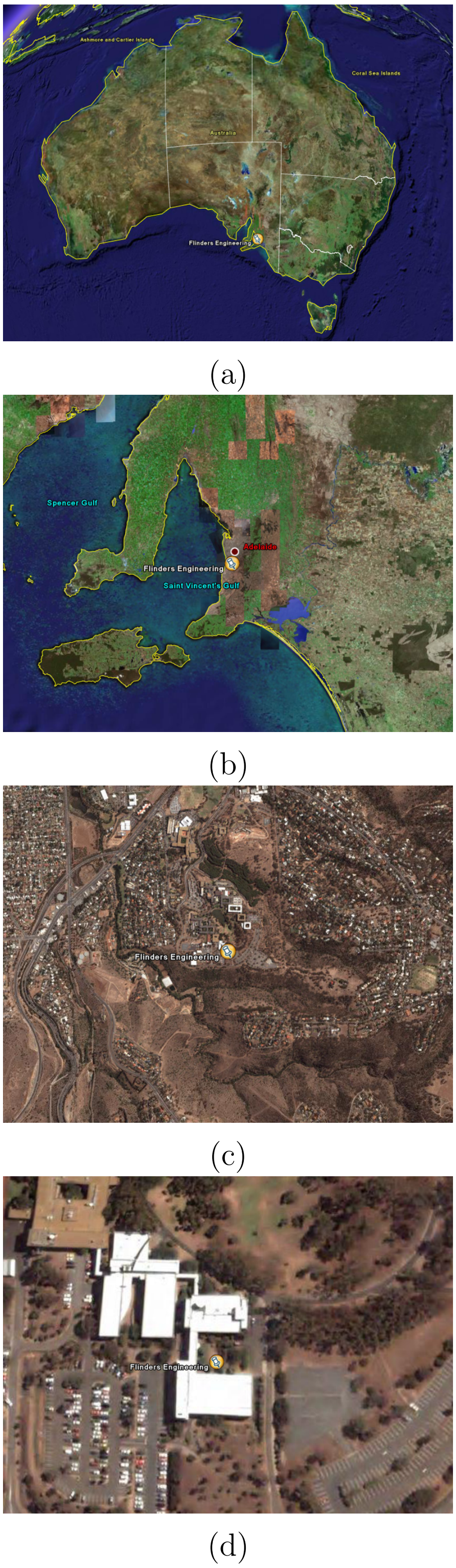
Since the structures within an x-ray image were not easily computed, a scale-space approach to the edge detection problem was used. Real world objects appear different depending on the scale of observation, and are only meaningful at a certain range of scales. For example, the concept of a country is appropriate at the scale of thousands of kilometres (Figure 4.3a), while the concept of a building is more appropriate at the scale of metres (Figure 4.3d). Similarly it is more relevant to talk of a long-bone at the scale of centimetres (Figure 4.4a), while fractures may be better examined at the millimetre scale (Figure 4.4c). Therefore when describing an object, the choice of scale at which it is observed is very important. The need for multi-scale methods is especially important when attempting to automatically analyse images and obtain information from them. The type of information obtained will be determined by the size of the actual structure of interest, as well as the size of the operator used [60]. Of course the task is made easier if the appropriate scales are known before the analysis is performed, although in many cases this is not possible. If no prior information is available, then the only reasonable approach is to represent the data at multiple scales.

To achieve this, a series of images in which fine scale structures are suppressed, are derived from the original image (Figure 4.5). Of crucial importance is the principle of causality, that structures at coarse scales are simplifications of the structures at finer scales rather than artifacts introduced by the fine scale suppression method [60].

To compute features within an image $I(x,y)$, $(x,y) \in R$, scale-space theory proposes that the image be replaced by a smoothed version $I(x,y,t)$. The degree of image smoothing allowed is determined by the scale parameter t, where $t = 0$ corresponds to the original image and $I(x,y,t)$ is the smoothed image at scale t. After smoothing, the points with high gradient intensity contain the most significant global information [69]. These points can be identified as the points where $\triangle I$ crosses zero [69], or the points where $\bigtriangledown I$ is maximal on the gradient line (the basis of the Canny edge detector) [25]. Depending on the scale chosen, local disturbing detail (due to noise and texture) can be removed and the features at that scale can be extracted.

4.3 Smoothing

Prior to obtaining images at different scales, it is necessary to examine what type of smoothing should be performed. However, before then some terms must be examined. The first of these is called locality, and is related to the problem of occlusion, in which different objects within an image partly hide each other. Locality requires that the analysis be as local as possible, since mixing these objects should be avoided. For example, a wide convolution would mix foreground and background objects, removing all depth information. The second term is iteration. From the viewpoint of maintaining locality, it is better to iterate a very local smoothing operator than directly apply the operator at a large scale [43]. The final term is invariance. In image analysis, invariance requirements are important because the objects must be recognisable under varying conditions of illumination (contrast invariance from the Mathematical Morphology theory [95]) and from different points of view (projective invariance). Invariance to contrast implies that the smoothing operator $T_t : u \rightarrow u(t) = T_t u$ commutes with any contrast change $u \rightarrow g(u)$, where g is any real non-decreasing function. The combination of the terms smoothing, locality and iteration implies the use of parabolic



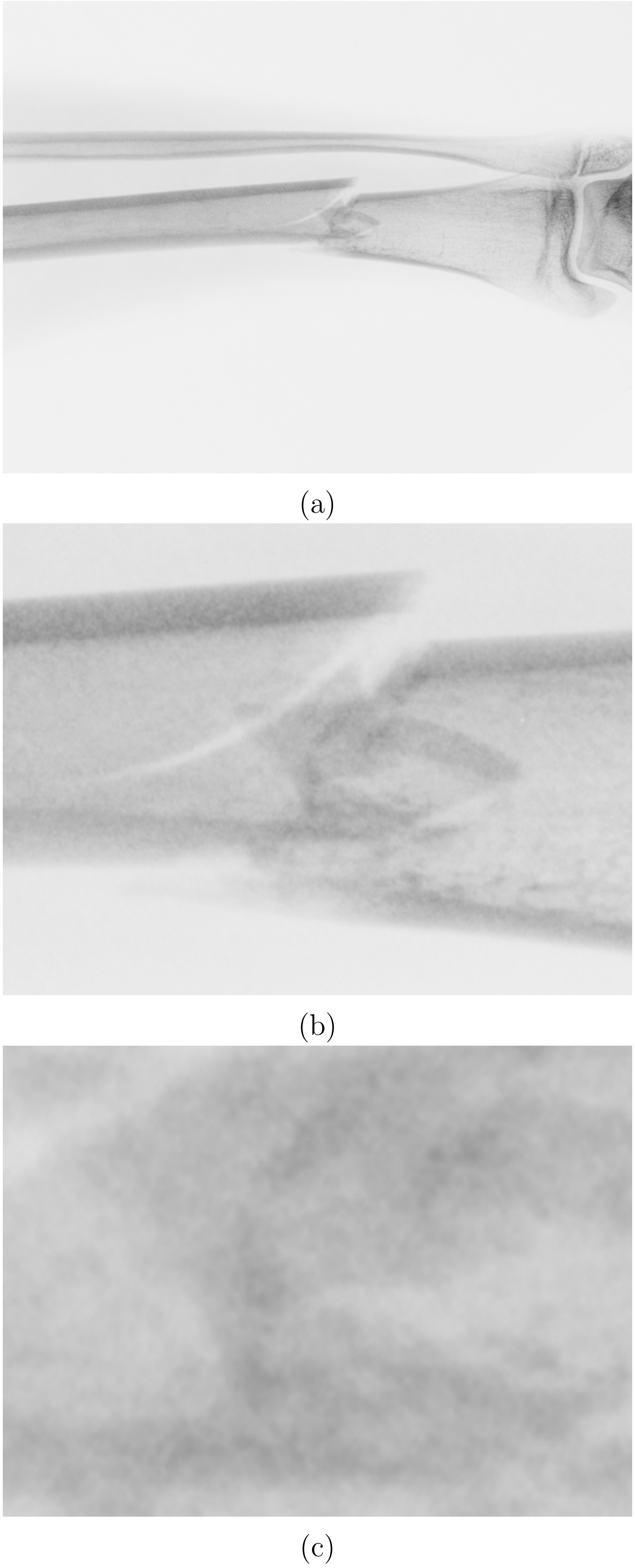
(a)

(b)

(c)

(d)

Figure 4.3: (a-d) Scale space example showing satellite images of the same building at four scales. The concept of a country is only meaningful at the scale shown in (a), while the concept of a building is only meaningful at the scale shown in (d). Replicated from Google Earth [1].



