ULTRA HIGH COMPRESSION FOR WEATHER RADAR REFLECTIVITY DATA

A Thesis

Submitted For the Degree of

Master of Science (Engineering)

in the Faculty of Engineering

by

Vishnu Vardhan Makkapati

Department of Aerospace Engineering

Indian Institute of Science

BANGALORE – 560 012

November 2006

TO

My Mother

Kamala Makkapati

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Prof. Pravas R. Mahapatra, Department of Aerospace Engineering, Indian Institute of Science (IISc). His wide knowledge and logical way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for my thesis. It was a great pleasure to me to conduct this research under his supervision.

I warmly thank my supervisor Dr. Sumit Basu for his valuable advice and friendly help. I wish to thank Mr. Yalla Veera Prakash, Mr. Mahalingam Vasudevan, Mr. Srinivas Raghavan, and Dr. Krishna Mikkilineni of Honeywell Technology Solutions Lab for permitting me to register for this program.

My special gratitude is due to the family of Prof. Pravas R. Mahapatra for their loving support during this time.

I owe my loving thanks to my mother Kamala, my wife Annapurna, and my son Shashi Vallabh. They have lost a lot of my personal time due to my research as an external registrant. Without their encouragement and understanding it would have been impossible for me to finish this work.

Needless to say that I am grateful to all of my friends at Indian Institute of Science, Birla Institute of Technology and Science, and Honeywell Technology Solutions Lab for their support.

Publications based on this Thesis

1. US Patent Application 2006/0145912 A1: Vishnu Makkapati and P. R. Mahapatra, "Compression and Transmission of Weather Data".

2. Vishnu Makkapati and P. R. Mahapatra, "Extreme Compression of Weather Radar Reectivity Data", submitted to *IEEE Transactions on Geoscience and Remote Sensing (TGARS)*, March 2006.

3. P. R. Mahapatra and Vishnu Makkapati, "Performance Enhancement of a Spline-based Method for Extreme Compression of Weather Radar Reflectivity Data", in Proc. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Athens, Greece, December 2005, pp. 147-152.

4. P. R. Mahapatra and Vishnu Makkapati, "Studies on a High-Compression Technique for Weather Radar Reflectivity Data", in Proc. *IEEE International Conference on Information, Communications and Signal Processing (ICICS)*, Bangkok, Thailand, December 2005, pp. 895-899.

5. Vishnu Makkapati and P. R. Mahapatra, "Radar Scan Conversion for Plan Position Indicator", filed in *United States Patent and Trademark Office*, June 2005.

6. P. R. Mahapatra and Vishnu Makkapati, "Ultra High Compression For Weather Radar Reflectivity Data Storage And Transmission", in Proc. *American Meteorological Society (AMS) 21st International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, San Diego, CA, January 2005.

Abstract

Weather is a major contributing factor in aviation accidents, incidents and delays. Doppler weather radar has emerged as a potent tool to observe weather. Aircraft carry an onboard radar but its range and angular resolution are limited. Networks of ground-based weather radars provide extensive coverage of weather over large geographic regions. It would be helpful if these data can be transmitted to the pilot. However, these data are highly voluminous and the bandwidth of the ground-air communication links is limited and expensive. Hence, these data have to be compressed to an extent where they are suitable for transmission over low-bandwidth links. Several methods have been developed to compress pictorial data. General-purpose schemes do not take into account the nature of data and hence do not yield high compression ratios. A scheme for extreme compression of weather radar data is developed in this thesis that does not significantly degrade the meteorological information contained in these data.

The method is based on contour encoding. It approximates a contour by a set of systematically chosen 'control' points that preserve its fine structure upto a certain level. The contours may be obtained using a thresholding process based on NWS or custom reflectivity levels. This process may result in region and hole contours, enclosing 'high' or 'low' areas, which may be nested. A tag bit is used to label region and hole contours. The control point extraction method first obtains a smoothed reference contour by averaging the original contour. Then the points on the original contour with maximum deviation from the smoothed contour between the crossings of these contours are identified and are designated as control points. Additional control points are added midway between the control point and the crossing points on either side of it, if the length of the segment

between the crossing points exceeds a certain length. The control points, referenced with respect to the top-left corner of each contour for compact quantification, are transmitted to the receiving end.

The contour is retrieved from the control points at the receiving end using spline interpolation. The region and hole contours are identified using the tag bit. The pixels between the region and hole contours at a given threshold level are filled using the color corresponding to it. This method is repeated till all the contours for a given threshold level are exhausted, and the process is carried out for all other thresholds, thereby resulting in a composite picture of the reconstructed field.

Extensive studies have been conducted by using metrics such as compression ratio, fidelity of reconstruction and visual perception. In particular the effect of the smoothing factor, the choice of the degree of spline interpolation and the choice of thresholds are studied. It has been shown that a smoothing percentage of about 10% is optimal for most data. A degree 2 of spline interpolation is found to be best suited for smooth contour reconstruction. Augmenting NWS thresholds has resulted in improved visual perception, but at the expense of a decrease in the compression ratio.

Two enhancements to the basic method that include adjustments to the control points to achieve better reconstruction and bit manipulations on the control points to obtain higher compression are proposed. The spline interpolation inherently tends to move the reconstructed contour away from the control points. This has been somewhat compensated by stretching the control points away from the smoothed reference contour. The amount and direction of stretch are optimized with respect to actual data fields to yield better reconstruction. In the bit manipulation study, the effects of discarding the least significant bits of the control point addresses are analyzed in detail. Simple bit truncation introduces a bias in the contour description and reconstruction, which is removed to a great extent by employing a bias compensation mechanism. The results obtained are compared with other methods devised for encoding weather radar contours.

Contents

List of Tables

List of Figures

Keywords

Extreme data compression, weather radar data compression, contour-based data compression, meteorological radar, high data compression, contour encoding.

Notation and Abbreviations

Symbols

c_{ij} – Element of the dispersion matrix

$d(\bullet)$ – Euclidean distance

dBZ – Reflectivity factor in decibels

$f(x, y)$ – Two-dimensional sequence

k_i – Number of control points

p – Degree of spline interpolation

p_i – Probability of occurrence of a symbol

r – Range corresponding to a display point

t_i – Knot values of spline

(x, y) – Coordinates of each pixel

$A_j(k)$ – Low frequency coefficient of wavelet decomposition

B – Total number of bits required for transmitting the encoded image

$C(t)$ – Curve defined by parameter t

$C(u, v)$ – Coefficient of DCT transformed sequence

$D_j(k)$ – High frequency coefficient of wavelet decomposition

H – Entropy of information source

I – Peak reflectivity value

L – Smoothing length

$M \times N$ – Dimensions of an image in pixels

N – Size (number of elements) of a sequence

$N_{i,p}(t)$ – Basis function of spline

P – Total number of contours traced

$\mathbf{P}, \mathbf{P}_i$ – Two-dimensional display points

Q – Queue containing the pixels of the external contour

R_b – Binary reflectivity value

$R_{threshold}$ – Reflectivity threshold value

$R(x, y)$ – Reflectivity value at a display point (x,y)

S – Information source

S_i – Symbol in the information source

T – Number of thresholds used for encoding

T – Knot vector of spline

$\mathbf{V}_j$ – Vector space

$\mathbf{W}_j$ – Vector space of difference signal

Z – Reflectivity factor

$\lceil \bullet \rceil$ – Ceiling value of $(\bullet)$

θ – Azimuth angle of a display point

μ_i – Component of the mean of the sample

ϕ – Elevation angle of radar beam

$\phi_{j,k}$ – Wavelet scaling function

$\psi_{j,k}$ – Wavelet function

Acronyms

BPP – Bits Per Pixel

CAPPI – Constant Altitude Plan Position Indicator

CR – Compression Ratio

DCT – Discrete Cosine Transform

DWR – Doppler Weather Radar

DWT – Discrete Wavelet Transform

IDCT – Inverse Discrete Cosine Transform

ISE – Integral Square Error

JPEG – Joint Photographic Experts Group

KLT – Karhunen Loeve Transform

LZW – Lempel-Ziv-Welch

LUT – Look Up Table

MBR – Minimum Bounding Rectangle

MPEG – Moving Picture Experts Group

MSE – Mean Square Error

NEXRAD – Next Generation Radar

NOAA – National Oceanic and Atmospheric Administration

NURBS – Non-Uniform Rational B-Splines

NWS – National Weather Service

PSNR – Peak Signal to Noise Ratio

PPI – Plan Position Indicator

RADAR – Radio Detection and Ranging

RHI – Range Height Indicator

RLE – Run Length Encoding

RMSE – Root Mean Square Error

SNR – Signal to Noise Ratio

UTC – Coordinated Universal Time

VQ – Vector Quantization

WSR-88D – Weather Surveillance Radar 1988 Doppler

Chapter 1

Introduction

1.1 General

Recent years have seen tremendous advances in the area of digital communication and storage systems. However, there is a simultaneous explosion in the volume of data generated and transmitted in various applications. This necessitates the development of techniques for reducing the volume of the data. Many schemes have been proposed to compress data by exploiting the redundancy therein. Imagery or pictorial data comprise a large fraction of digital data handled today, and its contribution is expected to rise in the coming years. Compression of pictorial data while retaining its visual content is therefore a subject of serious research.

1.2 Compression of Digital Data

Early schemes designed for image compression were based on classical information theory. These techniques were pixel-based and did not utilize the nature of pictorial data itself [9]. Hence these schemes offered very low compression ratios. Lossless and lossy schemes have been proposed to compress pictorial information [10, 11]. Lossless compression schemes offered compression ratios of the factor of 2, while lossy schemes could offer a factor approaching 10 at the expense of some image quality.

Recent schemes exploit the human visual system to achieve higher compression ratios without perceptually degrading the image quality [12, 13, 14]. These methods classify parts of an image into visually significant and insignificant components and apply appropriate compression techniques to these parts.

Several approaches have been proposed to identify visually important pictorial information. Edges of objects or features within imagery have been considered to be critical to human perception and thus convey maximum information. Many techniques have been proposed to preserve edge information while encoding. These techniques separate information into edge and texture and encode them independently.

Many situations demand lossless image compression e.g., medical imaging. Lossless coding retains an image as is without any reduction of information. Certain lossy techniques can be made lossless if the stage where loss of information occurs is handled appropriately.

1.3 Ground-Air Transmission of Digital Data

Exchange of digital information between the cockpit and ground is critical for modern flight operations. The typical information transmitted comprises:

- Near-real-time tracking and reporting of aircraft position
- Text messaging to and from the cockpit
- Air-ground-air data transmission via the data center
- Data from aircraft systems such as engine health, etc.

However, the bandwidth of ground-air data links is very limited and even when bandwidth is available it entails significant cost. Hence information transferred over these links is required to be highly compacted.

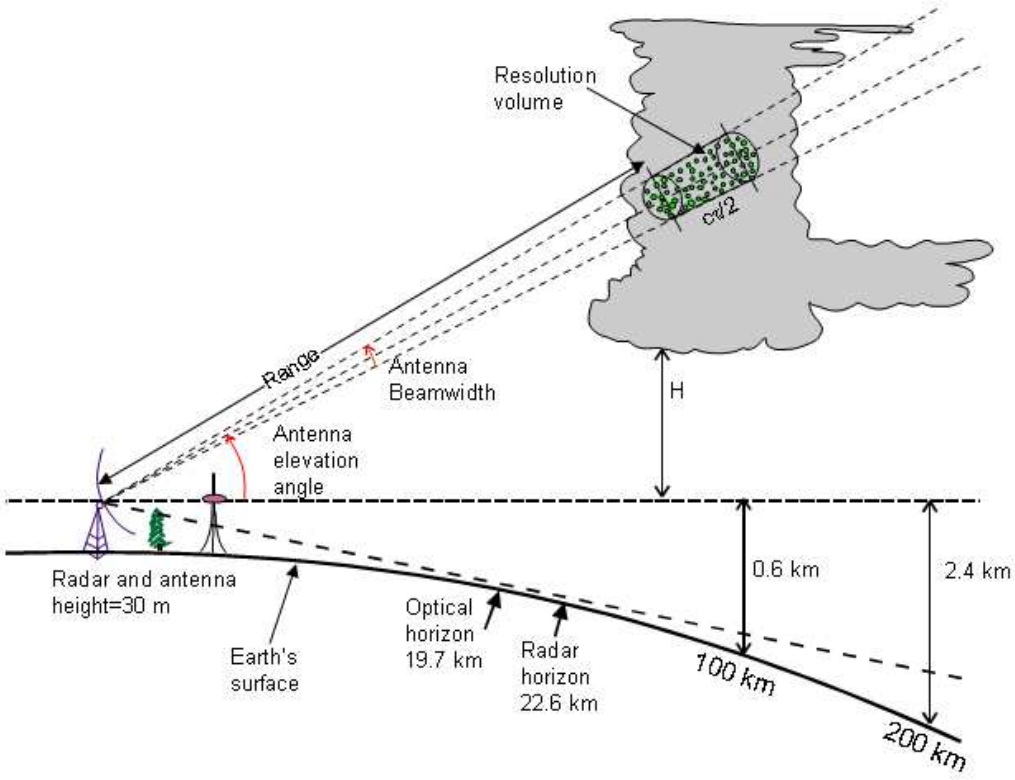


Figure 1.1: Schematic showing the geometry of operation of a weather radar [1]

1.4 Basics of Weather Radar

RADAR is the acronym for **RA**dio **D**etection **A**nd **R**anging. Radars use radio waves to determine the size, strength and location of a target. Typically, transmitted radio waves appear as a sine wave. A radar emits a coherent train of microwave pulses along a directed *beam* and processes the reflected pulses or *echoes* to derive the parameters of the scatterers within its detection range [15]. The volume within the radar beam sampled at a given instant constitutes a *resolution volume* or *bin*. A radially contiguous set of bins is termed as a *radial* or *ray*. The angle of a radial with respect to a fixed reference (typically north) is called its *azimuth*. A weather radar beam generally scans circularly at a fixed *elevation* angle; a set of the successive rays accumulated at an elevation angle is called a *sweep*. Azimuthal scans are usually performed at a sequence of elevation angles, resulting in a *volume scan*. The choice of the elevation angles, the resolution of azimuth scan, and the number of bins depend on the type of radar used, the mode of operation,

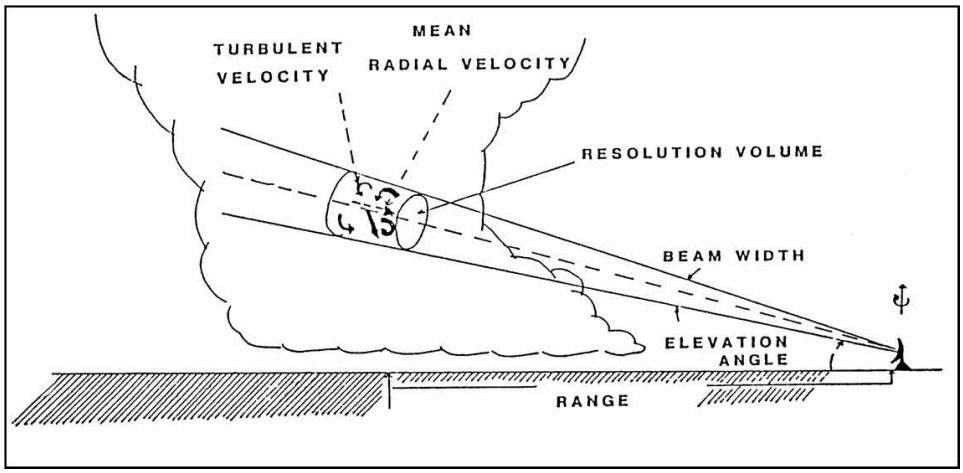


Figure 1.2: Schematic showing the operation of a Doppler weather radar [2].

and the intended application. Figure 1.1 shows a schematic of the geometry of operation of weather radars.

Weather radars have been a primary source of weather information due to the capability of microwaves to penetrate cloud and rain and generate detailed and accurate depiction of weather fields [16]. Cameras placed on satellites cannot probe the internal structure of clouds and the hazardous phenomena embedded in weather events such as storms.

Doppler radar is the only instrument that can remotely detect and trace winds and measure their radial velocities. The Doppler effect can be described as the observed change in the frequency of sound or electromagnetic waves due to the relative motions of the source and observer [17].

The operation of Doppler weather radars is illustrated in Fig. 1.2. Doppler radar is capable of detecting the changes in frequency resulting from the motion of atmospheric scatterers toward or away from the radar. This is the principle behind the generation of all velocity data products which are extremely useful to weather observers, pilots, meteorologists and researchers. Such observations permit weather forecasters to provide better warnings and help them in understanding the life cycle and dynamics of weather events such as tornadoes, cyclones, microbursts, and other storm hazards.

Table 1.1: NWS Reflectivity Thresholds

Level	**Reflectivity Interval dBZ**	**Rainfall Category**
1	18 - 30	Light
2	30 - 41	Moderate
3	41 - 46	Heavy
4	46 - 50	Very heavy
5	50 - 57	Intense
6	> 57	Extreme

1.5 Nature of Weather Radar Data

The composite echoes from hydrometeors constitute a weather signal. Many radar data products are available to meteorologists. These are classified as *base* and *derived* products. A base product is the digital data obtained after initial processing of radar echoes from each resolution volume. Base products are used for direct and instantaneous viewing of weather patterns and analyzed for significant meteorological features. Derived products are obtained after application of meteorological algorithms to base data and often entail significant delay or data latency.

The product legend is generally displayed alongside weather radar products. The information typically includes the name of the product, current date and time, location, maximum and minimum data values, range resolution, elevation angle, centre of the display (i.e., radar location coordinates relative to display), data value range and the corresponding color codes.

Doppler weather radars generate three important base products: reflectivity, radial velocity and spectrum width [15]. These parameters are explained in the following subsections.

1.5.1 Base Reflectivity Product

Reflectivity is a scalar parameter indicating the strength of the echo returned from a given radar bin, and is used as a measure of the intensity of the instantaneous precipitation rate

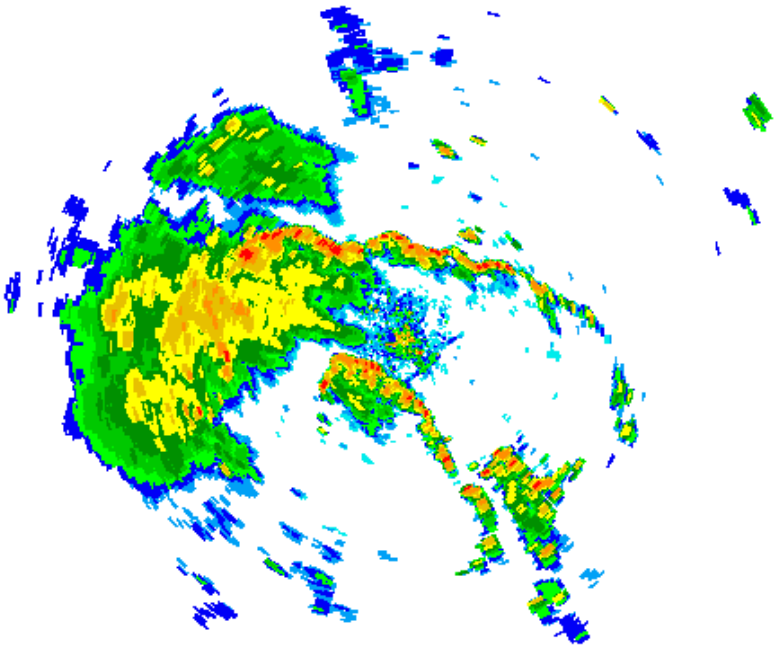


Figure 1.3: Sample reflectivity field

at the location [16]. The *reflectivity factor Z* has a very large dynamic range and hence it is normally represented in decibels *dBZ*. A sample radar reflectivity field is shown in Fig. 1.3. The reflectivity intervals depict various levels of rainfall intensity. Table 1.1 shows the classification of reflectivity intervals for various levels of rainfall activity as specified by the U.S National Weather Service (NWS) [18].

1.5.2 Base Velocity Product

Weather radar captures the radial component of wind velocity relative to the radar, commonly called *radial velocity*. In the base velocity product, inbound and outbound velocities are assigned negative and positive signs respectively. A true wind velocity vector can be constructed when base velocity data from three or more non-collinear radars are available at a given point [19, 20]. Products derived from velocity provide exact location of high wind speeds, wind shear and microburst activity, and also help predict their movement. This helps Air Traffic Controllers (ATC) to anticipate hazards, and aids in managing airfield and terminal area traffic. Fig. 1.4 shows a sample radial

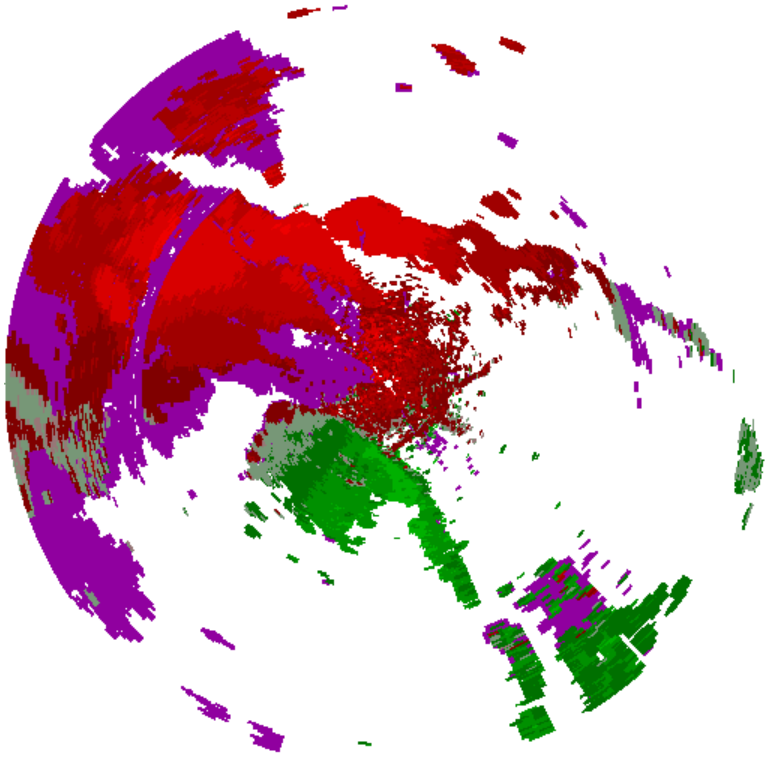


Figure 1.4: Sample radial velocity field

velocity field.

