Proceedings of the SURVEYING & SPATIAL SCIENCES BIENNIAL CONFERENCE 2011
21-25 November 2011, Wellington, New Zealand

Compressing Images Using Contours

Gabriel Scarmana
University of Southern Queensland
Toowoomba, Australia
gabriel.scarmana@usq.edu.au

ABSTRACT

A method for the vectorisation of digital images into contour maps with subsequent conversion of the contours to a pixel format is presented. This method may offer an alternative spatial image compression which is computationally inexpensive and can be directly applied to compressing certain classes of grey-scale and/or colour (RGB, 24 bits) imagery of any size.

The feasibility of this study is based on research which shows that pixel based imagery can be sufficiently and accurately represented by their contour maps if a suitable contour model and scale selection method is used (Elder and Goldberg, 2001).

The compression process is based on filtering and eliminating those contours that may contain redundant information. Contours extracted from digital images may contain multiple redundant data (i.e. intersecting or nested contours), any of which might logically be used as a basis for discrimination or, on the other hand, used in the reconstruction of an original captured image.

As per current image compression techniques the proposed method does not require special hardware and, if combined with existing encoding schemes, it can be efficiently used for image transmission purposes due to its relatively modest storage requirements. Depending on the applications, and the amount of contour lines employed in the reconstruction of an image, the process allows for various levels of image accuracy while preserving visual integrity and reducing compression artifacts.

KEYWORDS: contouring, image compression, image reconstruction, image processing, image editing.

1 INTRODUCTION

Image compression schemes are based on the fact that any set of data can, and generally does, contain redundancies. In digital images, data redundancies can exist as repeated patterns and other forms of common pixel intensity information between multiple pixels of the image. The goal of image compression is to characterise these redundancies and code them to a new form that requires less data than the original (Weeks, 1996).

All image data compression schemes are basically twofold as they involve both a compression operation and an inverse decompression operation. The compression operation converts the original image into a compressed image data form. The decompression operation converts the compression image data back to its original uncompressed form (Weeks, 1996).

The measure of the amount of compression achieved in an image compression operation is obtained from dividing the data size of the original image by the data size of the compressed image. The result is called the compression ratio. The higher the compression ratio is, the smaller the compressed image has become. In all cases, the aim is to maximise compression ratios in

image compression operations while still meeting application requirements (Duperet, 2002). These requirements generally involve compressed image quality (for lossy compression schemes), time to compress and decompress the image, storage and the computational effort (Russ, 2007).

The type of compression scheme where the compressed data is decompressed back to its exact original form is called lossless data compression. It is devoid of losses, or degradation, to the data. The counterpart to lossless data compression is called lossy data compression. Lossy compression schemes introduce degradations to the data they compress (Gonzalez, Woods and Eddins, 2009). However, they do so in a way that is tolerable for the intended application. A lossy compression scheme which has become an industry standard is referred to as the Joint Photographic Expert Group (JPEG).

The JPEG image data compression standard handles grey-scale and colour images of varying resolution and size. It is intended to support many industries that need to transfer and archive images. This standard is commonly used in a lossy mode. However, there is also a lossless mode with reduced compression performance. The JPEG scheme is also adjustable. For instance, the amount of retained information can be changed producing variable compression ratios and inversely proportional decompressed image quality. Hence the JPEG algorithm can be fine-tuned to meet an application's requirements of compressed image data size and decompressed image quality (De Jong and van der Meer, 2004).

Other relatively new techniques for lossy image compression include the use of wavelet transforms (ECW-Enhanced Compressed Wavelets) and fractals. Compared to JPEG, these methods have not gained widespread acceptance for use on the Internet at the time of this writing. However, these techniques are very promising as they produce higher compression ratios and higher visual image quality than JPEG. A detailed explanation of lossless and lossy image compression schemes and techniques is beyond the scope of this work, and the reader is referred to Russ (2007), Gonzalez and Woods (2008) and Cunha, Zhou and Do (2006) for the theory and applications related to this important aspect of image processing.