(a)

(b)

(c)

Figure 4.4: Scale space example showing a long-bone x-ray at three scales. The concept of the bone is most meaningful at the scale shown in (a), while the fracture is more meaningful at the scales shown in (b) and (c).

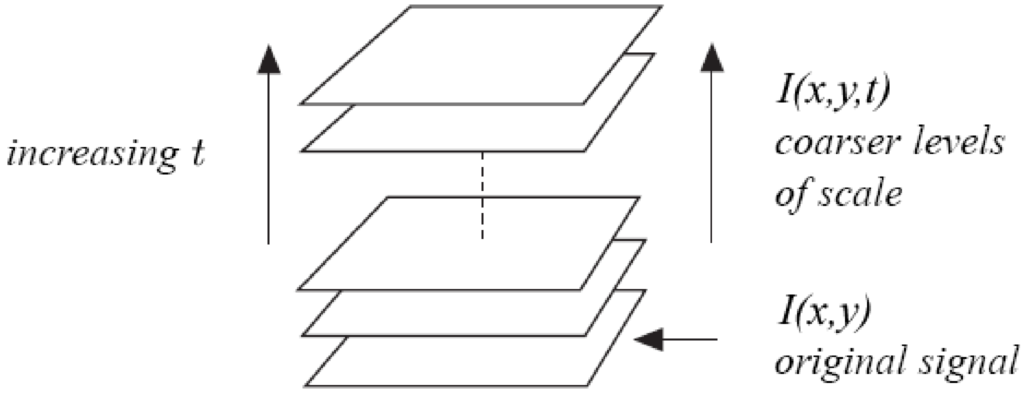


Figure 4.5: The derived multi-scale representation of an image. Replicated from Lindeberg [60].

partial differential equations [43], which will be discussed in the following sections.

4.3.1 Linear smoothing

Image details such as boundaries, corners and other singularities cannot be computed prior to smoothing, since they are derivatives of a non-smooth function. As previously stated, because the image is multi-scale, the smoothing must also be multi-scale [43]. If the smoothing function is required to be linear (and isotropic), then the only suitable candidate is convolution with a Gaussian function:

$$G(x,y,t) = \frac{1}{2\pi t} e^{-\left(x^2+y^2\right)/2t} \tag{4.4}$$

$$I(x,y,t) = G(x,y,t) * I(x,y) \tag{4.5}$$

This was noted by Koenderink [56], to be equivalent to the solution of the heat equation:

$$\frac{\partial I}{\partial t} = \Delta I \tag{4.6}$$

where $\Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ is the Laplacian of $I(x,y,t)$ with respect to the spatial coordinates (x,y). From a locality perspective, the heat equation is the worst possible choice for the task of smoothing because it produces a wide ranging mixture of grey levels. In addition, iterating the convolution of small Gaussian kernels is identical to directly convolving with a single large Gaussian, thereby also contradicting the iteration statement. Since the heat equation is isotropic, smoothing occurs across the boundaries within the image, resulting in blurring of the features to be detected and the loss of pertinent information. Figure 4.6 shows an example of the smoothing produced by the heat equation. For a small number of iterations (such as 10, as shown in 4.6b), the smoothing successfully removes the irrelevant edges, but for a larger number of iterations (such as 50, as shown in 4.6c), the edge definition is lost. Eliminating smoothing across edges prevents the loss of edge information. Fortunately, there are many non-linear ways to smooth an image.

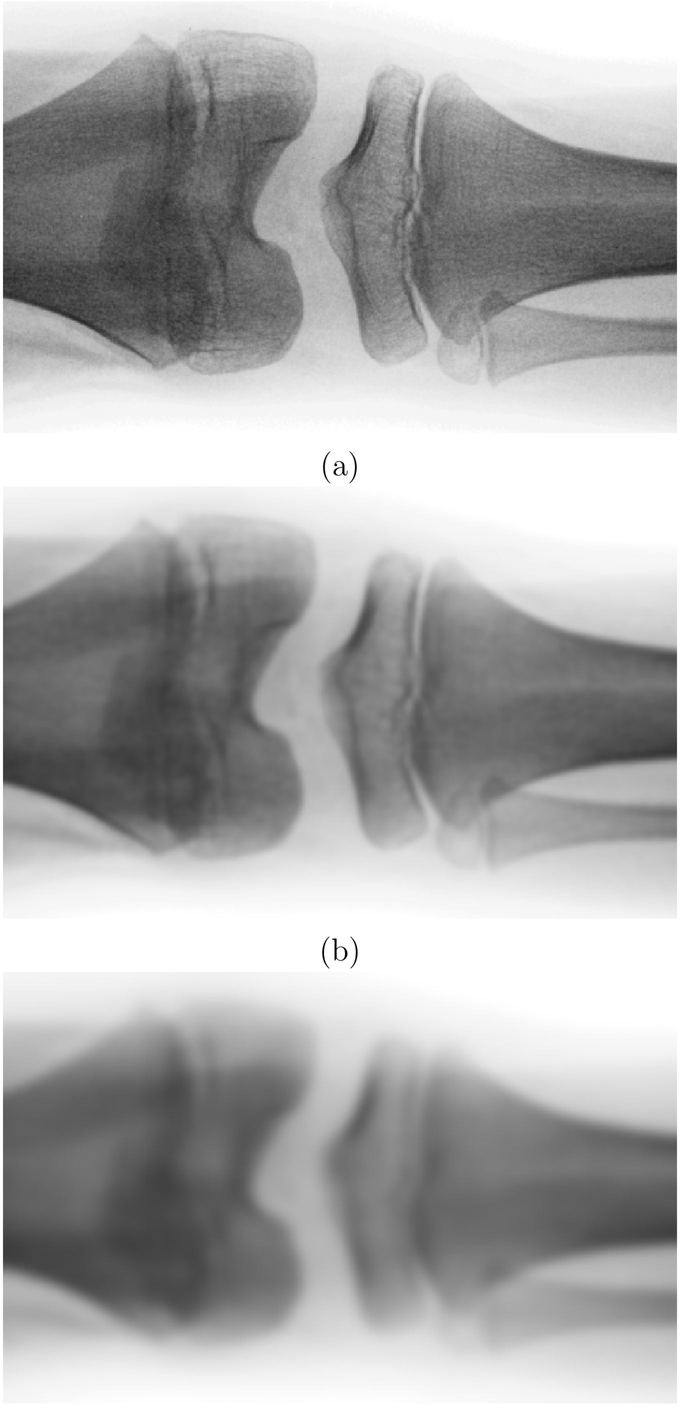
4.3.2 Anisotropic diffusion

Perona and Malik [83] proposed a non-linear partial differential equation (PDE) model as an improvement to the classical linear scale-space smoothing. In their model the edge detection is included within the filtering step, allowing an interaction between scales from the start of the diffusion. Accordingly only the regions requiring smoothing (irrelevant, homogeneous regions) are smoothed, while the remaining regions (the boundaries) are enhanced, with no interaction between these regions. As a result, the smoothing should look like the heat equation when $|\bigtriangledown I|$ is small, and an inverse heat equation when $|\bigtriangledown I|$ is large. In their model, the heat equation is replaced by the following non-linear equation:

$$\frac{\partial I}{\partial t} = div\left(\mathcal{C}\left(x,y,t\right)\left|\bigtriangledown I\right|\right) \tag{4.7}$$

where $\bigtriangledown I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$ is the spatial gradient of I, $|\bigtriangledown I|$ is the Euclidean norm of $\bigtriangledown I$, and $div = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}$ is the divergence operator. When the conduction coefficient $\mathcal{C}\left(x,y,t\right)$ is a constant, Equation 4.7 reduces to the isotropic heat equation. Adjusting the conduction coefficient based on the location of the boundaries at a particular scale t encourages smoothing within regions, rather than across boundaries. Although the location of the boundaries at each scale is not known in advance, Perona and Malik found that the best estimate of the edge positions is a function of the gradient of the intensity:

$$\mathcal{C}\left(x,y,t\right) = g\left(\left|\bigtriangledown I\left(x,y,t\right)\right|\right) \tag{4.8}$$



(a)

(b)

(c)

Figure 4.6: The Gaussian scale space. Results of the linear smoothing of the input image (a) using the heat equation after (b) 10 iterations, and (c) 50 iterations.