1.5.3 Base Spectrum Width

Spectrum width is a measure of radial velocity dispersion among scatterers in spatial of scales upto the dimensions of the resolution volume. It provides a measure of the variability of the radial velocity estimates primarily due to turbulence, with contributions from wind shear, insect/bird movement, etc. It is commonly used to estimate turbulence associated with thunderstorms and mesocyclones. Fig. 1.5 shows a base spectrum width display.

1.5.4 Display Formats

Visualization of 3-D weather radar data is important for analysis and interpretation [21]. Volume scan data are generally spatially sparse since certain elevation angle intervals

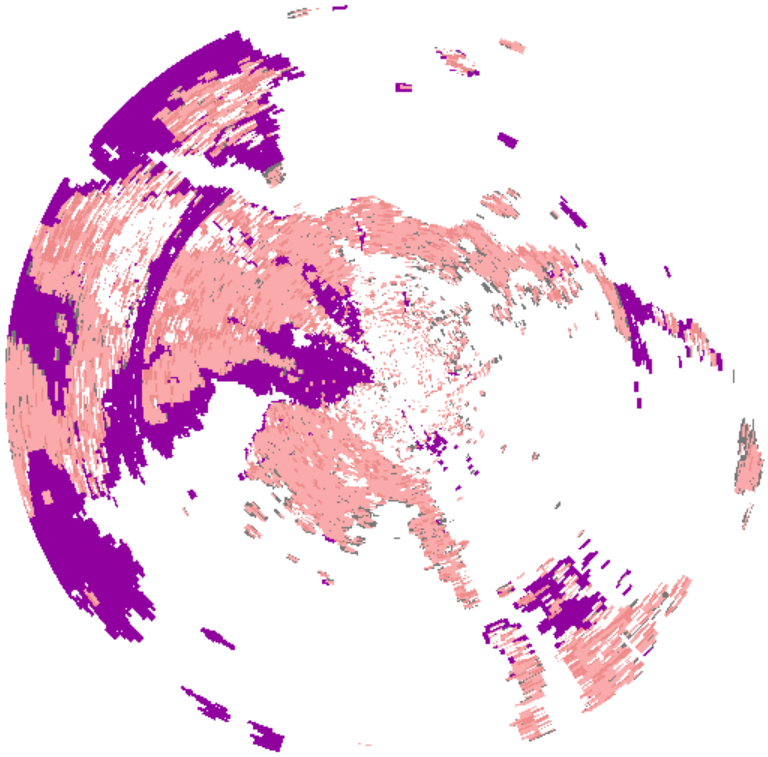


Figure 1.5: Sample spectrum width field

are usually skipped during the scan cycle in order to perform the volume scan within reasonable time limits. Further, the tangential distance between the centroids of resolution volumes in successive radials in a scan increases with the range, effectively reducing the spatial resolution. Since display formats have at best two dimensions, 3-D data are commonly visualized by taking appropriate planar cross sections. Three popular planar display formats of weather radar data are briefly explained in the following sections.

1.5.4.1 Plan Position Indicator

Since radar generates data along radials, each of which is along a distinct azimuth, individual resolution volumes within the radar's field of view are readily designated in (r,θ) coordinates. A convenient and highly informative display is therefore achieved by depicting a radar-derived parameter corresponding to the resolution volume in the same (r,θ) coordinates about a designated 'origin' on the display. Such a display is known as Plan Position Indicator (PPI) and effectively displays a plan view of the distribution

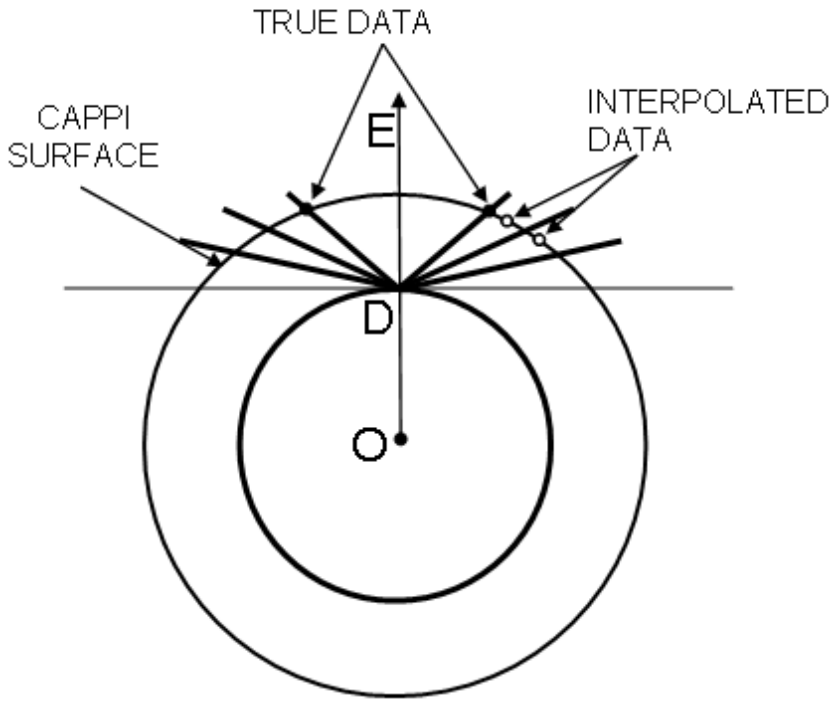


Figure 1.6: Schematic of CAPPI generation. O is the centre of the Earth, D is the radar location on the surface, and E is a point vertically above the radar.

of radar scatterer parameters in an area around the antenna. The parameter displayed on a PPI may be any of the base or derived radar products. On a color display the parameter is usually color-coded while on a monochrome display the modulation is in terms of intensity or gray scale.

Radar scan data may be converted to a 2-D format appropriate for presentation in polar coordinates with respect to a radar location. In a centered PPI mode, the antenna location is mapped at the center of the circular area of the display. In an offset PPI mode, the antenna location is represented at a distance from the center of the display area. The offset may in general have both x and y components. Figures 1.3, 1.4, and 1.5 show the PPI displays of the three base parameters corresponding to the same weather scene.

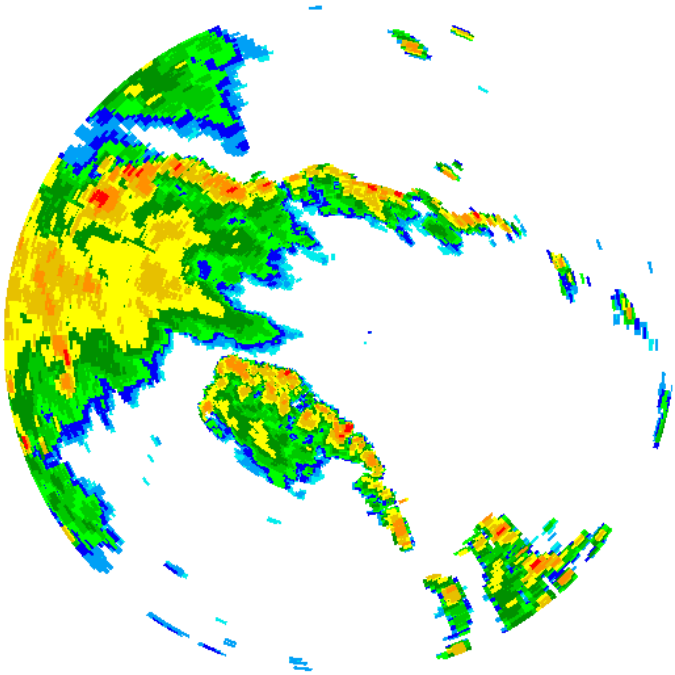


Figure 1.7: Sample CAPPI display of reflectivity

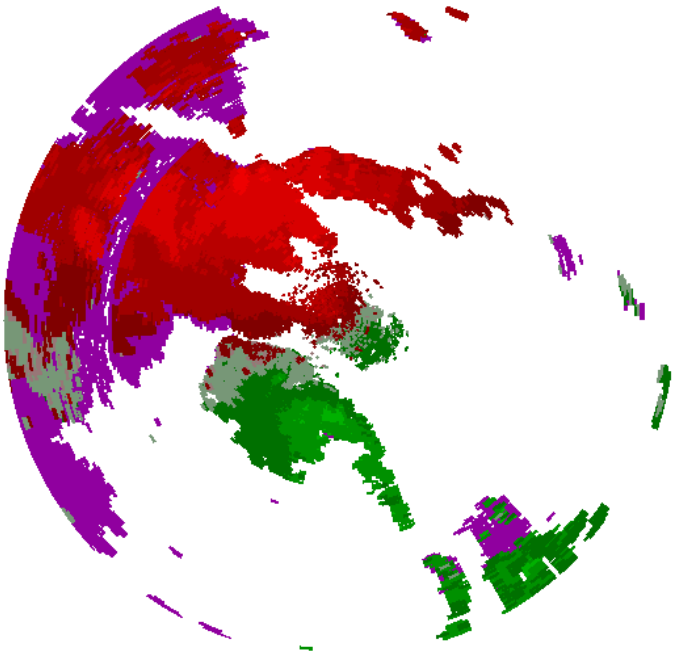


Figure 1.8: Sample CAPPI display of velocity

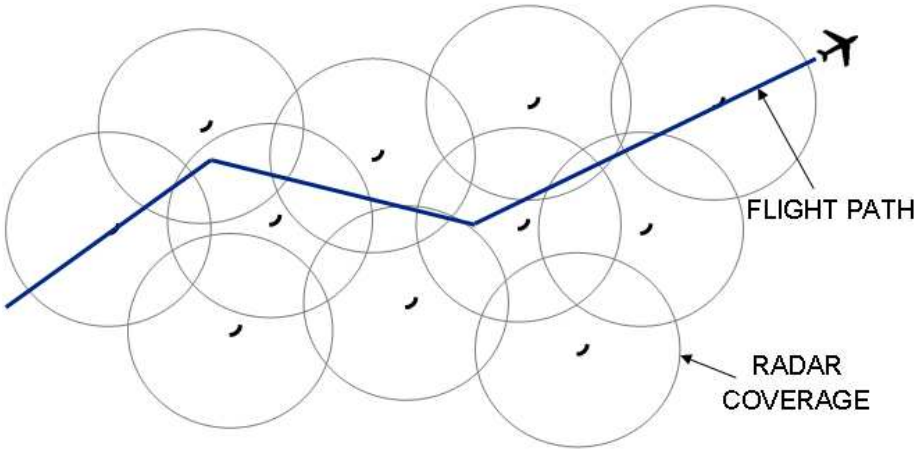


Figure 1.9: Schematic showing need for multiplanar offset RHI along a flight path

1.5.4.2 Constant Altitude Plan Position Indicator

At all beam elevation angles other than zero, an increasing range (distance from radar) also means a greater height with respect to the radar location. In order to display weather at a constant height, a Constant Altitude Plan Position Indicator (CAPPI) display may be synthesized from the volume scan data [22]. For CAPPI display, a horizontal slice is taken through the radar volume scan data, i.e., the display parameter data are retrieved from the 3-D scan data at a constant altitude above the earth's surface (Fig. 1.6). These 2-D constant-altitude data are presented in polar coordinates on a computer display, paper printout, or any other surface. Since there are significant gaps between the intersections of the CAPPI surface (designated horizontal surface at which CAPPI data are extracted) with the actual radar scan surfaces (cones generated by the scanning radials), an appropriate algorithm is used to fill in the data over these gaps. Sample CAPPI displays for reflectivity and velocity parameters are given in Fig. 1.7 and Fig. 1.8 respectively.

1.5.4.3 Range Height Indicator

Aircraft flight paths can be represented as straight lines or curves (typically straight-segmented) on a PPI or a mosaiced PPI (Fig. 1.9). Such a plot enables the visualization of the expected weather along flight paths planned to be flown by pilots. Because flight

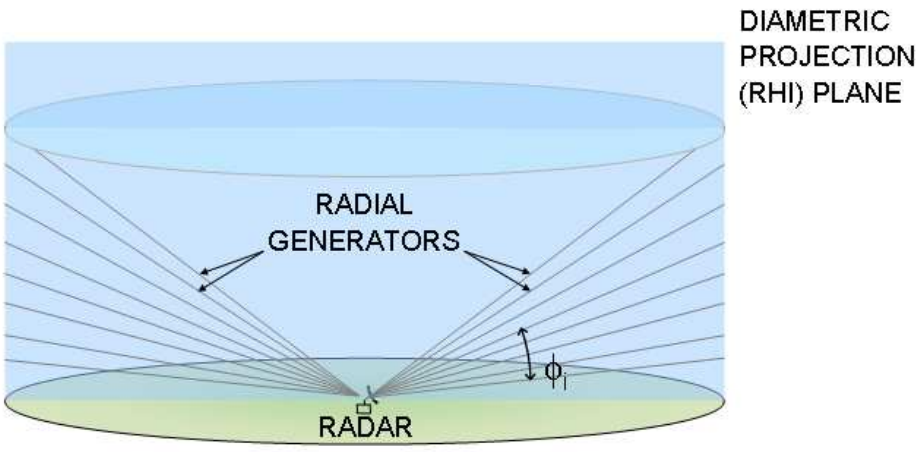


Figure 1.10: Schematic of radial RHI generation

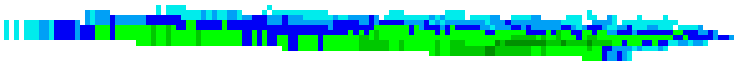


Figure 1.11: Sample radial RHI synthesis

operations cover a range of altitudes, information on the vertical variation of weather parameters is important for aviation. However, PPI displays are incapable of depicting the vertical dimension of weather fields. To display the vertical structure of weather, a common format employed is the range height indicator (RHI) which shows the radar data along each azimuthal direction as the beam sweeps through various elevation angles in a vertical plane (Fig. 1.10).

The conventional RHI depiction essentially represents a vertical section of the 3-D weather field through the radar location (Fig. 1.10). However, in practice an aircraft flight path may have any general orientation and offset with respect to a given radar. A long flight path would typically consist of multiple straight segments each of random heading and length (Fig. 1.9). It is extremely useful to pilots, controllers and aviation meteorologists to be able to see displays of weather parameters along a vertical surface passing through the aircraft flight path. Accurate and efficient schemes for the generation of such displays have been designed [23]. Figure 1.11 shows a sample radial RHI picture.

1.6 Need for Extreme Data Compression

Because of the strong role played by weather on aviation, weather surveillance systems are an integral part of aviation infrastructure [24]. The Doppler Weather Radar (DWR) [15, 16] is a potent tool for obtaining accurate and high-resolution weather data which can be used by pilots and meteorologists for hazard assessment and flight planning. Extensive 3-D weather data are generated by numerous ground-based radars such as the US WSR-88D network and similar radars installed or planned in other areas of the world. To be useful to the pilot, these data must be transmitted to the cockpit in real time via air-ground communication links. Most large aircraft also carry weather radars on board which provide a view of the local weather at the flight altitude. These data may be transmitted to the ground via air-ground links for viewing by meteorologists and also for integration with ground radar and other meteorological data.

Bandwidth is a premium resource for data transmission between ground and aircraft. Since DWR data is inherently voluminous, its transmission over such links is very expensive, if possible at all. It also generally involves significant data latency when transmitted over commonly available links. It therefore becomes necessary to compress radar-generated weather pictures as far as possible for transmission to and from aircraft.

Airborne weather radars have become more powerful and versatile in recent years. They are now fitted on most large and many medium aircraft. With Doppler capability (in addition to rainfall detection and estimation) and significant range of observation, they can act as independent sensors for detecting hazardous weather infront of the aircraft. Specifically, they can sense and estimate the severity of rain and hail, wind shear, and turbulence in a sector in front of the aircraft. However, by the very nature of airborne radars and certain constraints imposed by airborne operations, a stand-alone airborne weather radar has the following limitations:

1. Relatively limited range, due to limits on onboard power and antenna size (power-aperture product)

2. Limited angular coverage, confined to a sector ahead of the aircraft. A more

complete picture may be desirable under certain conditions, e.g., maneuvers.

3. Relatively wide beam (e.g., 3^0 cone in case of Honeywell RDR-4B airborne radar, compared to $<1^0$ for ground-based WSR-88D), resulting in poorer angular resolution, which translates to poorer absolute (spatial) resolution at longer ranges.

4. Relatively high frequency of operation (typically X-band, i.e., transmitted wavelength of the order of 3 cm, as in the case of Honeywell RDR-4B), causing rapid absorption of electromagnetic energy by rain. Absorption reduces the storm penetrating power of the radar beam, resulting in underestimation of reflectivity (rain/hail intensity) deep inside storms. In extreme cases, deep interiors of large storms or entire storm cells in the lee of large and heavy rain patches may not be detected at all.

5. The high operating frequency also proportionately aggravates the range-velocity ambiguity problem whereby the radial velocity is folded and appears to be much lower than its actual value if a useful unambiguous range is desired. Compared to ground-based long-range weather radars which typically operate at 10 cm wavelength, the ambiguity problem is over 3 times worse, i.e. for a given unambiguous range of operation, the unambiguous velocity will be over 3 times less, and vice versa. For this reason the Honeywell RDR-4B airborne radar has a maximum range of 40 nm for turbulence observation and even less for wind shear detection.

6. Degraded ability to observe and estimate the parameters of hazardous weather (rain rate, wind shear, turbulence) close to ground, where such observation is of critical importance.

7. Ability to observe only the radial component of the wind relative to the aircraft. Cross-components of the wind will be missed by the radar.

8. Corruption of wind velocity and turbulence estimates by navigational errors and aircraft body motions. This is especially significant for small and less sophisticated aircraft.

These limitations notwithstanding, airborne radar data has some advantages over ground-based weather radar data, which itself is of high quality and is often integrated (mosaiced) over national or regional scale radar networks. The advantages are:

1. Proximal View: the airborne radar has opportunity to see the weather from close quarters, which is a distinct advantage, especially if the aircraft is flying far away from ground radars

2. Flight-level Observation: The airborne radar observes weather as it exists at the flight altitude. The ground radar (network) often has a spatially sampled view of the weather at altitude, with a constant-altitude picture (CAPPI) generated by data synthesis.

3. No-break Observation: Ground radars usually have blind zones overhead, which may extend to tens of kilometers at the flight altitude in case of normal long-range radar scan cycles. Airborne radars, in contrast, continuously observe the weather ahead.

4. Turbulence Detection: Airborne radars have short range capability for turbulence detection, but ground radars also have limitations in detecting turbulence (though their range is more than that of airborne radars). Hence airborne observation of turbulence can usefully supplement observation by ground radars.

In summary, airborne radar provides useful local weather data at the flight scene, while the ground radar (network) provides accurate, reliable and extensive weather picture. It is useful to be able to combine data from both types of radars. There is need for two-way transmission as follows:

Ground-Air Transmission (Data Uplinking): Availability of ground-based weather radar data / images / mosaics in the cockpit enables the pilot to be aware of the larger weather scenario and helps him/her to plan route changes and take evasive action in advance. This transmission mode will be especially important in the emerging free-flight scenario where the pilot is expected to have complete situational awareness. For aircraft

without on-board radar, ground-transmitted radar data are the only source of real-time pictorial weather information for the pilot.

Air-Ground Transmission (Data Downlinking): The availability of airborne radar data on the ground and its fusion with ground radar data helps in creating composite pictures of greater current utility to aviation. Airborne radar data can fill in gaps in ground radar coverage, provide data of higher resolution (than ground radar alone) over terminal areas and along flight paths, and improve the quality of flight-altitude weather data. Data from radar-equipped aircraft can aid the flight of small aircraft (without weather radar) immediately preceding or following them.

Transmission of raw weather radar data, however, is bandwidth-intensive. At present and into the foreseeable future, bandwidth is and will remain a premium resource in air-ground and ground-air links. This limitation all but precludes the up/downlinking of raw weather radar data. In view of the above-listed advantages of such data transmission, it is necessary to process the data carefully to be able to transfer its maximum information content within the limited bandwidth available or affordable. This constitutes the goal of this thesis.

1.7 Approach to Extreme Data Compression

General-purpose digital data compression methods lend themselves for ready use in reducing weather data bandwidth. However, the compression factors achievable by such methods are not adequate to accommodate Doppler weather radar data within the available bandwidth for real-time transmission. It is therefore necessary to take into account the nature of these data, their mode of usage, and their intended application to achieve very high compression ratios. Such an approach is taken in this thesis.

1.8 Scope of the Thesis and Organization

An extreme compression scheme for weather radar data that achieves compression ratios over an order of two is presented. This method, based on contour encoding, extracts control points from the contours of the imagery and transmits them to the receiving end. A B-spline interpolation is used to reconstruct the contours from the control points. Enhancements to improve the compression ratio and quality of reconstruction are also presented.

The thesis is organized as follows. Chapter 2 summarizes some of the general purpose image compression schemes and prior work on weather radar data compression. An overview of the techniques used in this work is given in chapter 3. The basic scheme for encoding weather radar data is discussed in chapter 4. The method is extensively studied and the observations are presented in chapter 5. Enhancements to the basic method are explained in chapter 6. Finally, conclusions are drawn in chapter 7 along with suggestions for future work.

