

Performance Enhancement of a Spline-based Method for Extreme Compression of Weather Radar Reflectivity Data

Pravas R. Mahapatra¹ and Vishnu V. Makkapati²

¹Department of Aerospace Engineering, Indian Institute of Science, Bangalore - 560 012, India
Email: pravas@aero.iisc.ernet.in

²Research and Technology Group, Honeywell Technology Solutions Lab
151/1, Doraisanipalya, Bannerghatta Road, Bangalore - 560 076, India
Email: vishnu.makkapati@honeywell.com

Abstract—Enhancements are carried out to a contour-based method for extreme compression of weather radar reflectivity data for efficient storage and transmission over low-bandwidth data links. In particular, a new method of systematically adjusting the control points to obtain better reconstruction of the contours using B-Spline interpolation is presented. Further, bit-level manipulations to achieve higher compression ratios are investigated. The efficacy of these enhancements is quantitatively evaluated with respect to achievable compression ratios and the root-mean-square error of the retrieved contours, and qualitatively assessed from the visual fidelity of the reconstructed contours. Results are presented for actual Doppler weather radar data sets.

Keywords - High data compression, weather radar data compression, contour-based data compression, spline based contour reconstruction.

I. INTRODUCTION

Doppler weather radars are now a very important source of data for general meteorological observations as well as for many specialized applications. Modern Doppler weather radars are characterized by their ability to generate very accurate estimates of basic weather parameters such as rainfall intensity, wind speeds and patterns, and atmospheric turbulence levels. They also provide a host of higher level data by automatically processing the basic or raw weather observations. Further they produce these data in 3-dimensions with very high spatial resolution and update rates. This not only helps in accurately identifying and delineating harmful phenomena and estimating their hazard potential, but also in tracking their movement and evolution in nearly real time. Such capability is of very high value in highly dynamic and intensive operations such as aviation.

Because of the very high quality of Doppler weather radar data many nations have either installed or are in the process of installing networks of such radars for seamless coverage of large geographical areas.

The volume scanning capability, high resolution, rapid update, data accuracy, and multiparameter output of Doppler weather radars result in extremely high data output rates. This calls for very high capacities for data processing, archival, transmission, retrieval, and display. While these requirements

are addressed substantially by parallel developments in low-cost and high-volume computing and data processing chains, the possibility of data overload still remains, especially when operating on national or regional scales of coverage.

There are certain applications where the large bandwidth associated with modern Doppler weather radar data pose especially severe constraints. A specific example is the transmission of the data from ground-based sources to aircraft in flight. Presentation of data from ground-based radar networks to the pilot in the aircraft cockpit would greatly assist him or her in taking timely decisions regarding the conduct of the flight. However, the low data capacity of existing ground-to-air data links poses severe constraints in transmitting radar data in raw form. Even where such bandwidth may be available, it would be a premium and expensive resource which must be maximally conserved. In such applications it is necessary to compress the weather radar data to the highest possible extent while maximally retaining its information content with regard to the nature, extent and hazard potential of the weather phenomena displayed.

In this paper we discuss a contour-based encoding method for highly compressing weather radar reflectivity data, and present certain techniques for further enhancing the efficacy of the compression scheme.

II. CONTOUR DISPLAY OF REFLECTIVITY DATA

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 reflectivity 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 (NWS) [1], [2]. 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 contoured representation are shown in Fig. 1.