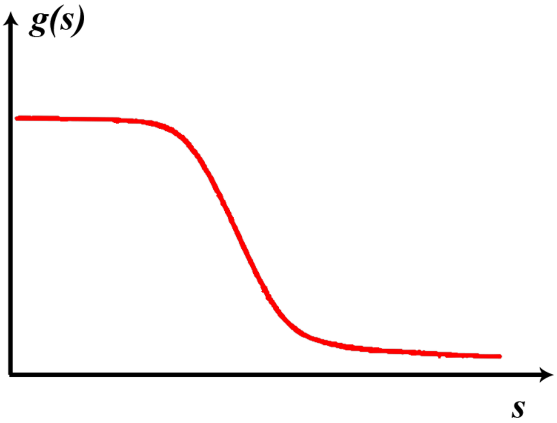


Figure 4.7: The qualitative shape of $g(s)$*. Adapted from Perona and Malik [83].*

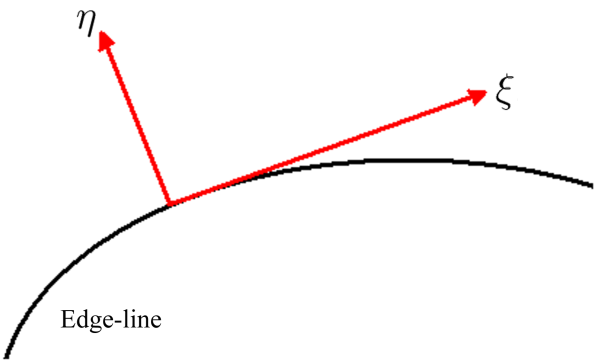


Figure 4.8: The local orthogonal coordinate system.

The function g has to be a non-negative monotonically decreasing function (Figure 4.7) with $g(0) = 1$, so that the diffusion takes place mainly in the interior of regions, and does not affect the boundaries where the magnitude of the gradient is large.

While the Perona Malik equation is undoubtedly an improvement over the heat equation, it still produces some diffusion in the direction of the gradient, rather than pure diffusion in the direction orthogonal to the gradient. This can be demonstrated by constructing a local orthogonal coordinate system (ξ, η) at a point in the image (Figure 4.8), such that the η axis is parallel to the direction of the gradient and the ξ axis is perpendicular to it. The unit vectors in these directions are:

$$\xi = (cos\beta, sin\beta) = \frac{1}{|\triangledown I|}(-I_y, I_x) \tag{4.9}$$

$$\eta = (sin\beta, -cos\beta) = \frac{1}{|\triangledown I|}(I_x, I_y) = \frac{1}{|\triangledown I|}\triangledown I \tag{4.10}$$

So the directional derivatives in the directions of ξ and η are therefore:

$$I_{\xi} = \bigtriangledown I \cdot \xi \qquad I_{\eta} = \bigtriangledown I \cdot \eta \tag{4.11}$$

Then the diffusion terms in the direction orthogonal to the gradient, and in the direction of the gradient are:

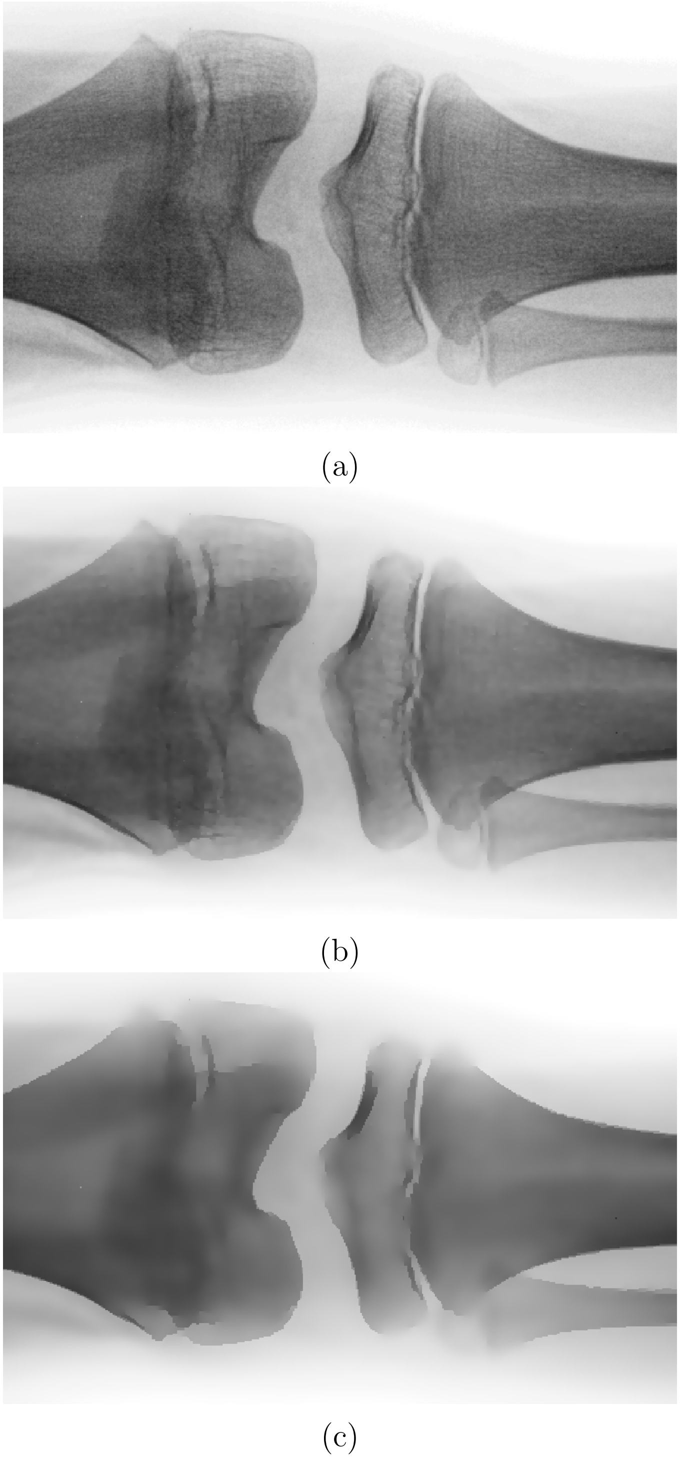
$$I_{\xi\xi} = \bigtriangledown^2 I\left(\xi,\xi\right) \qquad I_{\eta\eta} = \bigtriangledown^2 I\left(\eta,\eta\right) = \frac{\bigtriangledown^2 I\left(\bigtriangledown I, \bigtriangledown I\right)}{\left|\bigtriangledown I\right|^2} \tag{4.12}$$

If the substitution $\hat{g}\left(s\right) = sg\left(s\right)$ is used, then the diffusion can be rewritten as:

$$\begin{aligned}\frac{\partial I}{\partial t} &= div\left(g\left(\left|\bigtriangledown I\right|\right)\left|\bigtriangledown I\right|\right) \\ &= g\left(\left|\bigtriangledown I\right|\right) \triangle I + g\prime\left(\left|\bigtriangledown I\right|\right)\frac{\bigtriangledown^2 I\left(\bigtriangledown I, \bigtriangledown I\right)}{\left|\bigtriangledown I\right|} \\ &= g\left(\left|\bigtriangledown I\right|\right) I_{\xi\xi} + \hat{g}\prime\left(\left|\bigtriangledown I\right|\right) I_{\eta\eta} \qquad (4.13)\end{aligned}$$

In this form, the first term is always a diffusion along the edges and the second term is the diffusion perpendicular to the edges. If $\hat{g}\prime\left(\left|\bigtriangledown I\right|\right) > 0$ then the equation produces diffusion in both directions. If $\hat{g}\prime\left(\left|\bigtriangledown I\right|\right) = 0$ then there is diffusion exactly in the direction orthogonal to the gradient. Finally, if $\hat{g}\prime\left(\left|\bigtriangledown I\right|\right) < 0$ then the equation produces a local inversion of the heat equation. An example of the anisotropic diffusion, using $g\left(\bigtriangledown I\right) = e^{-\left(\frac{\|\bigtriangledown I\|}{K}\right)^2}$ and $K = 10$ as suggested by Perona and Malik [83], is shown in Figure 4.9.

