

5

Feature Detection, Extraction and Representation

5.1 Introduction

The fundamental problem in feature detection and extraction is generically defined as the location of some set of pixels in an image that correspond to some features of interest in the image and the extraction of those features from the background information. The features could be for example, simple objects such as hot spots (containing just a few pixels of similar intensity) or more highly complex objects such as urban regions (containing many thousands of complex interrelated pixel intensities). The later could be used in its own right or as a cueing aid for more detailed post processing, such as the detection of road networks, junctions and buildings in order to allow registration of imagery to maps for navigation, image distortion correction, change detection as well as map-updating. It could also be an attention/region cueing device for image interpreters or indeed as feature extraction for subsequent inclusion into an image database or for content based compression. In this chapter examples of both types of feature extraction will be given followed by representation of the extracted complex regions.

5.2 Low Level Feature Detection

One of the most powerful features in any modern battlefield surveillance system is the ability to be able to automatically detect and track targets. The amount of information being presented to operators is increasing at an alarming rate and needs to be reduced and managed more appropriately. Any system that can intelligently filter the data to present results of possible targets without all the intermediate information is likely to be of significant benefit.

The problem considered here, is generically defined as the detection of small point sized 'hot' targets in the imagery on the basis of their emitted thermal radiation. These potential targets being within some operator specified sizes, but typically just 2×2 to 10×10 pixels. The resulting detection can then be used as a cueing aid either for a narrow field of view imager and/or a human operator for subsequent target recognition. The requirement from

the current algorithm in this instance was not necessarily to locate all the targets or indeed just the targets, but to locate possible areas of interest from a chosen portion of the imagery for further analysis. The wide field of view sensor produces very small potential targets. These targets have low contrast and signal to noise ratios and can make their detection a difficult problem.

5.2.1 *The Process*

Most stages of a process like this may need to be deliberately chosen due to their suitability for future implementation in special DSP hardware modules for real-time performance. The process is shown schematically in figure 1. There are a number of stages only the main ones are described in more detail below.

5.2.2 *Destriping*

Due to sensor imperfections (or the imaging conditions) the imagery that is available for a particular task may be unsuitable, however it may be all that is available so it has to be used. The imagery used in this example contained marked fixed pattern stripe artefacts (top of figure 2). A destriping stage of the algorithm is therefore required to reduce or even to eliminate the banding effects that result in this type of imagery. It is likely that the next generation of infra red imager for this application will produce images of improved quality with imperceptible striping artefacts and hence the destriping algorithm will become unnecessary.

The destriping algorithm (Booth & Radford 1992) employed removes bias variation between adjacent scan lines. Two adjacent scan lines i and $i+1$ are equalised by determining the distribution differences between adjacent pixels along the scans. The majority of entries should reflect differences in the baseline between the two scan lines. An additive correction to scan line $i+1$ is obtained from the median of the difference distribution, the median being used as it is a more statistically robust measurement. This process is then repeated using scan lines $i+1$ and $i+2$ (bottom of figure 2).

The disadvantage with this particular approach is that a software implementation can be relatively slow. An alternative scheme was developed as an intermediate measure (*prior to hardware design and implementation of the full destriper*). The approach is to model the sensor responses in order to estimate a set of corrective factors. If we can assume that the image is uniform over a number of neighbouring scan lines then any differences should be due the sensor itself. The median of each image row is obtained and the maximum of all medians is taken as the maximum sensor response. The difference of the maximum response and the median of each row can then be used as an additive amount for that row. The analysis can be done on the first frame in the sequence and then at successive frames, the corrective amounts (stored in a look up table) are then added to each row as frames from the image sequence arrive for processing.