1.9 Summary and Conclusion

Weather is a critical factor affecting the operation of flights. The network of ground-based weather radars provide detailed weather picture and it will be helpful if these data are transmitted to the cockpit. The basic operation of weather radars and parameters of data captured by them are explained. Various display formats for these data are presented. The need for compressing these high-voluminous and information-rich data by high orders is discussed in detail.

Chapter 2

Literature Survey

2.1 Introduction

Compression of images, besides being *lossless* or *lossy* may also be classified as *transform*-based and *non-tranform*-based. The choice of a particular compression scheme depends on the application. For example, lossy methods may not be helpful in compression of handwritten scripts as any degradation may impede character recognition. However, lossless methods do not in general permit high orders of compression. General lossy methods, though offering higher compression, are also limited in their performance. Hence lossy methods that exploit the psychovisual redundancy of an image have been devised over the years to achieve even greater compression. A broad overview of image compression methods can be found in [10, 11].

2.2 Lossless Compression

Lossless compression schemes are based on principles of information theory. The entropy of an information source **S** as defined by Shannon is

$$H(\mathbf{S}) = \sum_{i=1}^{n} p_i \log_2 \left(\frac{1}{p_i}\right) \tag{2.1}$$

where p_i is the probability that a symbol S_i in $\mathbf{S}$ will occur. $H(\mathbf{S})$ is the lower bound on the bit-rate that can be achieved with any lossless compression method. The term $\log_2(\frac{1}{p_i})$ denotes the number of bits needed to code S_i.

2.2.1 Huffman Coding

Huffman coding has been shown to the achieve minimum bit-rate among the entropy encoding algorithms [25]. This method uses a variable-length code for encoding a source symbol based on the estimated probability of occurrence of each possible value of the source symbol. It uses a particular method for choosing a bit representation for each symbol, which results in prefix-free code (i.e., the bit string for a particular symbol is never used as the initial part of the bit string for any other symbol). It assigns shorter and longer strings to frequently occurring and less frequent source symbols respectively. The efficiency of Huffman coding depends strongly on a good a priori estimate of the probability of each input symbol. Several variations of this method namely adaptive, limited-length, template, n-ary etc. have been proposed [26, 27].

2.2.2 Run-length Encoding

The run-length encoding (RLE) scheme stores runs of data (i.e., sequences in which the same value occurs in successive data elements) as a single data value and count instead of the original run. This method is very useful for encoding graphic images and binary images, e.g., transmission of data in facsimile machines. However, this method is not well suited for continuous-tone images such as photographs.

2.2.3 Lempel-Ziv-Welch Method

The Lempel-Ziv-Welch (LZW) method generates a string translation table from the data being compressed [28]. This table maps fixed-length codes to strings. First, the string table is initialized with single characters. Next, the method examines the text, character

by character, and stores every unique two-character string into the table by code concatenation: The code is mapped to the corresponding first character which is displayed when each two-character string is stored. This scheme is repeated for incremental string lengths (i.e., three, four and so on). The decoding scheme uses the compressed data as input and builds an identical string table from the compressed data as it is recreating the original data. This method is designed for fast implementation, but does not analyze the input data and hence may not offer optimal performance.

2.3 Lossy Compression

Certain lossy compression schemes take into account the human visual system to achieve higher levels of compression [29]. These systems are usually evaluated by the perceptual quality. Hence, the features that have higher impact on psychovisual perception are given priority. These techniques can be broadly classified as *transform*-based and *non-transform*-based.

2.3.1 Transform based Compression

Transforms decorrelate the image data and represent it in a compact form suitable for efficient encoding. They better preserve subjective image quality, and are less sensitive to statistical image property changes.

2.3.1.1 Karhunen-Loeve Transform

An optimal transform is the one that minimizes the distortion between the original and reconstructed images. This transform is the Karhunen-Loeve Transform (KLT) [30, 31, 32]. KLT is a linear transformation that maps a sample of points in L-dimensional space into another orthogonal space that exhibits the property of the sample most clearly along the coordinate axes. The sample variances are the extrema along the new axes and are uncorrelated. KLT uses the dispersion matrix calculated as

$$c_{ij} = \frac{1}{N} \sum \left[(x_i - \mu_i) (x_j - \mu_j) \right] \tag{2.2}$$

where x_i is the i^{th} component of the point coordinates, μ_i is i^{th} component of the mean of the sample, and the sum is over the N points of the sample. The eigenvectors and the eigenvalues of this matrix correspond to the principal axes and the variance along each one of them respectively. KLT reduces the dimensionality by discarding less significant eigenvectors and thus achieves compression.

KLT, however, is highly data-dependent and does not permit fast implementation. It also requires the knowledge of the second-order statistics of an image. Since this is not always available, sub-optimal techniques that approximate KLT have been developed [33].

2.3.1.2 Discrete Cosine Transform

The discrete cosine transform (DCT) converts a signal into elementary frequency components [34, 35]. The two-dimensional DCT is a sequential extension of one-dimensional DCT, i.e., the one-dimensional DCT is applied to each row and then to each column of the result. This transform for a two-dimensional sequence $f(x, y)$ of size $N \times N$ is given by

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) cos \left[\frac{\pi(2x+1)u}{2N} \right] cos \left[\frac{\pi(2y+1)v}{2N} \right] \tag{2.3}$$

for u, $v = 0, 1, \ldots, N\text{-}1$ where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} \text{ for } u = 0 \\ \sqrt{\frac{2}{N}} \text{ for } u \neq 0 \end{cases} \tag{2.4}$$

The two-dimensional DCT is called *separable* since it can be computed by applying one-dimensional transform independently to the rows and columns. The image is retrieved by applying the inverse DCT (IDCT) as:

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)cos\left[\frac{\pi(2x+1)u}{2N}\right] cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{2.5}$$

for x, $y = 0, 1, \ldots, N\text{-}1$.

DCT is an orthogonal transform and offers excellent decorrelation and energy compaction. For this reason it is chosen in the popular JPEG standard for image compression [36]. Instead of applying the DCT to an entire image, it is usually applied to separate $n \times n$ blocks of an image; this is commonly called *blocked* DCT. JPEG, for example, uses blocks of 8×8 pixels. However, blocking in DCT introduces certain artifacts.

2.3.1.3 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) offers a multi-resolution representation of an image. It uses two basis functions $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ to decompose an image. The function $\phi_{j,k}(t)$ is a basis for vector space $\mathbf{V}_j$ where $\mathbf{V}_j = \mathbf{V}_{j-1} + \mathbf{W}_{j-1}$. $\psi_{j,k}(t)$ is a basis for $\mathbf{W}_{j-1}$, where $\mathbf{W}_{j-1}$ is the difference signal lost in moving from a finer scale to the next coarser scale. The input signal $x(t)$ in $\mathbf{V}_j$ can thus be expressed as

$$\begin{aligned} x(t) &= \sum_k cA_0(k)\phi_{j,k}(t) \\ &= \sum_k cA_1(k)\phi_{j-1,k}(t) + \sum_k cD_1(k)\psi_{j-1,k}(t) \end{aligned} \tag{2.6}$$

The coefficients $A_0(k)$ at scale j are analyzed to produce two sets of coefficients $A_1(k)$ and $D_1(k)$ at scale $j-1$ and this step is commonly referred to as wavelet analysis or decomposition. The inverse procedure reconstructs $A_0(k)$ using the two sets of coefficients $A_1(k)$ and $D_1(k)$ and this process is termed as wavelet reconstruction or synthesis.

2.3.2 Non-transform based Compression

2.3.2.1 Vector Quantization

Vector Quantization (VQ) is a classical method of image compression and is based on the principle of block coding [37, 38, 39]. The input to a VQ is a set of $m \times n$ image blocks. The goal of VQ is to design a *codebook* consisting of *codevectors* that can faithfully represent an image. The design of a codebook is not unique and several methods have been proposed for this purpose. VQ exploits the linear and non-linear redundancies inherent in an image and is optimal among block encoding schemes. However, the computational and memory complexity of VQ increases with the size of the codebook and the dimensions of the codevectors. Several schemes have been proposed to reduce these costs [40, 41, 42].

In its initial days, the design of a vector quantizer was considered to be a challenging problem as it required multi-dimensional integration. The use of a training sequence for VQ design was proposed by Linde, Buzo and Gray [43] thereby eliminating the need for multi-dimensional integration. Since then several variants of VQ have been proposed.

2.3.2.2 Shape Encoding

Shape is a concept which is widely understood and yet difficult to define [44, 45]. Shape can be considered to be a function of the position and direction of a simply connected curve. In a two-dimensional field, every point on a simply connected curve has at most two neighbors that lie on the curve. The coding of shape involves encoding its boundary and/or the pixels that lie inside it.

The boundary of a shape has to be detected before coding can be performed. This is commonly obtained by either edge detection or segmentation. Further, the shape has to be described in mathematical terms, and this definition is application-dependent. Shapes in general are stored for two primary purposes – recognition and data compression. For recognition purposes, it is enough to employ a description that is sufficient to distinguish a given shape from similar ones within the competing family. However, it is not necessary

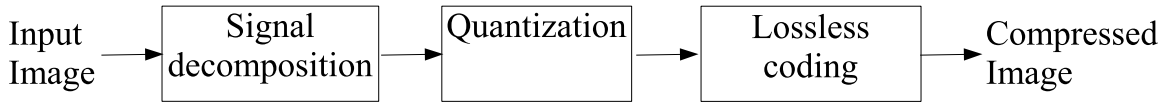


Figure 2.1: A typical lossy image encoder

to be able to reconstruct the original shape from its description [38]. In the case of data compression, it is essential to faithfully reconstruct the original shape from its compact representation.

Lossless or lossy encoding methods may be employed depending on the application. Chain coding is a convenient way for lossless representation of a boundary. Shape descriptors may be classified as either internal or external depending on whether they encode the area inside the boundary or the boundary itself. Object encoding in MPEG-4 uses both these modes of compression [46].

2.3.3 Hybrid Methods

The current generation of image compression schemes employs a combination of the methods described in the previous sections to achieve maximum compression with minimum degradation in quality. These systems comprise three steps, namely, signal decomposition, quantization and lossless coding, as shown in Fig.2.1. The goal of signal decomposition is to decorrelate the signal and concentrate the energy into a few coefficients. The quantization step represents the decomposed coefficients with a relatively small number of symbols. This step is typically lossy (nonlinear and noninvertible). The lossless compression step minimizes bit requirements by assigning codewords with a few bits to more likely symbols and more bits to less probable symbols.

KLT, DCT and DWT are commonly used techniques for signal decomposition. Methods such as scalar quantization and vector quantization are generally used for the quantization process. Huffman coding and LZW are popular choices for lossless compression. The JPEG standard [36] uses DCT while the more recent JPEG-2000 [47] is based on DWT. VQ is a favored choice to encode subband coefficients [48, 49, 50, 51].

2.4 Encoding of Weather Radar Data

General purpose image compression methods may not be suitable to encode weather radar pictures, especially for transmission over low-rate data links and high-density archival. These methods do not take into account the specific characteristics of the data and hence may not yield the desired compression ratios for such applications. Higher compression ratios can be achieved if the nature of the data and their utilization scenario are factored in effectively during the encoding process. Attempts have therefore been made to devise effective schemes to compress weather radar data. These include several lossless and lossy methods [52, 53, 54].

It is common to represent weather radar reflectivity data in binary form using certain threshold values. Meteorologists, airline pilots and air traffic controllers are trained to interpret these thresholds or contoured displays in order to infer weather phenomena and their hazard potential. Further, thresholding imparts a certain degree of neatness to weather radar imagery which otherwise often present a randomly granular or cluttered texture.

The contours of these binary regions may be encoded for further compression gain. A contour and its associated threshold are sufficient to construct a binary image. Once the contour is delineated the region inside the contour can be filled with a color corresponding to the threshold value. Compression of a thresholded image consists of extracting only that information of a contour which is necessary to represent it with a chosen level of fidelity. Weather data contours are generally extremely random and these undulations need to be maximally preserved in the encoded and reconstructed data besides the major areas of the contours. Two methods have been recently proposed in the specific context of encoding weather contours. These are based on polygon-ellipse representation [55] and polygonal approximation [56]. The polygonal approximation scheme identifies contiguous straight line segments of a contour, the end points of which are termed *vertices*. Only the vertices need to be transmitted to the receiving end. The polygon-ellipse method approximates a contour using straight-line segments and elliptical arcs.

Weather imagery contours in general display rapid and random undulations of various

scales. Polygonal and polygon-ellipse approximations may not be ideal for modeling such erratic contours. This thesis takes a different approach to the contour encoding and retrieval problem which, in principle, is capable of capturing more generalized contour shapes while yielding high compression ratios.

2.5 Summary

Many schemes for image compression have been proposed in the past. These methods exploit the spatial redundancy in images to achieve compression. The psychovisual information present in an image should be preserved for faithful reconstruction. Many hybrid schemes that use a transform(s) followed by a quantization procedure are embedded in image compression standards. A summary of the important methods for image compression has been presented in this chapter. It is inferred that while general-purpose compression schemes can yield only modest to moderate degrees of compression, the limited number of schemes developed for encoding weather radar imagery may have limitations in capturing the random fine structure present therein. This gives rise to the need for developing alternate and more versatile compression schemes for compressing weather radar imagery. This forms the focus of the work presented in the thesis.

Chapter 3

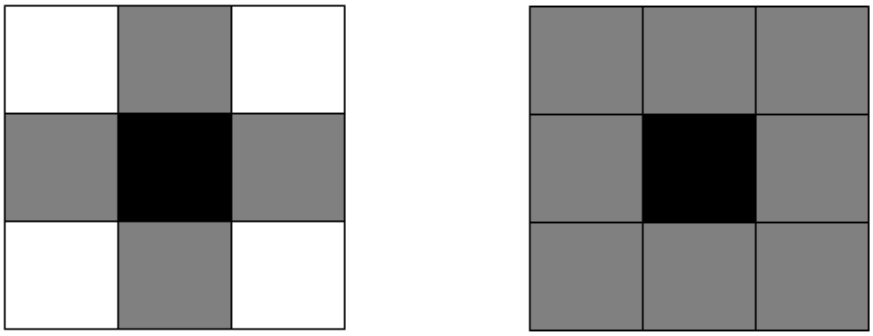
Background

3.1 Introduction: Contour Tracing

This chapter explains some of the techniques used in this work for compressing weather radar data. A review of general purpose contour encoding algorithms is also included.

A contour in an image is the locus of points on which a certain parameter, e.g., intensity or color value, remains constant. It may also represent a sharp or rapid transition of a given parameter.

In digital images the process of extracting boundaries of patterns is called *contour tracing*. It is also known as *boundary tracing* or *boundary following*. The boundary of a given pattern is the set of its border pixels. The boundary pixels may be classified as 4-border pixels (Fig. 3.1(a)) and 8-border pixels (Fig. 3.1(b)). Mere identification of the boundary pixels of a pattern is not sufficient for contour processing. It is necessary that an ordered sequence of the boundary pixels be obtained. In addition, the nature of the boundary, i.e., what it encloses, may be necessary for further processing steps such as retrieval of the image.



(a) (b)

Figure 3.1: Schematic showing (a) 4-connectivity and (b) 8-connectivity. The central black pixel is the pixel of current interest and the gray pixels are the neighboring pixels considered.

3.2 Contour Extraction

Contour tracing is a common preprocessing step applied to digital images to extract information about their general shape. Different characteristics of a contour (e.g., aspect ratio, perimeter etc.) may be used as features for pattern classification. Incorrect extraction of a contour leads to inaccurate feature points which may hinder proper classification of a given pattern. Features of a pattern can be extracted directly from the input image. However, the pixels on the contours of an image represent a small subset of the total number of pixels representing a pattern. Applying a feature extraction method based on a contour is usually much more efficient than performing the same on the original pattern. Contours retain much of the significant information about a pattern and hence feature extraction methods that operate on them will be generally accurate. Hence it may be said that contour tracing is an essential contributor to any feature extraction process. Several methods have been proposed for tracing contours.

3.2.1 Single Contour Techniques

3.2.1.1 Square Tracing Algorithm

The square tracing algorithm is one of the early methods to extract the contour of a binary image. Given a digital pattern consisting of a group of black pixels on a

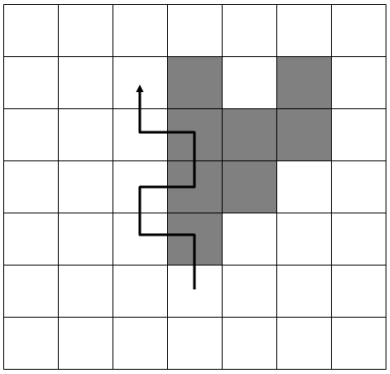


Figure 3.2: Schematic showing progress of the square tracing algorithm [3]

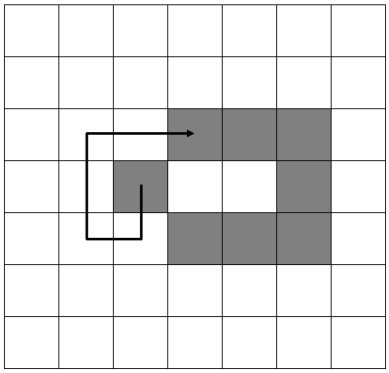


Figure 3.3: Schematic showing progress of the Moore-neighbor tracing algorithm [4]

background of white pixels, the scheme first identifies a black pixel and designates it to be the start pixel. Next, motion is initiated along a given cardinal direction (i.e., along x or y). While moving forward, the method turns in the left (right) direction every time a black (white) pixel is encountered (Fig. 3.2). This process is repeated till the start pixel is revisited. The black pixels traversed will be the contour of the binary pattern. The major weakness of the method is that it misses pixels lying diagonally with respect to a given pixel.

3.2.1.2 Moore-Neighbor Tracing

The Moore neighborhood of a pixel is defined as a set of 8 pixels which share a vertex or edge with that pixel (Fig. 3.1(b)). This scheme also finds a black pixel and labels it as start pixel. The method then traverses the binary pattern in clockwise or anti-clockwise

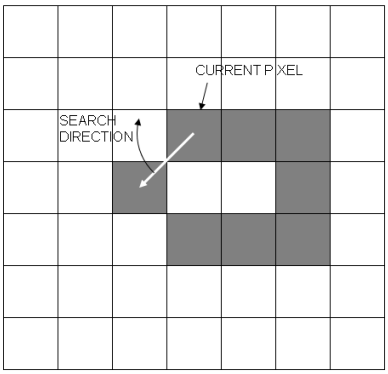


Figure 3.4: Schematic showing progress of the radial sweep algorithm [5]

direction. Every time a black pixel is found, the scheme backtracks i.e., goes to the white pixel that was previously hit. Then it goes around the black pixel in a clockwise direction, visits each pixel in its Moore neighborhood, until a black pixel is hit (Fig. 3.3). This process is terminated when the search returns to the start pixel. The main drawback of the method is the lack of an efficient stopping criterion. If this criterion is based on visiting the start pixel for a second time, the method will fail for a large variety of patterns.

3.2.1.3 Radial Sweep

Radial sweep is a popular algorithm for tracing contours [57]. This method also locates a black pixel first and tags it to be the start pixel. The method designates the latest boundary pixel located during the tracing operation as the current pixel $\boldsymbol{P}$. An imaginary line segment is drawn between $\boldsymbol{P}$ and the previous boundary pixel (Fig. 3.4). This segment is then rotated about $\boldsymbol{P}$ in a clockwise direction until it hits a black pixel in the Moore neighborhood of $\boldsymbol{P}$ sequentially. It should be noted here that rotation of the segment is identical to checking each pixel in the Moore neighborhood of $\boldsymbol{P}$. Two stopping criteria are commonly used to terminate the algorithm. The first criterion terminates the algorithm when the start pixel is visited for the second time. However, the performance of the algorithm is greatly enhanced when the second criterion is used, with the following two conditions:

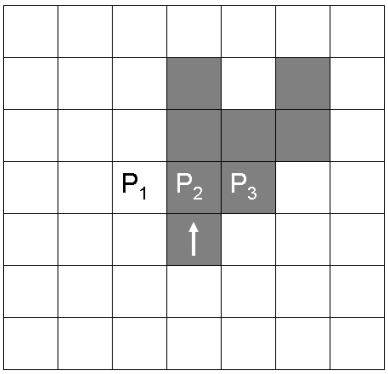


Figure 3.5: Schematic showing the three frontal search pixels in Theo Pavlidis' algorithm [6]

1. The current boundary pixel $\boldsymbol{P}_i$ has appeared before, i.e., it has occurred previously as pixel $\boldsymbol{P}_j$ where $j < i$

2. $\boldsymbol{P}_{i-1} = \boldsymbol{P}_{j-1}$

3.2.1.4 Theo Pavlidis' Algorithm

Theo Pavlidis' algorithm [58] is one of the recent methods designed for contour tracing. The method first identifies a start pixel on the binary pattern such that its left adjacent pixel (relative to a certain direction of assumed motion along x or y direction) is not black. The scheme then considers three pixels: the one in front of it, P_1, and the pixels adjacent to it on the left and the right, designated as P_2 and P_3 respectively (Fig. 3.5). The method first checks if P_1 is black and if true, it is declared as the current boundary pixel. If P_1 is not black, then the method finds if P_2 is black and if so, tags it to be the current boundary pixel. If both P_1 and P_2 are not black and if P_3 is black, the scheme designates it to be the current boundary pixel. If all the three pixels are white, the scheme considers the set of 3 pixels in front of the current pixel obtained after rotating clockwise by an angle of 90^o. If this process is repeated three times and still the next boundary pixel cannot be located, the scheme designates the current boundary pixel to be an isolated one, and the search process stops. The other, and more deliberate, stopping criterion checks if the start pixel is hit for the second time. The method works

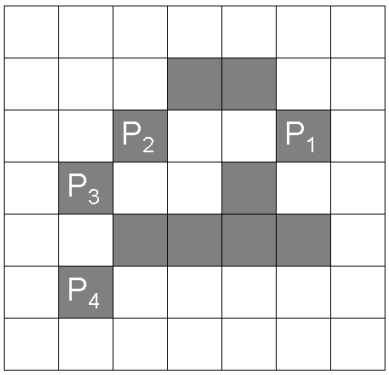


Figure 3.6: Schematic showing a pattern containing pixels (P_1, P_2, P_3, P_4) that are 8-connected but not 4-connected [7]

very well for 4-connected patterns (Fig. 3.1(a)) but fails for 8-connected patterns that are not 4-connected (Fig. 3.6). This can, however, be overcome by modifying the algorithm in two ways. The stopping criterion can be applied when the start pixel is visited for a third or even the fourth time. Another criterion is that the start pixel can be identified by using a designated procedure [58].