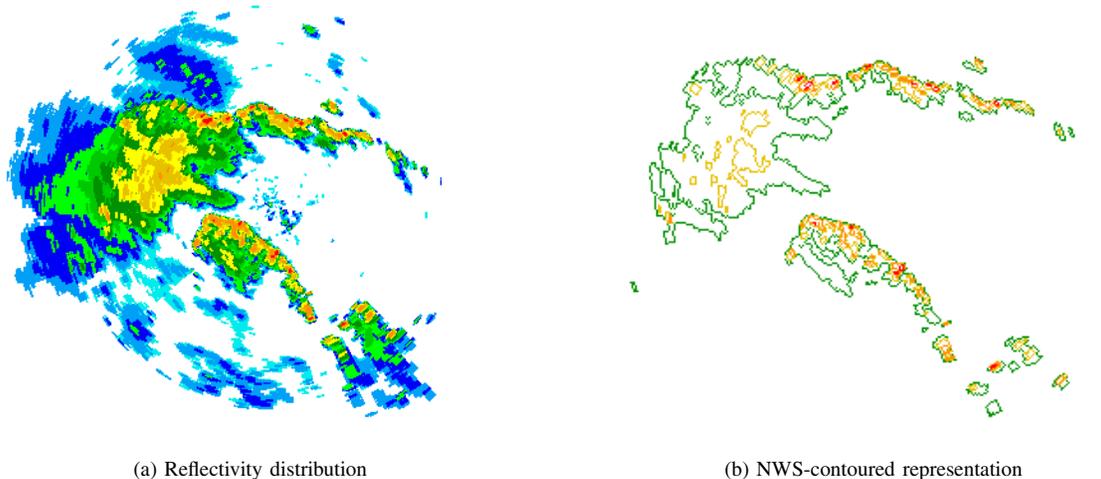


Fig. 1. Sample PPI reflectivity field from WSR-88D Doppler weather radar in Level II format (elevation 0.35° , display 512×512 pixels with 8-bit depth)

III. CONTOUR-BASED COMPRESSION OF WEATHER DATA

Transmitting the contours in place of all the pixels constituting a reflectivity field results in a fair degree of data compression. Contoured reflectivity data obtained in binary form may be transmitted using general-purpose encoding schemes such as *run-length encoding*. However, the degree of compression achieved in such methods is not very high. It would be more efficient to transmit the contours alone, which consist of far fewer points than any 2-dimensional description. Further reduction can be achieved by encoding the contours themselves. This may be achieved in several ways.

A. Chain Coding

Chain coding [3] is an established method of compressing data corresponding to linear features. The method represents each point on the contour as a designated neighbor of its previous point, and encodes the incremental position change between successive pixels on the contour. This is a lossless encoding scheme. However, since the reconstruction is performed by adding the incremental changes to a starting point on the contour, any error introduced into the reconstruction process (e.g. due to communication channel errors) cannot be recovered, and the remaining part of the contour would be represented erroneously. Thus this method is not recommended under conditions of low signal-to-noise ratios or where possibilities of channel malfunction exists.

B. Polygonal Encoding

Small segments of a contour may be represented by straight-line segments, providing a polygonal approximation to the contours. Such a method had been proposed by Burdon [4] specifically for encoding weather radar reflectivity contours. The method has been found to provide fairly high degree

compression, but may result in either inaccurate representation or reduced compression ratios in the presence of strong randomness in the contours which are typical of weather data.

C. Polygonal-Ellipse Encoding

In this method [5] local approximations to contour segments may be made by elliptical arcs in addition to straight segments. This offers greater flexibility in describing gently curved portions of weather contours, but still experiences difficulties when the contours are highly jagged. The quantitative behavior of this as well as the Polygonal Encoding method are included in the results cited later in this paper.

D. Control-Point Based Encoding and Spline Reconstruction

The authors had proposed a compression scheme that utilizes the same contour representation of the weather data field as often used for display of the data [6]. The main steps in the compression, data transmission and contour retrieval are outlined in the subsections below.