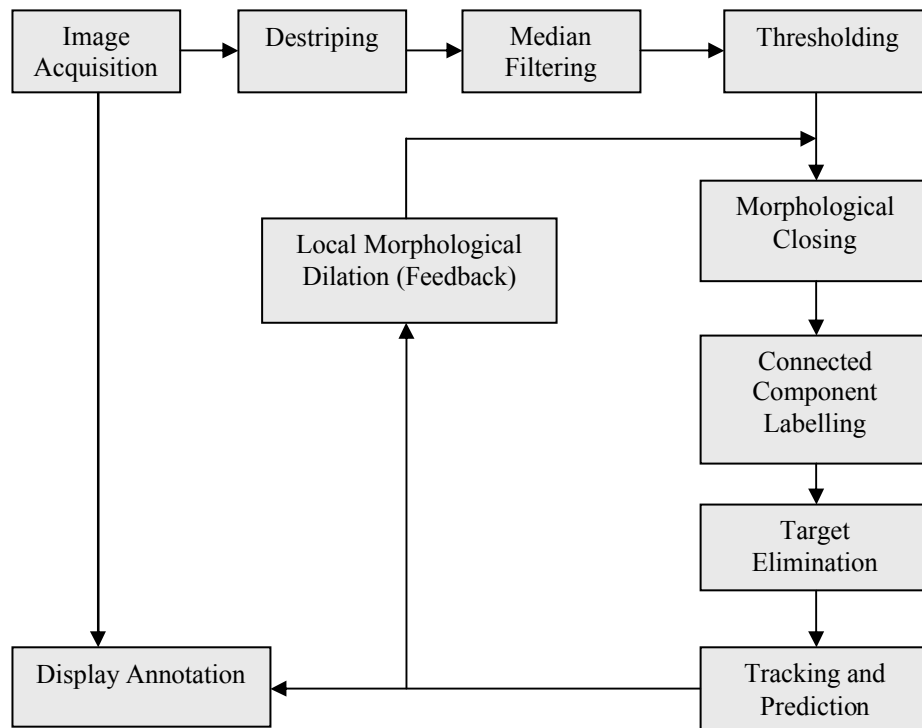


Figure 1, Algorithm Block Diagram

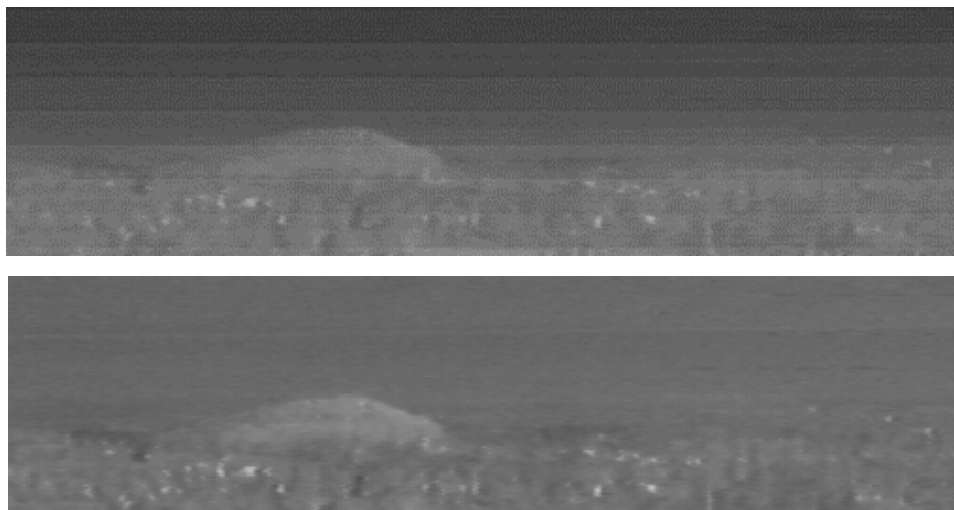


Figure 2; Destriping

5.2.3 Median Filtering

The use of a median filter is to remove salt-and-pepper type of noise. Given a small window over an image the pixel values are likely to be homogeneous with only a small number of them being attributable to noise. These noisy pixels tend to be at either extreme of the grey level distribution and therefore are unlikely to be selected as the output from the filter (*the output being the median of the ranked input values*). The filter has the advantage of reducing noise without a smoothing effect on the image and various fast implementations are available. In this instance since the possible targets are very small only a 2×2 median filter¹ is applied, however this does remove a number of points in the imagery which can be classed as noise.

5.2.4 Thresholding

There are various adaptive thresholding schemes that have been used, these rely on computing some localised measure which is then used as the basis for the thresholding. Unfortunately these approaches can have a tendency to produce a significant amount of noise. This can result in the problems of over and under segmentation caused by incorrectly chosen thresholds. In this application a significant amount of noise resulting from the thresholding will pose a major problem due to the small sizes of some of the possible targets.

Because of this difficulty a thresholding scheme based upon the global mean and variance of the portion of the imagery being processed is used. A threshold could typically be set at $\mu + 3\sigma$. This proved to be an acceptable level (figure 3) for this application domain but an option can be provided in the algorithm for this to be varied interactively during run time. An alternative to the simple thresholding schemes which is a thresholding selection method (Otsu 1979) that aims to minimise the ratio of the between class variance to the within class variance of the two classes².

¹ An important point to be considered is that a median filter when used for noise suppression can in fact be replaced by grey scale morphology. This point does not really apply to this particular algorithm as the filter we use only has a small kernel, it is however important and worth mentioning. A morphological opening followed by a closing operator can achieve the same effect as a median filter. Morphology has two distinct noise suppression stages, the opening suppresses positive noise impulses whilst the closing suppresses negative noise impulses.

² i.e. the higher the variance between the classes the better the separation (in a simple example the classes could be defined as being the region of interest and background)



Figure 3; Thresholded image

5.2.5 Morphological Closing

Mathematical morphology is a shape or geometrical approach to image processing which is based upon set theory. Two basic operations are dilation and erosion, these being used to form higher level operations such as closing and opening. Morphological closing is defined as a dilation followed by an erosion and smooths contours, fuses narrow breaks and long thin coves and eliminates small holes. The closing of a function (image) f with a kernel g is defined as

$$f^g(z) = (f \oplus g) \circ g \quad (1)$$

where dilation $\oplus$ is defined as

$$(f \oplus g)(z) = \max \{f(y - z) + g(z) : y \in Z^2\} \quad (2)$$

and erosion as

$$(f \circ g)(z) = \min \{f(y + z) - g(z) : y \in Z^2\} \quad (3)$$

A fuller and more detailed description of morphology can be found in numerous papers in the literature (Sun & Rubin 1987, Haralick 1988, Maragos 1990, Hussain 1991). The reasons for applying a morphological closing operator are twofold. Consider the thresholding of an image this can and is likely to result in the fragmentation of objects. Firstly, an object, which is a target, could be fragmented into several parts thus leading to possibly several targets being detected instead of one. Secondly an object which is not a target (*perhaps by virtue of its size*) could be fragmented such that small parts of it are then likely to be identified as possible targets.

To resolve this problem a morphological closing operator³ is applied in an attempt to piece the fragments back together, figure 4. The kernel that is applied is deliberately kept small enough to try to avoid the problem of merging several genuine targets together.

³ The morphological closing operator is defined as an erosion followed by dilation. The dilation of an image X by a structuring element K is the union of the translates of X taken over all points of k . The erosion of an image X by a structuring element K is the set of all elements c of E^N for which K translated to c is contained in X . There is also a morphological filter known as the COMOC filter, which is the Mean of a Close-Open and a Open-Close as well as adaptive morphology which is a structure preserving form of morphology (Wu 1994).



Figure 4; Morphology

5.2.6 *Connected Component Labelling*

Given that an image has been pre-processed and thresholded each of the clusters of pixels in the resulting binary image needs to be examined in turn to decide if it is a potential target. The objective of connected component labelling (CCL) is to take a binary image and then apply a segmentation in order to obtain a set of connected regions each of which is uniquely labelled (Smith & Radford 1993, Klette & Zamperoni 1996). Typically 8-connectivity is used⁴ and all clumps (disjoint regions) of connected pixels within the image are labelled with a unique identifier, figure 5. Once labelled, the regions can be extracted and further analysis can be performed based on size and shape etc. This fundamental process is important in many applications and can be used as an input to shape recognition tasks.



Figure 5; Labelled objects

5.2.7 *Target Elimination*

Once a labelled image has been obtained regions can be discarded according to a number of criteria. The operator may initially have specified a set of bounds for targets of interest. Regions, which have a width, height, aspect ratio or total area outside these constraints, are discarded. In this example due to the limited number of on target pixels it is not possible to discard objects based upon a true description of shape. In addition to this there is no knowledge of the type of targets, which are unknown in this application domain, therefore a simple rejection based on area and aspect ratio is performed.