3.2.2 Multiple Contour Technique

The schemes discussed in the previous subsection do not trace all the contours in a binary image. The multiple contours traversal scheme described in [58] may be used for this purpose. The algorithm assumes that an external contour obtained by using a contour tracing method is placed in a queue Q. The availability of both x and y coordinates and the chain code values is also assumed. It marks the pixels belonging to the contour on the picture during the traversal process. The method assumes that the pixels of the region (defined in Sec. 4.2.1) are marked by 1 and those not in the region are marked by 0. The pixels on the contour are incremented by 1 during the tracing process, i.e., the pixels of the contour will have values 2 or greater at the end of tracing. The method utilizes these values when the interior of the region contour is searched for holes. The method avoids searching the picture outside the region of interest in an efficient manner.

The contents of the queue Q containing the pixels of the external contour are examined by removing one pixel at a time from it. If the current pixel is located on a

downward arc, the search is conducted to the right. These pixels are characterized by the chain code values: the previous element of the chain code must have values 4 to 7 while the next element should be in the range 5 to 7. However, the problem becomes complicated when there are pixels common to both the external contour and hole contour(s) inside it. To solve this problem, the search is carried out in the x-direction from the current pixel by examining the triplets of successive pixels A, B, and C. If A is 0, and B is 1, or if A is 0, B is 2, and C is 0, then B is used as the starting pixel for tracing hole contour and its pixels are appended to queue Q; upon returning from this step the search continues to the next pixel. Else if A is 0, B is greater than 2, and C is zero, or if A is 1, B is 2, and C is zero, then the search continues to the next point. This procedure is repeated till the queue Q is empty.

3.3 Contour Encoding methods

3.3.1 General Contour Encoding Methods

Several methods have been proposed for encoding contours. These may be broadly classified depending on whether they operate on the chain code values (Fig. 3.7) or the original contour points. A further classification is possible depending on whether they use curvature or any other criterion. Many of these methods detect *dominant* (also known as *key* or *corner*) points on the contours. The dominant points are transmitted to the receiving end, where they are joined using an appropriate interpolation mechanism. The common interpolation methods are linear segments, splines, and circular arcs. The dominant point detection schemes identify points such that the chosen interpolation method(s) yield(s) good reconstruction.

3.3.1.1 Chain Code Based Encoding

An early scheme proposed in [59] detects sharp corners in a chain-coded plane curve. A corner is defined as the chain node to which an identifiable discontinuity in the mean curvature of the curve can be associated. The method is based on the measurement of

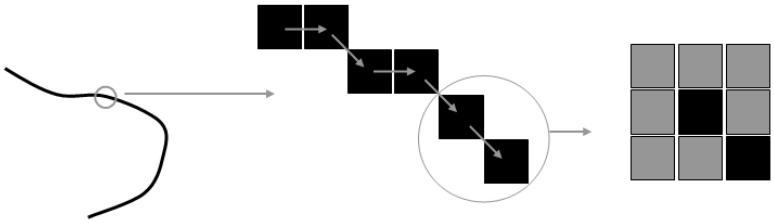


Figure 3.7: Schematic showing chain code

mean slope at every point of the curve. The mean slope is computed by the smoothing the local slopes obtained over a certain length of the curve. A corner is characterized by the presence of two runs of points at which the mean slope is below a threshold, separated by a run of the appropriate length over which the curve makes a significant turn.

A contour coding scheme based on the decomposition of the original chain code into a smoothed code and its difference code is presented in [60]. Suitable schemes are used to encode each of the decomposed data. The smoothed data consist of longer straight line segments compared to the original one, and an appropriate fixed or variable-length coding scheme is employed.

A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves is proposed in [61]. Four propositions are suggested to identify these vertices: 1) No change in chain codes from the i^{th} link to $i + 1^{th}$ link implies that the vertex is not a significant one. 2) If the absolute value of the difference between chain codes of the i^{th} and $i + 1^{th}$ links is 4, then it is significant. 3) If the chain codes for the i^{th} and $i + 1^{th}$ links are such that the angle between the links is less than or equal to 90^o, then the vertex is significant. 4) If the chain codes for the i^{th} and $i + 1^{th}$ links are such that the angle between the links is obtuse, then no immediate conclusion can be made and further steps are performed.

3.3.1.2 Curvature Based Encoding

A curvature based scheme for detecting dominant points is presented in [62]. This method addresses two major problems associated with dominant point detection on

digital curves – the precise definition of discrete curvature and the determination of the region of support for the computation of curvature. The region of support for each point is determined based on its local properties. Once the support region is determined, various measures of significance for each point are computed accurately and the dominant points are detected by a process of non-maxima suppression.

A method based on k-cosine is proposed in [63] to determine the region of support and detection of significant points. A measure of significance based on k-cosine is proposed. The procedure detects both maximal and minimal curvature points. A minimum curvature point that falls within the region of support of a curvature maximum is discarded while the maximal point is retained. The region of support implicitly fits a circular arc in the neighborhood of each point.

A method of fitting circular arcs to a digital curve is presented in [64]. A Gaussian filter is used to suppress noise and quantization error of the curve. An adaptive smoothing scheme is used in order to avoid the drawbacks due to large and small standard deviation values. A breakpoint is detected by calculating the curvature function of the smoothed curve, then taking the absolute value of derivative of the curvature function, and finally detecting the local maximum of the resulting function. Properly designated circular arcs are used to join successive breakpoints.

3.3.1.3 Polygonal Approximation

A generalized encoding scheme for two-tone (e.g., black-and-white) image contours is presented in [65]. This method detects digital line segments on the contour and encodes them using codewords of fixed or variable length. The conventional contour run length coding using 8- or 4- direction chain code is shown to be a special case of this scheme.

An efficient globally optimal approximation algorithm to find dominant points is described in [66]. It finds an endpoint bounded polygonal curve such that the sum of the errors over all its edges is minimum. A recursive formulation of this problem is presented and a solution is derived using dynamic programming. It is shown that any error measure can be used with the algorithm.

A split-and-merge method for polygonal approximation is proposed in [67]. The accuracy of this class of methods depends on the tolerance, i.e., the error threshold value. An adaptive threshold value that depends on the grid constant and the length of line in a collinearity test is used to reduce the quantization error while retaining the original contour shape. A contour tracing method is also discussed in the paper.

The research reported in [68] addresses the problem of finding the optimal polygonal approximation with the minimum number of segments using the sum-of-squared-error measure. An efficient graph search strategy based on the A* framework is presented for this purpose. The A* algorithm creates a graph with nodes corresponding to the end points of possible line segments and with arcs corresponding to the costs of line segments. The cost is defined as the sum of two terms: a high constant value associated with the creation of a line segment, and the error of the line segment. The method determines a path with minimum cost between the first and the last points of the digitized curve.

3.3.1.4 Hierarchical Representations

The method in [69] represents a curve at its natural scales. This approach makes the scheme suitable for any reasoning process, e.g., matching. Instead of representing the curve exhaustively over an entire range of chosen scales, the method identifies the significant scales for a compact and efficient representation. The scheme first obtains a set of descriptions of the curve by Gaussian-smoothing it over a range of scales. It then calculates the significance of each scale, and the natural scales are identified using these values. The natural scales are again ranked by a second significance criterion and the near-duplicate ones are eliminated.

A hierarchical representation of digitized curves based on the detection of dominant points is presented in [70]. The method uses integral square error (ISE), defined as the sum of the squares of perpendicular distances of each point from its approximating straight line. The method recursively splits a curve into two subcurves at such a point where the ISEs corresponding to the two subcurves are equal, and hence is given the name Equal Error Tree. However, the splitting procedure may not always be possible

and hence the scheme is suitably modified. The splitting is performed at a point where the absolute value of the difference between ISEs of the segments on its either side is minimum. The following propositions are adopted to divide the curve into subcurves recursively: 1) Determining if a subcurve requires further segmentation 2) Performing no further subdivision of a subcurve if it is a straight line. Thus the method results in a binary tree that is not necessarily balanced.

3.3.1.5 Circular Arc Approximations

A globally optimum approximation using circular arcs designed between each pair of adjacent dominant points is proposed in [71]. The method uses dynamic programming algorithm to detect these points. A circular arc can be designated to pass through three consecutive points and hence a significant number of points can be represented without error by an optimal choice of vertices. The approximation error is computed by using the area of the curve instead of its length. The method is generalized to approximate closed curves.

3.3.1.6 Straight Line and Circular Arc Approximations

A perceptually significant representation of digital curves is presented in [72]. This method uses a perceptual error measure based on two subjective criteria instead of a geometric error measure: 1) Straight-line segments are preferred over circular arcs; circular arcs are adopted only if they significantly simplify the representation. 2) Longer line segments or circular arcs are preferred, i.e., less segmentation is preferred in order to produce simple and clear representations. The perceptual error is minimized using a dynamic programming algorithm to find the optimal representation of contours.

A genetic algorithm based method for approximating digital curves with line segments and circular arcs is proposed in [73]. The representation task is considered to be an optimization problem and the optimal break points are obtained using a search process. The quality of such an approximation depends on the precise estimation of the break points. The search process is performed using a technique called chromosome

differentiation. The method does not require any threshold value to make a decision and is independent of the starting point.

3.3.1.7 Other Methods

The method proposed in [74] first finds two points that are farthest apart on the contour and joins them by a straight line. This divides the contour into two segments and the points farthest away from the straight line in each segment are identified, further splitting each segment into two. This process is applied recursively till the designated number of dominant points is found.

A compact representation of contours on the basis of sets of coordinates of equidistance points along the contour is reported in [75]. The method associates a vector with an ordered set of points which are approximately equidistant on the contour. The components of this vector consist of the angles between the straight lines passing thorough successive points. A component of this vector extracts information about the local features of the contour such as concavities, convexities, corners and straight lines.

A Bezier approximation to represent contours is proposed in [76]. This method decomposes a contour into arcs and straight line segments, the end points of which are designated as key pixels, to be transmitted. The reconstruction is based on the Bezier method, using Bresenham's line drawing algorithm [77] to fill in any gaps. Deletion and shifting of undesirable pixels generated by the Bezier approximation and insertion of new pixels is performed to maintain connectivity of pixels and to preserve the key points.

A parametric model to represent contours is presented in [78]. This method uses superquadrics to approximate contours typically found in medical images. Superquadrics model a wide variety of shapes by the introduction of tapering and bending deformations. However, this scheme is suitable for representing smooth contours only, which is not the case with contours obtained from weather data.

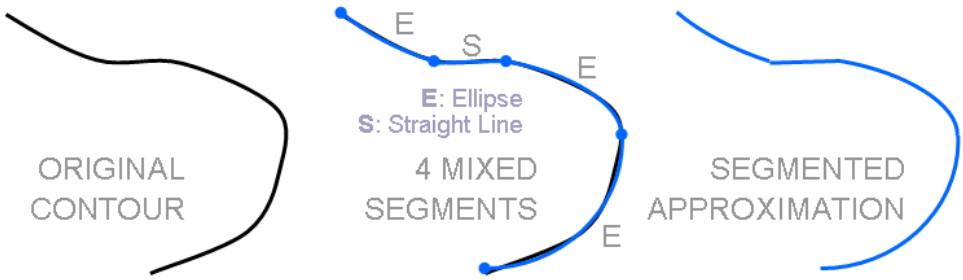


Figure 3.8: Schematic showing polygonal-ellipse approximation

3.3.2 Contour Encoding Methods for Weather Radar Data

Two contour encoding schemes have been proposed specifically for compression of weather data. These methods aim to describe contours as a chain of simple primitive segments consisting either of straight lines only or of straight line segments and elliptical arcs. This enables the contour to be described in terms of the parameters of each primitive. These methods differ from each other in the technique used for selection of vertices.

3.3.2.1 Polygon-Ellipse Encoding

This method uses segments of ellipses or straight lines (Fig. 3.8) to represent a contour [55]. It first determines which of the two shapes is suitable to approximate a given segment of the contour. If an elliptical arc is found suitable, the scheme determines its centre and major and minor axes. The contour segment is reconstructed at the decoding end by using an ellipse drawing algorithm using these parameters. The polygonal algorithm first identifies collinear points on the contour by using a maximum deviation threshold. The collinear points thus identified are approximated by a straight line. The scheme then checks for the next set of collinear points in the traverse direction (clockwise or counterclockwise). Two or more successive line segments are merged into one when two criteria are satisfied: 1) the angle between the current line segment and the previous one is less than a specified threshold, and 2) the angle between the current line segment and the initial segment (of the set sought to be merged) formed is within a tolerance limit. However, since ellipses and straight lines are smooth lines, this method may not be suitable to encode weather contours which are highly random.

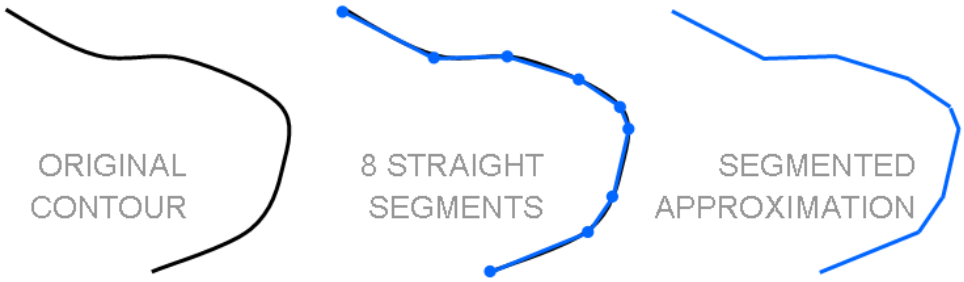


Figure 3.9: Schematic showing polygonal approximation

3.3.2.2 Method of Burdon

The scheme in [56] encodes a contour as a polygon (Fig. 3.9). This method searches only for collinear segments in a contour. The distance of the points between the current vertex and the vertex candidate from the line joining these points is calculated. If this distance is found to exceed a certain threshold, then the previous vertex candidate is added to the polygon list. It also checks if the length of the side being developed is starting to decrease. If so, the vertex candidate is added to the polygon list. Although proposed independently, and has some differences in the details of straight line segment determination algorithm relative to the corresponding step in the polygon-ellipse method, Burdon's method may be considered to be a special case of polygon-ellipse encoding. Like the polygon-ellipse method, Burdon's method may not be best suited for encoding randomly undulating weather contours.

3.4 Spline Interpolation Method

In this thesis the contour is retrieved from the control points by using spline interpolation. The principle of the interpolation method is outlined in this section.

3.4.1 Basic Principle

Spline interpolation is a popularly used form of interpolation where the interpolant is a special type of piecewise polynomial called spline. It is preferred over other forms of interpolation (e.g., polynomial) because the interpolation error is minimal even when a

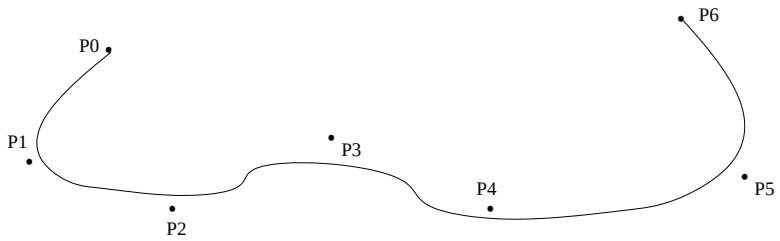


Figure 3.10: B-Spline representation of a curve

low degree polynomial is used for the spline.

The curve defined by a parameter t:

$$C(t) = \sum_{i=0}^{n} P_i N_{i,p}(t) \tag{3.1}$$

is a B-spline [79]. $C(t)$ denotes the point on the curve at the parametric value t. The curve is drawn for various values of t, ranging from t_{min} to t_{max}, typically having values 0 and 1 respectively. The shape of the B-spline curve can be controlled by $(n+1)$ control points, denoted by P_i, which are points in the object space. The positions of the control points are chosen so as to achieve the shape of the curve being modeled. The B-spline curve does not necessarily pass through the control points, but always lies within the convex hull of these points (Fig.3.10).

3.4.2 Knot Vector

A 'knot vector' is defined as

$$\mathbf{T} = \{t_0, t_1, ..., t_m\} \tag{3.2}$$

where the constants $t_i \in [0, 1]$, called 'knot values', are strictly in nondecreasing order. The knot values $t_{p+1}, ..., t_{m-p-1}$ are called internal knots. A uniform B-spline curve is one with equally spaced internal knots.

3.4.3 Degree of Interpolation

The degree p of the curve is defined as

$$p \equiv m - n - 1 \tag{3.3}$$

The curve becomes smoother as the degree of the curve increases. However, the departure of the curve from the control points (Fig. 3.10) is likely to increase with p.

3.4.4 Basis Function

The basis function $N_{i,p}(t)$ decides the extent to which a control point controls the curve at a parametric value t.

$$N_{i,0}(t) = \begin{cases} 1 \text{ if } t_i \leq t < t_{i+1} \text{ and } t_i < t_{i+1} \\ 0 \text{ otherwise} \end{cases}$$
$$N_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} N_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1,p-1}(t) \tag{3.4}$$

These functions have a non-zero value only in the open interval $t \in (t_i, t_{i+p+1})$. The sum of the control points is an affine combination, where the sum of all the basis function values of the control points is equal to 1. However, the sum of basis function values is not equal to 1 in the intervals $[t_0, t_p)$ and $(t_{n+1}, t_{n+p+1}]$ and no affine combination of the control points is possible here. The curve is thus defined only in the interval $[t_p, t_{n+1}]$, and end point interpolation is not possible. However, interpolation at the ends may be achieved by considering the first $p + 1$ and last $p + 1$ knots to be equal to t_{min} and t_{max} respectively. A curve is $p - k$ times differentiable at a point where k duplicate knot values occur.

3.5 Contour Filling

To reconstruct images from retrieved contours, a filling operation is necessary. Area filling algorithms operate on a raster while being completely unaware of any other primitives previously drawn [77].

3.5.1 Parity Fill Algorithm

The parity fill algorithm processes the raster one row or scan line at a time. Each scan line starts at a point outside the figure. As it marches, the state of the scan line alternates between the inside and the outside of the figure. The pixels along the scan line are filled with the current fill color when it is inside. However the method suffers from significant disadvantages. It processes each figure in isolation, works only for ideal primitives, needs special handling for lines parallel to scanning direction, and does not work for self-intersecting contours. However, it is popularly used because it is fast and well suited for hardware implementation.

3.5.2 Seed Fill Approach

3.5.2.1 Boundary Fill

A boundary fill algorithm requires the coordinates of a starting point, a fill color, and a background color. The method starts at a point inside the figure and paints it with an assigned color. The filling process continues until a boundary color is encountered. Four-connected and eight-connected algorithms are proposed for this purpose. These are recursive routines that may call themselves four and eight times respectively. If the current pixel was not on the boundary or already filled, then that pixel is first filled and then its neighbors are filled. However, there are cases when these methods fail.

3.5.2.2 Flood Fill

The recursive flood fill algorithm replaces all connected pixels of a selected color with a fill color. Four- and eight-connected versions of the algorithm exist. The algorithm requires

that the old color be read before it is invoked. This is commonly overcome by dividing the routine into parts that fill the east, south, north and west pixels of the current pixel. The eight-connected version gets into the nooks and crannies that the four-connected one cannot. But there are situations where the eight connected algorithm complicates the filling process.

3.6 Summary

Some of the traditional techniques used for different aspects of the work reported later in this thesis are explained in this chapter. Contour tracing has been found to be an important step in many feature extraction methods and popular algorithms for this purpose are discussed. General contour encoding methods are summarized and the ones specific to weather contours are explained in more detail. In particular, the principles and characteristics of popular and powerful spline interpolation method are explained. Schemes to fill the interior of a contour with a specific color value are briefly discussed.

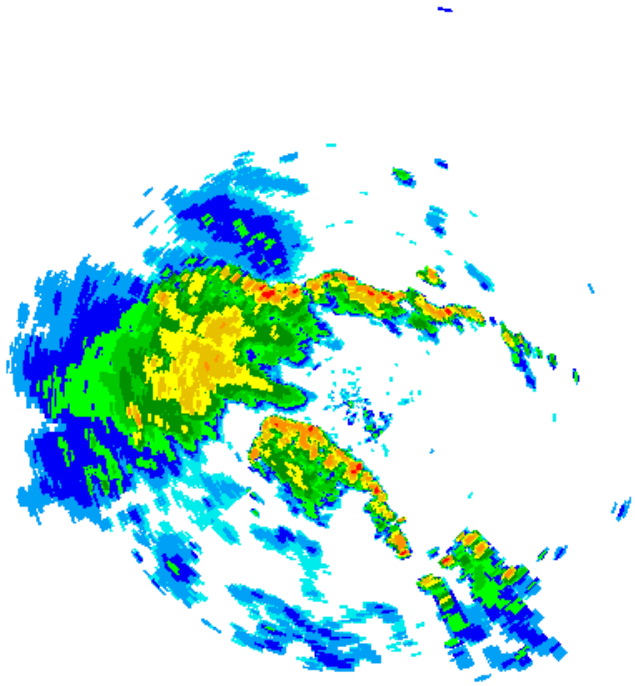
Chapter 4

A Novel Method of Contour Encoding

4.1 Introduction

Weather radar data are often presented for visual observations in a contoured form. This is especially true for reflectivity data which essentially depict the intensity of precipitation. In such a depiction, contours of constant values are drawn through the reflectivity distribution over a given area. The specific values for the contours are open to choice, but a frequently used set of thresholds is the one stipulated by the US National Weather Service (Table 1.1) [24, 16]. The contour at each assigned level is filled with a specific color for visual appreciation of the intensity value. A sample of a reflectivity field and its NWS-contoured representation are shown in Fig. 4.1. The arbitrary general shape as well as the random undulations of the contours are clearly apparent in Fig. 4.1(b). This illustrates the fact that weather contours have far less 'structure' compared to many other classes of images, e.g., in the medical domain.

