15A.4 Lossless Differential Compression of Weather Radar Data in Universal Format Using Motion Estimation and Compensation

W. David Pan*
Dept. of Electrical & Computer Engineering, University of Alabama in Huntsville, Alabama

Paul R. Harasti
Visiting Scientist Programs, University Corporation for Atmospheric Research, Boulder, Colorado

Michael Frost, Qingyun Zhao and John Cook
Marine Meteorology Division, Naval Research Laboratory, Monterey, California

Timothy Maese
Basic Commerce and Industries, Moorestown, New Jersey

Lee J. Wagner
Atmospheric Propagation Branch, SPAWAR Systems Center, San Diego, California

1. INTRODUCTION

A meteorological radar data assimilation system has been developed at the Marine Meteorology Division of the Naval Research Laboratory (NRL) to provide environmental information to enhance the safety of ship and aircraft operations. Radar observations are assimilated into the Navy's Coupled Ocean/Atmosphere Mesoscale Prediction System (COAMPS®) to improve the forecasts of hazardous weather (Zhao et al. 2006; 2008) and to provide decision makers with timely products to help exploit or mitigate those predictions. The system takes advantage of future Navy vessels having weather processors for their tactical radars (e.g., SPS-48E/G: Hazardous Weather Detection and Display Capability (HWDDC); SPY-1 Tactical Environmental Processor). The ships in the battle fleet having this capability will be able to digitally generate full-resolution, full-volume weather radar data, and archive those data in Universal Format (UF – Barnes (1980)) files approximately every 5 minutes. Several UF files will be transmitted in near-real-time per hour to Fleet Numerical Meteorology and Oceanography Center (FNMOC) where the data assimilation into COAMPS® is conducted.

UF file sizes range from ~5 MB (SPS-48E) to ~13 MB (SPY-1), which would be too large a load on the operational bandwidth of the ships' communication systems. To overcome this obstacle, we have developed a novel UF file compressor that typically reduces UF file sizes by a factor of forty, thus permitting their transmission from a ship to FNMOC (Pan et al. 2009). NRL has delivered the UF file compressor to Basic Commerce and Industries, Moorestown, NJ, who is the developer of the HWDDC under contract for Space and Naval Warfare Systems Command (SPAWAR) Systems Center, San Diego, CA. From May to August 2009, BCI and SPAWAR conducted tests using the land-based SPS-48E at Navy facilities in Dam Neck, VA on version 2 of the HWDDC, which will soon be installed on ~10 Navy ships. One of these tests was the successful transfer of compressed UF files from the HWDDC to NRL in real-time (Harasti et al. 2009).

The current UF file compressor achieved large compression by exploiting only intra-UF file correlations. Since UF files were taken approximately 5 minutes apart, temporal correlations are expected to exist between radar data in neighboring UF files. In this work, we investigate ways in which these inter-UF file correlations could be exploited to achieve compression.

We analyzed archived SPS-48E UF data obtained from an at-sea experiment onboard the USS PELELIU (LHA5) in February 2006 (Harasti et al. 2006; Maese et al. 2007). This data set contains a wide range of precipitation echoes spanning 22 hours of observations. The UF files contain full-resolution (915 m in range; 1° in azimuth) reflectivity factor (DZ), raw radial velocity (VE), de-aliased radial velocity (VD), spectrum width (SW), signal to noise ratio (SN) and valid radial velocity indicator (VV) data. These data are available within PPI scans at 22 different elevation angles ranging from 0.2° to 24° out to 275 km range, except for VE, VD, SW and VV, which are only available from the first three elevation tilts (up to 1.6° elevation) out to 52 km range.

As the first step, we consider the problem of lossless compression of the original raw data of DZ, SW and VE in the UF dataset. This paper presents a case study of the dataset to demonstrate that inter-UF file coding, based on motion estimation and compensation, can provide more efficient lossless compression than intra-only UF file compression.

2. TEMPORAL CORRELATION EXPLOITATION

Efficient compression of UF files could be achieved by exploiting the temporal correlations existing between radar data in adjacent UF files. For example, Fig. 1 shows the similarly looking PPI images of the DZ, SW, and VE data taken from three UF files that were created

*Corresponding author address: Dr. W. David Pan, Dept. of Electrical and Computer Engineering, University of Alabama in Huntsville, Huntsville, AL 35899. Email: dwpan@eng.uah.edu.