5.2.8 *Tracking and Prediction*

Once an acceptable set of possible targets has been obtained the x and y co-ordinates of the centre points for each are passed to the tracking process (Blackman 1986). The tracking is

⁴ For a 3x3 neighbourhood the centre pixel has 8 neighbours.

used to introduce an element of temporal consistency into the algorithm. This is used to resolve a number of issues such as false targets (*due to segmentation errors or genuine noise in the imagery*), targets appearing and disappearing, and overlapping targets etc. Once these issues have been resolved then a prediction stage is performed to estimate the target's position in the next frame. Targets develop a history after n frames and therefore isolated noise, which appears for $n-1$ frames, or less will not be tracked and can be eliminated.

The initial part of the tracking is actually an association stage where observations are associated with tracks. This uses a standard assignment optimisation algorithm (Shamblin & Stevens 1974) which was modified (Brown et al 1993) to deal with targets, which appear and disappear. It was also modified by the author resolve a problem of several observations being identical distances from a given track but outside the permissible (*gated*) regions for all other tracks. This condition appeared to cause the standard algorithm to fail to converge to an optimum assignment.

A Kalman filter (Kalman 1961, Salmond 1981) is a classical approach for the prediction of a targets new position. It is the optimal filter for performing tracking when the equations of target motion are known. If the x and y target co-ordinates can be decoupled then it has been shown, (Bridgewater 1978) that the Kalman filter can reduce to one called the α - β filter which is much simpler and requires no matrix multiplication.

5.2.9 Local Morphological Dilation (Adaptive Feedback)

One important point arising from thresholding operations is the difficulty in setting a threshold level at just the correct level for detection of all targets. As was mentioned previously it is possible for noise to be included and also for a genuine target (*but maybe one which is very small and/or emitting low thermal radiation*) to be excluded from the thresholded image. Incorporating feedback from the tracking algorithm can reduce the effect of this. Essentially we are using the track confidence to adapt the output from the segmentation. This is achieved by performing a local morphological dilation in an area around known targets (*targets that have developed a history*). This attempts to enlarge the thresholded output to a point where it would be accepted by the target elimination stage and effectively reduces the number of dropouts due to weak targets. As expected if a target genuinely disappears then this approach will have no effect.

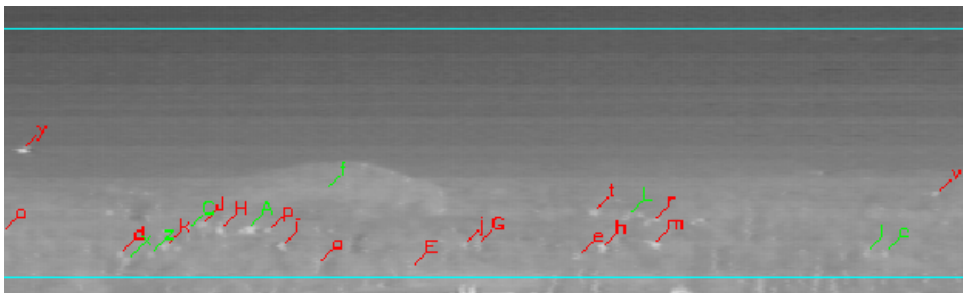


Figure 6, Target detection

Figure 6 shows the final results with the detected targets indicated by their track identifiers, red marks objects which have developed a history and have some degree of consistency,

whilst green signifies those that currently have not. These detections would then cue a narrow field of view sensor.

5.3 High Level Feature Detection

Here we consider more sophisticated approaches to feature detection which rely on the use of Pearl Bayes (or Belief) networks (Pearl 1986) to combine several sources of evidence in order to achieve feature detection. A Pearl Bayes network is a directed acyclic graph (figure 7). In this graph nodes B, C and E represent different statistical information, whilst node A represents the belief in detecting an urban patch. A graph G is a pair of sets (V, A) for which V is non-empty (figure 7). The elements of V are vertices (*nodes*) and the elements of A are pairs (x, y) called arcs (links) with $x \in V$ and $y \in V$.

Consider the simple network shown in figure 7, the equations for computing the belief and propagation of information are as follows⁵. Consider the link from node B to A then the graph G consists of the two subgraphs G_{BA}^+ and G_{BA}^- . These two subgraphs contain the datasets D_{BA}^+ and D_{BA}^- respectively.

5.3.1 Belief Equations

From figure 7 it can be observed that node A separates the two subgraphs $G_{BA}^+ \cup G_{CA}^+ \cup G_{EA}^+$ and G_{AF}^- . Given this fact equation we can write the equation:

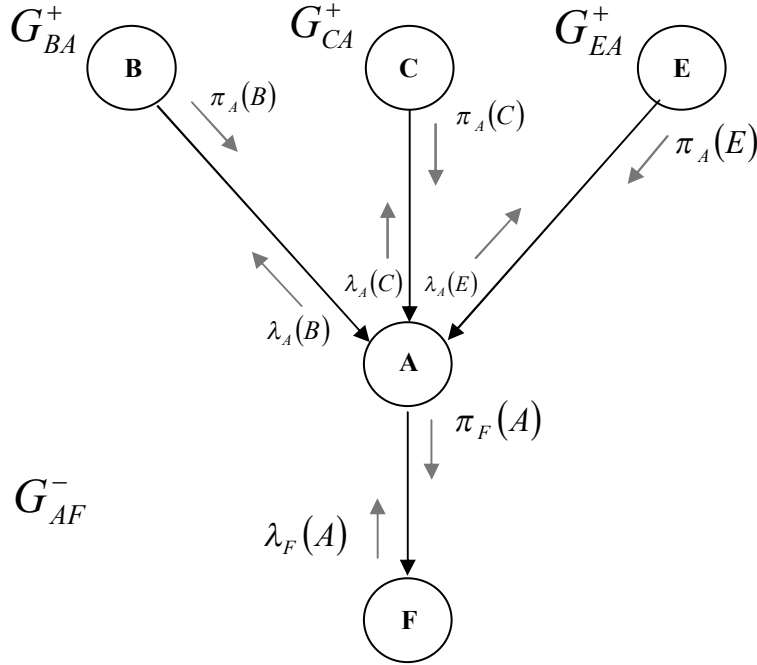


Figure 7, Example Belief Network

⁵ The equations are derived along similar lines to those derived by Pearl where in his example node A has two predecessors and two successors.

$$P(D_{AF}^- | A_i, D_{BA}^+, D_{CA}^+, D_{EA}^+) = P(D_{AF}^- | A_i) \quad (4)$$

by using Bayes rule the belief in A_i can be defined as:

$$\begin{aligned} BEL(A_i) &= P(A_i | D_{BA}^+, D_{CA}^+, D_{EA}^+, D_{AF}^-) \\ &= \alpha \cdot P(A_i | D_{BA}^+, D_{CA}^+, D_{EA}^+) P(D_{AF}^- | A_i) \\ &= \alpha \cdot P(D_{AF}^- | A_i) \left[\sum_{j,k,l} P(A_i | B_j, C_k, E_l) P(B_j | D_{BA}^+) P(C_k | D_{CA}^+) P(E_l | D_{EA}^+) \right] \end{aligned} \quad (5)$$

where α is taken to be a normalising constant. It can be seen that this equation is computed using three types of information:

- Causal support π (from the incoming links).
- Diagnostic support λ (from the outgoing links).
- A fixed conditional probability matrix (which relates A with its immediate causes B , C and E).