1) *Thresholding and Contour Tracing*: Basic reflectivity data fields are generally available in the form of scan lines corresponding to multiple elevations. A plot of the reflectivity values as function of range and azimuth of radar scan would result in a 2-D reflectivity image similar to the one shown in Fig. 1(a). The thresholding algorithm converts the data field into a binary image by comparing each data point with an assigned reflectivity value (e.g. one of the NWS levels). A contour tracing algorithm based on [7] and [8] is then applied which traces the locus of the points of transition of the binary image. Contours enclosing unit values in the binary image are called *region contours*. Closed region contours may enclose areas of zero value, whose boundary would be called *hole contours*. Region and hole contours may be nested to

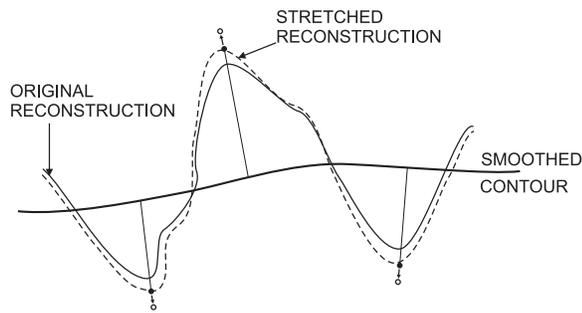


Fig. 2. Schematic showing principle of control point stretch

any order. Our contour tracing algorithm traces both types of contour of all orders and labels them as such.

2) *Extraction of Control Points*: The core of our contour compression algorithm is to express each contour in terms of a discrete set of control points. There is no unique way of defining these points (e.g. [5], [4]). Our method searches for peaks of local undulations in the contour. To achieve this, the contour is running-averaged to obtain a smoothed contour which acts as a reference line for determining points of maximum departure. Based on experience we have adopted an averaging length of 10% of the contour length in the case of the larger contours. Points of maximum departure from the reference line (smoothed contour) lying between successive points of intersection of the original and smoothed contours are declared as control points. In cases where the crossing points are far apart, additional control points are introduced midway between the extremum-based control point and the crossover points on either side (see Section IV). Control points for small contours are obtained by uniformly dividing the contours.

3) *Control Point Encoding and Transmission*: The 2-D position coordinates of these control points are required to be transmitted over the communication channel. To minimize the data requirements for transmission, the absolute values of the coordinates are not transmitted. Instead, the bounding rectangle of each contour is determined, and the coordinates of the control points for that contour are referenced relative to the corner of the rectangle. To derive absolute coordinates at the receiving end, the coordinates of two opposite corner points of each bounding rectangle need to be transmitted in the data stream. The stream also includes the threshold value, tag bits indicating the nature of the contour (region or hole), and the number of control points on the contour.

4) *Retrieval and Display of Contours*: Received data files are decoded for the relative coordinates of the control points, which are converted into absolute coordinates by adding those of the reference corners of the respective bounding rectangles. The contours are reconstructed by a spline interpolation of the control points [9]. A second-order spline fit has been found to be optimum from the point of view of accuracy of contour reconstruction. The root-mean-square (RMS) departure of the reconstructed contour from the original is used as the figure of merit for the fidelity of reconstruction.

The retrieved contours are filled with colors representing

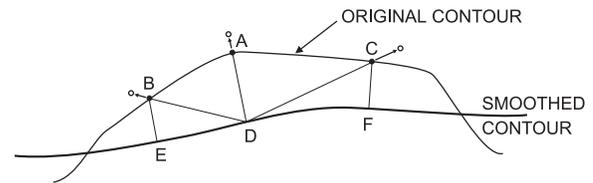


Fig. 3. Schematic showing the geometry of stretching the additional control points

the respective threshold values using a boundary fill algorithm [7]. The retrieved tag bit identifying region and hole contours is utilized here to fill only the space between region contours and any hole contours it may enclose. In case of multi-level contouring, each non-overlapping region is filled only once to minimize computational needs at the receiving end. In fact the entire processing at the retrieving end is performed in a lean and computationally efficient way keeping in mind the limited processing capability available on board many aircraft.

5) *Scope for Refinement*: 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. Two methods of performance improvement are discussed in the following sections.

IV. 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. The higher the order of the curve, 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. This drawback may be partially overcome by stretching the control points artificially about the smoothed contour prior to transmission. Figure 2 schematically depicts the principle involved. The solid dots in Fig. 2 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 control points.