COAMPS® is a registered trademark of the Naval Research Laboratory.

approximately 5 minutes apart on 22 February 2006. In principle, given the data in a previously obtained UF file as the reference, the corresponding data in the current UF file can be reconstructed by adding to the reference data update information that describes the differences between the current data and the reference data. Depending on the way the update is obtained, coding the update information might be more efficient than coding the data in the current file directly. We call the former approach inter-file (or inter-volume) coding method, and the latter intra-file coding method.

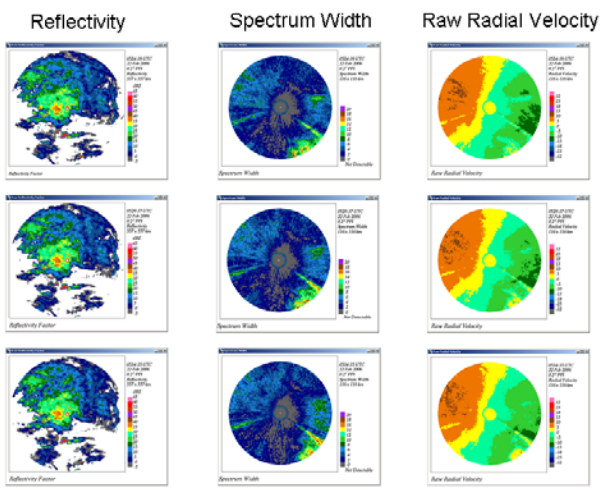

Fig. 1. PPI images of the DZ, SW and VE data in three successive UF files taken approximately 5 minutes apart, at 05:24:39 UTC (top row), 05:29:37 UTC (middle row), and 05:34:35 UTC (bottom row), respectively.

In order to assess the effectiveness of the inter-file coding approach, we considered two methods of extracting the update information between data from two temporally adjacent UF files. (i) Simple subtraction of the reference data from the current data; (ii) A more sophisticated method known as *delta compressor*, which is concerned with efficient file transfer over a bandwidth-constrained link in the case where the receiver already has a similar file. Specifically, we experimented with a version of delta compressor called *zdelta**, which has been used to increase the efficiency of distributing updated versions of software over a network, or synchronizing personal files between different accounts and devices.

Unfortunately, neither the simple subtraction method nor the delta compressor could provide more efficient compression than intra-only compression, which is based on *bzip2*. As shown in Fig. 2, using *bzip2* to compress the differences between DZ data from adjacent UF files (approximately 5 minutes apart) led to less compression achieved by directly applying *bzip2* on the DZ data from each individual UF file. Employing more sophisticated differential compressing method such as the delta compressor could not offer much help – as can be seen in Fig. 3, more bits are required to code the updates between neighboring UF files using *zdelta*, than coding each individual UF file using *bzip2*.

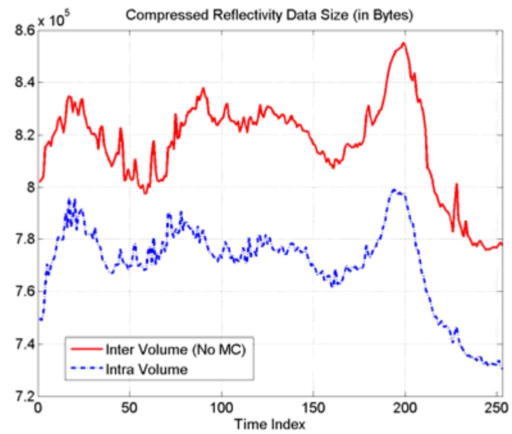

Fig. 2. Compressed DZ data sizes. Inter-UF file compression based on simple subtraction results in 6% on average lower compression than intra-compression based on *bzip2*.

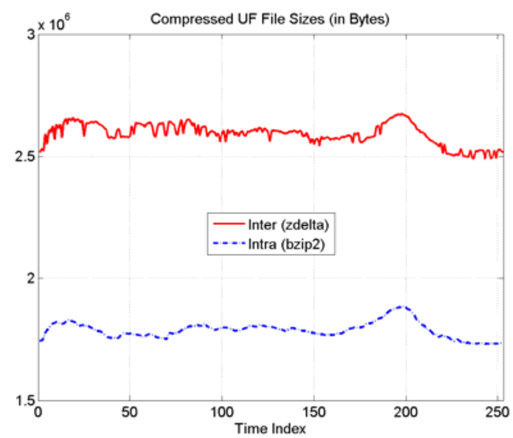

Fig. 3. Compression of the compressed UF file sizes between intra-coding based on *bzip2*, and inter-coding based on *zdelta*.