The equations, which form the above information, are given as follows. Firstly the causal support equations:

$$\pi_A(B_j) = P(B_j | D_{BA}^+) \quad (6)$$

$$\pi_A(C_k) = P(C_k | D_{CA}^+) \quad (7)$$

$$\pi_A(E_l) = P(E_l | D_{EA}^+) \quad (8)$$

Secondly the diagnostic support equation is given by:

$$\lambda_F(A_i) = P(D_{AF}^- | A_i) \quad (9)$$

Finally the conditional probability matrix is defined to be:

$$P(A | B, C, E) \quad (10)$$

The belief equation can now be rewritten to obtain the belief at node A:

$$BEL(A_i) = \alpha \lambda_F(A_i) \cdot \sum_{j,k,l} P(A_i | B_j, C_k, E_l) \pi_A(B_j) \pi_A(C_k) \pi_A(E_l) \quad (11)$$

The belief at nodes B, C and E can be obtained from the equations:

$$BEL(B_j) = \alpha \pi_A(B_j) \lambda_A(B_j) \quad (12)$$

$$BEL(C_k) = \alpha \pi_A(C_k) \lambda_A(C_k) \quad (13)$$

$$BEL(E_l) = \alpha \pi_A(E_l) \lambda_A(E_l) \quad (14)$$

5.3.2 Propagation Equations

The propagation equations for the network are derived as follows firstly the diagnostic ones. From a previous analogy with equation (9) it can be formulated as:

$$\lambda_A(B_i) = P(D_{BA}^- | B_i) \quad (15)$$

by partitioning the D_{BA}^- into its component parts, namely A , D_{AF}^- , D_{CA}^+ , D_{EA}^+ we can obtain

$$\lambda_A(B_i) = \alpha \sum_{j,k} \left[\pi_A(C_j) \pi_A(E_k) \cdot \sum_l \lambda_F(A_l) P(A_l | B_i, C_j, E_k) \right] \quad (16)$$

likewise for $\lambda_A(C_j)$ and $\lambda_A(E_k)$

$$\lambda_A(C_j) = \alpha \sum_{i,k} \left[\pi_A(B_i) \pi_A(E_k) \cdot \sum_l \lambda_F(A_l) P(A_l | B_i, C_j, E_k) \right] \quad (17)$$

and

$$\lambda_A(E_k) = \alpha \sum_{i,j} \left[\pi_A(B_i) \pi_A(C_j) \cdot \sum_l \lambda_F(A_l) P(A_l | B_i, C_j, E_k) \right] \quad (18)$$

5.3.3 Causal Equations

These are defined using a similar analogy as follows.

$$\pi_F(A_i) = P(A_i | D_{BA}^+, D_{CA}^+, D_{EA}^+) \quad (19)$$

and from this we then derive the equation

$$\pi_F(A_i) = \alpha \left[\sum_{j,k,l} P(A_i | B_j, C_k, E_l) \pi_A(B_j) \pi_A(C_k) \pi_A(E_l) \right] \quad (20)$$

An important point to realise from these equations (16-18, 20) is the fact that they demonstrate that the parameters λ and π are orthogonal to each other i.e. perturbation of one will not affect the other. Hence evidence propagates through a network and there is therefore no reflection at boundaries.

The above Belief network approach has been adapted (Ducksbury 1993) for the detection of urban regions, using parallel processing systems for improved performance. The belief network is used in a multi-resolution sense (figure 8) to combine statistical measures into the detection (or belief) of the required hypothesis (in this case an urban region)⁶.

The problem is approached by taking several statistical measures from small patches of an image these are treated as a set of judgements about the content of the patches. These statistics are a count on the number of edges, the number of extrema and a classification of the grey level distribution type. These statistics are quantised down into a smaller (more manageable) number of levels. The number of edges and extrema are both reduced to 5 levels, whilst the distribution type remains with 4 possibilities. It is important to stress that any suitable measure that provides the required textural discrimination could have been used. In fact corner detection has also been used as that can provide a good clustering around regions of interest.

The statistics are then used to produce a set of judgements; for example an expert might upon looking at a particular window issue a report of the form (0.0,0.7,0.9,0.6,0.0). This means that he believes there is a 70 % chance that level 2 describes the number of edges, 90% chance that its level 3 and 60% for level 4. But he believes there to be no chance of it being levels 1 or 5.

⁶ Some work has also been carried out into using the Belief Network approach at a higher level of abstraction, i.e. for the combination of several urban region finding algorithms to generate a single confidence measure. This is illustrated on the top left of figure 9, in which each coloured outline represents the result of applying a different feature detection algorithm. The belief network being shown is in red (this being just a coarse approximation from a region cueing perspective), yellow – a wavelet based algorithm, green – a fractal based algorithm and pale blue represents the probabilistic combination of all three results.

An alternative to actually combining feature detection algorithms together is to make an informed choice between the algorithms based upon some search technique, such as an evolutionary approach using a genetic algorithm. This is described in (Ducksbury 1998a and 1998b) for the above case as well as that of optimising the parameters for a particular application of a single feature detection algorithm. The worse case scenario from a genetic search would be the choice of a set of parameters, which result in no feature detection (when there should be according to the ground truth), this particular set of parameters would be evolved out of the population.