The Perona Malik equation has several serious practical and theoretical difficulties that are solved in other PDE implementations. Firstly, in the case of an image containing noise, large oscillations of the gradient $\bigtriangledown I$ are produced, so the conditional smoothing produced by Equation 4.7 will not produce good results, since all these edges will be retained [26]. Perona and Malik considered this, and proposed to eliminate this problem by first smoothing the image with a low pass filter. Although practically this works well, it introduces another parameter, the variance of the Gaussian with which the preliminary smoothing should be performed. Essentially their theory adopts what it was initially attempting to avoid, the non-adaptive isotropic filtering that corrupts edge information. Secondly, to ensure existence and uniqueness of the solution to Equa-



(a)

(b)

(c)

Figure 4.9: Results of the non-linear smoothing of the input image (a) using the Perona Malik anisotropic diffusion after (b) 10 iterations, and (c) 50 iterations.

tion 4.7, g must be chosen such that $sg(s)$ is non-decreasing [26]. If this condition is not met and $sg(s)$ is non-increasing, then for some g a non-deterministic and unstable process can result, so that similar images will produce divergent solutions and therefore different edges. While it is possible to carefully choose g that is quickly decreasing, such as $g(s) = e^{-s}$, more robust and well-posed solutions exist.

4.3.3 Mean curvature motion

Any non-linear diffusion can be used as an image analysis model, so a large number of non-linear parabolic PDEs have been both proposed and tested over the last few years. In their paper [10], Alvarez, Lions and Morel propose and study a class of non-linear parabolic equations for image processing, based on the Perona Malik equation:

$$\begin{aligned} \frac{\partial I}{\partial t} &= |\bigtriangledown I|\, curv(I) \qquad (4.14) \\ &= |\bigtriangledown I|\, div\left(\frac{\bigtriangledown I}{|\bigtriangledown I|}\right) \end{aligned}$$

where $curv(I) = \kappa$ is the curvature, a measure of the deviation of the curve away from the tangent. Using Equation 4.13 with $g(s) = \frac{1}{s}$ and Equation 4.12 for the diffusion term in the direction orthogonal to the gradient, produces:

$$\begin{aligned} \frac{\partial I}{\partial t} &= |\bigtriangledown I|\left(\frac{\triangle I}{|\bigtriangledown I|} - \frac{1}{|\bigtriangledown I|^2}\frac{\bigtriangledown^2 I(\bigtriangledown I, \bigtriangledown I)}{|\bigtriangledown I|}\right) \\ &= \triangle I - \frac{\bigtriangledown^2 I(\bigtriangledown I, \bigtriangledown I)}{|\bigtriangledown I|^2} \qquad (4.15) \\ &= I_{\xi\xi} \qquad (4.16) \end{aligned}$$

As demonstrated by Equation 4.16, the smoothing produced by Equation 4.14 is only in the direction orthogonal to the gradient, with no smoothing in the direction of the gradient. Comparing Equation 4.6 to Equation 4.15 shows that the first term, the Laplacian, is the same, while the second inhibits the smoothing in the direction of the gradient. The geometrical interpretation of Equation 4.14 is that the level sets of the solution move in the normal direction with a speed proportional to their mean curvature

[10], an effect termed mean curvature motion (MCM). This property indicates that Equation 4.14 satisfies the main morphological axiom, grey-scale invariance, implying that the mean curvature motion is a morphological operator. The main difference between the model in Equation 4.14 and the Perona Malik equation in 4.7, is that the smoothing is scaled by $\frac{1}{|\bigtriangledown I|}$ such that it is lowered when $|\bigtriangledown I|$ is large. An example of the smoothing produced is shown in Figure 4.10.

4.3.4 Affine and morphological scale space

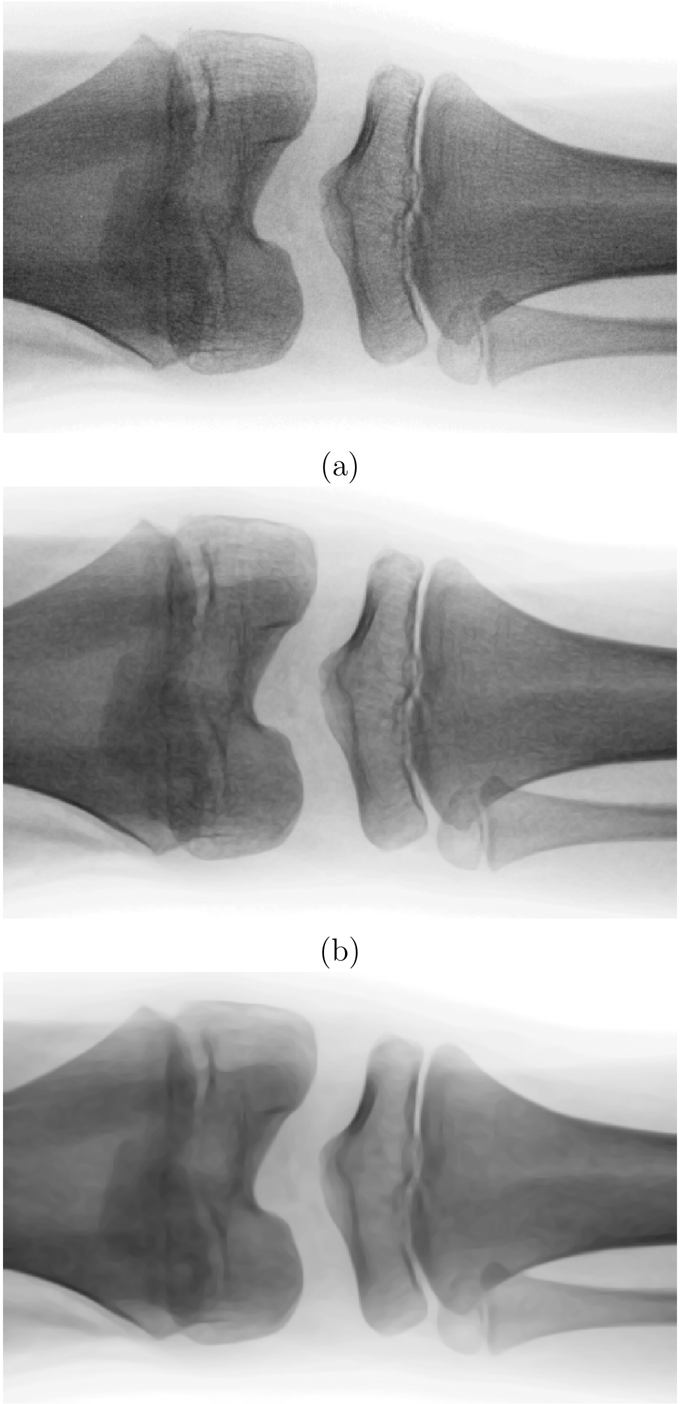
Not all of the proposed PDE models meet the short list of essential invariance requirements for image analysis, including properties like Euclidean invariance (translation and rotation invariance). It was proven by Chen et al. [29] and Alvarez, et al. [10] that to meet all invariance requirements, all multi-scale image analysis techniques should have the form of a curvature motion:

$$\frac{\partial I}{\partial t} = |\bigtriangledown I|\, F\left(div\left(\frac{\bigtriangledown I}{|\bigtriangledown I|}\right)\right) \tag{4.17}$$

where F is increasing with respect to its argument. The specific case $F(s) = s^{\frac{1}{3}}$, called the affine and morphological scale space (AMSS), plays a prominent role:

$$\begin{aligned} \frac{\partial I}{\partial t} &= |\bigtriangledown I|\left(div\left(\frac{\bigtriangledown I}{|\bigtriangledown I|}\right)\right)^{\frac{1}{3}} \\ &= |\bigtriangledown I|\,(curv(I))^{\frac{1}{3}} \end{aligned} \tag{4.18}$$

This case is chosen because it diffuses purely in the direction orthogonal to the gradient, plus it is the only case that produces an additional invariance termed affine invariance. A curve is said to be affine invariant if the coordinate system in which it is represented can change without affecting the relative geometry of the curve. As a result the geometry of the curve remains consistent when the curve is translated, scaled or rotated. This means that the analysis of planar shapes is not dependent on their orientation or location in three-dimensional space [9]. The AMSS therefore retains the noise-elimination properties of the heat equation, but also has the shape-preserving properties of morphological operators. The example in Figure 4.11 shows the same



(a)

(b)

(c)

Figure 4.10: Results of the non-linear smoothing of the input image (a) using the mean curvature motion (MCM) after (b) 10 iterations, and (c) 50 iterations.

image smoothed using the AMSS. Visually there is little difference between the MCM and AMSS smoothing, while both are very different to the anisotropic diffusion and heat equation.