The inability of these two inter-file coding methods to capture the temporal correlations could be attributed to the change of the position of the ship where the radar data was collected. The ship movement can be calculated by using the ship position data (latitude, longitude, etc.) found in the headers of UF files. However, there was a hardware problem during the NPEL SPS-48E/HWDDC weather observations on 22 February 2006, which caused invalid ship position data to be stored in the UF files. See Harasti et al. (2009) for a discussion of the method developed to estimate the position of the ship over the 22-hour period on this day, and the resulting ship position measurement uncertainty. Alternatively, we can estimate the ship movement indirectly by establishing the correspondence between radar data collected at different times. Since the end goal of this work is to achieve efficient compression on radar data, we considered motion estimation techniques that have been employed by many video data compression standards such as MPEG-4 and H.264 (Richardson 2003). Motion

* http://cis.poly.edu/zdelta/

estimation without relying on ship position data allows us to directly capture the temporal correlations between radar data from successive UF files, and to compensate for the ship movement and other motion of weather prior to applying differential coding codes.

3. MOTION ESTIMATION AND COMPENSATION

The objective of motion estimation is to establish the correspondence between similar radar data collected at different time points. To this end, we employed a block-based approach. In this approach, the polar coordinate system used to display the circular PPI images of data at a certain radar scan elevation (such as those shown in Fig.1) is first converted to a Cartesian coordinate system, where data at the same elevation are organized as a rectangular image (such as those shown in Fig. 5), with horizontal axis representing azimuth (with the unit of degree) and the vertical axis representing the range (measured in number of bins, with each bin spanning approximately 0.91 *km*). Each rectangular image will be partitioned into equally sized square blocks of $bs \times bs$ pixels. For each block $C(i,j)$, where $i, j \in [0, bs-1]$, in the PPI image from the current UF file, we search though all the candidate blocks $R(i+dx, j+dy)$ within a search window in the reference PPI image, so as to find a best match to $C(i,j)$ (Fig. 4). In this work, the best-matched block is defined as the one that would minimize the following sum of squared differences (SSD):

$$SSD = \sum_{i=0}^{bs-1}\sum_{j=0}^{bs-1}\left[C(i,j) - R(i+dx, j+dy)\right]^2 \qquad (1)$$

The displacement of the best-matched block relative to the current block is represented by the motion vector (dx, dy) of the match block.

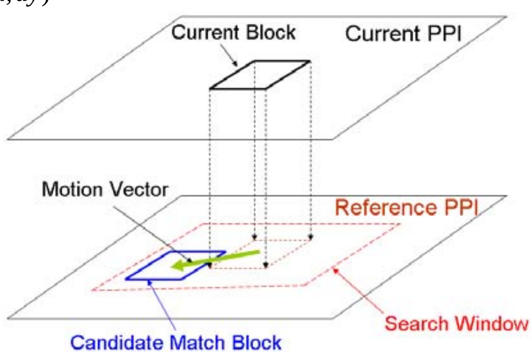

Fig. 4 Block-based motion estimation. Note here that we adopted the jargon in video coding, where motion vectors point to the past as far as motion compensation is concerned. In reality, the reflectivity motion is from past to present.

In general, the choice of the block size and the size of the search window controls the tradeoffs between the granularity of the objects (which depend on the type of weather and resolution of the radar) whose motion are to be estimated, the maximal magnitude of the motion vectors, and the computational complexity of the algorithm. After extensive experiments with our dataset, we chose the block size to be 10×10 pixels (equivalent to $9.15\,km \times 10^{\circ}$), and search window size to be ± 15 pixels (in range and azimuth units about the position of the current block). Fig. 6 shows the motion vectors of the best-matched blocks of the DZ data blocks at the first elevation of the data volume taken at 05:29:37 UTC (which is shown at the bottom of Fig. 5).