For the purpose of this paper, the approach to solving the problem of image compression adopts techniques from the compression of contour maps. Compression here is achieved through the sequential filtering of contour data so as to remove redundant information which may have no or little effect in reconstructing an original digital image (Wang and Liu, 2006). Filtered contour data is then encoded to improve storage requirements, speed of transmission and reconstruction. As in the case of JPEG, the proposed compression technique is adjustable because, depending on the amount of discarded data, the storage capacity may vary at the expense of image quality. Two basic filtering techniques were tested in the proposed contour-based compression process, that is, string filters and contour exclusions.

String filters are designed to remove surplus vertices from contours strings. By way of a user defined tolerance, string vertices that are within the tolerance-based offset of two straight lines are removed. The tolerance value used normally depends on the data set and the job that the data is being used for. That is, the string filter removes vertices from contour lines that do not deviate by more than a specified offset tolerance from straight lines joining successive strings.

Contour exclusions can be achieved by comparing the centroid values of the area that these contours occupy to determine the shorter and longer distance to the centroids. These parameters may offer a useful constraint for deciding whether a contour is very similar to another and thereby could be removed. Also, for any group of at least three closely nested contours, it may possible to test for redundant contours by removing the middle contour and interpolating and/or diffusing between the other two. However, this process may have the disadvantage of smoothing and/or distorting particular high frequency information (sudden changes of pixel intensity values within an image) that may be necessary for a more accurate image reconstruction.

The statistical tests presented in the ensuing sections were carried out to determine the accuracy and integrity of images reconstructed from their contour representations. A comparison with the JPEG compression protocol was also conducted to evaluate the efficiency of the proposed contour compression technique in relation to storage requirements, accuracy and quality of visual details.

STANDARD DEFINITIONS

Digital image: A digital image is a discrete approximation of an image obtained by sampling points with discrete coordinates and quantising the values of each sample. It is formed by a finite number of sample elements equally spaced over a square grid with a rectangular shape. Each element is called a pixel and has an intensity value. The rows and columns of elements determine the spatial coordinates (x, y) of the pixel; and the intensity determines its grey-scale or colour value. In a grey-scale image, all pixels have shades of grey ranging from black to white. In the case of colour images, the intensity determines the colour of each pixel according to some colour model, such as RGB. See Figure 1 for an example of gray-scale continuous and digital images.

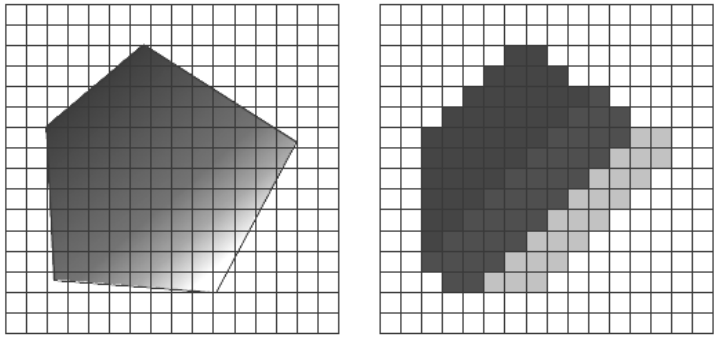

Fig. 1: Continuous image (left) and sampled digital image (right).

Pixel-based contour representations (Figure 2) require a lot of space because many pixels are needed; they are not scale independent since the number of pixels required changes with the size; and they are discrete. Pixel-based contours are generally not smooth unless numerous pixels are used. Given these downsides, a geometric representation or vectorisation provides a better way to represent contours, for example, with line segments defined by pairs of points. This representation does not have the problems associated with the discrete counterpart [3].