This thesis is premised on the observation that transmitting the contours in place of all the pixels constituting a reflectivity field has the potential to offer high degree of data compression. Contoured reflectivity data obtained in binary form may be transmitted using general-purpose encoding schemes such as *run-length encoding* (Sec. 2.2.2).



(a) Reflectivity distribution

(b) NWS-contoured representation

Figure 4.1: Sample PPI reflectivity field from WSR-88D Doppler weather radar in Level II format (elevation 0.35^o, display 512×512 pixels with 8-bit depth)

However, the degree of compression achieved in such methods is not very high. It would be more efficient to transmit the contours as linear structures, which would demand far fewer points than any 2-dimensional description. Further data volume reduction can be achieved by encoding the contours themselves. This may be achieved in several ways as discussed in Section 3.3.1. In this chapter a contour-based encoding method for highly compressing weather radar reflectivity data is proposed [80].

4.2 An Extremum-Based Contour Encoding Scheme

4.2.1 Thresholding and Binary Image Generation

The 2-D reflectivity distribution is converted into a chosen binary image by applying a threshold, e.g., one of the values specified by the US National Weather Service (NWS) (Table 1.1) corresponding to different qualitative levels of weather phenomena [24]. The first step in the processing chain is to convert the reflectivity image into a binary form. This is achieved by assigning a value of 0 to all pixels where the reflectivity value falls below the threshold, and a value of 1 to those pixels where the reflectivity equals or exceeds the threshold. Symbolically,

$$R_b(x,y) = \frac{1}{2}\left[1 + \mathrm{sgn}\{R(x,y) - R_{threshold}\}\right] \tag{4.1}$$

where (x,y) are the coordinates of each pixel, R is the original (raw) reflectivity value, and R_b is its binary form. $R_{threshold}$ is the reflectivity threshold value.

The boundary separating an area of 0's from an area of 1's now becomes the contour corresponding to the chosen threshold. Contours are of two types: those enclosing a zone of 1's are called 'region contours', and those that have only 0's within are called 'holes'. It is possible for region contours and holes to be nested, i.e. there may be region contours within holes, and *vice versa*, multiple times (Fig. 4.2). The outermost contour of a nested set, however, will always be a region contour.

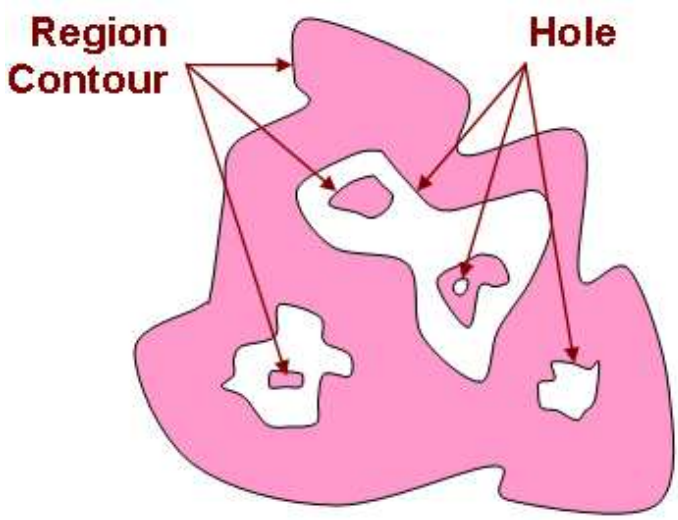


Figure 4.2: Sample binary image showing region contours, holes, and their nesting

4.2.2 Contour Tracing

4.2.2.1 Delination of Contours

To trace the contours of the binary images, the radial sweep algorithm of Sec. 3.2.1.3 is used [57]. This algorithm does not automatically trace internal structures of contours such as holes, and region contours nested inside them. Further, it traces only one contour at a time, and only after the contour has been initiated through a search procedure. To overcome these limitations, the scheme proposed in [58] is used (Sec. 3.2.2), applying it recursively to trace nested holes and region contours.

4.2.2.2 Initiation of Contour Tracing Scheme

Several methods can be used to locate a start pixel on a contour. In this work, the binary image is scanned by starting from the bottom-left corner of the image, searching through the column of pixels from the bottom to top, and searching subsequent columns from the left to right until a '1' is encountered. This pixel is labeled as the 'start' pixel for the contour. This processing may be performed starting from any of the corners of the binary image.

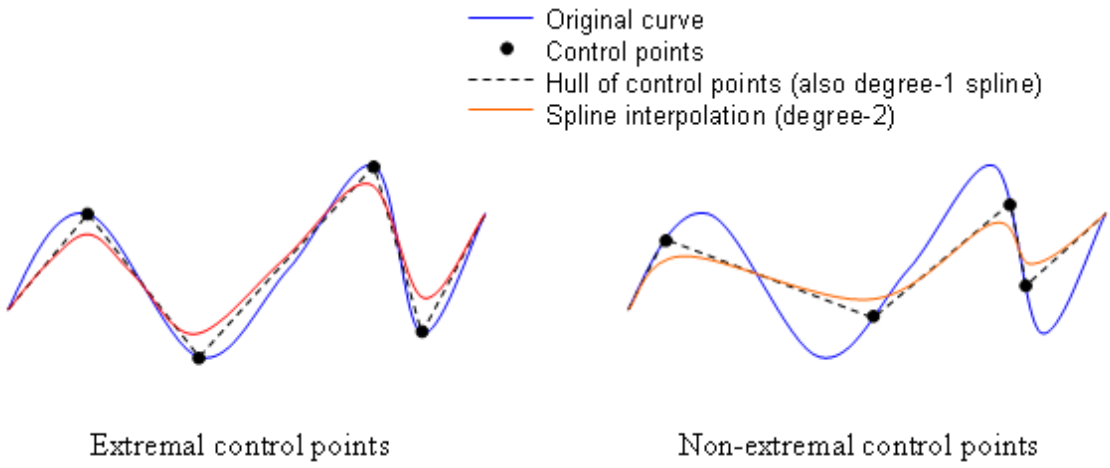


Figure 4.3: Schematic showing extremal points

4.2.3 Extraction of Control Points

When trying to represent a contour by a relatively small number of points, the obvious criterion would be to choose the dominant points, i.e., those points which would maximally retain the information contained in the contour. Attneave [81] made the profound observation that information on the shape of a curve is concentrated at points having high curvature. A large body of research and algorithms has been based on this criterion (e.g., [62, 82]). The rationale for Attneave's criterion is that arc segments between points of high curvature would have relatively low curvature and hence can be approximated by straight-line segments while reconstructing the contour. Such an approach serves very well to describe the outlines of objects of daily life which are often relatively smooth. Weather contours, in contrast, generally present a highly jagged appearance, with random high-frequency undulations (Fig.4.1(b)). Determining local curvatures in such cases becomes extremely cumbersome. Since radar display frames, each with numerous contours, are generated every few seconds, real-time encoding of the contours requires simple and fast algorithms. It may be noted that other natural contours such as those derived from vegetation and topography are also fractally undulating, but their static nature lends them to computation-intensive encoding on a one-off basis.

Algorithms based on distance measure are expected to be computationally more

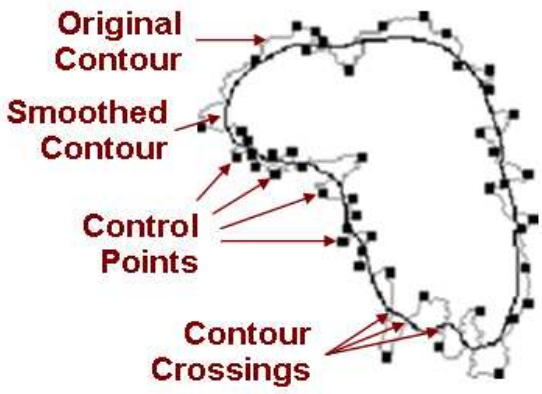


Figure 4.4: A raw contour (gray line) and its smoothed version (black line). The small black squares are the control points.

tractable than those requiring curvature computation, especially for rapidly and randomly undulating contours. This has been the basis of the previous methods for encoding weather radar contours [55, 56]. The proposed method is based on the premise that vertices signifying local extrema in specified scales naturally define the perceptually significant points in a contour [83]. Further, while high curvature points serve as good key points for straight line reconstruction, the proposed approach utilizes a spline-based reconstruction which permits curvilinear interpolation. Since a spline interpolation confines itself to the convex hull of the control points (Fig.3.10), it will not track maximal points, if non-maximal points are taken as control points (Fig. 4.3). We therefore use the local extremum criterion to choose control points for encoding the contours.

4.2.3.1 Smoothing and Reference Contour Generation

The aim here is to capture the shape of each of the contours with a limited number of significant points. A novel scheme for extracting the control points is developed in this work. The scheme first obtains a smoothed version of a chosen (original) contour by averaging the contour over a certain neighborhood of each point (Fig. 4.4). The smoothed contour, also called *reference contour* is obtained as

$$x_p^s = \frac{1}{L+1} \sum_{i=p-\frac{L}{2}}^{p+\frac{L}{2}} x_i$$
$$y_p^s = \frac{1}{L+1} \sum_{i=p-\frac{L}{2}}^{p+\frac{L}{2}} y_i \tag{4.2}$$

where (x_p, y_p) are the coordinates of the p^{th} point on the original (unsmoothed) contour, $L+1$ is the number of points over which the contour is smoothed (also called *smoothing length*), and (x_p^s, y_p^s) are the coordinates of the p^{th} point on the smoothed contour. The smoothing length may be specified absolutely or as a fraction of the overall contour length.

4.2.3.2 Distance Metric and Extreme Departure Detection

The deviation of each point on the original contour from the nearest point on the smoothed curve is calculated. The Euclidean distance d defined as

$$d\{(x_i, y_i), (x_p^s, y_p^s)\} = \sqrt{(x_i - x_p^s)^2 + (y_i - y_p^s)^2} \tag{4.3}$$

is used to calculate the deviation.

To start with, the scheme determines a *crossing point* between the original and smoothed contours by searching for a point where the distance between the two contours is one pixel or less. Then a search is performed to find a point on the original contour with maximum deviation from the smoothed contour before the next crossing point is encountered. Such a point is designated as an *extremal* control point.

Since the control points constitute nearly the entire transmitted data, their number for a given contour determines the compression ratio. This ratio may be altered within limits by adjusting the number of control points. Control points may be prioritized based on their distance from the smoothed contour, with higher importance given to farther points. Elimination of the lowest-priority points would enable achieving higher

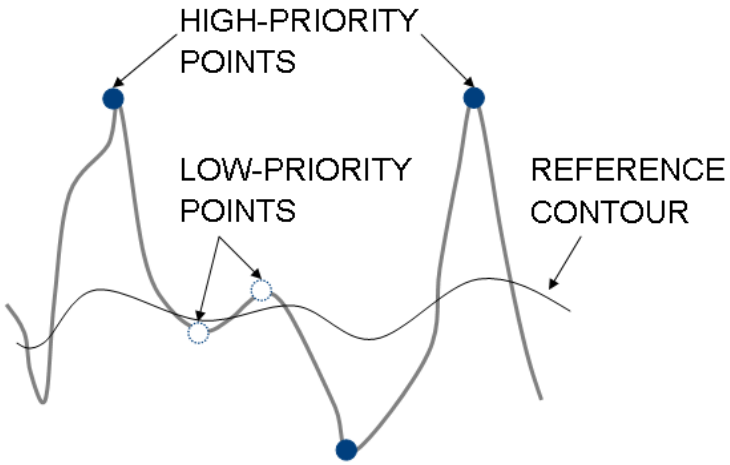


Figure 4.5: Schematic showing low-priority and high-priority control points

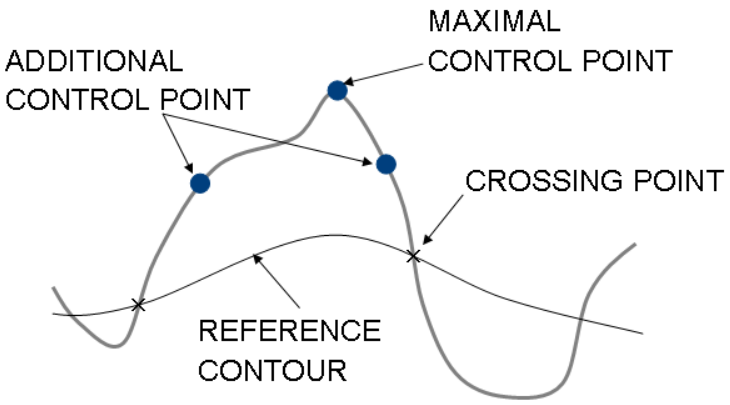


Figure 4.6: Schematic showing addition of control points

compression ratios with minimal loss of contour fidelity (Fig. 4.5). The departure of a control point from the reference contour may be used as a measure of its significance, with priority being given to those that have more deviation.

4.2.3.3 Need for Additional Control Points

If the smoothing length were very high, successive crossing points may be far apart. Extracting only one control point for the long segment between them may result in sparse spatial sampling of the contour and hence poor reconstruction of the contour at the receiving end, irrespective of the method of interpolation. To overcome this drawback, an extra control point is added between the maximal control point as determined above

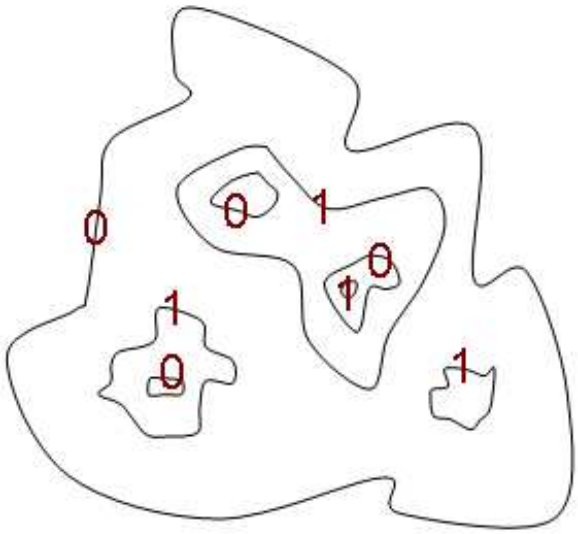


Figure 4.7: Tagged region and hole contours

and the crossing points on its either side, if the length of the original contour between any two crossing points exceeds a certain threshold (Fig. 4.6).

4.2.3.4 Control Points for Smooth and Small Contours

The scheme may not extract proper control points for smooth contours and small contours (e.g. contours with < 20 pixels because they do not display pronounced departure maxima). The control points for such contours are determined by considering uniformly spaced points along the contour. This spacing may be taken to be a given absolute length (in pixels) in case of large smooth contour segments. In case of small contours, the control point spacing may be fixed at a fraction of the overall contour length in order to ensure a certain number of control points to describe the contour.

4.2.4 Data Formatting for Transmission

The reflectivity thresholds used for generating contours are stored in a look-up table (LUT) available at encoding and decoding ends, and only the indices need to be transmitted to minimize bit requirements. Since there may be multiple closed contours for any given threshold, each contour is numbered before transmission. To distinguish region

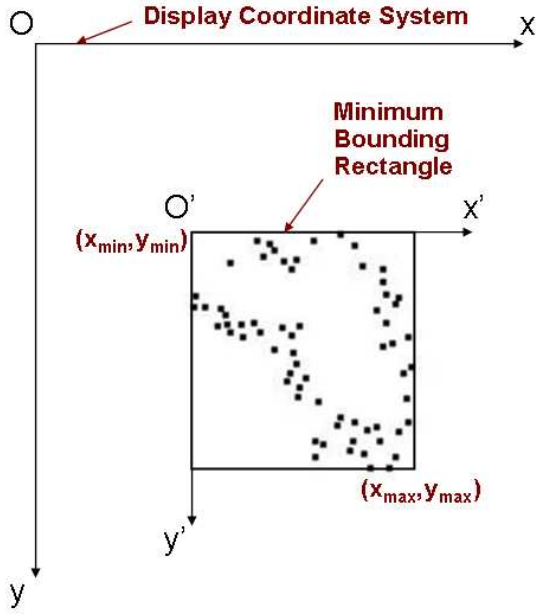


Figure 4.8: Schematic showing the overall display coordinate system and the coordinates used to minimize bit requirements

contours from hole contours, a tag bit of 0 and 1 respectively is used (Fig. 4.7).

To store or transmit the control points, each point would require $\{\lceil \log_2(M) \rceil + \lceil \log_2(N) \rceil\}$ bits where M and N denote the width and height of the input reflectivity field respectively, in terms of number of pixels, and $\lceil \bullet \rceil$ denotes the ceiling value of $(\bullet)$. Since control points constitute the maximum percentage of the data load, they are referenced to top left corner (x_{min},y_{min}) of the contour's minimum bounding rectangle (MBR) in order to save on the bit requirements (Fig. 4.8). Then each point would require $\{\lceil \log_2(x_{max} - x_{min}) \rceil + \lceil \log_2(y_{max} - y_{min}) \rceil\}$ bits where (x_{max},y_{max}) are the coordinates of the bottom right corner of the MBR. The coordinates of the top left corner, however, have to be transmitted once for each contour.

4.3 Decoding of Compressed Weather Radar Data

4.3.1 Pre-Processing of Received Data

At the retrieving end the hole and region contours are identified using their appended tag bits. The control points of each tagged contour are restored to the original coordinate system (with origin at the top left corner of the entire original image) by adding the coordinates of top-left corner of the MBR to those of the individual points.

4.3.2 Contour Retrieval

4.3.2.1 Closed Contour Reconstruction Using B-spline Interpolation

The contours are reconstructed from the control points using a B-spline interpolation [79]. As pointed out in Sec. 3.4, B-splines provide a common mathematical representation of standard analytic shapes as well as free-form curves, and are projection invariant. Such a representation is chosen here for the following advantages:

1. Changes to a control point only affects the curve only in its vicinity
2. Any number of control points can be added without increasing the degree of the curve
3. Adding multiple points at or near a position draws the curve towards that position
4. Closed curves can be represented by making the first and last points of the point sequence identical.

4.3.2.2 Choice of Knot Vector

The degree of spline interpolation and choice of the knot values determine the quality of the reconstructed contour. The scheme developed in the thesis uses uniform knot vector values for spline interpolation.

4.3.2.3 Resolution of Interpolation Parameter

Let k_i+1 denote the number of control points received for the i^{th} contour. The parameter t in equation (3.1) is incremented in steps of $\frac{1.0}{k\times(k_i+1)}$ where k is an integer.

4.3.2.4 Filling of Missing Points

A low value for k may cause gaps between successive points. Any gaps resulting between successive reconstructed points are filled using Breshenham's line drawing algorithm [77].

4.3.2.5 Discarding Duplicate Points

Higher values of k generate duplicate points because of rounding off to integer pixel values, which may be discarded. Through repeated numerical experiments, a range of 2-10 is found to be appropriate for k, and a 'safe' value $k = 10$ has been adopted, which almost completely avoids gaps while generating some duplicate points.

4.3.3 Filling of Region Contours

After the contours have been reconstructed, their interiors have to be filled with a color corresponding to the threshold value used for generating the contour. This is performed using a region-filling algorithm. For weather radar data the region contours for smaller thresholds are always larger than those of the higher thresholds. On the contrary, the hole contours at lower thresholds are always smaller than those at higher thresholds. An efficient scheme should fill these contours without any redrawing i.e., only the area inside the region contours but outside the hole contours should be filled with the color corresponding to the given threshold.

4.3.3.1 Contour Drawing

The proposed scheme first draws the region and hole contours at the lowest threshold value and the next higher threshold, and fills the area between them with the color

corresponding to the lowest threshold. This process is then repeated for each pair of consecutive threshold values to generate the complete multi-threshold weather picture.

4.3.3.2 Identification of Seed Point

An arbitrary point inside a region contour at the lowest threshold value is first identified [58]. Similarly an arbitrary point inside the hole contour at the next higher threshold level is also obtained. These points are used as seed points for the first-level boundary fill algorithm. This scheme is repeated for the next higher threshold, and so on. A composite contoured image of the reflectivity is generated when all the thresholds are exhausted.

4.4 Metrics for Performance Evaluation of the Compression Scheme

The performance of a compression scheme may be evaluated using the resulting compression ratio and the fidelity of reconstruction. Compression ratio is defined here as the ratio of the number of bits required to store the raw reflectivity image to the number of bits required to transmit the contour-coded image. The fidelity may be evaluated quantitatively and qualitatively. The metrics commonly used for quantitative assessment are Mean-Square-Error, Root-Mean-Square-Error (RMSE), Signal-to-Noise-Ratio (SNR), and Peak-Signal-to-Noise-Ratio (PSNR) [84, 85, 86]. In this work, the RMSE between the original and reconstructed contours is used as a quantitative metric for evaluation of the scheme [84]. SNR and PSNR, which are of great importance in continuous-tone image reproduction, are not appropriate criteria in the case of contour reconstruction. For qualitative assessment, a visual comparison is made between the original and reconstructed reflectivity fields.

