Chapter 5

Long-bone shaft parameter approximation

In many imaging applications, the complete image analysis problem can be reduced to one of extracting features and analysing shapes. Shape representation techniques essentially attempt to produce a non-numerical representation of the shapes present in the image, while still retaining the important characteristics about the shape. As with edge detection, there are many ways to extract features of interest from the image.

As mentioned in Chapter 4, the extraction of primitives such as edges from an image is one of the first stages of analysis, and can in itself be regarded as a type of feature extraction. In order to further understand the contents of the image, segmentation and extraction of higher-level features is often the next important stage. Segmentation is the process of dividing an image into multiple regions based on chosen criteria, with the goal of distinguishing foreground objects from the background. However, the same techniques that are applied to image segmentation can be extended to feature extraction: the problem of detecting and isolating the desired portion of an image.

This chapter describes a novel method of approximating the long-bone shaft using a set of straight line parameters, rather than attempting to apply traditional segmentation methods that often produce poor results with x-ray images. As a result, this method can accurately distinguish the foreground object—the long-bone—from the image background.

5.1 Common segmentation and extraction methods

By 1994 over 1000 segmentation algorithms had been published [111] and this number has more than likely doubled since. Segmentation and feature extraction methods can be categorised into two main approaches: clustering-based algorithms and global segmentation algorithms. While there are many methods of segmenting images that fit into these categories, many are too simple to solve most imaging problems. This is especially true in images where there is poor contrast between the foreground and background objects, such as the x-ray images used within this thesis.

Clustering-based algorithms

The oldest and simplest of the segmentation techniques is thresholding an image [81] such that each pixel $I(x, y)$ is assigned to either object I_1 or background I_2 classes depending on whether $I(x, y) > T$. This type of thresholding is referred to as bi-level. Algorithms which deal with three or more classes are called multi-level thresholding techniques. The threshold T can either be fixed independent of the image data, or automatically selected from a histogram. This automated selection can be done in many ways. For bi-level thresholding, Prewitt and Mendelsohn [84] repeatedly smoothed the histogram until a single minimum existed between two maxima, and chose T as the intensity that corresponded to this minimum.

For multi-level thresholding, the k-means algorithm [45] and the iterative selection thresholding (ISODATA) method [87] can be used. These two methods are similar and involve separating the histogram into two parts using a starting threshold value of half the maximum dynamic range. The sample means of the foreground and background pixels are computed, and a new threshold value is computed as the average of these two sample means. The process is repeated, based upon the new threshold, until the threshold value does not change any more. A disadvantage of this type of histogram-seeking method is that it may be difficult to identify significant peaks and valleys in the image, thereby affecting the quality and usefulness of the final solution. As shown in Figure 5.1c, segmenting a long-bone from an x-ray image using a single threshold chosen using this method works well for some regions, but also works very poorly for others.

Since spatial information is ignored, the use of thresholding for the segmentation of long-bones from an x-ray image is not appropriate.

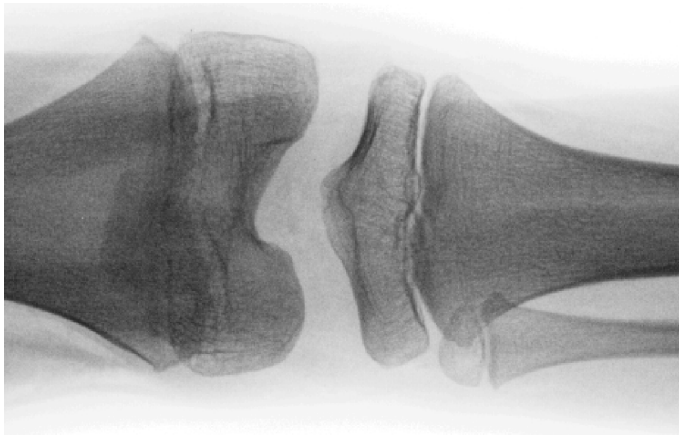
Global segmentation algorithms

Global segmentation algorithms include edge-based methods, region-based techniques and deformable models. These algorithms take spatial information into consideration, so they usually outperform the clustering-based algorithms (thresholding, k-means and ISODATA). This is because most segments corresponding to real world objects consist of pixels that are spatially connected.

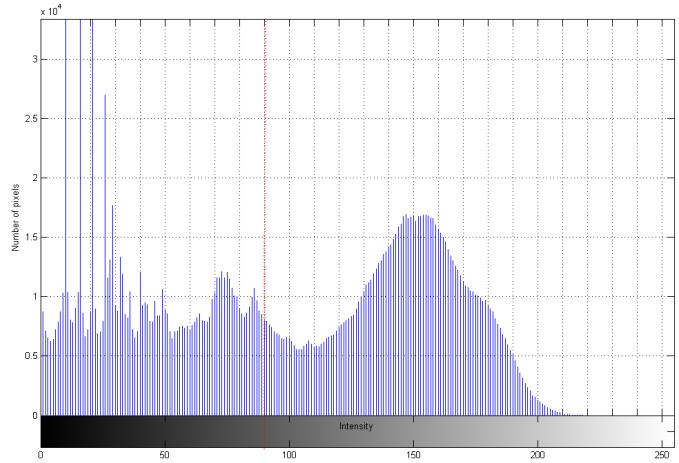
The edge-based detection techniques generally use edge and contour tracking, and exploit this spatial information because the edges correspond to discontinuities in the homogeneity criterion for segments. There are many edge-based algorithms in the literature, but the application of edge models to local regions is unlikely to be successful as local regions contain too few pixels, of which most have incomplete neighbour information. Edge-based techniques are also prone to failure in the presence of blurring due to breaks in the contour lines.

In region-based methods the image is split into connected regions by grouping or clustering neighbouring pixels with similar intensity levels. That is, they identify homogeneous, spatially connected sets of pixels within an image. The adjacent regions are then merged together using criteria based on sharpness of region boundaries or homogeneity of the regions. The choice of the criteria is critical, since lenient criteria overlook blurred boundaries and over-merge, while over-stringent criteria create unnecessary image fragmentation. Region growing methods also require a seed (an initial contour that approximates the shape being segmented) to begin, and the resulting segmentation depends heavily on the initial seed chosen.

The final method is more commonly referred to as a deformable model, or connectivity-preserving relaxation-based method (discussed in Section 3.2.1). Becoming trapped in a local minimum that does not correspond to the global solution is a difficult problem that these models face. Other advanced segmentation algorithms utilise wavelet or Fourier techniques to separate regions based on texture. In many cases it is necessary



(a)



(b)

(c)

Figure 5.1: Thresholding (a) the input image. (b) Histogram of the input image showing the threshold of $T = 90$, automatically chosen using the ISODATA method, and (c) the result of applying the threshold to the image.

to utilise both local and global criteria to improve the segmentation.

5.1.1 Feature extraction in long-bone x-ray images

Before choosing a segmentation or feature extraction method, it was pertinent to examine the features of interest, as these would determine the method used. As previously discussed in Section 2.5.1 on page 25 and shown in Figure 2.10b, the diaphysis was the region of interest for fracture detection, so a robust method of determining the location of the diaphysis within an image was required. Based on the AO method, the first critical feature that had to be extracted was the location and orientation of the bone within the image. One method of determining this position information was to segment the bone from the background using one of the methods just outlined in Section 5.1, and then examine the segmented result to determine the region of interest. Each of the methods described had flaws, and a poor segmentation could make the task of identifying the diaphysis very difficult. Rather than attempting to accurately segment the bone from the background, it was proposed that only the information about the location and orientation of the bone needed to be accurately determined. Then a series of identifiable landmarks on the bone could then be used for segmentation according to the AO protocol (Section 2.5.1 on page 25).

To extract the position information, prior knowledge about expected human long-bone morphology was utilised. Regardless of their anatomical location, long-bone diaphyses containing no major pathology, were typically very straight. This was true even though there was a large degree of natural variation in bone shape, and the presence of pathologies could cause the long-bones to become less straight or even fragmented. Making this assumption allowed the diaphyseal bone edges to be approximated by line parameters that were useful for both diaphysis segmentation and fracture detection.

5.2 The Hough Transform

To extract global information for approximating the edges corresponding to the long-bone shafts within the x-ray image, the Hough Transform was used. The Hough Transform is a very powerful pattern detection method first suggested by [47] in 1962. It

assumes that there are an infinite number of potential lines that pass through any point in the image, each at a different orientation. The purpose of the transform is to determine which of the theoretical lines pass through the most features in the image—that is, which lines fit most closely to the data.

The transform is essentially an evidence gathering procedure where every point (x, y) within the input image I votes for all the possible straight line parameter combinations that could have produced it. The votes are added to an accumulator array H, and the number in each cell indicates the relative likelihood of a pattern described by those particular parameters being present within the image. The Hough Transform therefore converts spatially extended patterns into spatially compact features within the space of possible parameter values, so that the problem is reduced from a difficult global feature analysis to a simpler local peak detection in the parameter space [50].

To determine if two points lie on the same potential line, it is necessary to create a line representation that allows meaningful comparison. In the original implementation, Hough chose to use the slope-intercept parameterisation $y = mx + c$, so the accumulator space was the two-dimensional slope-intercept plane. However both the slope and intercept are unbounded, making the technique hard to apply in practice. In the normal parameterisation of the Hough Transform, suggested in 1972 by Duda and Hart [37], a straight line with parameters (ρ, θ) is specified by:

$$\rho = x\cos\theta + y\sin\theta \tag{5.1}$$

where ρ is the length of the normal to the line segment from the image origin (taken to be the centre of the image), and θ is the angle of the normal with the positive x axis. Unlike the original Hough implementation, Equation 5.1 maps every line in the (x, y) plane to a unique point in the bounded (ρ, θ) plane.

For every point (x, y) within the image, and for each sampled value of θ, the value of ρ is calculated using Equation 5.1. After ρ is quantised to the nearest value on the ρ-axis, the corresponding cell $H(\rho, \theta)$ in the accumulator is incremented. A simple example of the Hough parameterisation is shown in Figure 5.2. The original image contained two lines of different length, which are displayed in the Hough accumulator

as two discrete points located at $(\rho_1, \theta_1) = (127, 76)$ and $(\rho_2, \theta_2) = (-100, 112)$. The intensity of the accumulator peaks is proportional to the length of the corresponding line. Figure 5.2a shows how θ and ρ relate the line and accumulator peak.