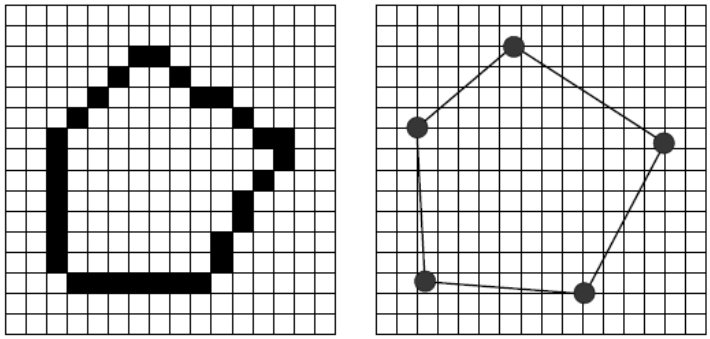

Figure 2: Pixel-based (left) and geometric-based (right) contour representations.

The definition of contour used in this work relates to the term used in cartography and surveying where a contour line joins points of equal elevation (height) above a given reference level (Watson, 1992). Note that this definition may differ from the one commonly used in image processing where contours relate to shapes, edges or lines and object boundaries (Elder and Goldberg, 2001).

CONTOUR DETECTION

The first step in the method is to generate contour lines for the gray-scale intensity values and assign coordinate values to pixels in the raster format image data. This is followed by interpolating between the pixels to find the coordinates of the points on the path of a contour with a

specific grey-scale intensity value. Hence, a digital image can be represented either as a three-dimensional surface with elevation z, or by using contours where each point (i.e. a vertex) of a given contour line corresponds to a position (x,y) with the same intensity value as the other points on the line.

Each pixel value can be assigned an x and y coordinate, which may for convenience be based on the coordinate of the centre of the pixel. Each pixel then has at least a pair of coordinates and a pixel grey-scale or intensity value. If two adjacent pixels have grey-scale values of 40 and 50 respectively, then a contour for grey-scale value 45 would lie between these two pixel centres. The coordinates in pixel space of a point on this contour can be estimated by interpolation, thus enabling the contours and/or the contour bounds to be estimated to sub-pixel accuracy.

For the class of images investigated in this work, the contour intervals or increments were selected based on the dynamic range of the image and/or on its histogram. Image histograms provide convenient, easy-to-read representations of the concentrations of pixels versus pixel intensity in an image. Using this graph it is possible to discern how much of the available dynamic range is used by an image (Russ, 2007). The dynamic range is the ratio of the highest (lightest) pixel intensity which a sensor records to the lowest (darkest) pixel intensity. The lightest pixel would correspond to the brightest intensity in an image; the darkest pixel to the deepest shadows.

In Figure 3 the pixel values of the .tif (taggered image file) image on the left, referred to as the *aerial view* (1000^2 pixels), fall between 15 and 215 (within a maximum range of 0 to 255) as shown on the right in Figure 3, with none in the other regions of the image. This indicates a relatively wide dynamic range of intensity values thus, in general, requiring a larger number of contours than an image with a narrow scale distribution (small dynamic range). The histogram in Figure 3 shows an example how many of the total number of pixels (3584 pixels) have an intensity value of 187 and the percentage (0.4%) they represent within the image of the *aerial view*.