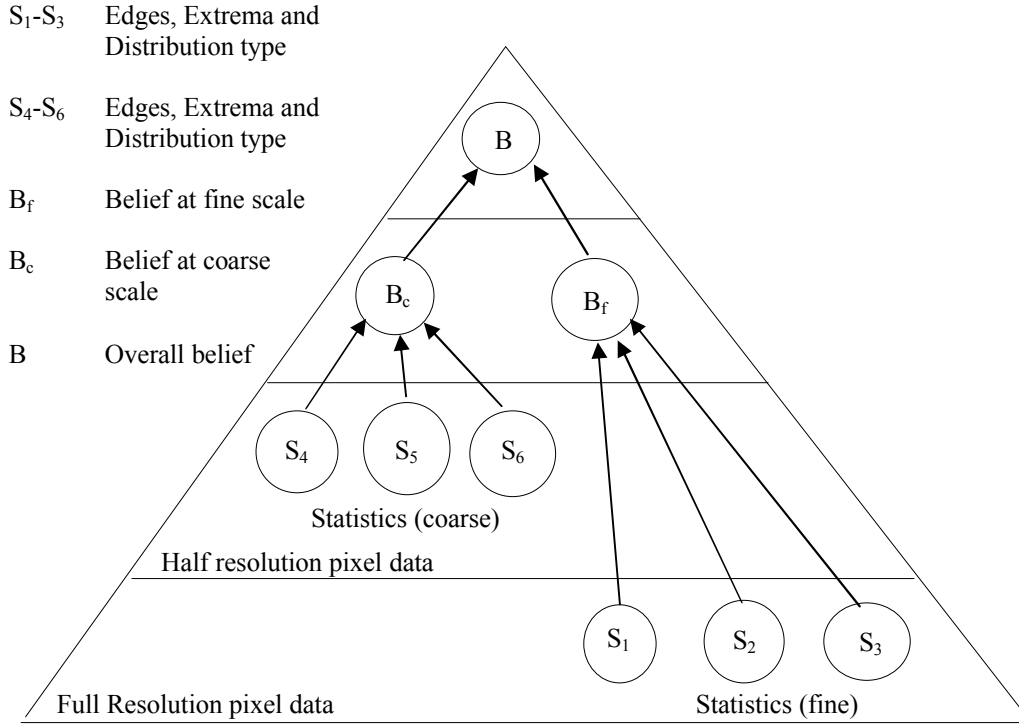


Figure 8, Multi-resolution Approach

For the Belief at nodes B_f , B_c and B in figure 8, 3 variables were chosen to denote the possible values that the region can have, namely (*low, medium, high*).

The fixed conditional probability matrices (eg $P(B_f|s1,s2,s3)$ etc) which are the prior information and relate the given node with its causal information are created along similar lines to that used in (Devijver 1989). They are based upon the natural assumption that the probability of an event at a given node should be greater if its causal information is tightly clustered together (all in agreement) than it should be if the causal information is further apart (disagreement). For the $P(B|B_f,B_c)$ matrix (which relates the beliefs from the fine and coarse resolutions) slightly more emphasis is given to the causal information received from the coarse resolution belief.

$P(B_{f_i}|s1_j,s2_k,s3_l)$ is described formally as

$$P(B_{f_i}|s1_j,s2_k,s3_l) = \begin{cases} 0.75 & \text{if } i = j = k = l \\ 0.25 / \alpha & \text{if } (i \neq j = k = l) \wedge (0 < |i - j| \leq C) \\ 1.0 / \beta & \text{if } \neg(j = k = l) \\ & \wedge (\max(j,k,l) - \min(j,k,l) \leq 2C) \\ & \wedge (\min(j,k,l) \leq i \leq \max(j,k,l)) \\ 0.0 & \text{otherwise} \end{cases} \quad (21)$$

such that $\sum_{j,k,l} P(B_{f_i} | s1_j, s2_k, s3_l) \leq 1 \forall i$

where $C = 1$ and i, j, k, l range over the number of variables in $B_f, s1, s2$ and $s3$ respectively. α and β represent the number of different values of i satisfying the constraint.

$P(B | B_f, B_c)$ is defined as

$$P(B_i | B_{f_j}, B_{c_k}) = \begin{cases} 0.9 & \text{if } i = j = k \\ 0.7 & \text{if } (i = j) \wedge (|i - k| \leq 1) \\ 0.3 & \text{if } (i = k) \wedge (|i - j| \leq 1) \\ 0.6 & \text{if } (i = j) \wedge (|i - k| > 1) \\ 0.1 & \text{if } (i = k) \wedge (|i - j| > 1) \\ 0.0 & \text{otherwise} \end{cases} \quad (22)$$

such that $\sum_{j,k} P(B_i | B_{f_j}, B_{c_k}) \leq 1, \forall i$ where i, j, k range over the number of variables in B, B_f and B_c respectively.

Figure 9 shows typical feature detection for a number of different applications using this approach. For urban region detection, experience suggests that the use of a corner detector as an additional (*or replacement*) source of evidence gives improved results. As one would expect corners will cluster more tightly around the area of interest. The belief network technique has also been applied (top right) as a means of reinforcement of existing classifiers (Ducksbury et al 1995) for the improved detection of vehicles, the evidence in this case being vehicle shadows, tracks and formations.

The output from equation (11) can be plotted directly to obtain a probability surface, which shows us the confidence in the particular hypothesis that is of interest. For the detection of driveable regions, as shown in figure 9 (lower left) a typical probability surface is shown in figure 10. Here we can see the confidence in a driveable region dropping off as we move away from the road. The peak on the upper right of this surface corresponds to an area of ‘road like’ (*smooth*) texture along the side of the building. This region has been eliminated by virtue of the fact that it is not connected (*in magnitude*) to the main driveable region⁷.

⁷ As a side issue to this, a cyclist passing in front of the camera would generate several statistical measures, which are then fused together resulting in a reduction in the probability surface, i.e. the cyclist would therefore not be considered to be a driveable area !.