5.2.1 Relationship between the Hough and Radon transforms

The Hough Transform is a special case of the Radon transform [35] suitable for implementation on images or functions of two discrete variables. The Radon transform is named after J. Radon who showed how a function could be described by its integral projections [85]. The mapping from the function onto the projections is the Radon transform. The most common formulation of the Radon transform $\mathcal{R}$ of the function $I(x, y)$ defined on the xy plane $\mathcal{D}$ is:

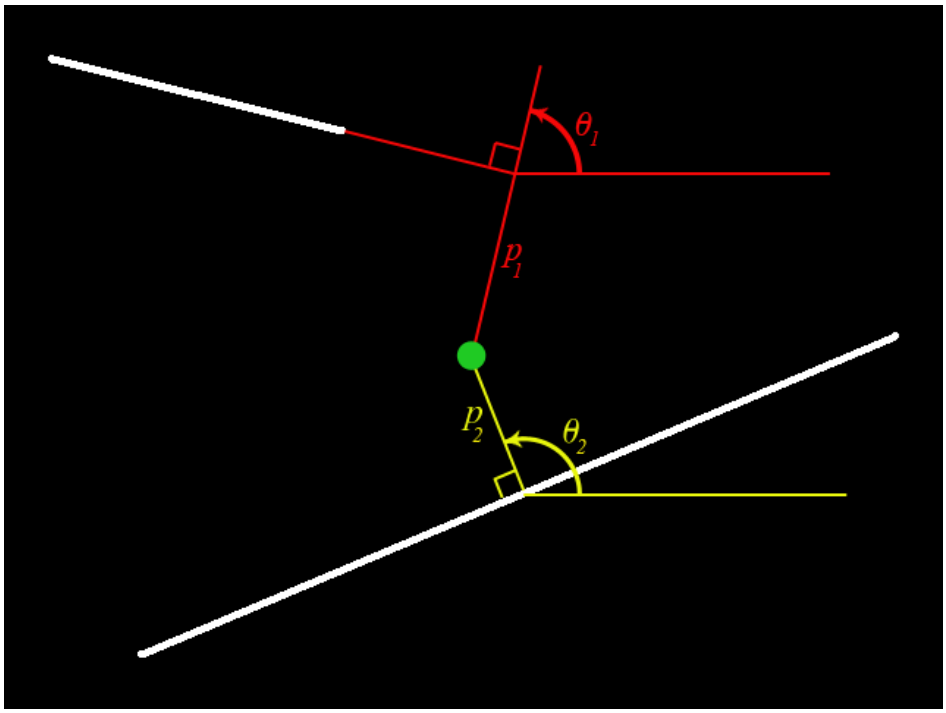
$$\mathcal{R}\{I\}(\rho, \theta) = \int_{\mathcal{D}} I(x, y)\, \delta(x \cos\theta - y \sin\theta - \rho)\, dx\, dy \tag{5.2}$$

The Radon transform is also known for its role in Computed Tomography, where it is used to model the process of acquiring projections of the original object using x-rays. The inverse Radon transform can be used to reconstruct images of the original object from the projection data.

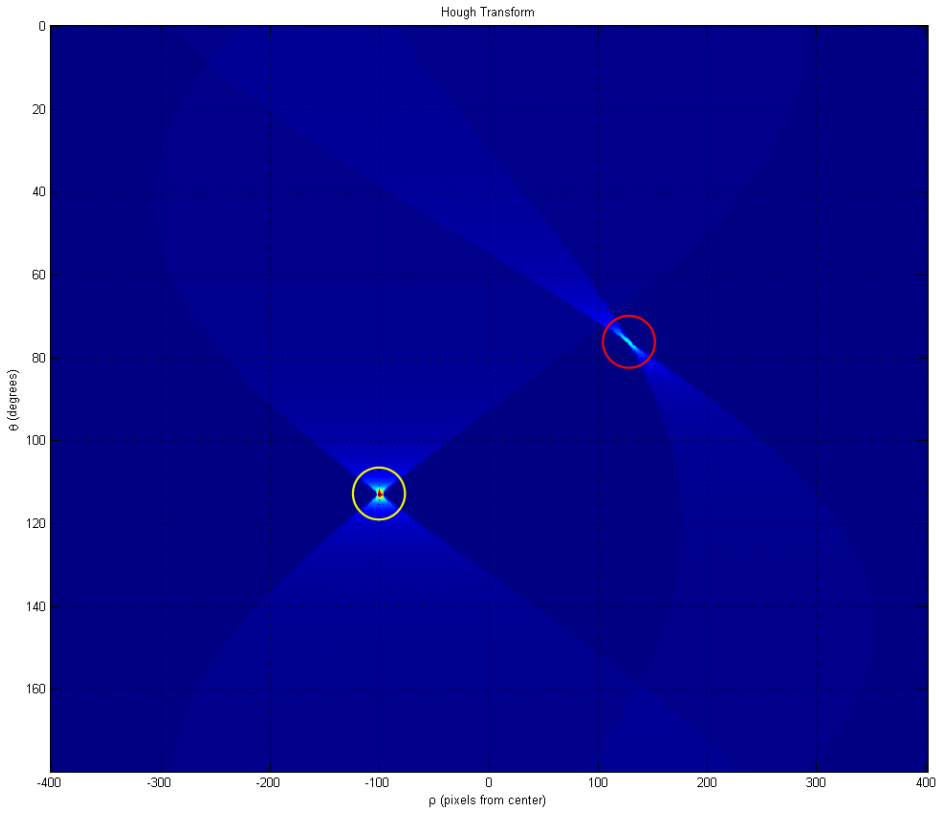
5.2.2 Advantages and disadvantages of the Hough Transform

Like the Fourier, wavelet, discrete cosine and Radon transforms, the Hough Transform is invertible. That is, all the information in the original image I is also present in H, simply in an alternative representation. The advantages of the Hough Transform are its robustness to noise within the image, and its insensitivity to discontinuities within the pattern of interest [12]. Random image points are very unlikely to coherently contribute to a single cell in the accumulator array, and as a result only produce a spread of low background counts. Another advantage is that the Hough Transform can be parameterised differently to detect other shapes such as circles and ellipses:

$$radius = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{5.3}$$



(a)



(b)

Figure 5.2: (a) Example image and (b) its Hough Transform. The construction lines show how the parameters (ρ,θ) are calculated for each peak.

However, increasing the number of unknown parameters ($radius$, x_0 and y_0) also increases the number of dimensions of the accumulator, and correspondingly increases calculation time. The major disadvantage of the standard Hough Transform implementation is that calculating Equation 5.1 at every pixel (x, y) for every θ in the range $[0, \pi]$ is very inefficient, and requires large processing and memory requirements.

5.2.3 The modified Hough Transform

For computer aided long-bone fracture detection, the implementation of the original Hough Transform was modified in two ways. Firstly, the difficult pre-processing problem of choosing an appropriate threshold for determining the feature points that should be included in the Hough Transform was eliminated. In the original implementation, a threshold was applied to the image (as described in Section 5.1 on page 83), producing a binary result to which the transform was applied. This meant that if a low threshold (for example $T = 20$) was chosen, then pixels with gradient magnitude slightly larger than the threshold (such as 30) and those much larger than the threshold (such as 220) were assigned the same level of importance, even though one edge was clearly much stronger than the other. In addition, choosing T was clearly still an important but difficult problem. It could, however, be eliminated by incrementing the accumulator by the magnitude of the gradient $|\bigtriangledown I(x, y, t)|$ at the chosen scale t_2 (i. e. using Figure 4.15c directly). As a result, strong edges automatically made larger contributions to the parameter space than weak ones.

Despite being robust against noise, the Hough Transform had a tendency of finding false lines in the image. To counter this, the number of feature points not belonging to the line of interest was reduced by using a variation of the method described in [79, 106]. For an image pixel $|\bigtriangledown I(x, y, t)|$ to cause the corresponding accumulator cell $H(\rho, \theta)$ to be incremented, the gradient direction $\phi(x, y, t)$ at that particular pixel had to be perpendicular to the current θ value. A small range of values r was allowed to ensure that a reliable match occured. This meant that for any given image point $|\bigtriangledown I(x, y, t)|$ the Hough Transform was only calculated for a small range of θ that satisfied the

following equation:

$$\phi(x,y,t) \pm \frac{\pi}{2} - r \leq \theta \leq \phi(x,y,t) \pm \frac{\pi}{2} + r \tag{5.4}$$

Where r was the range delimiter, and the gradient direction was given by:

$$\phi(x,y,t) = arctan\left(\frac{\partial I}{\partial y} \cdot \frac{\partial x}{\partial I}\right) \tag{5.5}$$

When using the standard Hough Transform, false accumulator peaks could be produced by spatially separated regions of high intensity that may have been in alignment at a particular transform angle θ, but did not necessarily belong to the same feature. Using a modified Hough Transform that had to satisfy Equation 5.4 prevented these false peaks from appearing in the accumulator. An added benefit of using the modified Hough Transform was the significant decrease in calculation time that occured, because for any given pixel there was a much smaller range of angles at which the transform needed to be calculated. Thus smaller values of r produced shorter calculation times, which are quantified at the end of Section 5.3. In addition to producing shorter calculation times, reducing r also increased the accuracy with which the angle of the gradient had to match θ, so it seemed natural to make r very small. However the accuracy of the angles calculated using Equation 5.5 was limited to a few degrees because they were calculated using a 3 x 3 stencil. Hough Transforms of the development images showed that when the match range was reduced below $r = 2^\circ$ some peaks disappeared because few matches occured. Therefore, the match range was set at $r = 2.5^\circ$, so that a five degree range centred on $\phi(x,y,t)$ was allowed.

A crude example of the false peaks effect is shown in Figure 5.3. The input image contains one long line segment, and seven short line segments that are spatially separated but aligned. The Hough Transform should therefore contain eight peaks, but Figure 5.3b contains nine peaks. The slightly dispersed false peak located at $(\rho,\theta) = (104,140)$ is produced by the alignment of the seven short line segments at the same θ as the longer line segment. The modified Hough Transform shown in Figure 5.3c was produced with a range $r = 2.5^\circ$, and the false peak due to the alignment of the

short line segments is no longer present. In addition, the characteristic butterfly shape produced by the standard implementation is also removed, leaving sharper peaks. It is worth noting that Figures 5.3b and 5.3c show the Hough Transform of the magnitude of the gradient of the image shown in 5.3a, resulting in double peaks at each point. This is different to Figure 5.2b which is the Hough Transform of 5.2a, not the magnitude of the gradient of 5.2a.