The total number of bits B required for storing or transmitting the contoured and

encoded image is given as

$$B = \sum_{i=0}^{P} \{k_i \left[\lceil \log_2(x^i_{max} - x^i_{min}) \rceil + \lceil \log_2(y^i_{max} - y^i_{min}) \rceil\right] + \lceil \log_2(k_i) \rceil\} + T \times \lceil \log_2(T) \rceil + P \times [1 + \lceil \log_2(M) \rceil + \lceil \log_2(N) \rceil] \quad (4.4)$$

where T, P and k_i respectively denote the actual number of thresholds used for encoding, the total number of contours traced (at all threshold levels), and the number of control points in i^{th} contour. The first term within the summation pertains to the bit requirement for transmitting the control points, and the second term corresponds to the number of control points. The remaining terms of equation (4.4) relate to the threshold indices, tag bits, and the corner points of the contour MBRs. Equation (4.4) does not include a few low-bit-rate terms that may have to be included in the file headers.

The compression ratio (CR) can then be expressed as

$$CR = \frac{M \times N \times \lceil \log_2(I) \rceil}{B} \quad (4.5)$$

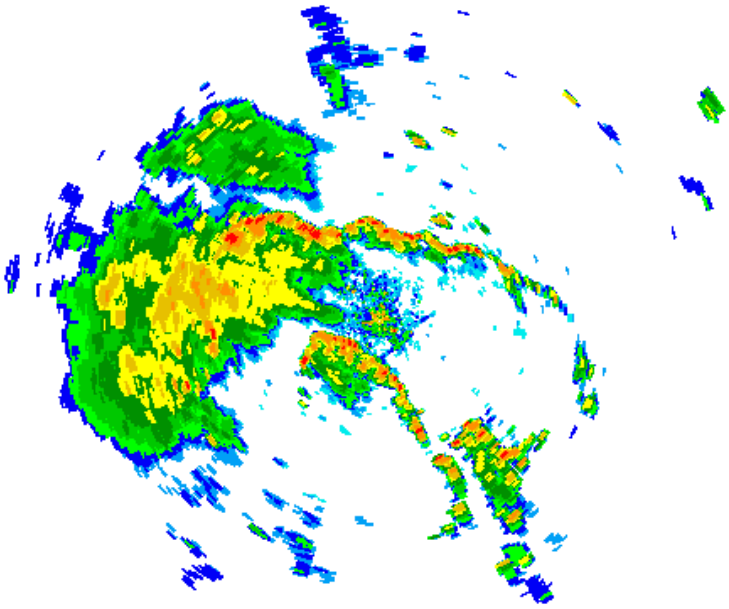
where I denotes the peak reflectivity value in the given image.

4.5 Application to Real Data

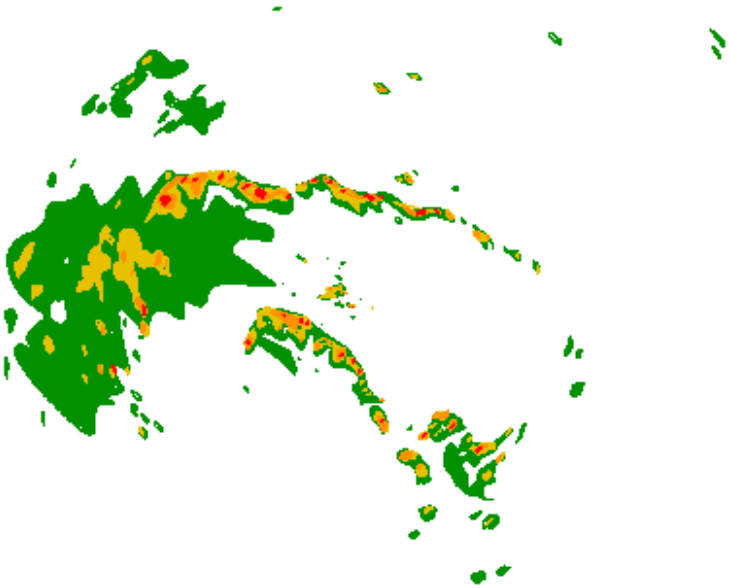
4.5.1 Application to Reflectivity

The compression algorithm has been tested using the Plan Position Indicator (PPI) images of radar reflectivity Level II data from a WSR-88D radar. The size of the image considered is 512 × 512 pixels. The reflectivity values are represented with 8-bit resolution, yielding a scale of 0-255 corresponding to a maximum reflectivity interval of 0-64 dBZ. The NOAA standard color table ('legend') is used for displaying the reflectivity distribution, with a slight modification to display the background as white, unlike the black background in the original display (Fig. 4.9(a)).

The compression ratio for the scheme as evaluated from equation (4.5) is 112. This



(a) Original PPI image



(b) Reconstructed PPI image using NWS thresholds

Figure 4.9: (a) Sample PPI Reflectivity field from WSR-88D Doppler weather radar in Level II format, and (b) its reconstructed version in NWS-thresholded form.

Figure 4.10: The raw contour of Fig. 4.4 (black line), and its reconstruction using spline scheme (red), polygonal method (Sec. 3.3.2.2) (green) and polygonal-ellipse method (Sec. 3.3.2.1) (blue).

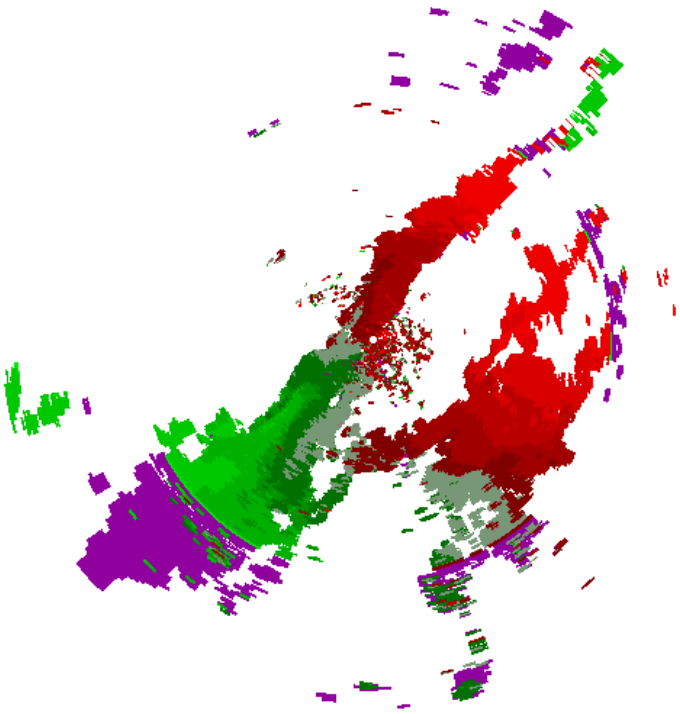
level of compression, over two orders of magnitude is very high compared to any of the general-purpose image compression methods. The reconstructed image (Fig. 4.9(b)) is found to retain the meteorologically significant features of the original image.

The largest-area contour appearing in the 30-dB-threshold section of the picture is encoded using 66 control points. A second degree spline fitted with these 66 control points is shown in Fig. 4.10. The Root Mean Square (RMS) departure of the reconstructed contour from the original is found to be 1.65. For comparison, the contours obtained by the methods outlined in [55] and [56] are also included in Fig. 4.10. These contours yield RMS departures of 1.95 and 2.12 respectively which are higher than the spline value.

4.5.2 Application to Other Data Fields

The contour-based compression scheme is readily applicable to other pictorial data fields as well. Besides reflectivity, the most common data field in the Doppler radar context is the distribution of radial velocities. Figure 4.11 shows a sample radial velocity field and its retrieved form using threshold values of -14, -6, -1, 1, 6 and 14 m s^{-1}. The compression ratio in this case is $\sim$40. The radial velocities smaller than the chosen threshold are used to generate a binary image for negative threshold values. The somewhat lower compression ratio compared to the reflectivity field quoted earlier is because of the use of a larger number of thresholds necessitated by the bipolar nature of radial velocity data.

The same method can also be applied to the other important Doppler moment, the spectrum width, in principle. However, in practice it does not yield an efficient solution because of the usually noisy nature of such fields which results in a very large number of small contours. Further, the operational utility of such a procedure for higher-moment fields is limited because it is normally only the reflectivity field which is viewed and interpreted by operators in the NWS-contoured form.



(a) Original Velocity PPI

(b) Retrieved Field

Figure 4.11: (a) Sample PPI velocity field from WSR-88D Doppler weather radar in Level II format (elevation 0.48^o, display 512×512 pixels with 8-bit depth). Radar location: Greer, SC, Date and time: 23 Jan 2006, 15:04:26 UTC [8], and (b) its reconstructed version in custom-thresholded form.

4.6 Qualitative Assessment of the Compression Scheme

The compression technique developed here is not lossless. Hence some very fine details of the contours will be smoothed. The extent of loss of fine structure is a function of the severity of compression, and methods for tradeoff between compression ratio and loss of detail are presented. At extreme compression ratios of the order of 100 or more, for which results are presented, the average fidelity of reproduction (rms departure of the reconstructed contour from the original) is of the order of 1-2 pixels, but locally the departure may reach a few to several pixels. Thus, kilometer or sub-kilometer scale phenomena such as tornado signatures may be partially smoothed. This is true not only for the proposed method, but also for the methods by Burdon [56] and Gertz and Grappel [55] for comparable compression ratios, as shown in Fig. 4.10.

In aviation application, which is the primary motivation for this study, kilometer-scale details are not of primary interest. This is because aircraft must avoid severe weather zones by safe margins of distance. Accurate delineation of high-precipitation areas assist the pilot and air traffic controller in planning detours around hazardous zones and locating safe flight corridors between such zones. For such use, the extremely fine structure of the contours is not of importance.

If, however, the contoured data are to be used for detection of fine features such as tornadoes, then the method developed here would have to be adaptively applied for compression. In such a method, the small feature of interest would first be identified and its location delineated on the original data field (PPI display) either by a human operator or by an automated recognition scheme [87]. Automatic recognition may be based on sharp gradients in reflectivity and/or velocity parameters as well as presence of characteristic patterns. Then a dense set of control points may be located on the contour in the area of the hazardous feature. In extreme cases, all the pixels over the critical segments may be included as control points. Since such areas are likely to be small, the fractional increase in the number of control points in the overall field will not be large, while the fidelity of the contour through the hazardous feature will be very high. Intelligent or recognition-based contouring is not within the scope of this thesis.

4.7 Summary

A novel method of compressing weather radar data for transmission over low-capacity communication links between ground and aircraft is presented. The method, based on contour encoding and spline reconstruction, achieves compression ratios over 100 as against general-purpose methods that give modest levels of compression. The compression ratio may be varied by addition or deletion of control points with a resultant decrease or increase in the reconstruction error. However, a drastic reduction in the number of control points will significantly degrade the performance of the scheme.

Chapter 5

Effect of Parameter Variation

5.1 Introduction

A scheme for high compression of weather radar data was presented in the last chapter. The performance of the scheme depends on the value of the parameters used in the method. In particular, the quality of the reconstructed image depends on the choice of the thresholds for contour generation, the degree of spline interpolation, and the extent of smoothing of the reference contour. This chapter presents the results of systematic variation in these parameters in order to help in optimizing the basic scheme, enabling an explicit tradeoff between compression ratio and fidelity of retrieval, and devising variants of the basic method.

5.2 Smoothing Length

An important parameter affecting the generation of control points is the length of the arc over which the original contour is to be smoothed in order to act as the reference curve for obtaining the control points (Sec. 4.2.3). Too small a smoothing length would make the reference curve almost as jagged as the original contour, and the control points obtained by the maximum departure criterion would not be reliable. Over-smoothing would make the reference curve overly biased inwards such that its intersections with the

original contour would be minimized, thus missing out many of the valid control points. Tables 5.1 and 5.2 respectively show the overall compression ratios and the RMSE for four reflectivity fields as function of the smoothing length, expressed as percentage of the original contour length, while using the NWS thresholds. The results in Table 5.2 are based on a second-degree spline interpolation.

Table 5.1: Compression ratio as function of percentage of smoothing length (bold numbers indicate the 'best' values)

Percentage	**Field 1**	**Field 2**	**Field 3**	**Field 4**
5	109.707	103.835	113.784	122.986
10	105.634	100.280	116.173	121.398
15	109.232	103.395	121.977	123.938
20	**111.717**	**105.448**	**124.927**	**128.683**

Table 5.2: RMSE of contour reconstruction as function of percentage of smoothing length (bold numbers indicate the 'best' values)

Percentage	**Field 1**	**Field 2**	**Field 3**	**Field 4**
5	1.249390	1.164300	**1.13833**	1.52562
10	**0.884273**	**0.884022**	1.31600	**1.18894**
15	1.292630	1.113530	3.78364	3.60833
20	1.405210	1.539360	3.19848	3.53206

An optimal choice of smoothing percentage should be such as to minimize the RMSE while maximizing the compression ratio. The Tables show that these parameters are data-dependent. However, as seen from Table 5.2, a smoothing length of 10% is found to be optimum or near-optimum in most cases. Since the optimum values in Table 5.1 are found for the highest tabulated smoothing length of 20%, it may be tempting to explore higher smoothing lengths in search of better compression ratios. However, such is not attempted since Table 5.2 shows that smoothing higher than about 10% leads to progressively worsening reconstruction errors.

5.3 Degree of Interpolation

Upon reception at the destination or retrieval from a storage medium, the compressed contour data need to re-composed into an actual contour. As mentioned before, a B-spline interpolation is used to regenerate the contour from the control points. This process offers some freedom of parameters which must be optimally chosen. Of importance here is the degree of the spline. Table 5.3 shows the effect of the degree of interpolation (over the range 1 to 5) on the RMSE of reconstruction of the contours for four reflectivity fields obtained at a radar elevation angle of 0.35^o. The standard NWS thresholds have been used together with a smoothing length of 10% for obtaining the respective reference contours.

Table 5.3: RMSE of contour reconstruction as function of degree of interpolation.

Degree	**Field 1**	**Field 2**	**Field 3**	**Field 4**
1	0.785433	0.816937	1.21413	1.02699
2	0.884273	0.884022	1.31600	1.18894
3	0.993228	0.988289	1.49077	1.29947
4	1.115750	1.094810	1.51259	1.37056
5	1.212050	1.187960	1.62094	1.51071

The error of reconstruction is seen to increase monotonically with the degree of interpolation. This is not unexpected, since the control points lie on the original contour itself, and higher degrees of smoothing takes the reconstructed contour farther from these vertices (Fig. 3.10). However, using a spline of degree 1 is not acceptable because this amounts to joining the successive vertices (control points) by straight-line segments, which gives the curve a jagged or saw-tooth-like appearance. It may therefore be concluded that a spline of degree 2 is the best compromise for reconstructing the contours smoothly while minimizing the error.

5.4 Choice of Thresholds

Although the NWS thresholds are commonly used for contouring reflectivity fields, in many applications these may be considered too coarse (i.e., too widely spaced). This is especially true in cases where the maximum reflectivity is not too high, since only one or two of the NWS thresholds may fall within the range of reflectivities present in the field. In such cases the reconstructed picture may not convey the full visual impact of the original reflectivity field. To obviate this difficulty, one may choose a higher number of thresholds, i.e., space the thresholds closer. One convenient possibility is to augment the number of thresholds by introducing an extra level midway between consecutive NWS levels.

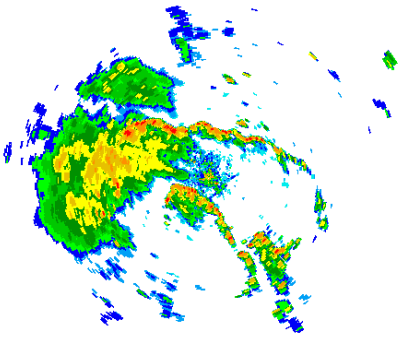
Table 5.4: Compression ratios as function of thresholds

Field	NWS Thresholds	Augmented NWS Thresholds
1	105.634	59.0498
2	100.280	56.4905
3	116.173	63.7161
4	121.398	69.0125

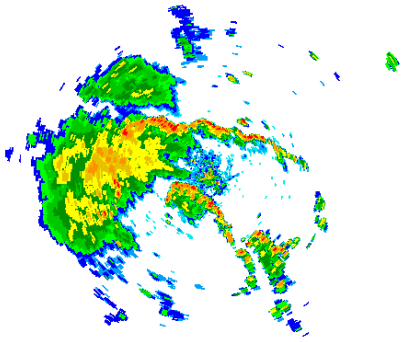
Table 5.4 shows the overall compression ratios achieved in four different reflectivity fields by using the basic and augmented NWS threshold levels. As expected, the introduction of additional thresholds reduces the compression ratios since additional thresholds amount to more 'planar cuts' through the 3-D reflectivity distribution, leading to more contours and hence control points to be generated and transmitted for the same original picture. This increase in the compressed data bandwidth is to be weighed against the enhancement in the quality of visual reproduction as shown in Fig. 5.1.

5.4.1 Summary

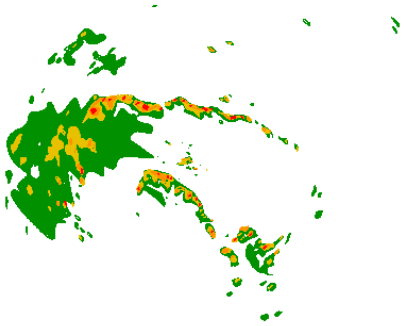
The performance of the basic scheme is evaluated by various choices of the smoothing lengths and thresholds employed during encoding, and the degree of spline reconstruction during decoding. The metrics used for comparing the various parameters are the



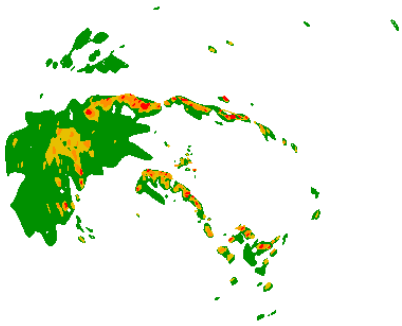
(a) Original Reflectivity PPI (Field 1)



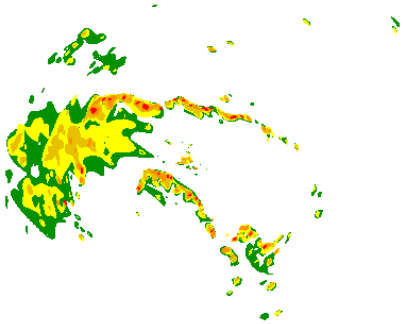
(b) Original Reflectivity PPI (Field 2)



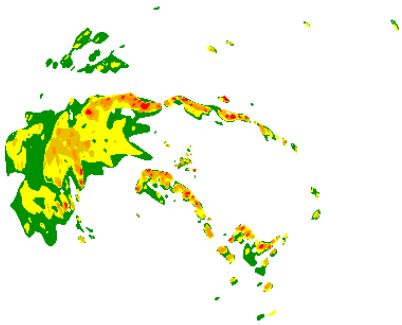
(c) Retrieved Field 1 (NWS Thresholds)



(d) Retrieved Field 2 (NWS Thresholds)



(e) Retrieved Field 1 (Augmented NWS Thresholds)



(f) Retrieved Field 2 (Augmented NWS Thresholds)

Figure 5.1: Fidelity of Reconstruction using NWS and Augmented NWS Thresholds

compression ratios achieved, root mean square error of the retrieved contours, and the visual fidelity of the reconstructed reflectivity images. A smoothing percentage of 10% during control point extraction has been found to be optimal for most of contours of significant size. A degree of 2 is found to be most suitable for spline reconstruction. The addition of intermediate thresholds improves the quality of representation but with a penalty on the compression ratio.

Chapter 6

Compression Algorithm Enhancements

6.1 Introduction

Although the basic compression scheme works very well and yields compression ratios better than two orders of magnitude while retaining all the vital meteorological information, there is still scope for further improving the performance of the scheme both in terms of compression ratio as well as fidelity of data retrieval by improving and fine-tuning the basic method. Two methods of performance improvement are discussed in the following sections.

6.2 Stretching of Control Points

A well-known property of spline interpolation is that for orders greater than unity, the reconstructed curve does not pass through the control points themselves (Fig. 3.10). The higher the order of interpolation, the smoother is the appearance of the curve but greater its departure from the control points. Since the control points were originally chosen from those lying on the contour itself, any departure of the reconstructed curve from these points inherently introduces an RMS error into the retrieval of the contour.

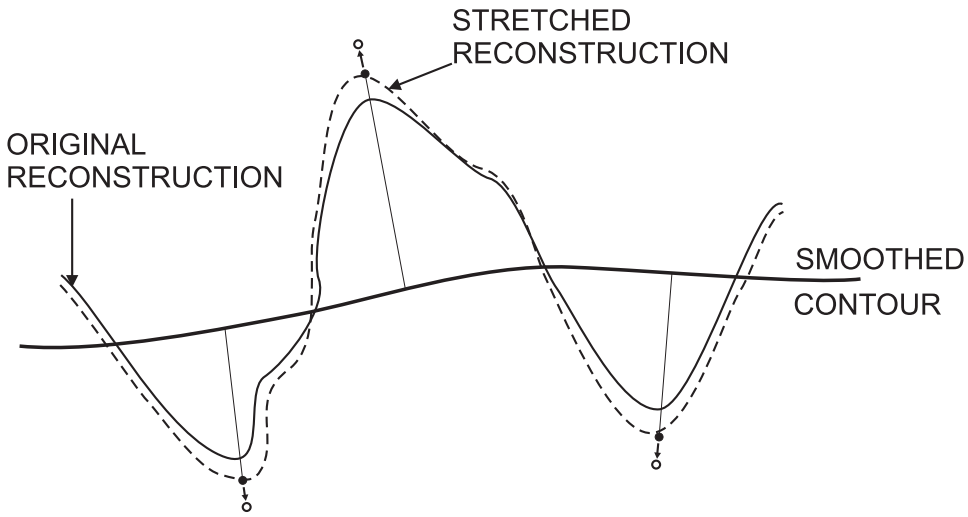


Figure 6.1: Schematic showing principle of control point stretch

This drawback may be partially overcome by stretching the control points artificially about the smoothed contour prior to transmission. Figure 6.1 schematically depicts the principle involved. The solid dots in Fig. 6.1 represent the original control points, and the reconstructed contour passes by these points by a small distance. Systematically displacing the control points farther from the reference line (smoothed contour) to the positions indicated by the hollow dots tends to compensate for the miss distance and pulls the reconstructed contour (dashed curve) closer to the original control points and hence the original contour.