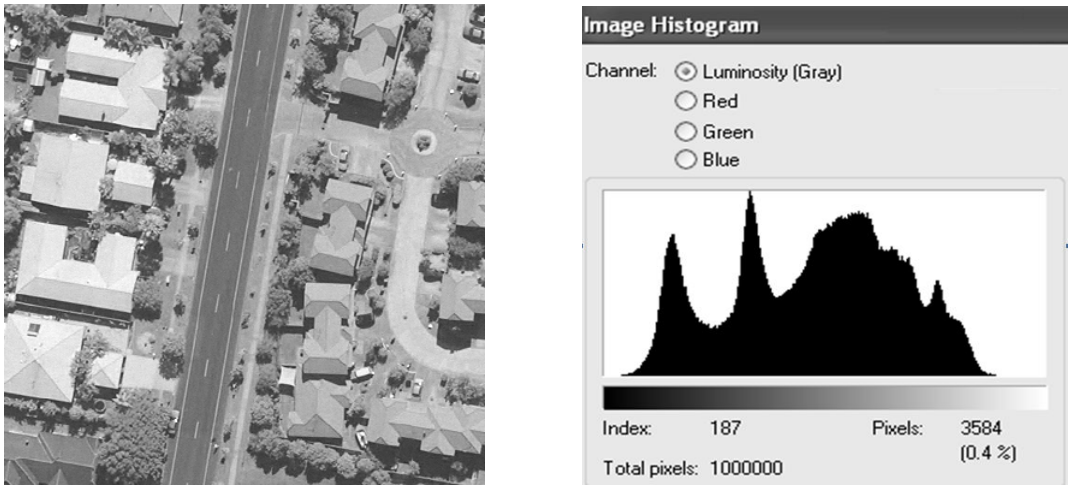

Figure 3: The histogram (right) shows that the distribution of pixel intensity values for the image (*aerial view*) on the left fall between approximately 15 and 215 within a range of 0-255.

Figure 4 illustrates a series of contour lines as produced for an array of uniformly spaced pixel intensity values. In this instance the contour increment was 4 units. The dots connecting the contour segments are referred to as vertices. There exist a number of methods and computer packages used for constructing contour lines from grid data points. The contours in Figure 4 were derived using a weighted average method of interpolation (Watson, 1992). Figure 4 (on the right) shows the contour lines for the *aerial view* in Figure 3 corresponding to the grey-scale intensity value of 100 only (full range being 0-255).

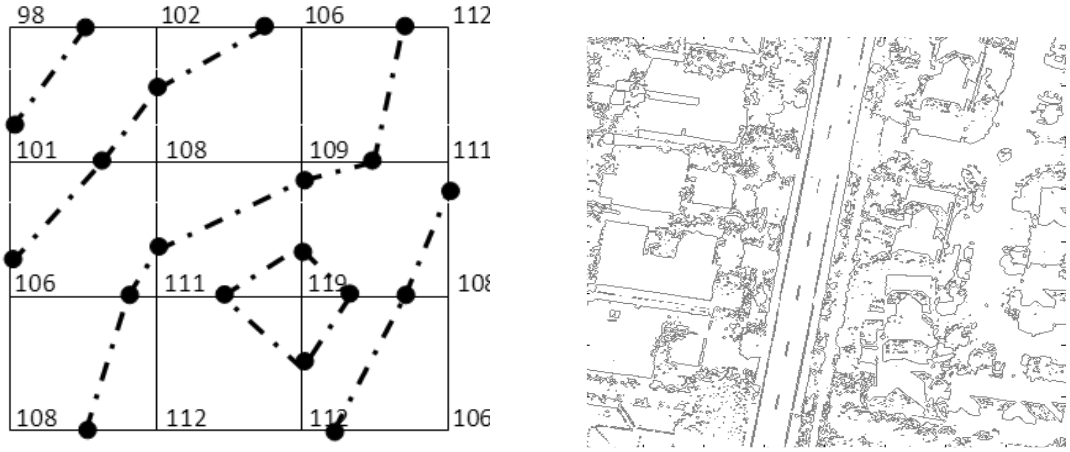

Figure 4 – Contours produced for a uniformly-spaced 4x4 grid of pixel intensity values (left) for contours incrementing in units of 4; the contour lines corresponding to the grey scale value of 100 in the *aerial view* of Figure 3 (left) are shown on the right.

IMAGE RECONSTRUCTION: FROM CONTOUR TO PIXELS

A number of tests were performed to investigate which contour increment would suffice to reconstruct images with a relatively large dynamic range such as aerial images, human faces, and landscapes. The tests were performed on a number of images by comparing the results of reconstructions via interpolating from respective contour representations with their original raster format.

Interpolating from contour lines to determine the value that would exist at the intersections of a regularly spaced grid is a classic problem in computational cartography and surface modeling. Traditional methods extend straight lines in a number of directions from the test point until these lines intersect the contour lines, the coordinates of these intersections are then used to interpolate the point of interest with a weighted average as described in Watson (1992) and Li and Heap (2006). These methods work efficiently if the contours are not nested.