5.3 Modified Hough Transform algorithm implementation

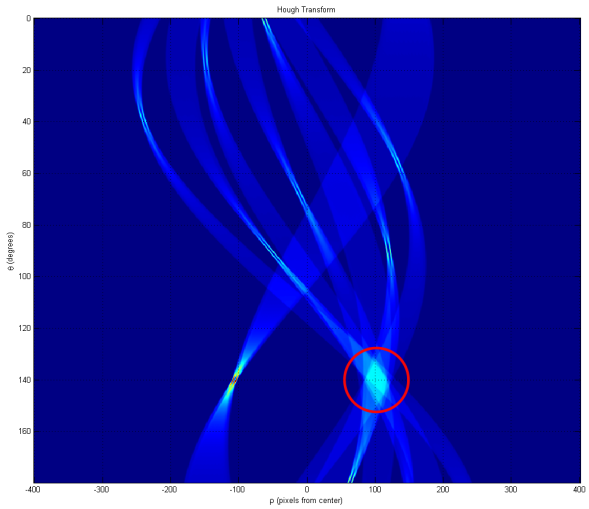
The Hough Transform was implemented using code that was based on the Matlab [72] Radon transform function and the description by Bracewell [22]. As suggested by Equation 5.1, the dispersion of each pixel along the ρ-axis is a nonlinear function of θ and the pixel location. To improve accuracy, each pixel was split into 4 sub-masses located to the north-east, north-west, south-east, and south-west of the original location, with a sub-mass spacing of 0.5 in the x and y directions, as shown in Figure 5.4. The algorithm proceeded by determining $x\cos\theta$ and $y\sin\theta$ for each of the four sub-masses:

$$\begin{aligned}
(x\cos\theta)_1 &= ((x_{origin} - x) - 0.25)\cos\theta \\
(x\cos\theta)_2 &= ((x_{origin} - x) + 0.25)\cos\theta \\
(y\sin\theta)_1 &= ((y_{origin} - y) - 0.25)\sin\theta \\
(y\sin\theta)_2 &= ((y_{origin} - y) + 0.25)\sin\theta
\end{aligned}$$

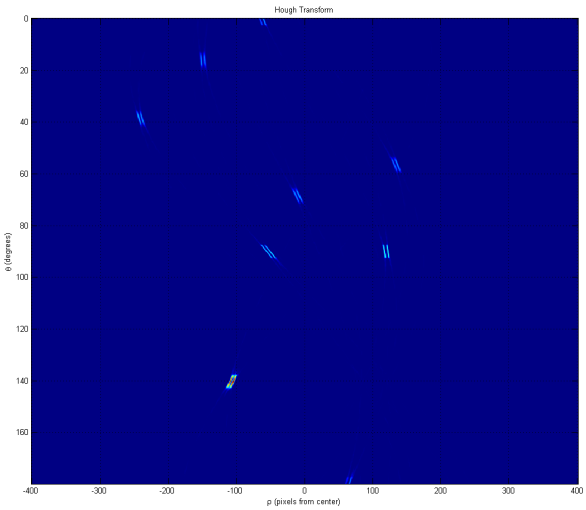
where (x, y) are the coordinates of the current pixel and (x_{origin}, y_{origin}) are the coordinates of the centre of the image. Four corresponding values of ρ were then determined using:

$$\begin{aligned}
\rho_1 &= (x\cos\theta)_1 + (y\sin\theta)_1 \\
\rho_2 &= (x\cos\theta)_2 + (y\sin\theta)_1
\end{aligned}$$

(a)



(b)



(c)

Figure 5.3: Example of the false peaks effect. The input image (a) contains aligned line segments, that produce a false peak in the Hough Transform in (b). The false peak is not present in the modified Hough Transform ($r = 2.5°$) shown in (c).

$$\begin{aligned}\rho_3 &= (x\cos\theta)_1 + (y\sin\theta)_2 \\ \rho_4 &= (x\cos\theta)_2 + (y\sin\theta)_2 \end{aligned} \tag{5.6}$$

At this stage each ρ was quantised to the nearest value on the ρ-axis, because a standard quantisation interval of $\triangle\rho = 1$ was used. Rather than simply rounding up or down and thereby losing some accuracy, a triangular smoothing window of length two was used to smooth the result in the ρ-direction. For example, in the case of ρ_1, the fractional part of ρ_1 was calculated using:

$$\delta_1 = \rho_1 - \lfloor \rho_1 \rfloor \tag{5.7}$$

Then the contributions made to the cells on either side of the actual ρ_1 value were proportional to δ_1, where in this case $\rho = \lfloor \rho_1 \rfloor$:

$$\begin{aligned} H(\rho,\theta) &= H(\rho,\theta) + 0.25(1-\delta_1)\left|\bigtriangledown I(x,y,t)\right| \\ H(\rho+1,\theta) &= H(\rho+1,\theta) + 0.25\delta_1\left|\bigtriangledown I(x,y,t)\right| \end{aligned} \tag{5.8}$$

As Equation 5.8 shows, once the coordinates (ρ,θ) for the four sub-masses were determined, those accumulator cells were incremented by one quarter of the intensity of the pixel at $|\bigtriangledown I(x,y,t)|$. The same process was repeated for the remaining sub-masses in Equation 5.6.

This code was modified to be more efficient, and to allow calculation of the modified Hough Transform described in Section 5.2.3. The original implementation calculated Equation 5.1 at every value of θ for every pixel (x,y) in the image, requiring $xy(\theta_{max} - \theta_{min}) / \triangle\theta$ iterations, where $\triangle\theta$ was the sampling interval (the step size in the θ direction). The code for the modified Hough Transform still calculated Equation 5.1 at every pixel (x,y) in the image, but rather than using every value of θ, only a small range based on Equation 5.4 and the direction of the gradient at (x,y) was used. This method required at most $2xyr/\triangle\theta$ iterations, and was dependent on the value r chosen for the allowed match range. The ratio of the number of iterations required by each method was therefore $\theta_{max} - \theta_{min} : 2r$, and for typically chosen values

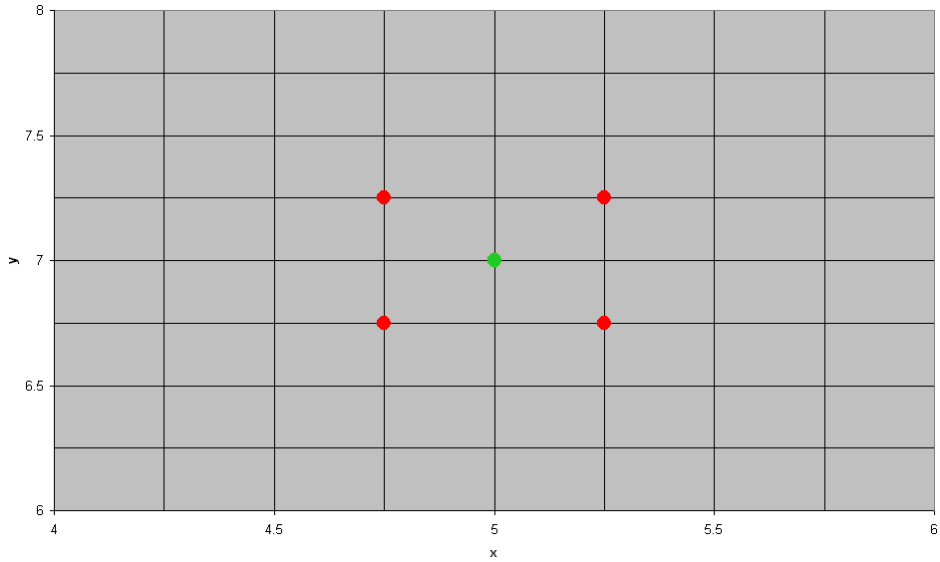


Figure 5.4: Four sub-masses located around the pixel $(x,y) = (5,7)$, with spacing 0.5 in the x and y directions. The corresponding locations (ρ,θ) of these four sub-masses in the Hough accumulator are incremented by one quarter of the pixel intensity at (x,y).

($\theta_{min} = 0$, $\theta_{max} = 180$, and $r = 2.5°$) this ratio was 180:5. Accordingly the modified method required less than three percent of the number of iterations required by the original method.

Calculation of the modified Hough Transform using this method had some overheads that were not present in the normal transform. The gradient direction image produced by the AMSS algorithm (Figure 4.15d) contained values in the range $[0,1]$, representing all the values (360°) on the unit-circle. Therefore after a pixel was read it had to be normalised to match the Hough Transform angle range (180°), including the subsequent shifting of all values in $[-180,0]$ into $[0,180]$ by adding 180° if the angle was less than zero. Once the normalised angle $\phi(x,y,t)$ of the gradient at that pixel was known, the Hough Transform only needed to be calculated at $\pm r°$ around that value. It was necessary to check that the range $\phi(x,y,t) - r < \theta < \phi(x,y,t) + r$ actually lay at least partly within $\theta_{min} < \theta < \theta_{max}$, and then the endpoints were modified if required. For example, with standard choices of $\theta_{min} = 0$, $\theta_{max} = 180°$, and $r = 2.5°$, a normalised value of $\phi(x,y,t) = 1°$ would require modification of the endpoints from $-1.5° < \theta < 3.5°$ to $0 < \theta < 3.5°$. Finally, once the required calculation range was established, the corresponding θ array indices for those angles had to be determined. This was achieved by simply searching the θ array for a match, and noting the index at which the match occured. Once the low and high array indices were known the transform proceeded as

described for the standard Hough Transform, but for a much smaller range of indices.

The choice of the sampling interval $\triangle\theta$ was a trade off between accumulator resolution and calculation time. To reduce parameter errors, smaller values of $\Delta\theta$ were desirable, although decreasing the sampling interval resulted in an increase in memory and computation requirements. Since there was often less than a few degrees difference between the lines being examined, a sampling interval of less than a degree was required. Calculations using the six development images showed that a sampling interval of $\Delta\theta = 0.25$ produced a resolution that was high enough to accurately distinguish the peaks in the accumulator.

Testing of the algorithm[1] revealed that for a typical x-ray image of size 3600 x 1200 pixels taken from the development set, and a usual choice of parameters ($\theta_{min} = 0$, $\triangle\theta = 0.25°$, $\theta_{max} = 180°$, $\triangle\rho = 1$), the standard Hough Transform required 273 seconds of calculation time, while the modified Hough Transform required only 94 seconds of calculation time. This corresponded to a speed increase of approximately 2.9 times.

5.4 Peak detection

Any cell (ρ, θ) in the resulting accumulator H that had a significant number of votes (that is, a high intensity) was a candidate for representing a line with those parameters in the input image I—that is, it was likely to correspond to a dominant feature such as the edge of a long-bone. Therefore the aim was to determine the coordinates of the accumulator peaks as accurately as possible. However the detection of peaks is a non-trivial exercise, especially with complicated input images. Simple peak detection algorithms that locate the absolute maximum or utilise a global threshold occasionally work well for artificial images, but generally perform poorly when applied to real x-ray images [12]. Accurate peak detection is even more difficult when the input image contains multiple straight lines that produce many peaks in the parameter space. The search for the peaks can be refined by searching for a local maximum within a window [99]. The choice of the window size is critical, however automatically determining the

[1] Performed on a 2.2GHz Dual Core Athlon 64 X2 with 2 x 1MB level 2 caches, 4GB RAM running Windows XP

correct window size is difficult. If the window size is set too small, then overlapping false peaks will be found, while peaks will be missed if the window is too large.