The basic contour encoding scheme used here has the freedom to introduce additional non-extremal control points in situations where the crossing points between the original and smoothed contours are far apart (Sec. 4.2.3.3). In such cases additional control points are introduced half-way between the crossings and the extremal control points as shown in Fig. 6.2. These additional control points are subjected to a different stretching algorithm than the 'regular' or extremal control points. In Fig. 6.2, for example, the additional points B and C are moved outward in the directions DB and DC respectively, unlike the 'regular' point A which is moved out along the direction DA which is normal to the reference line (smoothed contour). This strategy minimizes computation and memory needs while not leading to any significant performance difference with respect

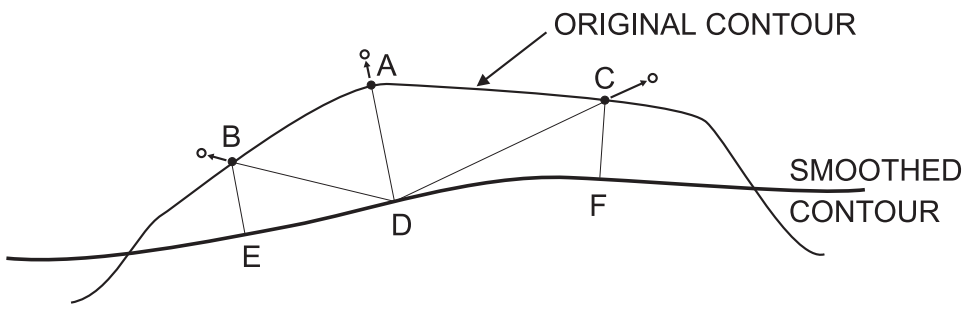
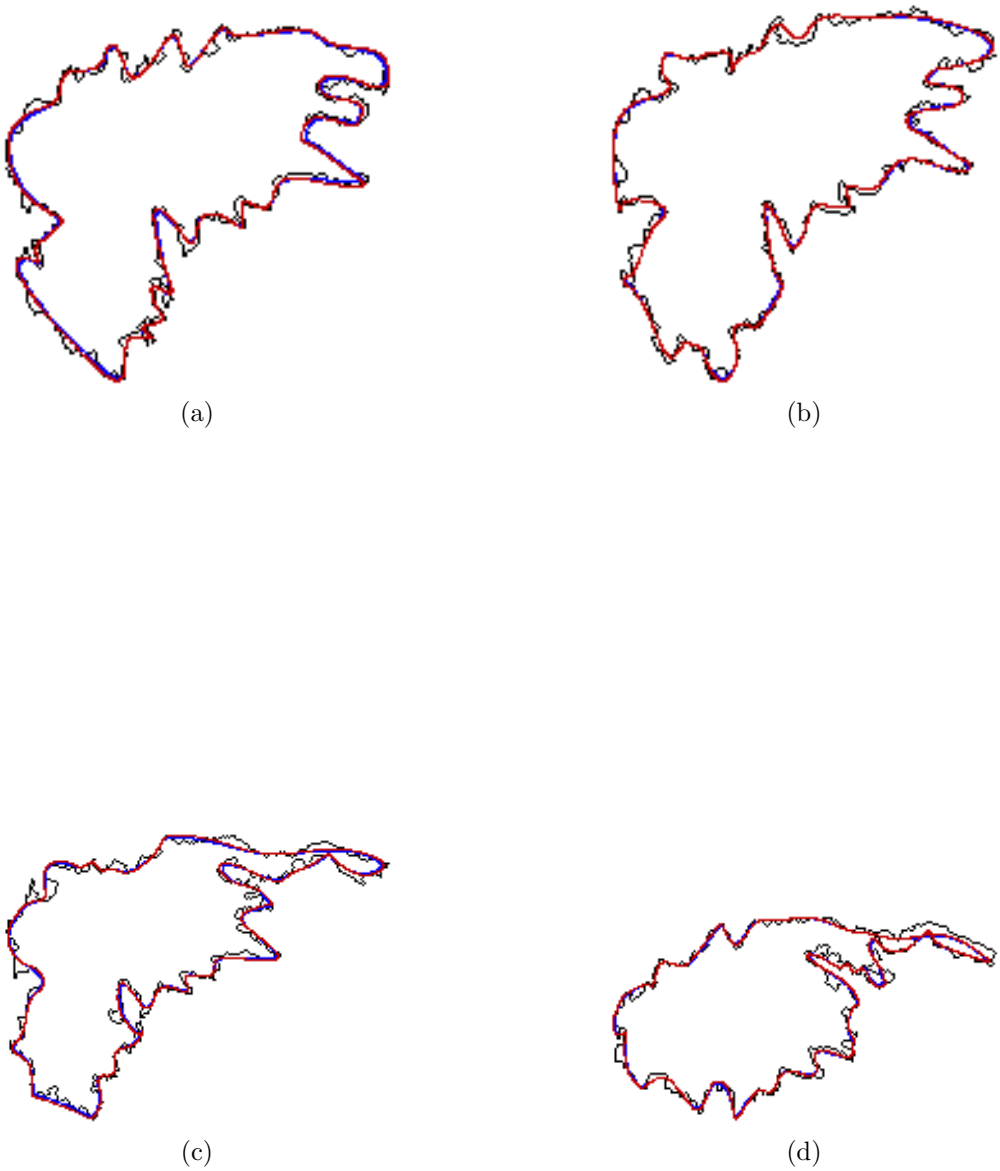


Figure 6.2: Schematic showing the geometry of stretching the additional control points to stretching along normal directions EB and FC.

Table 6.1: RMS contour retrieval error as function of percentage of stretch (bold numbers indicate the 'best' values)

Stretch %	**Field 1**	**Field 2**	**Field 3**	**Field 4**
0	0.884273	0.884022	1.31600	1.18894
1	0.884273	0.884022	1.31600	1.18894
2	0.884273	0.884022	1.31600	1.18894
3	0.881816	0.883848	1.31313	1.18822
4	0.877065	0.882626	1.30654	1.17727
5	0.875231	0.882975	1.30622	1.18185
6	0.874312	0.879651	1.29831	1.17509
7	0.865074	**0.875698**	1.29073	1.17757
8	0.861537	0.881052	1.29611	1.16144
9	0.858454	0.878862	1.27919	1.11312
10	0.855078	0.888546	**1.26470**	**1.11003**
11	0.854420	0.892701	1.26477	1.11581
12	**0.854326**	0.894082	1.26503	1.11573
13	1.459560	0.888893	1.27045	1.13251
14	1.460060	0.890193	1.27197	1.12598
15	1.461980	0.888199	1.27183	1.12651

Since the reconstructed contour weaves about the reference line at variable distances, it appears logical to stretch the points away from the reference line by a certain factor of their original displacements rather than by a fixed distance. A range of stretch factors have been tried out in order to optimize the scheme. Table 6.1 shows the variation of the overall quality of retrieval of a contoured image (as given by the RMS departure)



(a) (b)

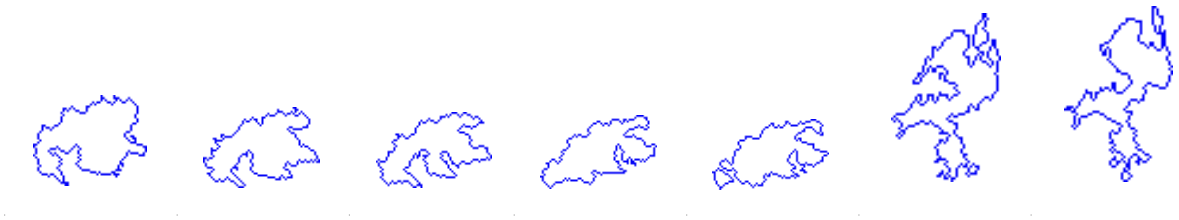
(c) (d)

Figure 6.3: Reconstructed contours for original and stretched control points (blue and red respectively) compared with the original contour (black) at four stages of evolution

with respect to the stretch factor using a second-degree spline interpolation. There would obviously be an optimum stretch factor, since excessive stretching would overcompensate the miss distance and pull the retrieved contour too far away from the reference line, beyond the original control points. From Table 6.1 the optimum stretch factor appears to be in the range of 7-12% and the corresponding improvement in the RMS departure lies between 1-7% over the original unstretched contour fit. Figure 6.3 shows a retrieved contour depicting the improvement in the fit due to stretching. If a uniform stretch factor is to be employed then a figure of 10% may be adopted. However, since the encoding process including stretching is carried out at the transmitting end where all the information regarding the original contours and the control points is available and high levels of computing power can be marshaled, it is possible to determine and adopt the optimum stretching factor exactly with modest computational effort. The optimum can be worked out for each data field, or even separately for each contour within data fields.

Table 6.2: Compression ratio as function of percentage of stretch

Stretch %	**Field 1**	**Field 2**	**Field 3**	**Field 4**
0	105.634	100.280	116.173	121.398
1	105.634	100.280	116.173	121.398
2	105.634	100.280	116.173	121.398
3	105.634	100.280	116.173	121.398
4	105.634	100.280	116.173	121.398
5	105.634	100.280	116.173	121.398
6	105.634	100.280	116.173	121.398
7	105.634	100.222	116.173	121.398
8	105.634	100.222	116.173	121.398
9	105.634	100.222	116.141	121.475
10	105.634	100.222	116.109	121.475
11	105.634	100.222	116.109	121.475
12	105.634	100.222	116.109	121.475
13	105.597	100.222	116.109	121.398
14	105.597	100.222	116.109	121.398
15	105.597	100.222	116.109	121.398



(a) (b) (c) (d) (e) (f) (g)

Figure 6.4: Original contours taken from NEXRAD data at Greer, SC, 23 Jan 2006

It is worth mentioning here that the stretching process will have negligible effect, if any, on the compression ratio of the contour coding scheme. This is because the data volume of the transmission is overwhelmingly dependent on the 'number' of control points, which is unaffected by the stretch process. There would, however, be a weak effect on the data volume if any of the contour size(s) are on the borderline such that a slight increase in their size would enhance the bit requirement of the bounding rectangle (Sec. 4.2.4) by an extra bit. This fact is corroborated from the results in Table 6.2, which shows small but discrete jumps in compression ratios.

Table 6.3: Comparison of RMS contour retrieval errors (bold numbers indicate the 'best' values)

Method	**Contour 1** (66)	**Contour 2** (69)	**Contour 3** (84)	**Contour 4** (66)
Spline	**1.53863**	**1.60564**	1.96551	1.97238
Polygonal	1.94767	1.83367	1.88074	1.77129
Polygonal-Ellipse	2.11772	1.61304	**1.51504**	**1.25601**

It is instructive to compare the quality of contour reconstruction obtained by the current method (incorporating 10% stretch) with that of the earlier methods used for reflectivity contour encoding. In particular the methods by Burdon [56] and Gertz and Grappel [55] are referred to since these have been specifically developed for weather radar contours (Sec. 3.3.2). Table 6.3 shows the RMS error of reconstruction of the contours shown in Fig. 6.3 by using the three methods. Each of the four contours in Table 6.3 is compressed to a common extent in the three methods. This is achieved by using, as far

as possible similar number of control points to represent each given contour (indicated in parentheses below the contour number) for all the three methods. However, it is not always possible to retain exactly the same number of control points in each case because these numbers are not directly selected but are indirectly determined by tailoring the departure threshold value. The results are mixed, suggesting that the proposed scheme performs better than the other two schemes for only two of the four contours.

Table 6.4: RMS contour retrieval error as function of percentage of stretch (bold numbers indicate the 'best' values)

Stretch %	**Contour 1**	**Contour 2**	**Contour 3**	**Contour 4**	**Contour 5**	**Contour 6**	**Contour 7**
0	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93498
1	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93498
2	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93498
3	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93498
4	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93326
5	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93326
6	0.76912	0.71774	0.67766	0.75461	1.1098	1.4162	0.93326
7	0.76912	0.71774	0.67766	0.75461	1.1098	1.3860	0.93326
8	0.76912	0.71774	0.67766	0.75461	1.1098	1.3860	0.93326
9	0.76912	0.71774	0.67449	0.75461	1.0932	1.3798	0.92806
10	**0.76299**	0.72970	0.67130	0.76073	1.0630	1.3839	**0.91581**
11	0.76299	0.72075	**0.66164**	0.74845	1.0763	**1.3757**	0.92457
12	0.76299	0.72075	0.66164	0.74845	1.0763	1.3757	0.92457
13	0.87210	**0.69941**	0.67130	0.72648	1.0554	1.3819	0.95208
14	0.87210	0.69941	0.67130	0.72648	1.0554	1.3819	0.95208
15	0.87210	0.69941	0.67130	0.72648	1.0554	1.3819	0.95208
16	0.87210	0.69941	0.67130	0.72648	1.0554	1.3819	0.95208
17	0.91416	0.70864	0.75552	**0.72008**	**1.0457**	1.4281	0.99027
18	0.91416	0.70864	0.75552	0.72008	1.0457	1.4281	0.99027

The RMS error for contours in Fig. 6.4 corresponding to a different reflectivity field are given in the Table 6.4, to show the effect of relatively smooth contours. As expected, because of the smoothness of the contours, stretching only marginally improves the fidelity of contour reconstruction. However, the best stretch even for this data set is

around 10%.

6.3 Bit Manipulation for Higher Compression

The primary data load in the contour transmission process arises from the need to designate the coordinates of the control points defining the contours. A certain minimum number of information bits would be required to transmit these coordinates accurately. For weather data intended for display, the required number of bits for each coordinate is determined by the dimensions of the minimum bounding rectangle of each contour in pixels. It is possible to save on storage and transmission bit requirements by truncating the bit stream for each coordinate if the resulting loss of accuracy can be tolerated. Since bit reduction would apply to all the points being transmitted, the gains in terms of data compression can be significant.

Table 6.5: RMS contour retrieval error as function of bits truncated

No. of Bits	Field 1	Field 2	Field 3	Field 4
0	0.884273	0.884022	1.31600	1.18894
1	1.046950	1.015760	1.42590	1.27986
2	1.432900	1.402440	1.66883	1.58341
3	2.915660	2.404190	2.81206	2.61941

It may readily be surmised that truncating the bit stream by dropping one or more of the least significant bits would impact on the data accuracy by increasing the quantization noise. However, since an efficient interpolating algorithm is employed at the retrieving end, a certain level of truncation error is expected to be smoothed out by the spline interpolator. Only when the quantization level is relatively severe will the overall reconstruction accuracy be impacted significantly. To test this hypothesis the overall RMS contour reconstruction error for 1, 2 and 3-bit truncation is calculated as shown in Table 6.5, and the corresponding compression ratios are tabulated in Table 6.6. In performing the truncation it is ensured that at least two of the most significant bits are retained in the data field. For example, if the width of the bounding rectangle of a

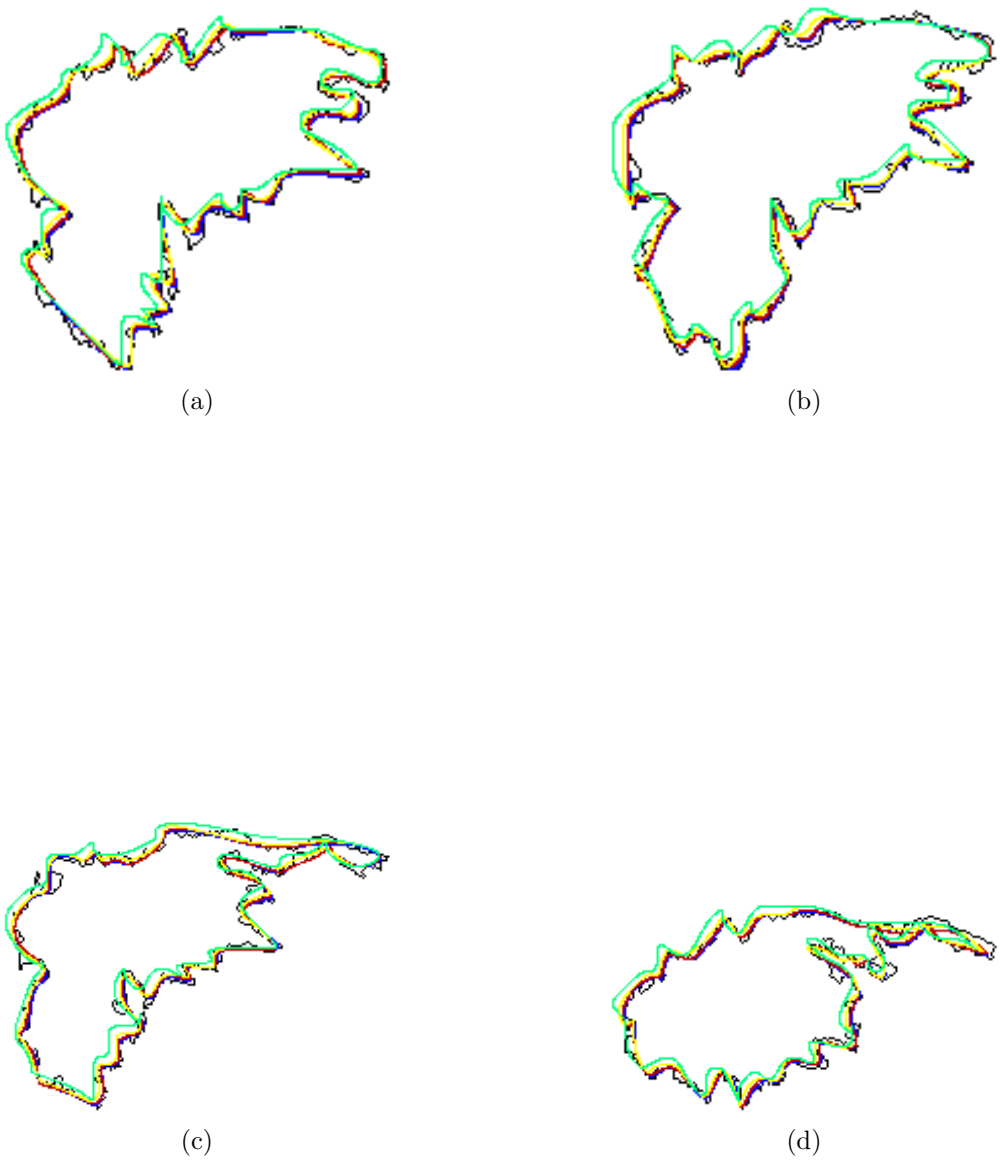
certain contour is 14 pixels, its minimum bit requirement is 4. Retaining the two most significant bits would permit truncation of not more than 2 bits. If the same rectangle has a vertical depth of 27 pixels requiring 5-bit description, that dimension would permit up to 3 bits of truncation. Such adaptive truncation is implemented for generating the results of Tables 6.5, 6.6 and 6.7.

Table 6.6: Compression Ratio as function of bits truncated

No. of Bits	**Field 1**	**Field 2**	**Field 3**	**Field 4**
0	105.634	100.280	116.173	121.398
1	120.811	114.586	132.832	139.198
2	132.622	125.061	145.464	152.001
3	142.363	132.748	153.446	161.096

The results in Table 6.5 show that 1-, 2- and 3-bit truncation causes the RMS error to increase by 8-18%, 27-62%, and 114-230% respectively among the four data sets considered. It should be noted that although the increase in RMS errors is high in percentage terms, especially for higher levels of truncation, their absolute values are still modest, being of the order of 1, 1.5 and 3 pixels for 1, 2 and 3-bit truncation respectively. Thus the reconstructed contours would still be fairly usable. Further, Tables 6.5 and 6.6 show that there is a direct correspondence between the increase in compression ratio and the degradation of RMS reconstruction error. This gives the user the choice of an explicit tradeoff between retrieval quality and extent of compression. In situations of extremely limited channel bandwidth, the user may opt for 2 bits of truncation (or even as much as 3 bits) and pack up to 25% more data (35% for 3 bits) while accepting an error of 1.5 pixels (3 pixels) on the contour definition.

Fig. 6.5 shows the quality of the retrieved contours for various levels of bit truncation. As expected, higher levels of truncation increase the departure of the reconstructed contour from the original. Further, it is also apparent that truncation shifts the reconstructed contour in a preferred direction, i.e., it introduces a bias in the contour shape. Again, this is expected as long as truncation is performed by simply dropping the least significant bits, i.e., by always rounding off the control point coordinates downwards.



(a) (b)

(c) (d)

Figure 6.5: Reconstructed contours for 0-bit (blue), 1-bit (red), 2-bit (yellow), and 3-bit (green) truncation compared with the original contour (black) at four stages of evolution

To compensate for this bias, the integer mean of the maximum possible quantization error is added to the truncated bit stream. Thus, for 1-, 2- and 3-bit truncation the bias compensation values are taken as 0, 2 and 4. Since this compensation is carried out at the receiving end, the unbiasing procedure leaves the compression ratio unchanged.