However, using the above technique on nested contours usually results in an artifact appearing in the form of a cone. Since the outer contour line are longer than the inner ones, the averaging rule causes the surface between these two contour lines to droop, as if pulled down by gravity, in a scalloping or terraced effect. To avert these shortcomings, the method used in this research uses the filtered contours approach to extract bi-directional “cross-sections” in x and y at regular intervals.

To determine the pixel intensity values which would exist at the intersections of a regular grid using the randomly spaced nodes of these cross-sections, several interpolators may be used depending on the application and accuracy requirements. The interpolation technique adopted here is the Matlab function “griddata” (www.mathworks.com). This function fits a surface of the form z = f(x,y) to the data in the (usually) non-uniformly spaced vectors x,y,z (i.e., the nodes of the cross sections). “griddata” interpolates this surface at specified x_i, y_i locations (a uniform grid) to produce the required z values using the Bi-harmonic Spline Interpolation method devised by Sandwell (1987). The surface always passes through the input data points.

Since the interpolation methods determine decimal values for the points at the grid intersections, the values are rounded to the nearest integer as this is the norm for representing pixels intensities within a digital image. In addition, these values should be restricted within the interval 0-255 for the case of standard (8 bits) gray scale imagery.

TESTS AND RESULTS

The first test was aimed at determining the degree of accuracy and memory storage requirements that can be expected when reproducing a digital image from its contour representation. The contours increments in this exercise varied between 2 and 10. The results are based on the *aerial view* in Figure 3. Table 1 shows the results of this test. The Root Mean Square

(R.M.S.) error (Spiegel and Stephens, 1999) refers to the error of the differences between the original *aerial view* and the same image reconstructed using the vertices of grey-scale contours.

The figures in Table 1 show a linear degradation of the R.M.S error as the contour interval is increased. The table also illustrates the amount of memory required to store the contour vertices needed for the reconstruction of the aerial view in Figure 3. As the contour interval increases, the memory requirement decreases.

Table 1: Accuracy and memory requirements needed to reconstruct the original *aerial view* using contour increments between 2 and 10.

Contour increments	Accuracy R.M.S.	Memory required (.txt)	Maximum Fluctuation value	Minimum Fluctuation value
2	3	0.50 Mb	+2	-2
4	9	0.33 Mb	+9	-11
6	14	0.20 Mb	+25	-23
8	18	0.17 Mb	+30	-32
10	20	0.13 Mb	+47	-55

By way of comparison, the .txt file needed to store the x and y coordinates of the vertices for reproducing the *aerial view* (for contour increment of 8 intensity values) used approximately the same amount of memory (0.13 Mb) of a JPEG compressed version of the same image with compression ratio 10:1, while producing a more accurate image with superior visual details (Figure 5). In this figure, the difference in visual quality between the aerial view reconstructed from contour data and the corresponding JPEG image is evident. Indeed, the blocking effects, typical of a lossy JPEG protocol, were almost completely eliminated.

In addition, upon subtracting the JPEG version of the aerial view in Figure 5 from the original aerial view in Figure 3 generated grey-scale differences fluctuating between +39 (maximum) and -44 (minimum) with an R.M.S error equal to +/-25. By contrast the contour approach produced grey-scale fluctuation values ranging between +30 and -32 respectively with an R.M.S. error equal to +/-18.

For easy extraction, processing and transmission over a digital link the contours data was encoded and saved in a .txt file. The .txt file contains a two-row matrix where each contour line defined in the matrix begins with a column that defines the value of the contour line and the number of (x,y) vertices in the contour line.

Figure 5: A JPEG version (right, section only) of the *aerial view* (1000^2) for a compression ratio of 10:1 and the same image reconstructed using the vertices of contour lines with increments equal to 8 (left, same section).