Thin line spread in the ρ direction:

It has been shown by Van Veen and Groen [106] that even for an infinitesimally thin line, oversampling in the parameter space (that is, using a small quantisation step) causes the peak in the accumulator to spread rather than remain localised to one cell. This means that in practice the votes do not fall into uniform and compact regions, so the peaks are often poorly defined and spread in both the ρ and θ directions. They showed that for an infinitesimally thin line, the maximum number of cells N_ρ over which the peak is spread in the ρ direction could be determined using:

$$N_\rho = \left\lfloor \frac{d \sin\left(\frac{\triangle\theta}{2}\right)}{\triangle\rho} \right\rfloor + 2 \tag{5.9}$$

where d is the maximum length of the line segment. The major assumption here is that the edges in $|\bigtriangledown I(x,y,t)|$ are infinitesimally thin, straight lines.

Thick line spread in the ρ direction:

Real world images containing thicker, non-straight lines—such as the edges $|\bigtriangledown I(x,y,t)|$ in a typical x-ray image—cause additional spread. As suggested in [12], the maximum number of cells N_ρ over which the peak is spread in the ρ direction can be better approximated using:

$$N_\rho = \left\lfloor \frac{d \sin\left(\frac{\triangle\theta}{2} + \arcsin\left(\frac{b_a}{d}\right)\right)}{\triangle\rho} \right\rfloor + 2 \tag{5.10}$$

where b_a is the width of the line. All parameters other than d and b_a are determined by the Hough Transform.

For all detectable lines of interest in $|\bigtriangledown I(x,y,t)|$, the length was at least two orders of magnitude greater than the width, so d was effectively the length of the line. To calculate the maximum spread a worst case scenario was assumed, in which the long-bone edges passed diagonally across the image, so the best estimate of d was the

diagonal length of the image $d = \sqrt{x_{size}^2 + y_{size}^2}$. Although in most cases this was an overestimate—since the bone shaft was not always oriented diagonally and completely filling the image—it allowed the maximum possible spread to be determined. The line width b_a was independent of scale because the smoothing was anisotropic, although it did vary between images due mostly to anatomical variations, and in part to differences in x-ray acquisition parameters.

Determining the line width:

The line width b_a varied between images, depending mainly on the thickness of the cortical bone in the shaft. To determine the amount of variation, the shaft thickness (line width) was measured (in pixels) on both sides of the long-bone axis, at the proximal, middle and distal portions of the shaft of each of the six images in the development set. For the distal long-bones, only the cortical widths of the larger bone (for example the tibia rather than the fibula) were measured. This was because Equation 5.10 was used to calculate the maximum spread, and the spread was larger when thicker lines were present. Recording the widths of the thinner cortical walls of the smaller bone would have decreased the resulting value of N_ρ too much.

The variation in the cortical width b_a was measured in the six locations across the development set, producing a total of 36 values. The width was measured from the magnitude of the gradient of the image smoothed to scale $t_2 = 20$, as this was the same scale at which the modified Hough Transform was calculated. It was found that the average edge width was $b_a = 40$ pixels (with a standard deviation of $\sigma = 12.3$), so the value $b_a = 40$ was set for all images. This resulted in a spread of $N_\rho = 50$ pixels, from a line length to width ratio of approximately $\frac{b_a}{d} = 0.01$. The linear relationship between the line width and the amount of resulting peak spread is shown in Figure 5.5, along with the chosen value of b_a.

Spread in the θ direction:

The maximum spread in the θ direction was determined by the parameters used in the modified Hough Transform calculation. Since Equation 5.4 allowed an error of $\pm r^\circ$ in

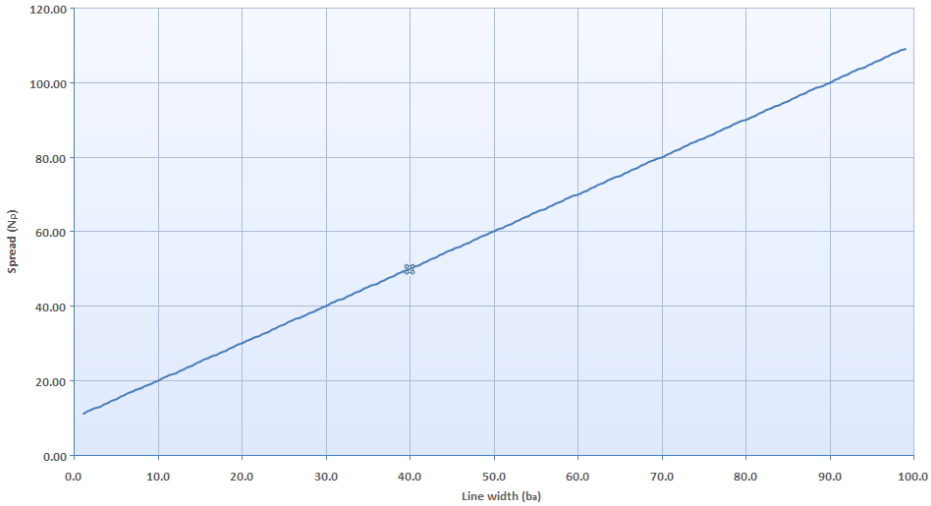


Figure 5.5: The effect of altering the line width b_a on the resulting spread N_ρ in the Hough accumulator. The chosen value $b_a = 40$ is marked.

the accuracy with which θ matched the gradient direction ϕ, the value:

$$N_\theta = \frac{2r}{\Delta\theta} \tag{5.11}$$

was chosen as the maximum spread in the θ direction.

5.4.1 The ranked sums method

Since the accumulator peaks spread, the large values around the maximum accumulator element were all likely to belong to the same image feature, even for an ideal input [42]. Therefore it was incorrect to simply assume that the largest cells were the peaks that were required. However, Equations 5.10 and 5.11 gave an estimate (N_ρ, N_θ) of how far the accumulator peaks were likely to spread, and this information could be used to iteratively calculate the location of the true peaks. The method by which this was performed for long-bone parameter approximation is illustrated using the simple example in Figure 5.6a, which shows a synthetic image containing four straight lines, and in 5.6b its modified Hough Transform ($r = 2.5°$) containing four discrete peaks.

Locating the maximum element:

Firstly, the location of the maximum element in the accumulator was identified using:

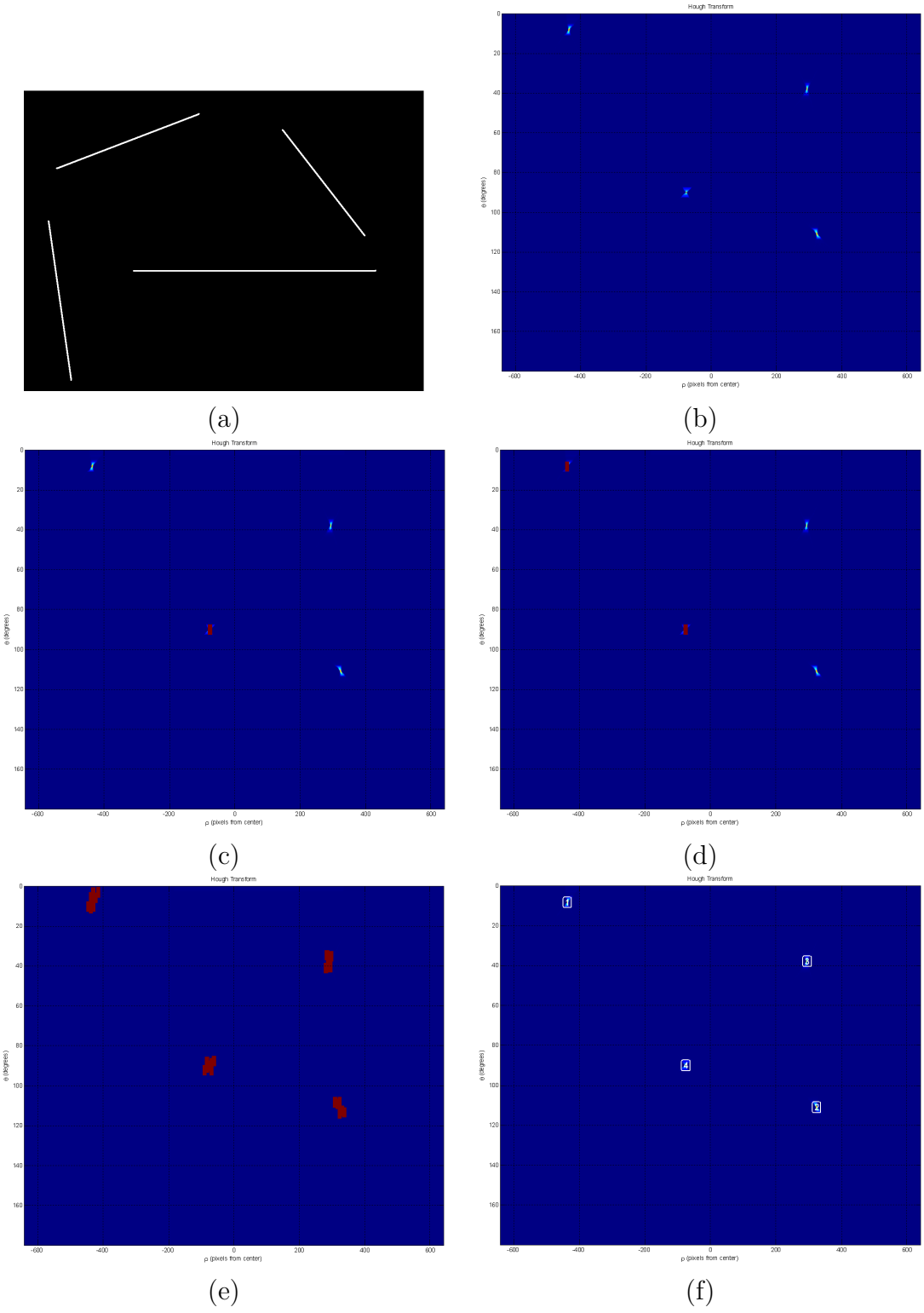


Figure 5.6: Example of the ranked sums method. (a) The original image, and (b) the modified Hough Transform ($r = 2.5°$). (c) The maximum value is calculated using Equation 5.12, the sum of the N_ρ x N_θ region around this detected peak at $(\rho_i, \theta_i) = (-74, 90)$ is calculated using Equation 5.14, before the region is cleared using Equation 5.15. This step is repeated in (d) for $i = 2$ up to (e) $i = 30$. (f) The result is the location of the four peaks.

$$(\rho_i, \theta_i) = \{a, b : H(a,b) = max(H(j,k); 0 \leq j < \rho_{size}, 0 \leq k < \theta_{size})\} \tag{5.12}$$

where in this case $i = 1$. This value was then used to normalise the accumulator with:

$$H_N = \frac{H}{H(\rho_1, \theta_1)} \tag{5.13}$$

so that the maximum value in the accumulator was $H_N(\rho_1, \theta_1) = 1$. Using Equation 5.12, the location of the maximum element in the accumulator in Figure 5.6b was identified as $(\rho_1, \theta_1) = (-75, 90)$.