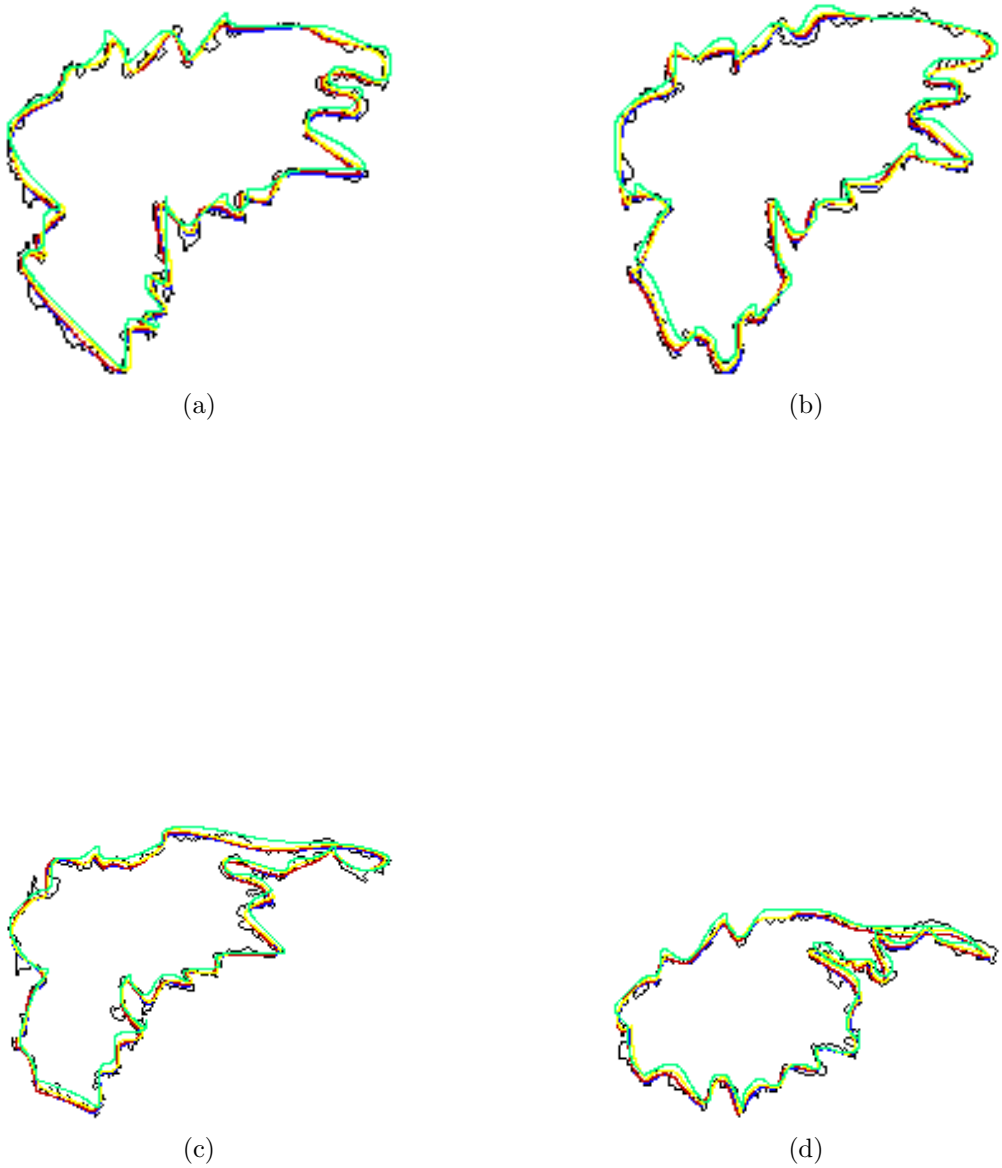
Table 6.7 shows the RMS errors of the contours reconstructed from the truncated control point bit streams after bias compensation. The RMS errors here for 1, 2 and 3-bit truncation are, respectively, 8-18%, 25-57% and 61-140% more than the retrieved contour without any truncation in absolute terms. These translate to about 1, 1.5 and 2 pixels of error respectively. These are found to be much lower than the errors caused by straightforward downward truncation (i.e., before bias correction). The improvement in the quality of contour retrieval because of bias removal is depicted in Fig. 6.6.

Table 6.7: RMS contour retrieval error with truncated bits after bias removal

No. of Bits	Field 1	Field 2	Field 3	Field 4
0	0.884273	0.884022	1.31600	1.18894
1	1.046950	1.015760	1.42590	1.27986
2	1.389160	1.304700	1.64999	1.53904
3	2.119670	2.126820	2.12012	2.12088

6.4 Summary

Two refinements to the basic scheme for weather radar data compression are presented. A systematic method of adjusting the position of control points to compensate for the smoothing introduced by spline interpolation is presented. The control points are stretched away from the reference (smoothed) contour by a fraction of their distance from it. An improvement of around 8% in the fidelity of contour reconstruction is achieved with no or negligible increase in the compression ratio. Bit manipulations to obtain higher compression ratios are also explored. An additional compression ratio of 25% or even 35% is shown to be obtained if small errors of 1-2 pixels in contour representation



(a) (b)

(c) (d)

Figure 6.6: Reconstructed contours for 0-bit (blue), 1-bit (red), 2-bit (yellow), and 3-bit (green) truncation after bias removal compared with the original contour (black) at four stages of evolution

can be tolerated. A bias compensation method has been suggested to minimize the increase in reconstruction error due to bit truncation while leaving the compression ratio unaffected.

Chapter 7

Conclusions

7.1 General Summary

The high-volume and multi-parameter weather radar data have to be compressed significantly for archival and transmission purposes. A lossy compression scheme would be an obvious choice to compress these data but such a method is expected to preserve the meteorological information in them even while delivering for more efficient compression than general-purpose methods.

Weather radar reflectivity data are commonly expressed in contoured form and hence a method for high compression based on contour encoding has been developed in this thesis. The method achieves compression ratios over two orders with minimal degradation in the contour shapes. The method compares well with other encoding methods developed for contour-based weather data compression. However, the loss incurred due to the present as well as earlier methods should be carefully evaluated with regard to the intended application.

The proposed scheme captures random undulations in a contour by extracting control points using a method that calculates the deviation of the contour from a reference derived from the contour itself through a smoothing process. The method can be scaled to achieve a range of compression ratios by deletion of control points but with corresponding degradation in the quality. The contour is reconstructed at the receiving end by spline

interpolation for which a degree of two has been shown to be the best choice.

A study has been conducted to determine the optimum parameters of the compression algorithm. A smoothing percentage of 10% is found to be optimal for extraction of control points. A degree 2 of spline interpolation has been shown to be most suitable for contour reconstruction. The addition of intermediate thresholds between the standard NWS thresholds has enhanced the visual reproduction but with a significant reduction in the compression ratio.

The basic scheme has been enhanced to offer better fidelity and compression ratios. In one approach, the control points are stretched 'outwards' in order to compensate for the smoothing introduced by the spline interpolation. As a second option, bit manipulations on the control points are proposed to further enhance compression ratios with minimal degradation in the fidelity.

7.2 Suggestions for Future Work

A uniform knot vector for B-spline interpolation is employed in the thesis to reconstruct a contour from its control points. The knot vector is a critical parameter affecting the interpolation process. Non-uniform rational B-splines (NURBS) offer a convenient representation where the control points can be weighed according to their significance. Methods may be designed to estimate appropriate weights for control points. However, since the weight value also has to be transmitted for each control point, this would significantly increase the data load and hence reduce compression ratios.

Uniform smoothing is used during the control point extraction process, i.e., each contour point within the smoothing interval is given equal importance. It may be appropriate to weight each contour point appropriately by using a smoothing windowing function. A Gaussian smoothing filter or other appropriate smooth filters may be used for this purpose. The resulting additional computing load can be easily handled on the ground where nearly unlimited computing power is available.

The algorithms presented in the thesis compress 2-D weather radar data. Some of

these schemes may be extended to compress 3-D weather radar fields. CAPPI slices through the 3-D weather radar may be taken and the control points for the contours in each slice may be obtained. B-spline method for reconstructing 2-D contours can be extended to reconstruct surfaces from the control points extracted from each CAPPI field.

Bibliography

[1] "Weather radar," http://www.umanitoba.ca/environment/envirogeog/course_notes/53−331/conventional_radar.ppt.

[2] "Doppler radar interpretation," http://www.umanitoba.ca/environment/envirogeog/course_notes/53−331/doppler_radar.ppt.

[3] "Square tracing algorithm," http://www.cs.mcgill.ca/∼aghnei/square.html.

[4] "Moore-neighbor tracing," http://www.cs.mcgill.ca/∼aghnei/moore.html.

[5] "Radial sweep," http://www.cs.mcgill.ca/∼aghnei/ray.html.

[6] "Theo pavlidis' algorithm," http://www.cs.mcgill.ca/∼aghnei/theo.html.

[7] "Defining connectivity," http://www.cs.mcgill.ca/∼aghnei/connectivity.html.

[8] "Integrated radar data services," https://www.radarservices.org/.

[9] A. Rosenfeld, "Picture processing by computer," *ACM Computing Surveys (CSUR)*, vol. 1, no. 3, pp. 147 – 176, September 1969.

[10] A. N. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366 – 406, March 1980.

[11] A. Jain, "Image data compression: A review," *Proc. IEEE*, vol. 69, pp. 349 – 388, March 1981.

[12] C. T. Chen and T. R. Hsing, "Review: Digital coding techniques for visual communications," *J. Visual Commun. Image Represent.*, vol. 2, pp. 1 – 16, March 1991.

[13] M. R. an R.J. Millar and N. Black, "Second-generation image coding: an overview," *ACM Computing Surveys (CSUR)*, vol. 29, no. 1, pp. 3 – 29, March 1997.

[14] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, "High-performance compression of visual information - A tutorial review - Part I: Still pictures," in *Proc. of the IEEE*, vol. 87, no. 6, June 1999, pp. 976 – 1011.

[15] M. I. Skolnik, *Introduction to Radar Systems*, 3rd ed. New York, NY: McGraw-Hill, 2000.

[16] R. J. Doviak and D. S. Zrnic, *Doppler Radar and Weather Observations*, 2nd ed. San Diego, CA: Academic Press, l993.

[17] "Doppler weather radar overview," http://www.nwas.org/committees/avnwinterwx/doppler_weather_radar_overview.htm.

[18] E. R. Hilgendorf and R. H. Johnson, "A study of the evolution of mesoscale convective systems using wsr-88d data," *Weather and Forecasting*, vol. 13, no. 2, pp. 437 – 452, 1998.

[19] P. Mahapatra and V. Makkapati, "Vector reconstruction and mosaicing from multiple doppler weather radar velocity data," in *Proc. International Geoscience and Remote Sensing Symposium*, vol. 6. Anchorage, Alaska: IEEE, September 2004, pp. 3868 – 3871.

[20] ——, "Constant altitude plan position indicator display for multiple radars," U.S. Patent 0 202 884, September 14, 2006.

[21] A. Ernvik, "3d visualization of weather radar data," Ph.D. dissertation, Department of Electrical Engineering, Computer Engineering, Linkping University, 2002.

[22] V. Makkapati and P. Mahapatra, "Constant altitude plan position indicator for radars," U.S. Patent 5 363 107, September 14, 2006.

[23] P. Mahapatra and V. Makkapati, "Synthesis of generalized vertical-plane weather radar imagery along aircraft flight paths," in *Proc. International Geoscience and Remote Sensing Symposium*, vol. 1. Seoul, Korea: IEEE, July 2005, pp. 344 – 347.

[24] P. R. Mahapatra, *Aviation Weather Surveillance Systems: Advanced Radar and Surface Sensors for Flight Safety and Air Traffic Management*. London, UK: IEE Press, 1999, ch. 6.

[25] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the I.R.E*, pp. 1098 – 1102, September 1952.

[26] D. R. McIntyre and M. A. Pechura, "Data compression using static huffman code-decode tables," *Communications of the ACM*, vol. 28, no. 6, pp. 612 – 616, June 1985.

[27] R. G. Gallager, "Variations on a theme by huffman," *IEEE Transactions on Information Theory*, vol. 24, pp. 668 – 674, 1978.

[28] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, pp. 8 – 19, June 1984.

[29] M. Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding*. IEE Press, 2003.

[30] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559 – 559, 1901.

[31] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417 – 417, 1933.

[32] G. H. Dunteman, *Principal Component Analysis*. London, UK: Sage Publications, 1989.

[33] M. Effros, H. Feng, and K. Zeger, "Suboptimality of the karhunen-loeve transform for transform coding," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1605 – 1619, 2004.

[34] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. on Computers*, vol. 23, no. 1, pp. 90 – 93, January 1974.

[35] G. Strang, "The discrete cosine transform," *SIAM Review*, vol. 41, no. 1, pp. 135 – 147, January 1999.

[36] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30 – 44, April 1991.

[37] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Kluwer, 1992.

[38] R. J. Campbell and P. J. Flynn, "A survey of free-form object representation and recognition techniques," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 166 – 210, February 2001.

[39] N. Nasrabadi and R. King, "Image coding using vector quantization: a review," *IEEE Trans. Comm.*, vol. 36, pp. 957 – 971, August 1988.

[40] V. Makkapati and M. Yeary, "Vector quantization of still images using reflected subcodevectors," in *Proc. Thirty-Eighth Annual Asilomar Conference on Signals, Systems, and Computers*, vol. 2. Pacific Grove, CA: IEEE, November 2004, pp. 1749 – 1752.

[41] V. Makkapati, "Swapping of reections of adjacent subcodevectors for vector quantization," in *Proc. International Conference on Systems, Man and Cybernetics*, vol. 3. Hawaii, US: IEEE, October 2005, pp. 2679 – 2684.

[42] V. Makkapati and P. R. Mahapatra, "Vector quantization using reflections of triangular subcodevectors," in *Proc. International Conference on Pattern Recognition*, vol. 3. Hong Kong: IAPR, August 2006, pp. 701 – 704.

[43] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. COM-28, pp. 84–95, January 1980.

[44] H. Freeman, "Computer processing of line-drawing images," *ACM Computing Surveys (CSUR)*, vol. 6, no. 1, pp. 57 – 97, March 1974.

[45] S. Marshall, "Review of shape coding techniques," *Image and Vision Computing (IVC)*, vol. 7, no. 4, pp. 281 – 294, November 1989.

[46] A. Katsaggelos, L. Kondi, F. Meier, J. Ostermann, and G. Schuster, "Mpeg-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE*, vol. 86, no. 6, June 1998.

[47] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103 – 1127, November 2000.

[48] P. Cosman, R. Gray, and M. Vetterli, "Vector quantization of image subbands: A survey," *IEEE Transactions on Image Processing*, vol. 5, no. 2, pp. 202 – 225, February 1996.

[49] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, pp. 205 – 220, April 1992.

[50] V. Makkapati and R. Kumar, "Improved encoding of wavelet cofficients extracted from multispectral and hyperspectral image data," in *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, Ahmedabad, India, December 2002, pp. 191 – 196, http://www.ee.iitb.ac.in/∼icvgip/PAPERS/282.pdf.

[51] R. Kumar and V. Makkapati, "Encoding of multispectral and hyperspectral image data using wavelet transform and gain shape vector quantization," *Image and Vision Computing (IVC)*, vol. 23, no. 8, pp. 721 – 729, August 2005.

[52] A. Kruger and W. Krajewski, "Efficient storage of weather radar data," *Software – Practice and Experience*, vol. 27, no. 6, pp. 623 – 635, June 1997.

[53] N. Wang, S. Madine, and R. Brummer, "Investigation of data compression techniques applied to awips datasets," in *Proc. 18th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*. Orlando, Florida: American Meteorological Society, January 2002, p. CDROM, http://ams.confex.com/ams/annual2002/18IIPS/abstracts/27337.htm.

[54] V. Lakshmanan, "Lossless coding and compression of radar reflectivity data," in *Proc. 30th International Conference on Radar Meteorology*. Munich, Germany: American Meteorological Society, July 2001, p. CDROM, http://ams.confex.com/ams/30radar/30radar/abstracts/21592.htm.

[55] J. L. Gertz and R. D. Grappel, "Storage and transmission of compressed weather maps and the like," U.S. Patent 5 363 107, November 8, 1994.

[56] D. Burdon, "System and method for the adaptive mapping of matrix data to set of polygons," U.S. Patent 6 614 425, September 2, 2003.

[57] M. Alder, *An Introduction to Pattern Recognition*. Osborne Park, Western Australia: HeavenforBooks.com, 1997, ch. 2.

[58] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Berlin, Germany: Springer Verlag, 1982, ch. 7.5.

[59] H. Freeman and L. S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. on Computers*, vol. C-26, no. 3, pp. 297 – 303, March 1977.

[60] W. Kim and R. Park, "Contour coding based on the decomposition of line segments," *Pattern Recognition Letters*, vol. 11, no. 3, pp. 191 – 195, March 1990.

[61] D. Sarkar, "A simple algorithm for detection of significant vertices of polygonal approximation of chain coded curves," *Pattern Recognition Letters*, vol. 14, no. 12, pp. 959 – 964, December 1993.

[62] C. Teh and R. Chin, "On the detection of dominant points on digital curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, October 1989.

[63] B. K. Ray and K. S. Ray, "Detection of significant points and polygonal approximation of digitized curves," *Pattern Recognition Letters*, vol. 13, no. 6, pp. 443 – 452, June 1992.

[64] S.-C. Pei and J.-H. Horng, "Fitting digital curve using circular arcs," *Pattern Recognition*, vol. 28, no. 1, pp. 107 – 116, 1995.

[65] M. K. Kundu, B. B. Chaudhuri, and D. D. Majumder, "A generalised digital contour coding scheme," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 3, pp. 87 – 92, June 1985.

[66] J.-C. Perez and E. Vidal, "Optimum polygonal approximation of digitized curves," *Pattern Recognition Letters*, vol. 15, no. 8, pp. 743 – 750, August 1994.

[67] Y. Xiao, J. J. Zou, and H. Yan, "An adaptive split-and-merge method for binary image contour data compression," *Pattern Recognition Letters*, vol. 22, no. 3-4, pp. 299 – 307, March 2001.

[68] M. Salotti, "Optimal polygonal approximation of digitized curves using the sum of square deviations criterion," *Pattern Recognition*, vol. 35, no. 2, pp. 435 – 443, February 2002.

[69] P. L. Rosin, "Representing curves at their natural scales," *Pattern Recognition*, vol. 25, no. 11, pp. 1315 – 1325, November 1992.

[70] B. Sarkar, L. K. Singh, and D. Sarkar, "Hierarchical representation of digitized curves through dominant point detection," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2869 – 2882, November 2003.

[71] S.-C. Pei and J.-H. Horng, "Optimum approximation of digital planar curves using circular arcs," *Pattern Recognition*, vol. 29, no. 3, pp. 383 – 388, 1996.

[72] J.-H. Horng and J. T. Li, "A dynamic programming approach for fitting digital planar curves with line segments and circular arcs," *Pattern Recognition Letters*, vol. 22, no. 2, pp. 183 – 197, February 2001.

[73] B. Sarkar, L. K. Singh, and D. Sarkar, "Approximation of digital curves with line segments and circular arcs using genetic algorithms," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2585 – 2595, November 2003.

[74] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a line or its carricature," *The Canadian Cartographer*, vol. 10, no. 2, pp. 112 – 122, 1973.

[75] A. Oliviero and G. Scarpetta, "A new approach to contour coding," *Computer Graphics and Image Processing*, vol. 15, no. 1, pp. 87 – 92, January 1981.

[76] S. Biswas, S. Pal, and D. D. Majumder, "Binary contour coding using bezier approximation," *Pattern Recognition Letters*, vol. 8, pp. 237 – 249, 1998.

[77] J. D. Foley, A. V. Dam, S. K. Feiner, and J. F. Hughes, *Computer Graphics: Principles and Practice in C*, 2nd ed. Reading, MA: Addison-Wesley, 1995.

[78] M. N. Salous, D. Pycock, and G. S. Cruickshank, "CBIT-contex-based image transmission," *IEEE Transactions on Information Technology in Biomedicine*, vol. 5, no. 2, pp. 159 – 170, June 2001.

[79] C. D. Boor, *Practical Guide to Splines*. New York, NY: Springer Verlag, 1978.

[80] P. Mahapatra and V. Makkapati, "Ultra high compression for weather radar reflectivity data storage and transmission," in *Proc. 21st International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*. San Diego, CA: American Meteorological Society, January 2005, p. CDROM, http://ams.confex.com/ams/Annual2005/techprogram /paper_82973.htm.

[81] F. Attneave, "Some informational aspects of visual perception," *Psychological Review*, vol. 61, no. 3, pp. 183 – 193, 1954.

[82] S. Biswas, "Contour coding through strectching of discrete circular arcs by affine transformation," *Pattern Recognition*, vol. 34, pp. 63 – 77, 2001.

[83] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *Proc. Sixth International conference on Multimedia*. Bristol, United Kingdom: ACM, 1998, pp. 211 – 218.

[84] P. L. Rosin, "Techniques for assessing polygonal approximations of curves," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 659 – 666, June 1997.

[85] S. Grgic, M. Grgic, and M. Mrak, "Reliability of objective picture quality measures," *Journal of Electrical Engineering*, vol. 55, no. 1-2, pp. 3 – 10, 2004.

[86] D. Pearson and J. Robinson, "Visual communication at very low data rates," in *Proc. of the IEEE*, vol. 73, no. 4, April 1985, pp. 795 – 812.

[87] P. Joe, D. Burgess, R. Potts, T. Keenan, G. Stumpf, and A. Treloar, "The s2k severe weather detection algorithms and their performance," *Weather and Forecasting*, vol. 19, no. 1, pp. 43 – 63, 2004.