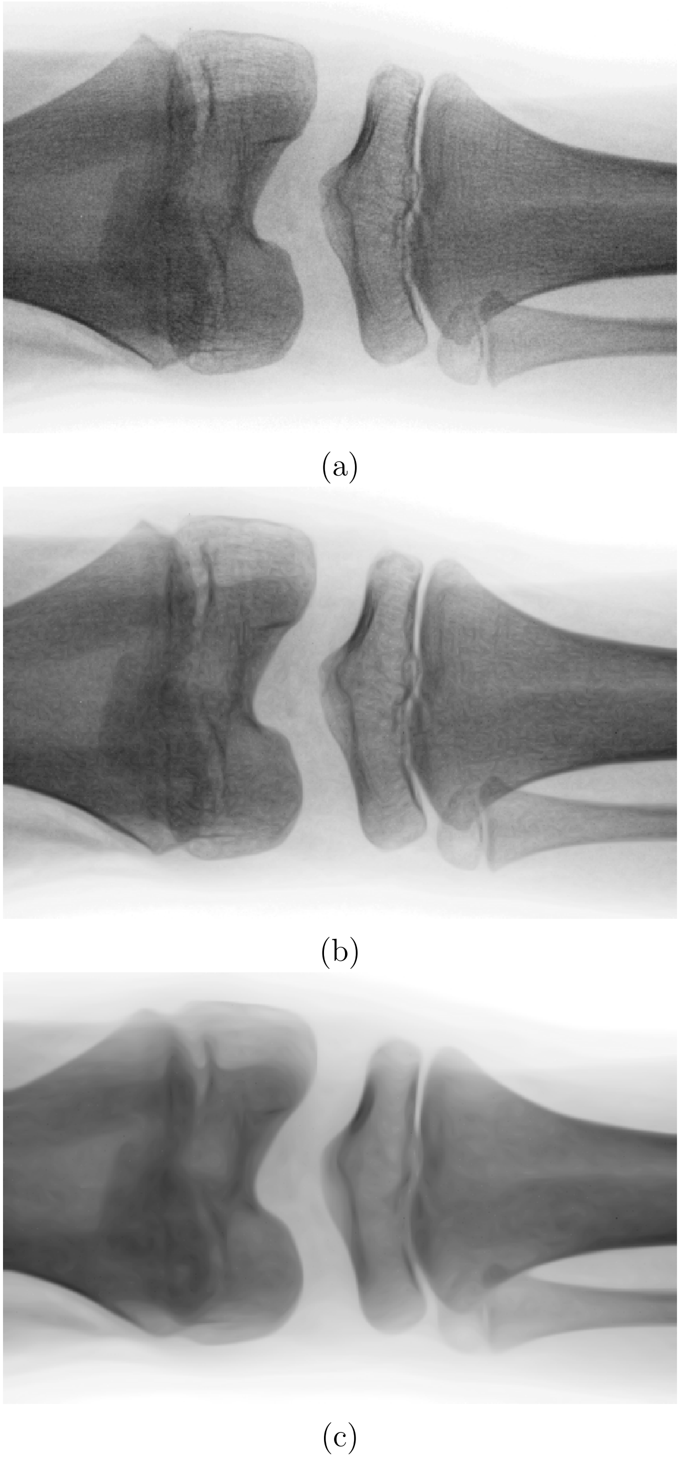
4.4 Choosing the best smoothing method

Section 4.2 established that because the x-ray images used were multi-scale, a scale-space approach to the edge detection problem was required. It also suggested that to obtain the image $I\left(x,y,t\right)$ at scale t, some type of image smoothing was required. Section 4.3 discussed four popular smoothing algorithms that could be chosen for smoothing x-ray images. Linear smoothing by convolution with a Gaussian was not appropriate because the edge information was lost by smoothing across boundaries, while the Perona Malik anisotropic diffusion was not suitable because oscillations occured in the output unless linear smoothing was first applied. The MCM produced visually impressive results, but was not fully affine invariant, unlike the AMSS. Computationally, there were few differences between the MCM and AMSS, so due to its well-posed solution, robustness, and full invariance, the AMSS was the smoothing method of choice for long-bone x-ray images. Indeed, it was unique in its ability to satisfy all classical invariance properties, as well as projection invariance.

4.5 Smoothing algorithm implementation

The nonlinear partial differential equations for the AMSS could not be solved analytically, so a discrete numerical method based on that by Froment, Guichard and Moisan [43] was implemented. Equation 4.18 could be simplified to:

$$\begin{aligned}
\frac{\partial I}{\partial t} &= \left|\bigtriangledown I\right|\left(curv\left(I\right)\right)^{\frac{1}{3}} \\
&= \left(\left|\bigtriangledown I\right| curv\left(I\right)\right)^{\frac{1}{3}} \\
&= \left(\left|\bigtriangledown I\right|^{2} I_{\xi\xi}\right)^{\frac{1}{3}} \qquad (4.19)\\
&= \left|\bigtriangledown I\right|^{\frac{2}{3}} I_{\xi\xi}^{\frac{1}{3}}
\end{aligned}$$



(a)

(b)

(c)

Figure 4.11: Results of the non-linear smoothing of the input image (a) using the affine and morphological scale space (AMSS) after (b) 10 iterations, and (c) 50 iterations.

where $|\bigtriangledown I|^{\frac{2}{3}}$ was the speed of the diffusion in the direction ξ. The value of $I_{\xi\xi}$ at the point (i,j) was then calculated using:

$$\begin{aligned}\left(|\bigtriangledown I|^2 I_{\xi\xi}\right)_{i,j} &= \frac{1}{\triangle x^2}\left(-4\lambda_0 I_{i,j} + \lambda_1\left(I_{i+1,j} + I_{i-1,j}\right) + \lambda_2\left(I_{i,j+1} + I_{i,j-1}\right) + \right.\\ &+ \left.\lambda_3\left(I_{i-1,j-1} + I_{i+1,j+1}\right) + \lambda_4\left(I_{i-1,j+1} + I_{i+1,j-1}\right)\right)\end{aligned}$$

where the spatial increment $\triangle x$ was the same in both the x and y directions, and the values of λ_0 to λ_4 were chosen by Guichard and Morel [43] to ensure consistency[1]. For each image pixel the algorithm estimated the gradient magnitude $|\bigtriangledown I(x,y,t)|$ and direction η, and therefore the orthogonal direction ξ using the nine points of a 3 x 3 stencil. A single iteration of the smoothing described by Equation 4.19 was implemented using the explicit finite difference scheme:

$$I(x,y)_{n+1} = I(x,y)_n + (\triangle t)\left(|\bigtriangledown I|_n^2 I_{\xi\xi}(x,y)_n\right) \tag{4.20}$$

where $\frac{\triangle t}{\triangle x^2}$ was the selected scale step, which had to be chosen carefully so that it was large enough to minimise the number of iterations, and small enough to retain stability. For the AMSS the maximum value could not be determined theoretically, but in practice had to be 0.1 or smaller [10]. With the inter-pixel spacing set to $\triangle x = \triangle y = 1$, n iterations of Equation 4.20 resulted in smoothing to a scale of $t = \frac{n}{\triangle t}$.

[1] The chosen coefficient values are:

$$\begin{aligned}\lambda_0 &= 0.25\left(2I_x^2 + I_y^2 - I_x I_y\right)\\ \lambda_1 &= 0.5\left(2I_x^2 - I_y^2 - I_x I_y\right)\\ \lambda_2 &= 0.5\left(I_y^2 - I_x I_y\right)\\ \lambda_3 &= 0.25\left(I_y^2 + 3I_x I_y\right)\\ \lambda_4 &= 0.25\left(I_y^2 - I_x I_y\right)\end{aligned}$$

with the first derivatives u_x and u_y, and the gradient $|\bigtriangledown I_{i,j}|$ calculated using:

$$\begin{aligned}(I_x)_{i,j} &= \frac{2\left(I_{i+1,j} - I_{i-1,j}\right) + I_{i+1,j+1} - I_{i-1,j+1} + I_{i+1,j-1} - I_{i-1,j-1}}{4\triangle x}\\ (I_y)_{i,j} &= \frac{2\left(I_{i,j+1} - I_{i,j-1}\right) + I_{i+1,j+1} - I_{i+1,j-1} + I_{i-1,j+1} - I_{i-1,j-1}}{4\triangle x}\\ |\bigtriangledown I_{i,j}| &= \left((I_x)_{i,j}^2 + (I_y)_{i,j}^2\right)^{\frac{1}{2}}\end{aligned}$$