Calculating the region sum:

It was previously determined that in a normal x-ray image the expected line width is $b_a = 40$ pixels, but in this example the lines were much thinner at approximately $b_a = 6$ pixels. Knowing the line width and that $r = 2.5°$ was used to calculate the Hough Transform, Equations 5.10 and 5.11 could be used to calculate that $(N_\rho, N_\theta) = (10, 20)$ was the amount of expected peak spreading. Next, the sum S_i of the cells in a rectangular N_ρ x N_θ region centred around the identified maximum was calculated:

$$S_i = \sum_{m=\rho_i - \frac{n_\rho}{2}}^{N_\rho} \sum_{n=\theta_i - \frac{n_\theta}{2}}^{N_\theta} H_N(m,n) \tag{5.14}$$

Using Equation 5.14 on the example in Figure 5.6b gave the sum of the first region as being $S_1 = 45.0$. In cases where the identified maximum was close to $\theta = \pm 180°$, the sum region was wrapped to include the opposite end of the accumulator, since the θ values were continuous.

Clearing the region:

To ensure that those cells in the rectangular region could not contribute to the sum of other regions—since they were all likely to belong to the identified maximum value—all accumulator cells within the N_ρ x N_θ region were cleared:

$$H_N(a,b) = 0;\ \rho_i - \frac{N_\rho}{2} \leq a \leq \rho_i + \frac{N_\rho}{2},\ \theta_i - \frac{N_\theta}{2} \leq b \leq \theta_i + \frac{N_\theta}{2} \tag{5.15}$$

At this stage, the order in which the peaks are cleared has no effect on the peak selection outcome, unless the rectangular N_ρ x N_θ regions overlap. In this (very likely) case, the cleared cells cannot contribute to any subsequently selected N_ρ x N_θ regions. That is, those cells belong to that maximum element (ρ_i, θ_i) rather than any of the subsequent ones, because it has a higher value than the subsequent ones. Figure 5.6c shows the result of clearing the region that contained the identified maximum value, using Equation 5.15.

Determining a ranking criteria:

If there was no peak spreading, and all votes for a particular line fell into a single accumulator cell, then it was likely that choosing the maximum values in the accumulator would yield the desired peaks. In a real image where the lines to be detected were not necessarily completely straight—such as the shaft of a long-bone—it was possible for the votes for a particular line to spread to such an extent that no single cell within the N_ρ x N_θ region had a high enough value for it to be selected as a peak. This situation could occur despite the region containing many cells with moderately high counts. As a result, because the peaks spread, the sum of all the cells within the N_ρ x N_θ region was demonstrated to be a better measure of the likelihood of that region containing the true peak than simply the peak value itself, since it took into account not only the maximum value, but also the values of the surrounding cells.

This necessitated calculating the sum S_i at a large number of i, in case a region with a lower peak value (one that occured at a large i) had a comparatively high region sum. Therefore, the process of calculating the sum of the region centred around the maximum value, followed by clearing that region, was repeated until a list of regions and their sums (ρ_i, θ_i, S_i) was produced. To ensure that no regions were missed, the steps were performed at $i = \{0, 1, 2, ..., 100\}$, where the maximum i was chosen to be sufficiently large such that all relevant regions were included. In the example in Figure 5.6, the first identified region (which contained the maximum value) had a sum $S_1 = 45.0$, the

second region had a higher sum of $S_2 = 51.6$, the third had a sum of $S_3 = 51.3$, and the fourth a sum of $S_4 = 47.3$. The next highest region sum was $S_7 = 7.8$. So in this case the same results would be achieved regardless of whether the region maximum or region sum was used as the ranking criteria, although this was due to the simplicity of the test image and the absence of any significant peak spreading. For more complicated images such as a long-bone x-ray image, the region sum was the better criteria for locating the peaks.

Ranking the identified peaks:

The list was then ranked according to the sum S_i, and depending on the number of peaks e to be detected, the maximum elements in the top e regions were selected as the required peaks. As a result, Figure 5.6f shows the peaks in a different order, because they were ranked in terms of their sums. Taking into account peak spreading by clearing an N_ρ x N_θ window around the i^{th} peak reduced the likelihood of subsequently selecting a maximum that belonged to that peak when examining the $(i+1)^{th}$ peak. Using this method the required peaks were able be accurately located with minimal user input.

To calculate e, the user was required to identify the anatomic location being examined, by selecting the horizontal value 1 through 4 from the AO fracture location in Figure 2.10. Alternatively, if available, this information could be directly obtained from the DICOM file header. For selections 1 and 3, the value $e = 2$ was set, and for selections 2 and 4, the value $e = 4$ was set, so that the number of peaks to detect corresponded to the number of bone edges. Figure 5.7 shows the detected peaks for a tibia and fibula (bone 4), using the corresponding value $e = 4$.

Unfortunately, due to the complexity of some fractures, it was not possible to utilise prior knowledge about the orientation of the long bones within the x-ray images to reduce the computational complexity of the peak detection. It was found that at this stage of the algorithm, making assumptions about the expected peak locations based on long-bone anatomy was too restrictive, and produced very poor peak detection performance. As a result, no further changes were made to the ranked sums method.

Evaluation of the ranked sums method using the six development images confirmed

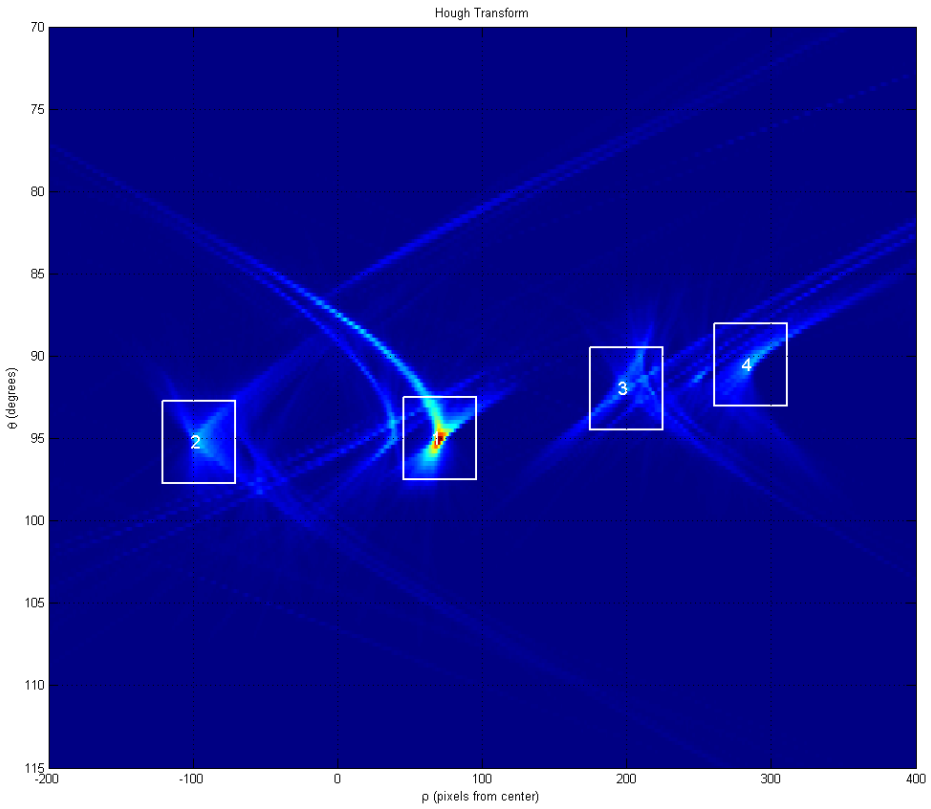


Figure 5.7: The modified Hough Transform ($r = 2.5°$) of the gradient image in Figure 4.15 on page 79c, showing the peaks (and the estimate of the peak spread) detected by the ranked N_ρ x N_θ sum method.

that peak detection was better when the region sum was used rather than the region maximum. Applying the ranked sums method to the six development images, in which 28 peaks were required to be detected, yielded 26 correctly chosen peaks and 2 incorrectly chosen peaks. Applying the same test, but ranking the regions based only on their maximum value, yielded 16 correctly chosen peaks and 12 incorrectly chosen peaks. That is, the ranked sums method selected 93% of the peaks correctly, as opposed to the maximum value method which only selected 57% of the peaks correctly. Therefore, the ranked sums method was well suited to detecting the accumulator peaks required for long-bone parameter approximation.

5.5 Back projection

Any parameter pair (ρ, θ) in the Hough accumulator represents a unique line in the input image and vice-versa, allowing a mapping between the two spaces. This means that the lines corresponding to the detected peaks (ρ_i, θ_i) could be drawn on either the

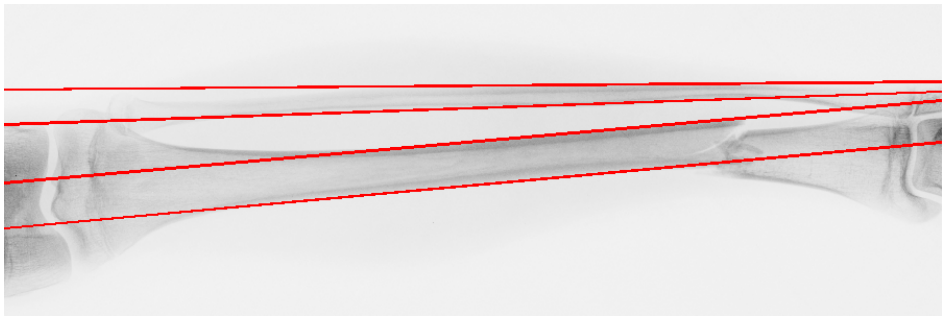


Figure 5.8: The lines detected from the original image that correspond to the identified peaks (ρ_i,θ_i) were a good approximation to the bone shaft edges.

input or gradient images to highlight the features that contributed to that particular peak, a process known as back projection. In this case, back projecting the lines corresponding to all the detected peaks should give a good straight-line approximation of the long-bone edges. The ranked sums peak detection evaluation described at the end of the previous section was performed by examining the back projected images, since this easily showed the accuracy with which the detected peaks matched the features of interest. Figure 5.8 shows the lines in the original image that correspond to the peaks detected in Figure 5.7.