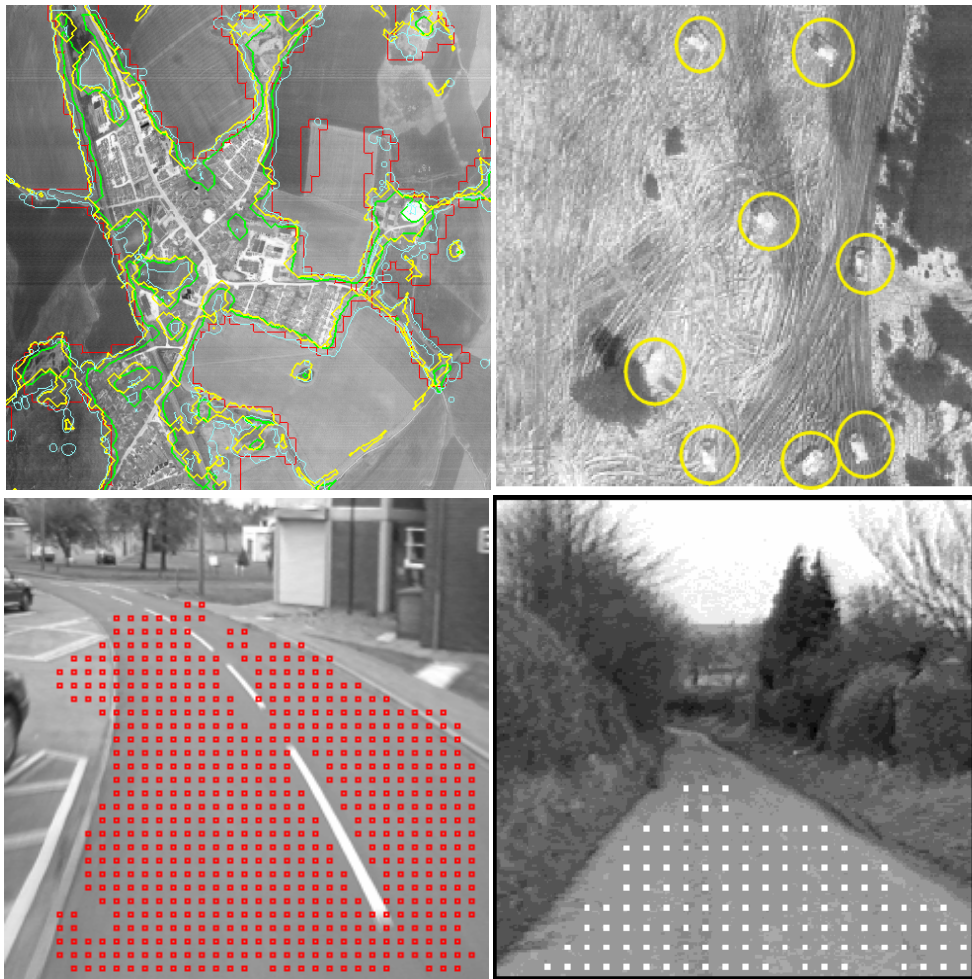


Figure 9 : Top left - Possible urban area outlines, Top right - Target reinforcement,
Bottom – Driveable region detection urban and rural

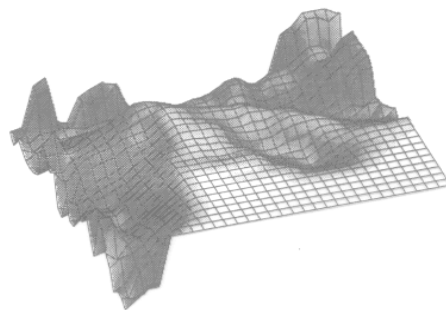


Figure 10; Probability surface for figure 9, bottom left

5.4 Feature Representation

Once a region is obtained then there may be the issue of how to represent its shape. In this instance we have chosen to use the curvature scale space approach (Mokhtarian 1995), which provides a very concise representation for objects. We initially explored this approach for the representation of objects such as aircraft and ships in an image database (Ducksbury 1997).

The technique requires some explanation and for this we refer to figure 11. For a given object (in this illustration an aircraft line drawing⁸) the raw outline is obtained as a simple (x,y) co-ordinate file⁹ (*lower left*). By iteratively convolving the (x,y) data with a gaussian function, the outline will be successively smoothed, this being shown along the left-hand side of figure 11. The outline ultimately reduces to a convex object at which point the process converges. As the gaussian convolution is repeated with larger values of scale parameter σ , at each iteration the zero crossings of curvature are located and marked on the u - σ plane. The curvature scale space (CSS) image itself, as shown (top right), plots contour position u against scale space σ . If a line is drawn horizontally across this image then this represents some particular scale, the points on this line effectively marking the zero crossings (*points of inflexion*) on the objects contour. Hence the first line near the bottom has numerous points (*corresponding to the many small features on the object*) whilst the next line has just 6, which mark the three concavities (*two major ones either side of the rotor blades, one small one below the tail*). The iterative process terminates when no more zero crossings are located (*essentially no more concavities are indicated*) and the object has reduced to a convex shape, illustrated at the top left of the figure.

From the curvature scale space image the major peaks are located. The (x,y) location of each of these peak marks the concavities relative contour position and scale, it is these peaks that are used as the shape description for subsequent searching. Hence for the line drawing in figure 11 the six largest peaks are chosen, typically therefore just a handful of pairs of integers are used to encode the objects shape. Figure 12 shows a curvature scale space image on the left for the urban region outline that is shown on the top left of figure 9, this also shows on the right one of the smoothed outlines (*in this case taken at scale $\sigma = 2.0$*) together with points of zero-crossing.

Figure 13, (top) shows an example of a database of approximately 200 aircraft and ships together with the outline, a smoothed contour and the curvature scale space image for a particular object. Figure 13 (bottom) shows an example of a database search, in this example for a particular type of helicopter (match result 0). The top six matches are shown together with their cost function indicating the degree of match (increasing from zero).

⁸ Aircraft outline data obtained by processing information supplied by Janes Information Group, Brighton road, Coulsdon, Surrey.

⁹ The input here being a set of line drawings of the aircraft. These are processed by thresholding, recursive filling of the background, inversion and selection of the main object, extraction of the edge, skeletonisation to obtain a single pixel wide boundary followed by the tracking of pixels on the boundary and output to the datafile.