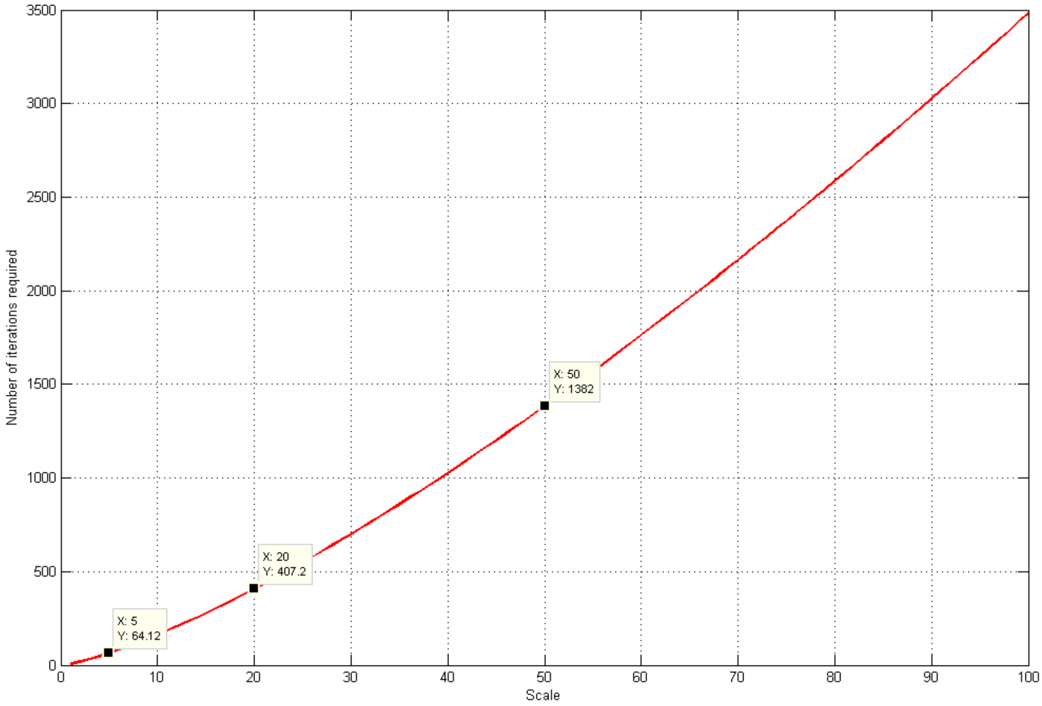


Figure 4.12: The number of iterations n required to reach a chosen scale t. For example, to reach the scale $t = 5$, 64 iterations are required.

It could therefore be shown[2] that for a disk of radius less than or equal to χ pixels to disappear, the number of iterations required was:

$$n = \frac{3}{4}\frac{\chi^{\frac{4}{3}}}{\triangle t} \tag{4.21}$$

Therefore Equation 4.21 could be used to determine the number of iterations of Equation 4.20 that had to be applied to produce an image $I(x,y,t)$ at a particular scale t. Figure 4.12 shows the relationship between the scale t and the number of iterations n of Equation 4.20 required to reach that scale.

So that it was processor and operating system independent, the algorithm described above was implemented in ANSI C. A set of image input and output functions was also created to read and write TIFF images (the image format selected in Section 2.6.1) and also a raw format. For the input and smoothed images TIFF was suitable (provided some quantisation was acceptable), since these images contained values within the range

[2] For the AMSS, the level sets of the solution move in the normal direction with a speed proportional to their mean curvature to the power one third. So the radius $R(t)$ of the disk changes at a rate that is proportional to its current radius, following $\frac{dR(t)}{dt} = -\frac{1}{R(t)^{\frac{1}{3}}}$ to yield $\frac{3}{4}\left(R^{\frac{4}{3}}(0) - R^{\frac{4}{3}}(t)\right) = t$,and a corresponding disappearing time of $t = \frac{3}{4}R^{\frac{4}{3}}$.

$[0, 255]$. However TIFF was not suitable for the gradient magnitude and direction images since these contained values in the range $(-\infty, \infty)$, so a simple raw format was used so that all values could be retained at their full precision. The raw format contained a simple header that specified information such as the image size, followed by the image data written in binary format.

4.6 Scale selection

Regardless of the image analysis model chosen, all scale-space models suffer the problem of determining how much smoothing is required to extract the required information. The multi-scale representation by itself contains no explicit information about what image structures should be regarded as significant, or what scales are appropriate for examining them. Unless it is known which image structures are important, and thus the scales that are important, there is a large amount of data to be interpreted by later stages. Most previous works handled this problem by using algorithms that rely on the information present in a small range of manually chosen scales, or even a single scale [62].

Like most images, the long-bone x-ray images contained information at multiple scales, and a single scale was not suitable for all analyses. For example, when segmenting objects based on their macroscopic structure a large scale t_2 was required. This was because only large features such as the bone edges were of interest, so smaller features like trabeculae should be highly smoothed. In contrast, when examining the texture produced by trabeculae or subtle fractures a small scale t_1 was required. In this case, too large a scale removed the important fracture detail, and made detection problematic.

Rather than examining the difficult problem of automatically selecting scales [62, 27, 61]—which is a significant project unto itself, and is beyond the scope of this thesis—they were fixed at values that produced the best results for this algorithm, and were shown to work well for all of the development images.

4.6.1 Choosing the correct scale

Firstly, it was important to note that the scale chosen was only optimal for an image at a particular resolution. For example, an image at a resolution of 640 x 480 pixels smoothed to a scale $t = 10$, would be smoothed to a greater extent than the same image scanned at a resolution of 2560 x 1920 pixels, also smoothed to the scale $t = 10$. This was because the smoothing scale was based on equation 4.21 in Section 4.5, which stated that a disk of radius χ pixels disappeared after n iterations of the smoothing equation. In images with large resolutions, χ pixels covered a much smaller portion of any given feature. It was for this reason that the image size was standardised to 3600 x 1200 pixels, as explained in Section 2.6.1 on page 29. It was assumed that the bones were approximately the same size within each image.

Secondly, the optimal scale varied for different features within an image, making the scale selection harder. To determine the best two scales t_1 and t_2 for analysis, the six development images were examined at a range of scales from $t = 0$ through to $t = 25$. This range was chosen so that it was large enough to exceed the anticipated values of both t_1 and t_2. The effect of the AMSS smoothing was quantified by examining the standard deviation of a total of eight manually selected 5 x 5 pixel regions throughout each image, at every integer scale. Since the standard deviation is the root mean square deviation of the values from their arithmetic mean, it was suitable for measuring how similar or dissimilar the values in the region became as a result of the smoothing. In each of the six development images, the eight regions chosen for analysis corresponded to similar anatomical locations. These were:

1. The centre of the epiphysis.
2. The centre of the manually identified fracture.
3. The shaft of the long-bone.
4. The cortical wall of the long-bone.
5. The soft tissue on one side of the bone.
6. The soft tissue between the two bones (if present).

7. The soft tissue on the other side of the bone.

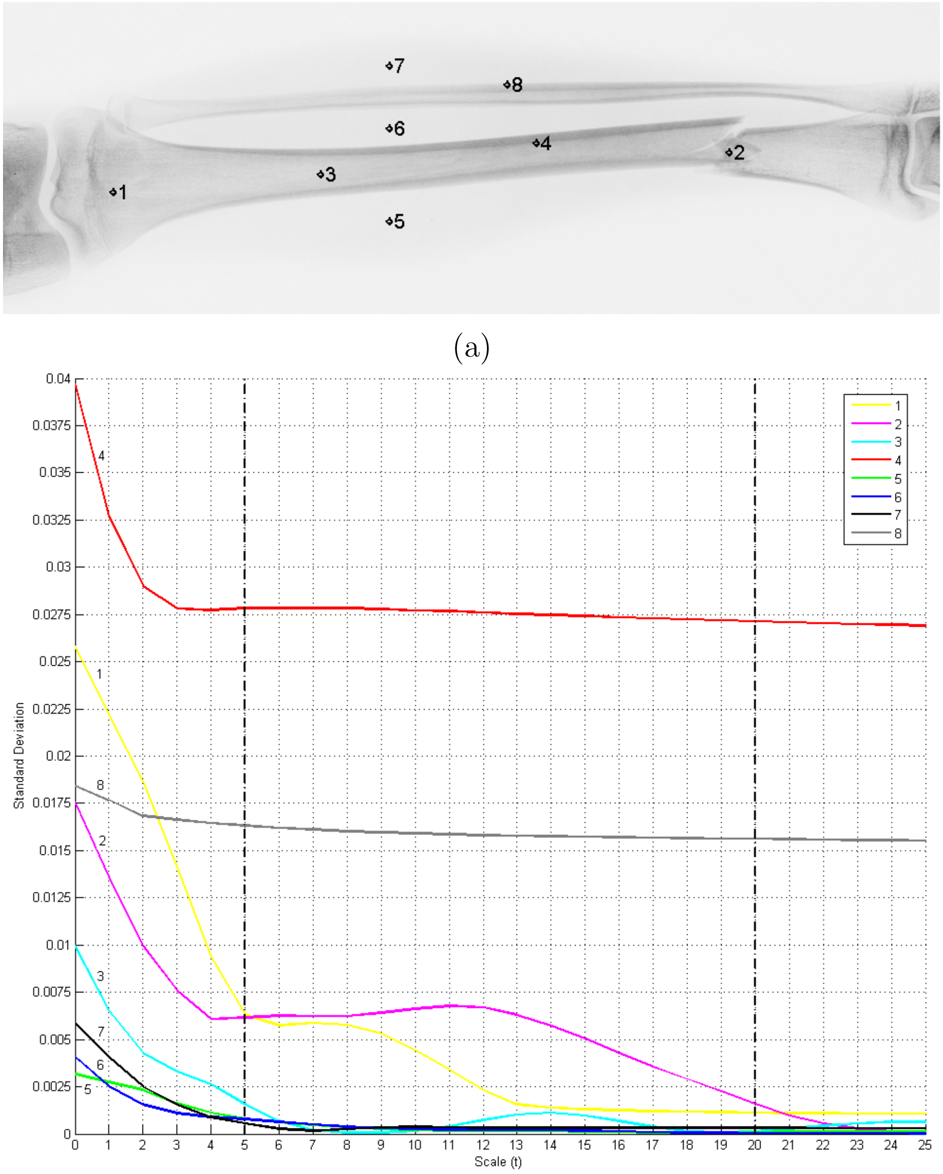
8. The interface between the bone and the soft tissue.