5.6 Improving poor peak detection

In some cases, the peak detection performed by the ranked sums method was not perfect, and the back projected lines did not always match the required features. Firstly, in some cases the value of e that was selected was not large enough to detect all the required edges. This could happen in a midshaft fracture where the bones were displaced in such a way that the segments were no longer aligned, or even in a non-fractured bone that was curved in such a way that its edges would be better approximated by multiple straight lines. In some of these situations it was necessary for six or even eight edges to be detected, where normally only four would be required. In addition, the values chosen for e were suitable for AP images, but not necessarily for ML images of the extremities, in which the bones could superpose. Accordingly for some ML images of the AO locations 2 and 4 (radius/ulna and tibia/fibula), the value $e = 2$ was more appropriate than the value $e = 4$. In these cases, the operator had the choice to increase

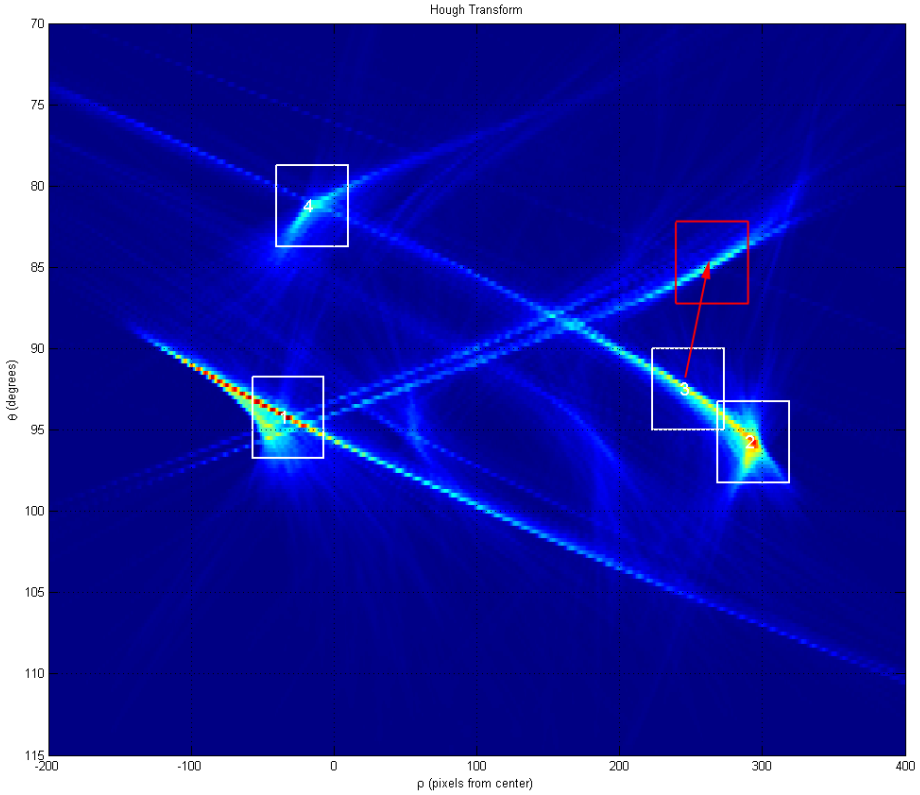
or decrease e manually if a poor detection resulted, and more or less edges should have been found. Often this choice was obvious upon visual inspection of the image, and could be made even before any analysis was performed.

Secondly, unsatisfactory peak detection could also occur when the gradient of a straight bone edge was not sufficient to produce a large peak in the accumulator, or if the bones were not straight enough to produce compact votes in the accumulator. Examples of these types of poor detection are shown in Figure 5.9 and 5.10, and correspond to the two peaks that were incorrectly detected in the 28 peak, six image development set. In these cases, increasing the value of e would have done little to improve the detection, because the incorrect peak was already one of those that had been selected. This means that the incorrectly selected peak(s) needed to be modified, rather than selecting additional—possibly also incorrect—peaks.

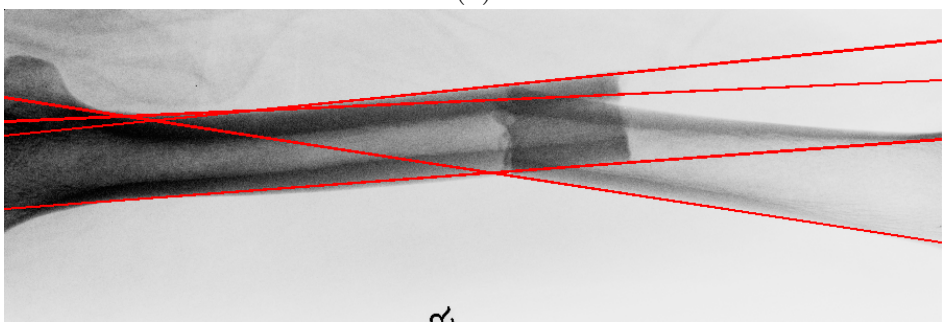
To allow this modification to occur, a small level of user input was required. The user was allowed to select the incorrect line on the back projected image, and also the location (x, y) of the edge where they believed the line should be located. This edge location should correspond to one of the i locations examined in the ranked sums method, so the list (ρ_i, θ_i, S_i) was ranked again based on how well each set of parameters matched the selected location. This was easy to perform since the parameters should match Equation 5.1 if the corresponding line passed through the selected point. Once this list was created, the parameters with the best match and the highest region sum were chosen as the new line parameters.

5.7 Finding line segments

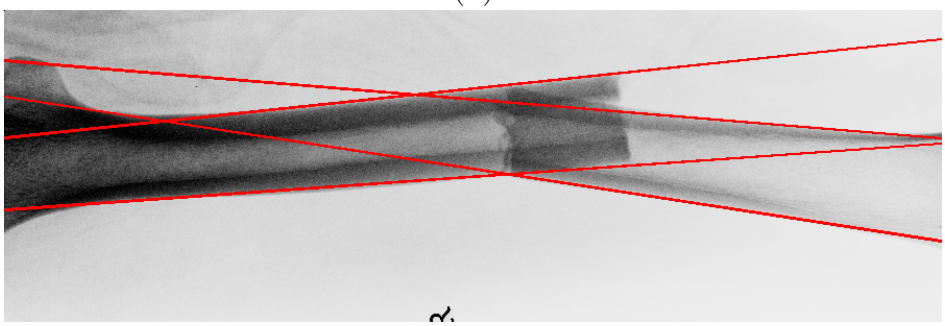
So far, the methods discussed in this chapter could detect straight lines represented by the parameters (ρ, θ), but they were unable to provide any information regarding the length or the endpoints of the lines. For some applications, such as long-bone centre-line approximation and diaphysis segmentation (described in Chapter 6), this type of information was crucial. The task of finding line endpoints has been the focus of a number of articles that are discussed in this section. The methods described are grouped into those that determine segment information from the Hough space, and



(a)

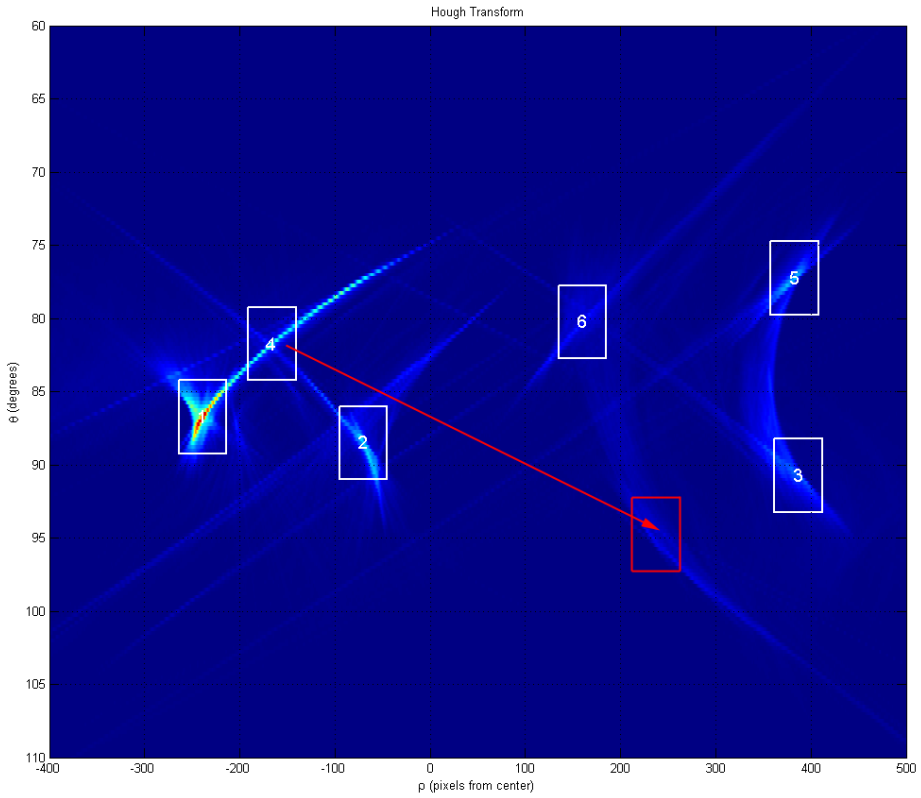


(b)

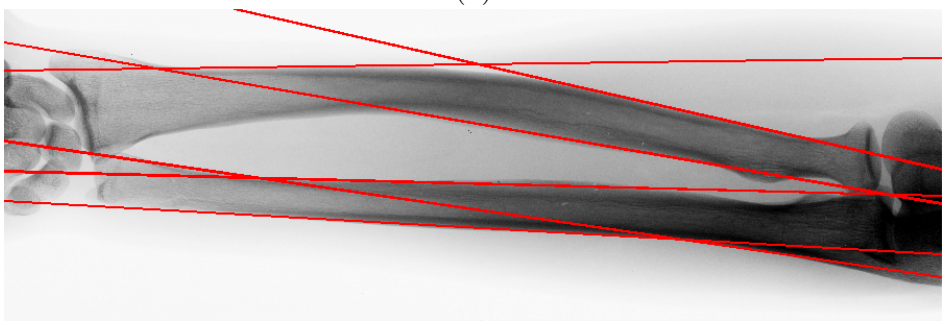


(c)

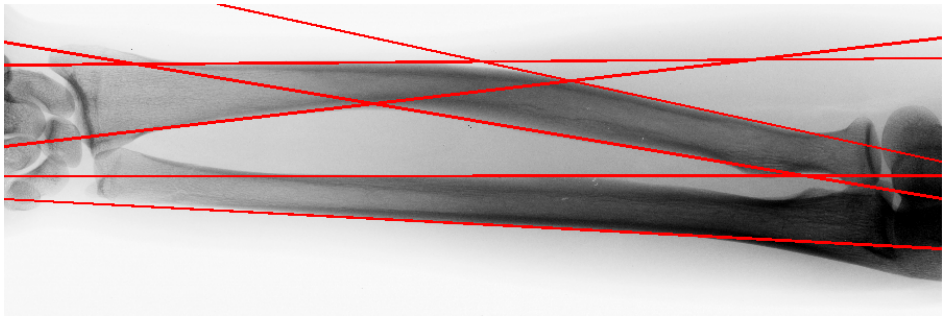
Figure 5.9: (a) Example of the modified Hough Transform of a development image in which a peak was incorrectly detected by the ranked sums method. This incorrect detection occurred due to the large gradient difference between the proximal and distal ends of the bone. The red rectangle shows the location of the correct peak. (b) The resulting incorrect line on the original image. (c) The correct line chosen by user input.