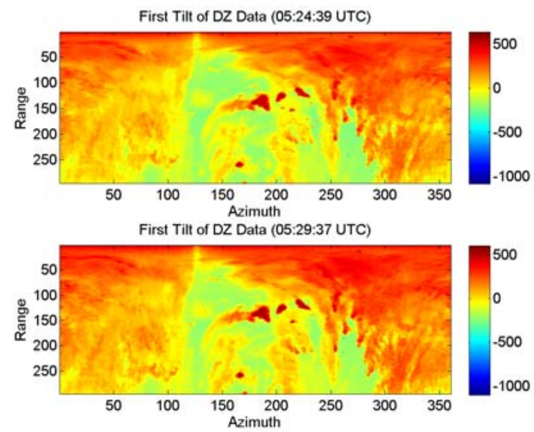

Fig. 5 First tilt of the DZ data from two adjacent UF files. The unit for the range is number of bins, and the unit for azimuth is in degrees.

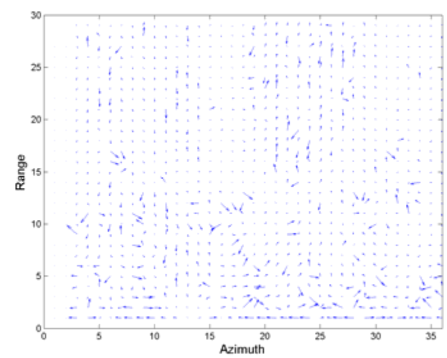

Fig. 6 Motion vector field obtained after applying block-based motion estimation on the two images of DZ data in Fig. 5 (using the data at 05:24:39 UTC as the reference). The unit for the range is in number of 10 bins, and the unit for azimuth is in 10 degrees, since the block size was chosen to be 10×10 pixels.

As a succeeding step to motion estimation, motion compensation refers to an attempt to predict the current block based on its best match block in the reference image. Given the reference image, we can retrieve the best-matched block of the current block by using the motion vector obtained by motion estimation. In order to perfectly reconstruct the current block, we just need to keep track of the difference between the current block and its best matched block, in addition to the motion vector. Such a difference represents the error (known also as the residue) of a motion-compensated prediction of the current block. In general, coding the prediction residue after motion-compensation would be more efficient than coding the residue without first compensating for the motion. As an example, Fig. 7 shows a visual comparison between absolute values of the residues with and without motion compensation. It can be seen that motion compensation could significantly reduce the magnitude of the residue associated with predicting blocks in the current image based merely on the co-located blocks in the reference image. The advantage of motion compensation can also be confirmed by an objective measure of the prediction error shown in the caption of Fig. 7.

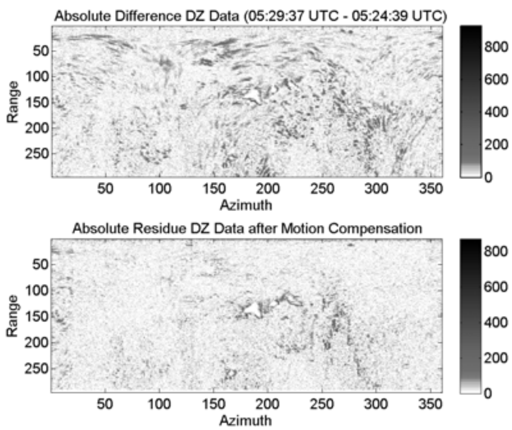

Fig. 7 Absolute values of the residues of predicting the DZ data at 05:29:37 UTC based on data at 05:24:39 UTC as a reference. The unit for the range is number of bins, and the unit for azimuth is in degrees. Top: the average difference in terms of RMSE (root mean squared error) between the data and their prediction (with simple difference without motion compensation) is 43.5; Bottom: RMSE = 32.2 (with motion compensation).

A block diagram for a proposed data encoding and decoding schemes based on motion estimation (ME) and compensation (MC) is given in Fig. 8, where the information describing the change of the data in the current UF file from the data in an earlier UF file consists of the residue data and the motion vectors. For the sake of lossless compression, lossless coding methods should be used. We employed the *Rice-Golomb* codes (Golomb 1966; Rice 1971), which have been widely used for coding prediction residues, and *bzip2* for coding motion vectors.