The remaining columns contain the data for the (x,y) pairs. The x, y values are stored to the first decimal place and separated by one tab space. In the coding shown below 112 is the grey-scale intensity of a selected contour and the contour line is formed by 7 vertices (Gonzalez et al., 2009). The x coordinate of the vertices are in the top row (i.e. 17.2 17.1...16.7) whereas the corresponding y coordinates are placed in the bottom row (i.e. 100 99...95.2)

112	17 2	17 1	17	17	16 9	16 7	16 7
7	100	99	98	97 9	97	96	95 2

A second test was devised to establish the effect that filtering the contour data (prior to the reconstruction phase) has on the memory required to store such data. The same *aerial view* in Figure 3 was used in this test. To generate the grey-scale intensity contours, an increment of 5 grey-scale intensity values was selected for a range of 15-215 pixel intensity values; this range is a multiple of the increment value and includes all the grey scale variations within the original image. The intervals ranging from 15 to 215 thus produced 40 grey-scale contour line values.

The resulting contours were then filtered in succession using the two filters introduced in Section 1. The first filter applied aimed at eliminating nested contours. Within any group of at least three closely nested contours, the middle contour was eliminated. Upon completing this elimination phase, the contour lines were further filtered using a string filter. Figure 7 illustrates the principle behind a string filter. Table 2 shows the results of the reconstruction upon varying the tolerance tube (shown in Figure 7) from 0.2 to 1.0 unit of intensity. The table also indicates the associated levels of memory requirements for the .txt files needed to store the contour data.

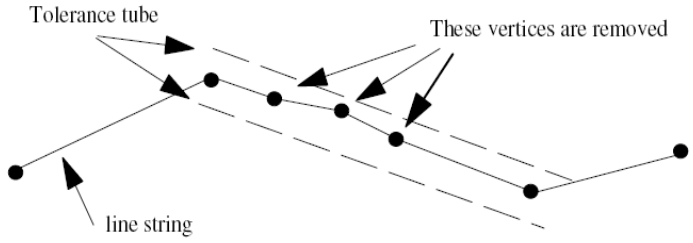

Figure 7: A string filter option is used to remove surplus vertices from contours lines.

By comparing Table 2 and Table 1 it may be concluded that a further reduction in memory requirements is obtained by filtering and eliminating contour data which may be considered as redundant and that for lower values of the tolerance tube width there are no major effects on the accuracy of the final reconstruction. For example, adding a string filter to the 5 grey-scale values contour, where the filter can have a tolerance tube size of up to 0.4, the resulting R.M.S. error (error 12, Table 2) is still lower than the R.M.S. error of the 6 grey-scale values contour (14, Table 1). The tables show that for contour increments of 5 grey-scale values, the memory required to store the *aerial view* as a .txt file may be reduced but at the expense of the accuracy of the image reconstruction process.

Table 2: The effect of increasing the tolerance tube on the accuracy of reconstruction and the memory required to store the contour data (in Mb) as a .txt file for the *aerial view* (for a contour increment equal to 5).

Tolerance tube	0.2	0.4	0.6	0.8	1.0
Accuracy R.M.S. error	10	12	18	25	36
Mem. required (.txt file) in Mb	0.30	0.17	0.15	0.11	0.10

COLOUR IMAGES

When processing colour images, the same contour and vectorisation concepts above hold true. However, instead of a single grey-scale intensity value, colour digital images have pixels that are generally quantised using three components (i.e. Red, Green and Blue). In general, all image processing operations can be extended to process colour images simply by applying them to each colour component.

Each of the three RGB components are processed, encoded and reconstructed separately as if they were three different grey-scale images. The results of the three reconstructions are then merged or fused to recreate the original colour image. In terms of image quality (R.M.S.) the results from the contouring approach, for the same classes of images (i.e. faces, landscapes and aerial views), were similar to those obtained from the grey-scale imagery described above.

On the other hand, the memory requirements to store/transmit contour data were also comparable to the memory needed to store colour imagery in JPEG format. The contour approach also provided more accurate results with no visible artefacts or blocking effects. However, the reconstruction process required to regenerate the colour image after transmission on a digital link required 2.5 times more time than reproducing the same image in JPEG format. In this context, additional research may be required to increase the speed of the image reconstruction from contour data when dealing, for example, with real time applications.