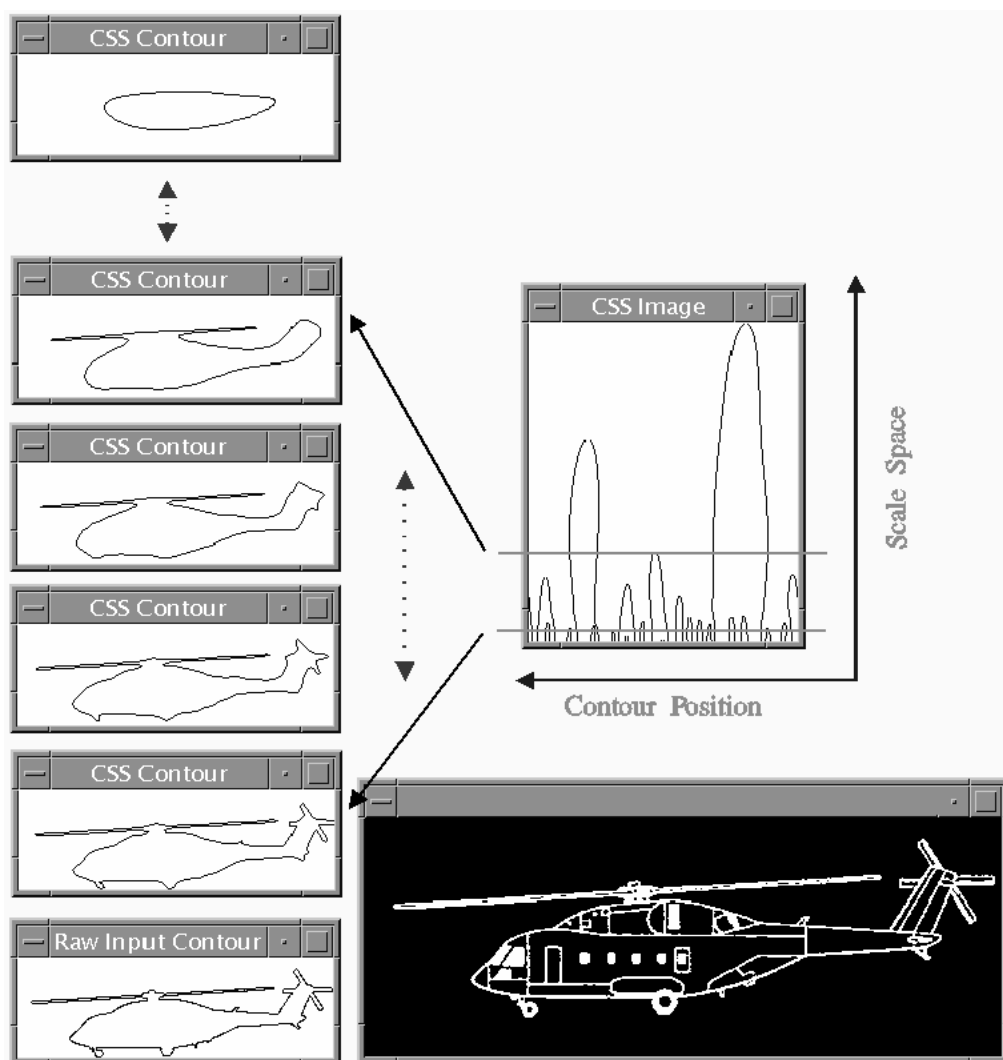


Figure 11: Curvature Scale Space Representation

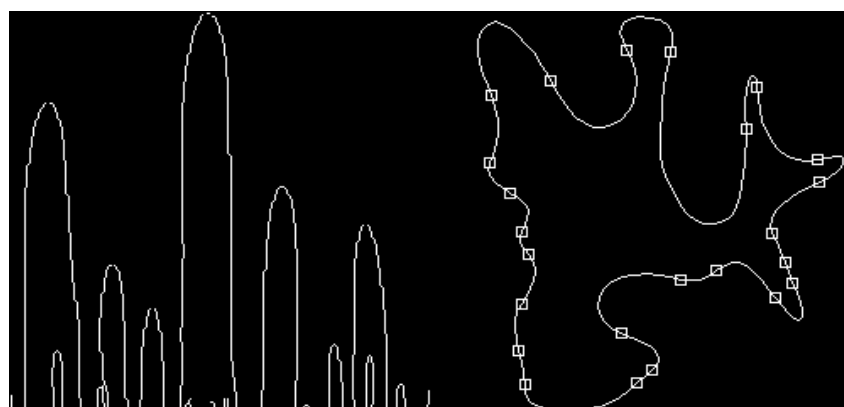


Figure 12 : Left - Curvature scale space image; Right - Outline representation at scale 2.0