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. In such cases additional control points are introduced half-way between the crossings and the extremal control points as shown in Fig. 3. These additional control points are subjected to a different stretching scheme than the 'regular' or extremal control points. In Fig. 3, for example, the additional points B and C are moved

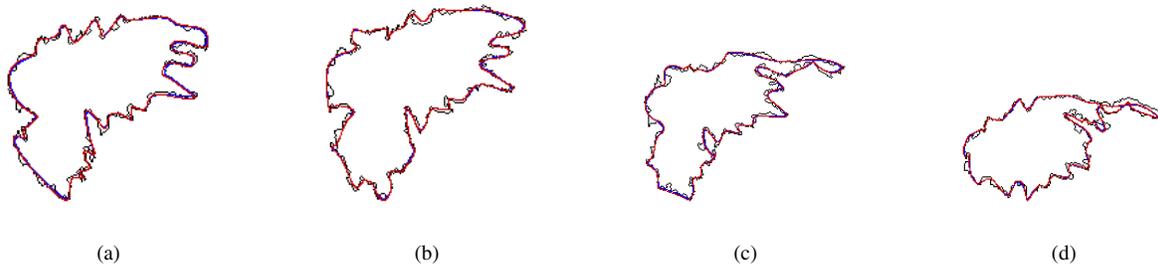


Fig. 4. Reconstructed contours for original and stretched control points (blue and red respectively) compared with the original contour (black) at four stages of evolution

outward in the directions DB and DC respectively, unlike the ‘regular’ point A which is moved out along the direction DA normal to the reference line (smoothed contour). This strategy minimizes computation and memory needs while not producing any significant performance difference with respect to stretching along normal directions EB and FC.

Table 1.

RMS CONTOUR RETRIEVAL ERROR AS FUNCTION OF PERCENTAGE OF STRETCH

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 1 shows the variation of the overall quality of retrieval of a contoured image (as given by the RMS departure) 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 control points. From Table 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 4 shows a retrieved contour depicting the improvement in the fit due to stretching. If a uniform stretch factor is to

Table 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

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.

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 to the next level. This fact is corroborated from the results in Table 2.

It is instructive to compare the quality of contour reconstruction obtained by our method (incorporating 10% stretch) with that of the earlier methods used for reflectivity contour encoding. In particular we refer to the methods by Burdon [4] and Gertz and Grappel [5]. Table 3 shows the RMS

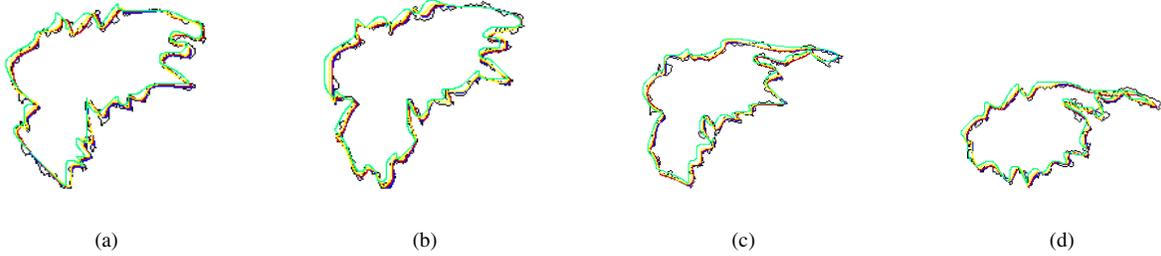


Fig. 5. Reconstructed contours for 0-bit (blue), 1-bit (red), 2-bit (yellow), and 3-bit (green) truncation compared with the original contour (black) at four stages of evolution

error of reconstruction of the contours shown in 4 by using the three methods. Each of the four contours in Table 3 is compressed to a common extent in the three methods. This is achieved by using the same number of control points to represent each given contour (indicated in parentheses below the contour number) for all the three methods.

Table 3. COMPARISON OF RMS CONTOUR RETRIEVAL ERRORS

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

V. BIT TRUNCATION FOR ENHANCING COMPRESSION

Table 4.

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

Table 5.