(a)



(b)



(c)

Figure 5.10: (a) Example of the modified Hough Transform of a development image in which a peak was incorrectly detected by the ranked sums method. This incorrect detection occurred because the bone edge is curved such that the votes are spread over a very large distance in the accumulator. The magnitude of the gradient at that point is also weaker. The red rectangle shows the location of the correct peak. (b) The resulting incorrect line on the original image. (c) The correct line chosen by user input.

those that determine segment information from the image space.

5.7.1 Finding segments by analysing the Hough space

The approach described by Niblack and Truong [77] computes a set of parameters to completely describe the line segment, by fitting a surface locally to the Hough array, where the parameters required are variables of the fit. Using this method, the algorithm analyses the Hough accumulator rather than the image space. The main limitation of this method is that only one segment at a given ρ and θ can be modelled, such that separate colinear segments are not detected. This was a major problem for long-bone images where there was a significant possibility that colinear segments were required to be detected. As a result, this method was unsuitable for long-bone line endpoint detection.

Akhtar and Atiquzzaman [8, 15, 14] determine the line length using a micro-analysis of the spreading of the votes in the accumulator array. As shown in Figure 5.2b, the pixels belonging to a line in the image space produce a butterfly shape in the parameter space. The length and position of the line determines the angle subtended by the wings of the butterfly, and this shape can be analysed to reveal a complete description of the line segment. The spread of the votes in the accumulator has already been described in Equation 5.9 on page 97, but this can be rearranged and modified to produce:

$$d = \frac{\left(N_{\rho}^{i} - 1\right) \triangle \rho}{\sin\left(\frac{\triangle\theta}{2} + d_{j,\rho} \triangle \theta\right)} \tag{5.16}$$

The papers by Akhtar and Atiquzzaman describe a method by which Equation 5.16 can be used to calculate the location of the line endpoints. While their method is demonstrated on simple images, it is not appropriate for complex real images such as long-bone x-rays. In addition, their fundamental assumption is that the butterfly shape be present to calculate the spreading in the accumulator. However as described in Section 5.2.3 on page 90, the modified Hough Transform removed the butterfly shape, so it was not possible to utilise this method.

Another method for locating the ends of the line segments is described by Costa

et al. [33], in which the Hough Transform is recalculated for only those angles and distances corresponding to peaks in the accumulator array. Each pixel in the input image that is found to contribute to that particular peak has its abscissae updated in a unidimensional histogram. The extremities of the straight lines are then determined using a connectivity analysis. A similar result is produced by simply computing a pixel-value cross-section along the line segment with parameters (ρ_i, θ_i) corresponding to the peaks in the Hough accumulator, and then projecting that line onto the x and y-axes before performing the connectivity analysis. Unfortunately, when implemented this method was found to produce very poor results, with the long-bone line endpoints very rarely detected correctly.

Thus, the unsuitability of the Niblack and Truong [77] method, and the poor results produced by the Costa et al. [33] and Akhtar and Atiquzzaman [8, 15, 14] methods, indicated that attempting to locate the long-bone line endpoints in the Hough accumulator was not a robust or reliable solution.

5.7.2 Finding segments by analysing the image space

The most reliable method is described by Skingley and Rye [97]. Their endpoint detection is a post-processing technique that focuses on the image space rather than the Hough accumulator. Each candidate line that corresponds to a detected peak is scanned using a one-dimensional gradient detector, consisting of a plus-minus mask that is moved along the line with parameters (ρ_i, θ_i) subtracting the sum of the following M pixels from the sum of the preceding M pixels. Homogeneous regions therefore produce a sum that is close to zero, while the line end points produce a large output. This procedure is then repeated for various values of M and averaging is performed to increase the weighting of the central pixels. The authors state that for their application typical operator sizes of $25 < M < 55$ were most successful.

To split the line into segments, the coordinates of the possible line end points are determined by thresholding the gradient detector output. To determine the existence of a line segment, the mean pixel value of each segment is calculated and compared to the mean of the pixel values along parallel segments on either side of the line.

A true segment will have a much higher mean than its parallel line segments, while a false segment will have a similar mean to its parallel line segment. This method is computationally intensive, since the swath along the line corresponding to every detected peak (ρ_i, θ_i) must be analysed with the gradient detector using multiple values of M, then the segments between endpoints must also be examined. Despite this, a modified implementation of this technique generally required only a small proportion of the calculation time of the modified Hough Transform.

When implemented, the Skingley and Rye [97] method was found to produce much more promising results than the methods described in Section 5.7.1. Thus it was decided that detection of the long-bone shaft endpoints would be performed in the image space.

5.7.3 The long-bone shaft endpoint detector

To accurately detect the long-bone shaft from the peak parameters, the Skingley and Rye method described in the previous section, needed to be altered. The first modification to their method was that it was not necessary to use a one-dimensional gradient detector, since accurate gradient information was already known from the smoothed image (as shown in Figure 4.15c on page 79), and could be used again here. Before checking where a detected peak (ρ_i, θ_i) matched a line in the image, feature points that did not belong to the line of interest could be removed from the gradient image. This was performed in a similar manner to the modified Hough Transform described in Section 5.2.3, where the direction of the gradient was utilised by ensuring that Equation 5.4 was satisfied. Here equation 5.4 was rearranged and used to modify the gradient magnitude $|\bigtriangledown I(x,y,t)|$ for any given i as follows:

$$\begin{aligned} E_i(x,y,t,\theta_i) = |\bigtriangledown I(x,y,t)| \qquad & \theta_i - \frac{\pi}{2} - r \leq \phi(x,y,t) \leq \theta_i - \frac{\pi}{2} + r \\ & \theta_i + \frac{\pi}{2} - r \leq \phi(x,y,t) \leq \theta_i + \frac{\pi}{2} + r \\ E_i(x,y,t,\theta_i) = 0 \qquad & Otherwise \end{aligned} \tag{5.17}$$

As a result, the magnitude of the gradient $E_i(x, y, t, \theta_i)$ remained the same when ϕ was close to θ_i, and was set to zero in all other cases. The same value of $r = 2.5°$ was used here. This effectively removed regions from the gradient image that could not be part of the line of interest, because they did not have the correct gradient direction.

Secondly, the described method assumed that the lines to be detected were very thin, which also implied that they were assumed to be completely straight. As a result, even the slightest deviation produced breaks in the segments and an inaccurate line detection. As discussed in Section 5.4, the presence of completely straight lines in long-bone images was a poor assumption, which often resulted in unsuccessful line endpoint detection. The Skingley and Rye method was modified to use a thicker strip—with a width of ± 20 pixels, chosen to match the average line width $b_a = 40$ pixels—around the line with parameters (ρ_i, θ_i). The use of a thicker strip ensured that even if the line was not completely straight, it should still be covered at least in part by the thick strip. Any time that the line deviated sufficiently outside the strip indicated that an endpoint had been located. To ensure that all pixels in the thicker strip contributed to the calculation, the sum across those 40 pixels was determined. The result was a one-dimensional signal in which a large value indicated the line was present in the strip at that point.

Thirdly, determining a good value of M for the Skingley and Rye method was problematic since it was found to vary significantly between images. More importantly, due to the modification to use a thicker strip, line deviation at the endpoints meant that the gradient did not always increase rapidly at those points. As a result, the plus-minus mask did such a poor job of detecting the ends of the lines that in many cases the largest outputs of the mask were at points on the line, rather than at its ends. Accordingly, the plus-minus mask was replaced with a dual stage thresholding method.

In the first stage of this new method, a threshold T_1 was applied to the one-dimensional signal—obtained from the thick gradient strip along the line with parameters (ρ_i, θ_i)—so that any regions with small values, and therefore not likely to be part of a line, were removed. In the second stage, only lines that were longer than

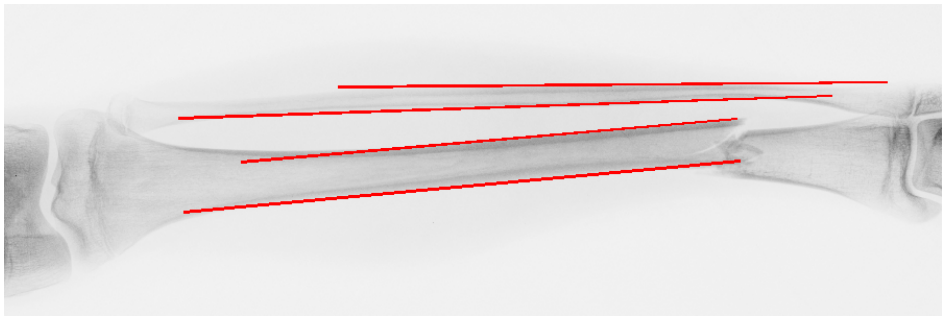


Figure 5.11: The lines detected using a modified version of the method described in [97], designed specifically to detect long-bone shafts.

the second threshold T_2 were selected. Since each signal was normalised, the thresholds were empirically chosen using the six image development set, to be $T_1 = 0.2$ and $T_2 = 0.02d$ where d was the maximum length of the line segment. Decreasing the first threshold tended to shorten the line segments and make them match the bone contour more closely at the end points, whereas decreasing the second threshold removed shorter lines from the group. The reverse was also true in both cases. In a test using the image development set, the end points of all 28 lines were correctly identified using this modified method with the selected thresholds, indicating that the poor performance with non-straight lines was effectively overcome.

Figure 5.11 shows that the lines detected by the modified algorithm were a good match to the bone edge. The endpoints were located where the bone edge deviates sufficiently away from the line. This line endpoint information has a useful application in the following chapter for long-bone diaphysis segmentation.

5.8 Parameter approximation evaluation

The algorithms developed in this chapter for long-bone parameter estimation were evaluated by testing them on the 44 images in the test set. The results of this evaluation are shown in Table 5.1.

The ranked sums method

When the modified Hough Transform and the ranked sums peak detection methods were applied using the standard values of e (chosen based on anatomic location) a total

of 160 peaks were detected. Manual inspection of these peaks in the image space showed that 142 (88.8%) were in locations that correctly corresponded to the edges of the bone shaft. However, the incorrect peaks were dispersed across the 44 images such that only 12 images (27.3%) were completely correct (all required peaks were correctly identified, with no false peaks present). The highest rates of detection occurred in the humerus and femur, where all the detected peaks corresponded to points of interest on the shaft. Unfortunately this did not necessarily make them completely correct, because in some cases the chosen value of e was insufficient for producing an adequate approximation. Disregarding the single correct humerus image, the tibia/fibula images were the most likely to be correct, with 7 out of 15 (46.7%) images completely correct. The incorrect peaks were spread fairly evenly between the radius/ulna and the tibia/fibula images (45.8% versus 46.7%, respectively).