Figure 4.13a shows the eight numbered regions selected in one of the development images, while Figure 4.13b shows the standard deviation of each of the eight regions at all scales between $t = 0$ and $t = 25$.

The results for the other five development images were very similar to those shown in Figure 4.13, and showed some interesting features. As a general rule, the standard deviation of the regions tended to decrease as the scale t increased, indicating that the values within the region were becoming more similar at higher scales. However, none of the lines were monotonically or strictly decreasing, because at some scales the standard deviation actually increased slightly. This occured due to the propagation of the AMSS function (which is explained in greater detail in Section 8.2.1 on page 187), which caused pixels at increasing distances from the region to influence it as the scale increased. In addition, in all eight regions the standard deviation decreased rapidly at low scales, and then tended to plateau as the scale increased further. In many cases, the standard deviation tended towards zero because the regions smoothed such that all pixels had the same intensity.

Not all regions rapidly tended towards a standard deviation of zero. The regions that contained complex textures such as the fracture site (location 2) and the trabeculae in the epiphysis (location 1), as well as those that lay on the boundary between high and low intensity at the cortical wall (location 4) and bone tissue interface (location 8), retained high standard deviations for many scales. These regions did not smooth to a single constant value due to the texture, edge or edges passing through the region. The major difference between the texture locations (1 and 2) and the boundary locations (4 and 8) was that the boundary locations did not tend toward zero within the range of scales that were tested, unlike the texture locations.

To retain the necessary information to segment the bones based on their macroscopic structure, extraneous information about texture could be removed, so that only the significant edge information remained. Regions 4 and 8 contained this significant information, so all other regions could be smoothed until their standard deviations



(a)

(b)

Figure 4.13: To determine how smoothing to a particular scale affected the image, the standard deviation was measured in eight 5 x 5 regions shown in (a), at the integer scales $t = 0$ through $t = 25$. The results in (b) showed that standard deviation tended to decrease with increasing t, and in many cases reached a plateau around the scale $t = 20$. The two chosen scales $t_1 = 5$ and $t_2 = 20$ are marked on the graph.

were close to zero and their pixel values were therefore relatively uniform. Figure 4.13b shows that this occured at a scale of approximately $t = 20$ to $t = 22$.

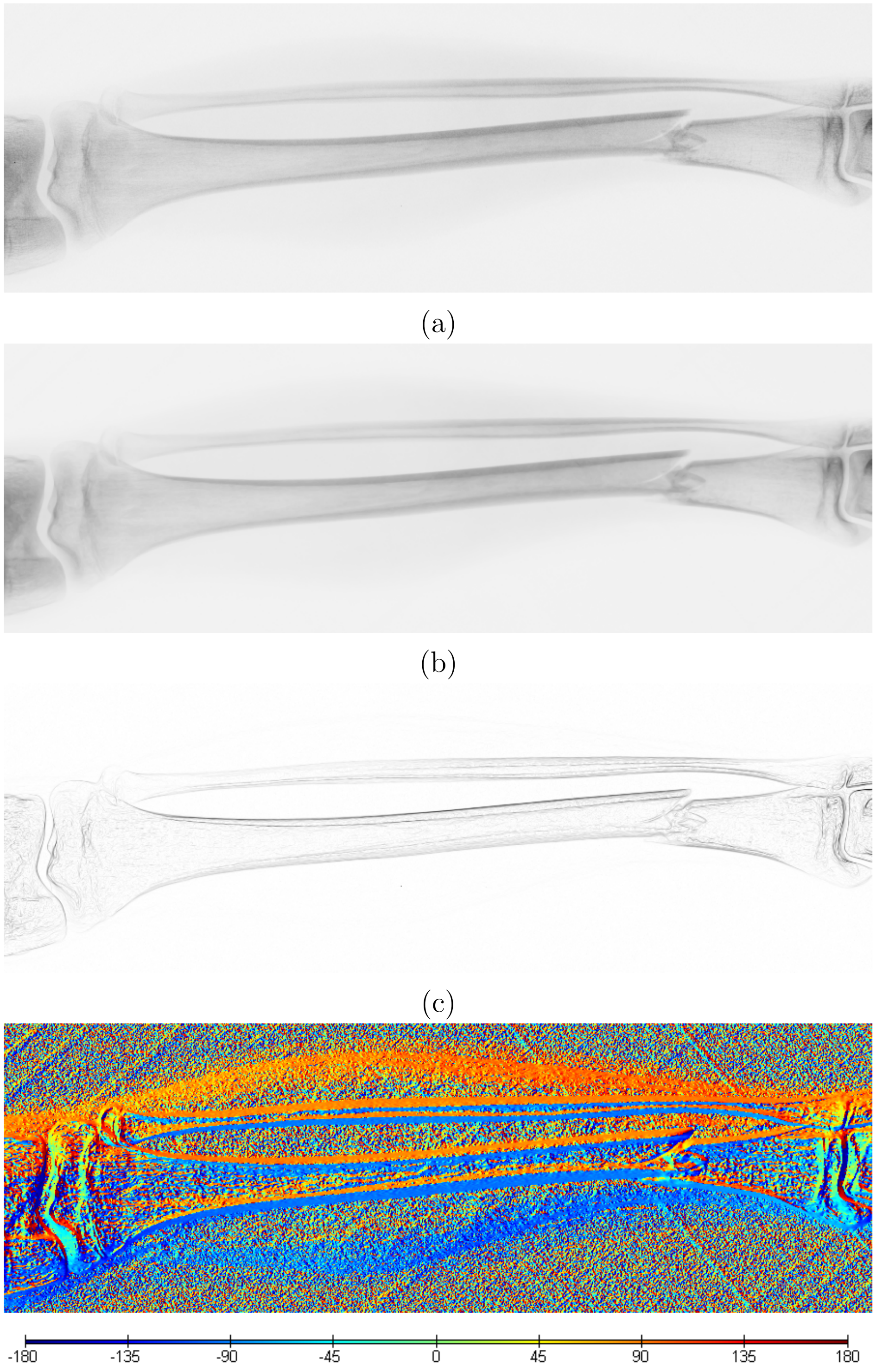
When examining smaller features such as fractures, a lower scale was necessary, but some smoothing of the initial image was still required. The most suitable point was that at which the tissue regions and shaft bone were smoothed, but the fracture region was not overly smoothed. Examining Figure 4.13b shows that this point occured around scale $t = 5$ to $t = 8$, where regions 1 and 2 still had a relatively high standard deviation, while regions 3, 5, 6 and 7 were close to zero. The results from the remaining five development images showed the same patterns and scales.

To minimise the number of iterations of Equation 4.20 that were needed, the lowest scales that provided the best amount of smoothing were selected. Therefore the two optimal scales for small feature (bone fracture detection) and macroscopic feature (bone segmentation) analysis were $t_1 = 5$ and $t_2 = 20$, respectively. These correspond to the two dashed vertical lines marked on Figure 4.13b.