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

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

Table 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

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.

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 we employ an efficient interpolating algorithm at the retrieving end, we expect a certain level of truncation error 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 we have calculated the overall RMS contour reconstruction error for 1, 2 and 3-bit truncation as shown in Table 4, and the corresponding compression ratios are tabulated in Table 6. In performing the truncation it is ensured that at least two of the most significant bits are left 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 4, 6 and 5.

The results in Table 4 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

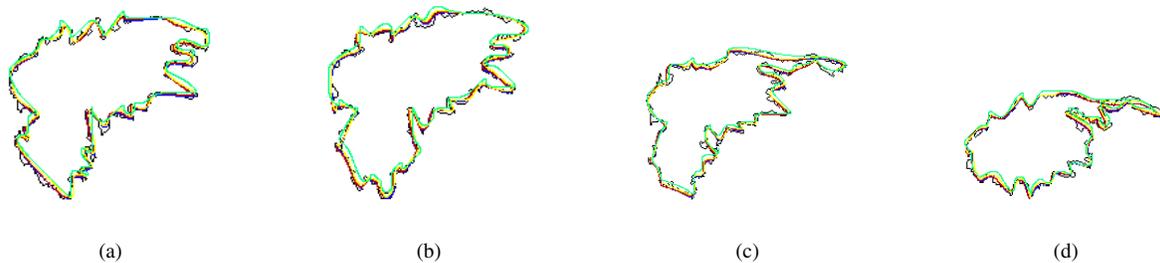


Fig. 6. Reconstructed contours for 0-bit (blue), 1-bit (red), 2-bit (yellow), and 3-bit (green) truncation after bias removal compared with the original contour (black) at four stages of evolution

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 4 and 6 show that there is a direct correspondence between the degradation of RMS reconstruction error and the increase in compression ratio. This gives the user the choice of a 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. 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 since the truncation is performed by simply dropping the least significant bits, i.e. by always rounding off the control point coordinates downwards. To compensate for this bias we have added the mean of the maximum possible quantization error to the truncated bit stream. Thus, for 1-, 2- and 3-bit truncation the bias compensation values are 0, 2 and 4. Since this compensation is carried out at the receiving end there is no need for extra bits to designate the control points during transmission. Thus the unbiasing procedure leaves the compression ratio unchanged.

Table 5 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. These are found to be much lower than the errors caused by straightforward downward truncation. Thus the levels of compression suggested by Table 6 are now achievable with only 1, 1.5 and 2 pixels of error for 1-, 2- and 3-bit truncation. The corresponding improvement in the quality of contour retrieval is depicted in Fig. 6.

VI. CONCLUSION

Contour-based compression schemes have been shown to yield very high levels of compression for high-volume weather

radar data with negligible loss of information in order to facilitate their transmission over low-capacity links. However, such basic compression schemes permit refinements to optimize their parameters and performance. Two specific directions of improvement are explored in this paper. A novel method of compensating the position of control points for the smoothing effects of the spline interpolation is found to yield up to 8% improvement in the fidelity of contour retrieval. It has also been shown that manipulating the bit representation of the control point coordinates can yield up to 25% or even 35% additional compression if small errors of one or a few pixels in contour representation can be tolerated. These errors can be even further reduced significantly by unbiasing the truncation process.

REFERENCES

- [1] 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.
- [2] R. J. Doviak and D. S. Zrnicek, *Doppler Radar and Weather Observations*, 2nd ed. San Diego, CA: Academic Press, 1993.
- [3] H. Freeman, "On encoding arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. 10, pp. 260 – 268, 1961.
- [4] 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.
- [5] 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.
- [6] 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 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.
- [7] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Berlin, Germany: Springer Verlag, 1982, ch. 7.5.
- [8] M. Alder, *An Introduction to Pattern Recognition*. Osborne Park, Western Australia: HeavenforBooks.com, 1997, ch. 2.
- [9] C. D. Boor, *Practical Guide to Splines*. New York, NY: Springer Verlag, 1978.