CONCLUSIONS

Notable findings encountered in this study are:

- Generating contour data from digital images involves assigning coordinates values to pixels in the raster format image data, and interpolating between the pixels to find the coordinates of points in the path of a contour having a given intensity value. This enables the contour bound to be found to sub-pixel accuracy if required.
- The conversion of certain classes of digital images into contour maps may be used to compress and reconstruct images in pixel format that are more accurate and with improved visual details than JPEG compressed versions of the same image, while requiring similar memory space for storage and speed of transmission over digital links.
- For the class of images investigated in this work, the contour approach to image compression requires contour data to be filtered and discriminated from the reconstruction process.
- Spline interpolation was used to reconstruct digital images from their contour representations. The process involves determining the pixel intensity value which would exist at the intersections of a regular grid using randomly spaced nodes of bidirectional cross-sections.
- The contour approach or vectorisation of digital images for the purpose of image compression can be applied to both grey-scale images and colour images. In the case of colour images each of the three channels (Red, Green and Blue) is processed and encoded separately. The results of the reconstruction for each colour component are then merged or fused to recreate the original colour image.
- Refinements to the proposed method are being undertaken to increase the accuracy achievable for a variety of scenes and dynamic ranges (including bi-tonal imagery).
- More research is required to assess the accuracy of the enhancement process in the presence of added random noise and/or video imagery.
- To compete with standard compression protocols for image compression additional study is required to speed up the process of reconstructing an image from its contour representations.

REFERENCES

Bonham-Carter G. F. (2006) Geographic Information Systems for Geoscientists: Modeling with GIS. Vol. 13 of Computer Methods in the Geosciences. Elsevier, Butterworth-Heinemann. 398 pages.

De Jong S. M. and van der Meer F.D. (2004) “Remote Sensing Image Analysis Including the Spatial Domain”, Kluwer Academic Publishers, pp. 360.

Duperet, A., (2002) “Image Improvements” Chapter 2, Digital Photogrammetry. In Michel Kasser and Yves Egels editors, Taylor & Francis Publishers. p.p. 351: 79-100.

Elder J. H. and Goldberg R. M. (2001) “Image Editing in the Contour Domain”. IEEE Transactions on Pattern Analysis and Machine. Vol 23 No. 3.
Gonzalez R. C. and Woods R. E. (2008) Digital image processing. Prentice Hall, 954 pages.

Smith M. Goodchild F. and Longley P.A. (2009) “Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools. 3rd Edition. The Winchelsea Press. pp. 560.

Spiegel M. R., Stephens L. (1996) “Theory and Problems of Statistics, Schaum’s Outline Series, McGraw-Hill Book Company. pp. 556.

Wang S., Ge F. and Liu T. (2006) “Evaluating edge detection through boundary detection,” EURASIP Journal on Applied Signal Processing, vol. 2006, pp. 1–15.

Watson D. F. (1992) “Contouring: A Guide to the Analysis and Display of Spatial Data”. Pergamon Press. Ney York. 321 pages.

Weeks, A. R. (1996) “Fundamentals of Electronic Image Processing. SPIE Optical Engineering Press, Bellingham, Washington. 570 pages.

Cunha A. L., Zhou J. and Do M. N. (2006) “The Non-subsampled Contourlet Transform: Theory, Design, and Applications” IEEE Transactions on Image Processing, VOL. 15, NO. 10, October.

Elder H. J., Goldberg M. R. (2001) “Image Editing in the Contour Domain”. IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. 23 Issue 3, March 2001. IEEE Computer Society Washington, DC, US.

Russ C. J., (2007) “The Image Processing Handbook”, Published by CRC Press, ISBN 0849372542, 9780849372544, 817 pages.

Gonzalez C. R. , Woods R. E. and Eddins S. L. (2009) “Digital Image Processing Using MATLAB” 2nd edition, Gatesmark Publishing. 827 pages.

Li J. and Heap D. A. (2008). “A review of Spatial Interpolation Methods for Environmental Scientists”. Record 2008/23. Geocat No. 68229. Geosciences Australia. Australian Government.

Sandwell, D. T. 1987), "Bi-harmonic Spline Interpolation of GEOS-3 and SEASAT Altimeter Data", Geophysical Research Letters, 14, 2, 139–142.