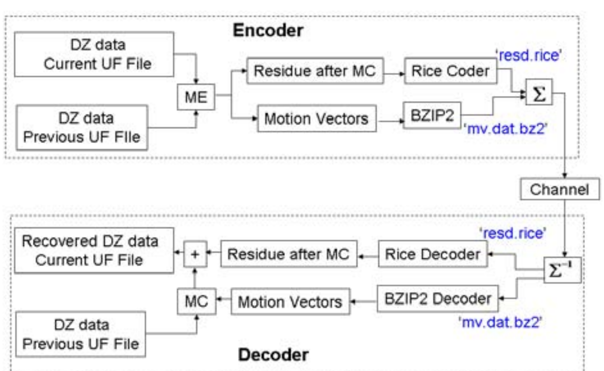

Fig. 8. Architecture of the differential lossless data encoder and decoder, given that the data in a previous UF file are available at both the encoder and decoder. ME and MC denote motion estimation and motion compensation, respectively.

The above ME/MC method can be applied to DZ data at all elevations. To compress the DZ data using the inter-coding mode, ME/MC references the DZ data at a co-located elevation from the UF file taken approximately 5 minutes earlier. Simulation results based on the UF file in the data set are summarized in Fig. 9, which shows that the proposed ME/MC-based method outperforms the intra-coding method based on *bzip2*. By comparing the results in Fig. 2 and Fig. 9, we can conclude that the ME/MC method could capture temporal correlations more effectively than the simple subtraction and delta compression methods discussed in Section 2.

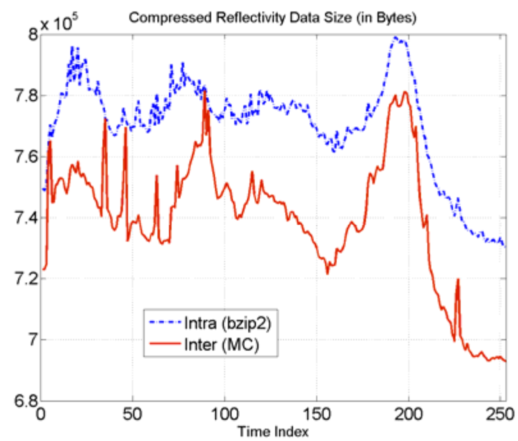

Fig. 9. Compressed DZ data sizes. On average, inter-file data coding based on motion estimation and compensation offers about 4% more reduction in the compressed data size, than intra coding based on *bzip2*.

4. INTER-FILE HEADER COMPRESSION

In UF files, parameters such as the radar name, location, volume date, sweep mode, and so on are archived as part of the record headers. Since many of these parameters tend not to change from record to record, in our prior work (Pan et al. 2009), the DPCM (Differential Pulse Code Modulation) method was employed to achieve very high compression on UF file headers, by taking advantage of the substantial amount of redundancy existing between neighboring records within an UF file. While this DPCM-based method is an intra-file compression method, in this work, we found that even higher compression could be attained by an inter-file compression method, which exploits the redundancy between the headers of two neighboring UF files. More specifically, the macro-header H_t^i (see Pan et al. (2009) for its construction) of record i in the current UF file at time index t will be aligned with the macro-header H_{t-1}^i of the co-located record in the reference UF file at time index $(t-1)$. Then the differential macro-block (D_t^i) between these two macro-headers can be found by applying the bitwise XOR (Exclusive OR) operation as follows:

$$D_t^i = H_t^i \oplus H_{t-1}^i \quad (2)$$

Hence if these two macro-headers differ in just a very small number of words, the difference D_t^i will be a very sparse vector of words with most of words being zero, thereby making D_t^i much easier to compress than the macro-header H_t^i itself.

Given a perfectly reconstructed differential macro-block D_t^i and the reference macro-block H_{t-1}^i, the current macro-block H_t^i can be faithfully recovered by

$$D_t^i \oplus H_{t-1}^i = H_t^i \oplus H_{t-1}^i \oplus H_{t-1}^i = H_t^i \quad (3)$$

The differential macro-blocks D_t^i obtained for all records in the current UF file are then concatenated into a single data file, on which the bzip2 is applied for lossless compression. Fig. 10 shows that this method of inter-file header compression could provide an average 39% more compression than the intra-file header compression method proposed in Pan et al. (2009). This result indicates that the inter-record header content variations within an UF file appear to be larger than those between two neighboring UF files.