4.7 Smoothing evaluation

The scale of smoothing was dependent on the image resolution, however the resolution was constant for all of the images, so the scales $t_1 = 5$ and $t_2 = 20$ were used for all images. Examples of the results produced on a development image at these two scales are shown in Figures 4.14 and 4.15 respectively. The time taken to perform this calculation was on the order of 20 minutes. The algorithm calculation time is discussed in more detail in Chapter 8.

Applying the algorithm to the six development images showed that the smoothing produced by the AMSS edge extraction algorithm was indeed contained within the boundaries of the bone, rather than smearing the edges like the heat equation. It was also clear when examining the smoothing results, that increasing the scale definitely resulted in greater smoothing of the image. The artificially coloured images showing the gradient direction were very different at the two scales. Many of the random fluctuations in the gradient direction that occured at $t_1 = 5$ did not occur at $t_2 = 20$, due to the higher level of smoothing. In addition, much more detail was removed from



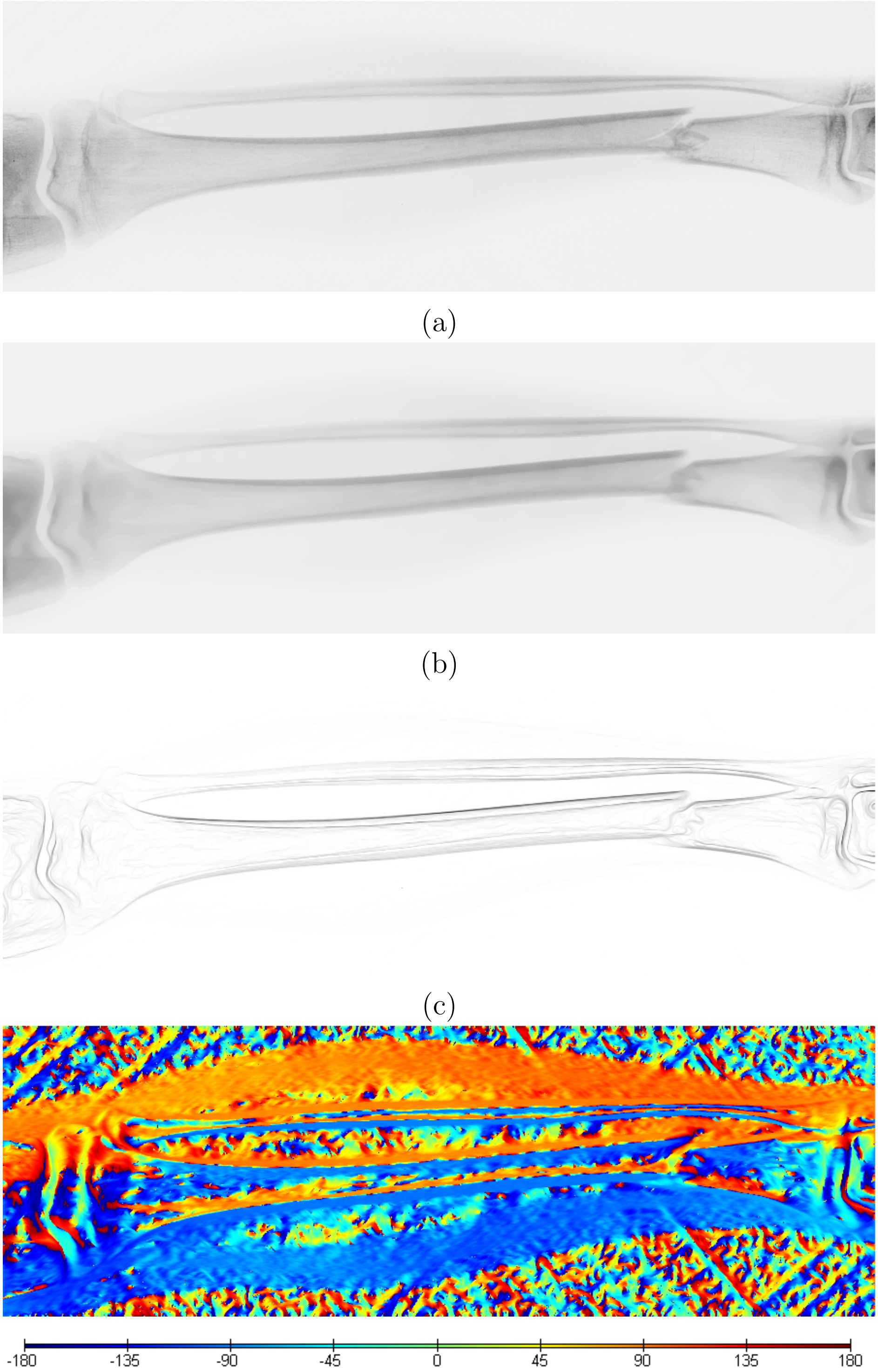
(a)

(b)

(c)

(d)

Figure 4.14: Example results produced by the AMSS algorithm. The original image (a) was smoothed to the scale $t_1 = 5$ in (b). The magnitude (c) and direction (d) of the gradient are also shown. The image in (c) was normalised to show the gradient lines more clearly.



(a)

(b)

(c)

(d)

Figure 4.15: (a) The same image shown in Figure 4.14a, smoothed to the scale $t_2 = 20$ *in (b). The magnitude (c) and direction (d) of the gradient are also shown with the same adjustments as Figure 4.14. This time the smoothing effect was much more pronounced, with the fracture and trabeculae detail removed.*

the gradient magnitude image at the higher scale, as the less pronounced edges were removed. This also meant that the fracture detail was also removed at the higher scale, requiring the lower scale to be used for fracture detection.

Comparing Figures 4.14c and 4.15c with the Canny edge detection shown in Figure 4.2 on page 56 shows that the edges are much clearer in the gradient of the AMSS smoothed images. Not only are the edges clearer, but they correspond to more important features within the image, such as the bone shaft rather than the small changes in texture within the bone. The application of a threshold to the AMSS smoothed image could be used to obtain only the most pertinent edges, or the Canny algorithm could even be applied to the smoothed image to obtain single pixel width edges. In addition, the AMSS smoothing also had the advantage that it was optimised for both fracture detection and global parameter estimation, through the choice of the two scales t_1 and t_2.

Although the AMSS did perform excellent edge detection, better results could possibly be achieved by implementing a scale selection algorithm that adjusted the scale of smoothing on a per image basis, rather than utilising a global pair of scales. As a result, the algorithm could be better tailored to each individual image, rather than assuming that the relationship between the standard deviation and scale was the same for every image. However, as mentioned, this was beyond the scope of this thesis.

4.8 Summary

The fundamental aim of low level image analysis is to locate discontinuities within an image. For example, the detection of edges such as the outline of a long-bone in a diagnostic x-ray image. Early vision research identified that edges could be found everywhere in the image, and that image smoothing is required to identify the true edges. This chapter formalised the smoothing process using the concept of scale space, and the idea of analysing a feature at a particular location and scale. Scale space theory proposed that an image $I(x,y)$ be replaced by its smoothed version $I(x,y,t)$ where the scale t determined the amount of smoothing. If the smoothing was required to be linear, then the only suitable candidate was convolution with a Gaussian function.

Unfortunately this linear smoothing removed edge definition, and was unsuitable for long-bone segmentation and fracture detection. Some non-linear PDE models were examined mathematically, starting with Perona and Malik's anisotropic diffusion. While it produced better results than the Heat equation by retaining edge definition, the Perona Malik equation had several practical and theoretical difficulties that also made it unsuitable for long-bone segmentation and fracture detection. The MCM and AMSS equations were also examined and both found to diffuse purely in the direction orthogonal to the gradient. The AMSS was the only case that was also affine invariant.

The AMSS satisfied all the classical invariance properties, so it was the smoothing method of choice when performing medical image analysis such as long-bone segmentation and fracture detection. The appropriateness of the AMSS for smoothing medical images has not been observed elsewhere in the literature, and was therefore a significant finding. The implementation of the AMSS algorithm using a finite difference scheme was described, along with the relationship between the scale and number of iterations of the smoothing equation. The difficult problem of automatic scale selection was not addressed in this thesis, but the choice of scales for the various analyses was justified by analysing the standard deviation of different regions within the development images, over a range of scales. This allowed the two scales $t_1 = 5$ and $t_2 = 20$ to be chosen for fracture detection and long-bone parameter approximation, respectively. Finally, an example of the smoothing of a typical diagnostic x-ray image of a long-bone was presented, showing the types of images produced at the two selected scales.

Once edges are extracted from an image, they must be interpreted to obtain some meaningful information about the content of the image. To further understand the bones within the x-ray image, the edges are used to determine a series of parameters that best approximate the long-bone shaft. This approximation is examined in the next chapter.