As one would expect, local contrast within the image affected accuracy. Where there was low contrast at the bone and soft-tissue boundary, the smaller gradient resulted in less of a contribution to the Hough accumulator, and a weaker peak. The contrast variation was affected by the type, thickness and density of the soft tissue surrounding the bone, with a high proportion of muscle lowering the gradient at the boundary. Of course the contrast could also vary significantly within an image, along the length of the bone. To compound this problem further, it was theoretically possible for the deep cortical wall to produce a stronger gradient than the superficial wall, so selecting the strong peaks in the Hough Transform produced a line inside the bone, rather than on the bone tissue interface. This situation was hard to identify automatically, but was shown not have a significant effect on the outcome of the fracture detection because the deep and superficial cortical walls were parallel, so the angle parameter θ_i was the same in either case. This was not a significant problem, because in the later stages of the algorithm it was more important to accurately identify the angle θ_i than the offset ρ_i.

		Bone location				View		Total
		Humerus	Radius / Ulna	Femur	Tibia / Fibula	AP	LAT	
Number of images		2	24	3	15	29	15	44
Initial number of peaks e		4	90	8	58	110	50	160
Correctly detected	Peaks	4 (100.0%)	79 (87.8%)	8 (100.0%)	51 (87.9%)	98 (89.1%)	44 (88.0%)	142 (88.8%)
	Images	1 (50.0%)	4 (16.7%)	0 (0.0%)	7 (46.7%)	8 (27.6%)	4 (26.7%)	12 (27.3%)
Incorrectly detected	Peaks	0 (0.0%)	11 (12.2%)	0 (0.0%)	7 (12.1%)	12 (10.9%)	6 (12.0%)	18 (11.3%)
	Images	0 (0.0%)	11 (45.8%)	0 (0.0%)	7 (46.7%)	12 (41.4%)	6 (40.0%)	18 (40.9%)
Adjustment of e required	Peaks	1 (25.0%)	21 (23.3%)	5 (62.5%)	4 (6.9%)	24 (21.8%)	7 (14.0%)	31 (19.4%)
	Images	1 (50.0%)	14 (58.3%)	3 (100.0%)	2 (13.3%)	14 (48.3%)	6 (40.0%)	20 (45.5%)
Final number of peaks e		5	109	13	58	134	51	185
Incorrect segments	Segments	0 (0.0%)	1 (0.9%)	1 (7.7%)	2 (3.4%)	3 (2.2%)	1 (2.0%)	4 (2.2%)
	Images	0 (0.0%)	1 (4.2%)	1 (33.3%)	2 (13.3%)	3 (10.3%)	1 (6.7%)	4 (9.1%)

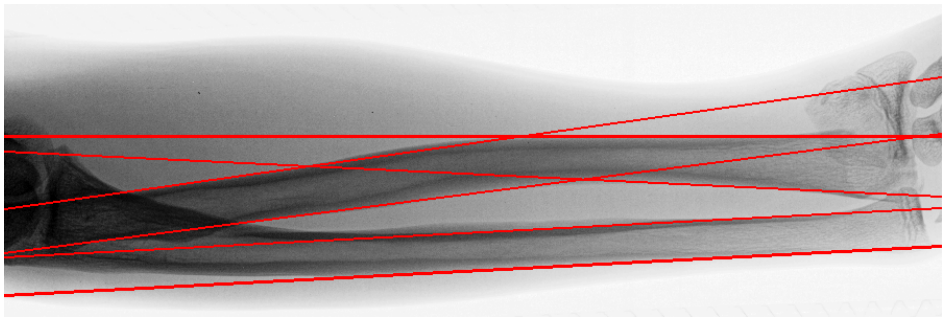
Table 5.1: The results obtained from testing the long-bone parameter approximation algorithm on the 44 image test set, broken down into categories based on bone location and view. The first two data rows show the number of images and the initial number of selected peaks in each category. The following two show the number of correct and incorrect peaks and images in each category. An image was deemed correct if all the peaks it contained were correctly detected, otherwise it was deemed incorrect. The fifth row shows the number of peaks not detected due to an incorrect value of e, and the number of images in which this occurred. The last two rows show the final number of peaks utilised, and the number of incorrect segments and images. Again, an image was deemed incorrect if it contained any incorrect segments.

Adjusting the value of e

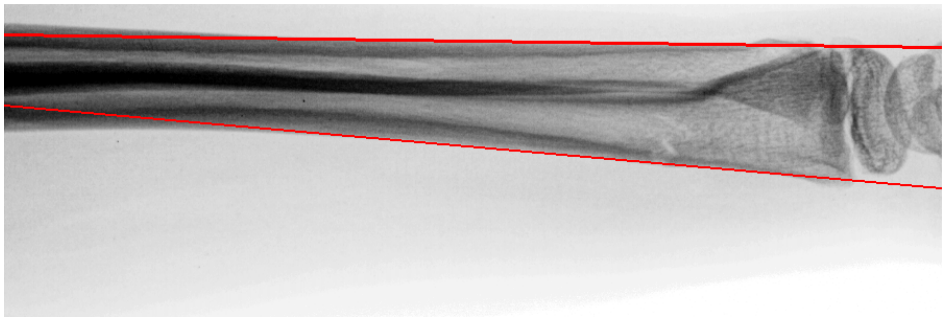
In some cases the number of peaks detected was either too large or too small, requiring e to be manually adjusted. This occurred in 20 images, and of those 18 required e to be increased (eight were increased by one and ten increased by two), while only two required e to be decreased. After e was changed in all the required cases, the final number of peaks detected was 185, an increase of 15.6%. The number of radius/ulna images in which insufficient peaks were detected was quite large (54.2% of all radius/ulna images, and 68.4% of all images with incorrect e), due to the slight curvature of the radius, as shown in Figure 5.12a. This meant that in some cases, the bones of the forearm would be better approximated using six lines rather than four. In those cases where the value of e was correct but a peak was still incorrectly chosen, the correct peak was selected by using the manual input methods described in Section 5.6. Thus in all cases where incorrect peaks were detected, the correct peaks were obtained using a small amount of user interaction.

As previously mentioned at the end of Section 5.4.1, the value of e was determined based on anatomic location. However it was also influenced by the x-ray view, because bones could superpose in a ML image such that identification of all the required bone edges would be difficult. In this situation the edges may have still been correctly detected, but alternatively it may have been necessary to either manually decrease e or manually correct the edges. Halving the value of e for some ML images of the extremities proved to be effective, since e was incorrect in only 14% of ML images. An example of superpositioning of the radius and ulna is demonstrated in Figure 5.12b, where the value $e = 2$ was used.

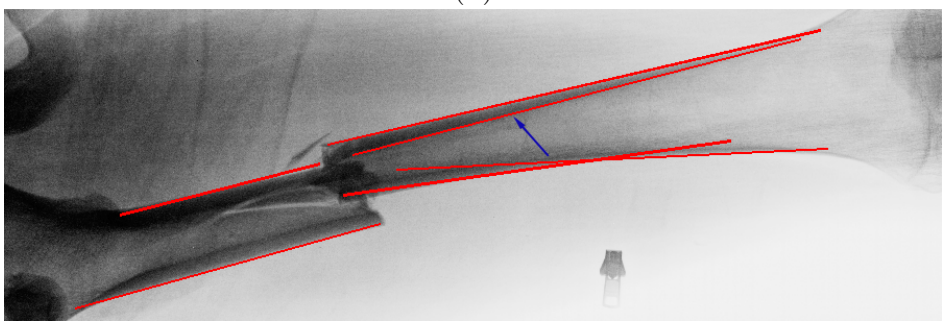
Parameters for all the bone segments were likely to be correctly determined, provided the edges in the image were of sufficient length and intensity to create large peaks in the accumulator. Midshaft fractures tended to produce peaks closer in magnitude, because each bone fragment was of a similar length. If the fracture was extremely proximal or distal then sometimes only the longer bone segments were accurately located (even when e was increased) since the small segments did not make sufficiently large contributions to the parameter space relative to the long segments. As each image



(a)



(b)



(c)

Figure 5.12: (a) In over half (54.2%) of the radius/ulna images, the curvature of the radius required e to be increased from four to six, to ensure a good detection occurred. In these cases the radius was approximated by four lines rather than two. (b) Overlapping of the radius and ulna in a ML image could make identification of the four bone edges difficult. (c) Incorrect line segment detection that resulted from the line matching the inner cortical wall of the femur. The incorrect segment is indicated by the arrow.

varied, it was not possible to reliably quantify this effect, or to determine where the fracture needed to be located so that all segments would be correctly detected. This occurred in 3 of the 44 test images, and one of the development images (shown in Figure 5.8).

Long-bone shaft endpoint detection

Once all 185 peaks were correctly located, the line endpoint detector was applied. Manual inspection of the resulting lines showed that only four of the 185 lines (2.2%) were incorrect. The incorrect lines were in four different images from the 44 image set, corresponding to 9.1% incorrect images. The incorrect lines all occurred as a result of good matches with edges belonging to other features within the image, that happened to lie underneath the line of interest. An example of an incorrect segment detection in a femur is shown in Figure 5.12c. The remaining line segments were all well detected, and the parameters of the straight sections were therefore known.

5.9 Summary

After the first stage of analysis where the important edges were extracted from the x-ray image, the edges had to be interpreted to understand the contents of the image. This chapter described a method where the long-bone shaft was approximated by a series of straight lines with parameters (ρ, θ), using the previously extracted edges. To determine these parameters, the Hough Transform was utilised to convert spatially extended patterns into spatially compact features, thereby reducing the analysis to a local peak detection in the parameter space. The relationship to the Radon transform was examined, along with the advantages and disadvantages of the Hough Transform. For this implementation the standard Hough Transform was modified in two ways. Firstly, instead of thresholding the input image, the accumulator was incremented by the magnitude of the gradient so that strong edges automatically made larger contributions to the accumulator space than weak ones. Secondly, to increment an accumulator cell, the direction of the gradient at that pixel had to closely match the angle θ being examined, thereby reducing the chance of false peaks appearing in the accumulator.

The software implementation of this modified transform was also discussed, along with approximate calculation times for a typical x-ray image.

Simply obtaining the Hough accumulator for a particular image did not yield significant information, since this came from detecting and analysing the peaks within the accumulator. The problem of peak spreading due to oversampling in the parameter space was examined, along with equations to quantify the maximum spread in both the θ and ρ directions. As a result of this spreading, the largest cells in the accumulator were not necessarily the peaks that were required. Therefore the ranked-sums method was introduced to detect the required number of peaks e in the accumulator. Once the parameters were determined, the lines could be projected back to the original image where they should approximate the long-bone edges. The parameters did not provide any information about the endpoints of the lines, but a method of determining these end points was also described.

Results from testing the modified Hough Transform and ranked sums method for peak detection showed that close to 90% of all peaks in the 44 image test set were accurately detected with no user input. In some cases it was necessary to manually alter either the position or number of lines detected to ensure that a good approximation occurred. Once this had been performed, almost 98% of the line segments corresponding to those lines were correctly detected by the long-bone shaft endpoint detector. These results provide valuable information that will be useful in the subsequent two chapters, so that semi-automated diaphysis segmentation and bone fracture detection can be performed.