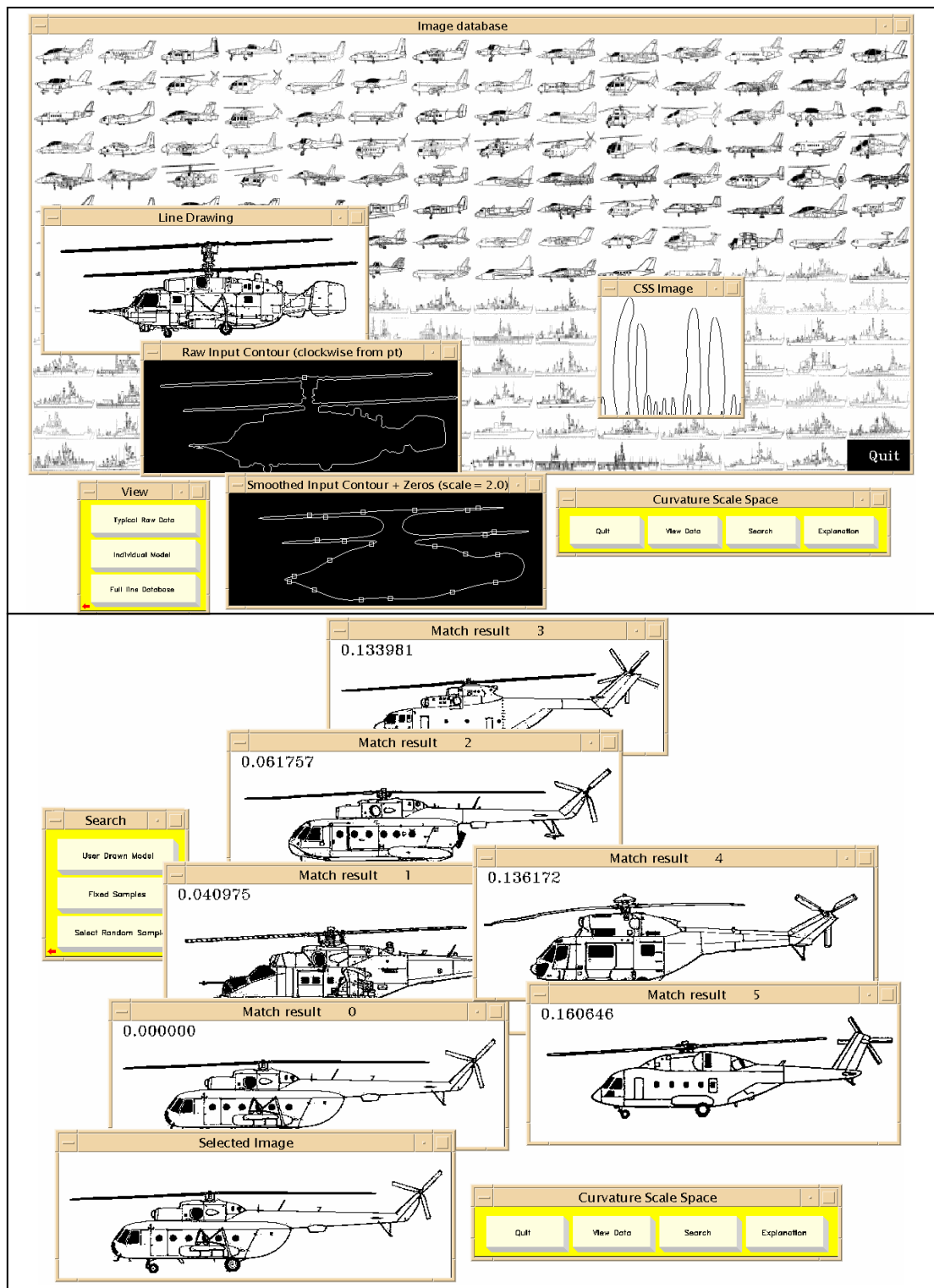


Figure 13; Object database and searching

5.5 References

- Blackman S.S., 'Multiple Target Tracking with Radar Applications', Artech House, 1986.
- Booth D.M., Radford C.J., 'The detection of features of interest in surveillance imagery: Techniques Evaluation', Unpublished DERA report, November 1992.
- Bridgewater A.W., 'Analysis of 2nd and 3rd order steady state tracking filters', *AGARD Conference Proceedings* no 252, 1978.
- Brown G., Smith R.W.M., Radford C.J., 'Target Acquisition and Tracking of a Staring Array Sequence on CHIP', Unpublished DERA report, September 1993.
- Devijver P.A., 'Real-time modelling of image sequences : Based on Hidden Markov Mesh Random Field Models' in 'Decision making in context', a Course on Statistical Pattern Recognition by P.A. Devijver and J. Kittler, Surrey University, 1989.
- Ducksbury P.G., 'Parallel texture region segmentation using a Pearl Bayes Network', *British Machine Vision Conference*, University of Surrey, pp 187-196, Sept, 1993.
- Ducksbury P.G., 'Driveable region detection using a Pearl Bayes Network', *IEE Colloquium on Image Processing for Transport Applications*, London, 9-Dec, 1993.
- Ducksbury P.G., 'Image Content Descriptors: Feature Combination and Representation', (unpublished) DERA technical report, March 1997, UK
- Ducksbury P.G., Varga M.J., 'Region based image content descriptors and representation', *6th IEE Image Processing & its Applications*, IPA-97, Trinity College, Dublin, July 1997.
- Ducksbury P.G., Booth D.M., Radford C.J., 'Vehicle detection in infrared linescan imagery using belief networks', *IEE 5th Int. Conf. Image Processing & Applications*, Edinburgh, 3-6 July 95.
- Booth D.M., Reno A., Foulkes S., Kent P.J., Hermiston K., Lewis S., Ducksbury P.G., 'Provision of an image intelligence feed into the strategic information fusion testbed', *SPIE Aerosense 97*, Orlando, April 20-25th, 1997.
- Ducksbury P.G. Varga M.J., Kent, P.J., Foulkes S., Booth D.M., 'Genetic algorithms for automatic algorithm and parameter selection in ATR applications', *SPIE Aerosense 98*, vol. 3371, 11th-17th April, Orlando, 1998
- Ducksbury P.G., 'Image segmentation and the use of genetic algorithms for optimising parameters and algorithm choice', Workshop on non-linear model based image analysis, *Noblesse workshop on non-linear model based image analysis (NMBIA-98)*, University of Strathclyde, 1-3 July, 1998.
- Haralick R.M., 'Mathematical Morphology and Computer Vision', *22nd Asilamer Conference on signals and computers*, Pacific Grove, CA, US, 31-Oct/2-Nov 88.
- Hussain Z., 'Digital Image Processing: Practical Applications of Parallel Processing Techniques', Ellis Horwood, 1991.
- Kalman, Bucy, 'New results in Linear Prediction and Filtering', *Journal of Basic Engineering*, vol. 83-D, pp 95-108, 1961.
- Klette R., Zamperoni P., 'Handbook of Image processing operators', John Wiley & Sons, pp 314-319, 1996.
- Maragos P., Schafer R.W., 'Morphological Systems for Multidimensional Signal Processing', *Proc. of IEEE*, vol. 78, no 4, April 1990.
- Mokhtarian F., 1995, 'Silhouette-Based Isolated Object Recognition through Curvature Scale' Space, *IEEE Trans Pattern Analysis and Machine Intelligence (PAMI)*, 17, 5
- Otsu N., 'A thresholding selection method from grey level histograms', *IEEE Trans Systems, Man & Cybernetics*, vol. 9, 1979, pp 62-66.
- Pearl J., 'Fusion, Propagation, and structuring in Belief Networks', *Artificial Intelligence* 29 (1986) pp 241 - 288.
- Salmond D.J., 'The Kalman Filter, The α - β filter and smoothing filters', Royal Aircraft Establishment, report TM-AW-48, February 1981.
- Shamblin J.E., Stevens G.T., 'Operations Research: A fundamental Approach', McGraw-Hill, 1974.
- Smith R.W.M., Radford C.J., 'Development of a connected component labeller DSP module for CHIP', Unpublished DERA report, October 1993.
- Sun F.K., Rubin S.L., 'Algorithm development for autonomous image analysis based on mathematical morphology', *Proceedings of SPIE 845: Visual Communications and Image Processing II*, 1987.
- Wu DeQuan., 'Morphological filters in image analysis', PhD Thesis, Dept. Engineering Science, Oxford University, 1994.