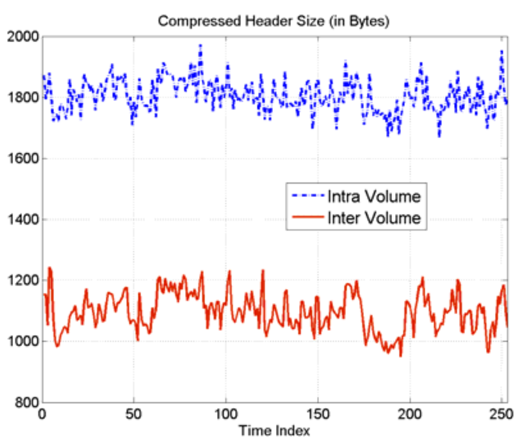

Fig.10. Comparison of the compressed header sizes between two header compression methods. On average, the headers in each UF file can be compressed to 1,095 bytes by the inter-file compression method, versus 1,802 bytes by the intra-file method.

5. RESULT OF DATA AND HEADER COMPRESSION

We applied the lossless compression method based on motion estimation and compensation discussed in Section 3 to the original raw data of reflectivity factor (DZ), raw radial velocity (VE), and spectrum width (SW) in the UF dataset. Each UF file was compressed differentially using the previous UF file in the dataset as its reference. The headers in each UF file were also compressed losslessly using the inter-file method described in Section 4. The overall sizes of the compressed data and headers of each UF file in the dataset is shown in Fig. 11. For comparison, the original raw data (DZ, VE, and SW) in each UF file were compressed individually using bzip2. Furthermore, the headers in each UF file were compressed using the intra-file DPCM method proposed in Pan et al. (2009). As shown in Fig. 11, the inter-file compression method outperforms the intra-file compression method consistently for all the UF files in the dataset, thereby demonstrating the effectiveness of the proposed method based on motion estimation and compensation in capturing temporal correlations in weather radar data.

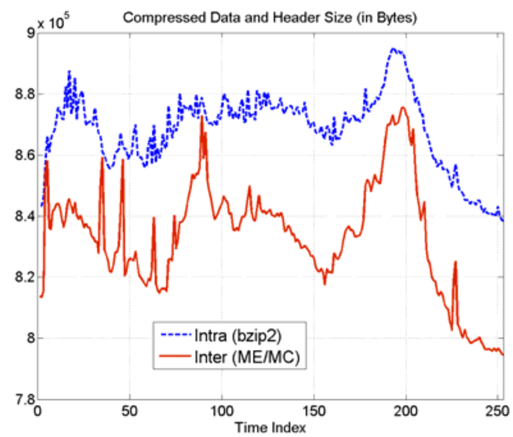

Fig. 11. Comparison of the sizes of the compressed data (DZ, VE, and SW) and headers between the inter-file compression method based on ME/MC, and the intra-file compression method based on *bzip2*.

6. SUMMARY AND FUTURE WORK

Although significant reduction on the UF file sizes could be achieved through an UF file compressor we developed earlier, the compressor has not taken advantage of inter-file correlations. In this work we considered differential compression techniques, which aim at efficient UF file transfer over a bandwidth-constrained link in the case where the receiver already has a previous UF file that has been perfectly reconstructed.

Simulations showed that neither differential compression based on straightforward UF file subtractions, nor the use of Delta Compressor *zdelta* would lead to more efficient compression than the existing intra-UF file compression technique. We thus employed block-based motion estimation technique to capture the temporal correlations between successive UF files and then compensate the motions of the ship and the weather. Strictly lossless coding was accomplished by compressing the motion parameters and the residue signals after motion compensation using lossless coding methods.

It should be noted that the proposed inter-file differential compression method is completely lossless and more than meets the 1 MB maximum file size requirement imposed by the operational bandwidth.

However, in terms of overall file size reduction, it cannot compete with the existing intra-file data reduction and compression technique (Pan et al. 2009), which achieves approximately 40% on average more reduction in file sizes, when data un-scaling, rounding and thresholding are included. On the other hand, the inter-file differential compression method tends to have considerably larger computational complexity than the intra-file compression method, mainly due to the exhaustive full search method we employed for motion estimation.

To the best of our knowledge, this could be the first attempt to apply motion estimation and compensation methods employed by lossy video compression standards (such as MPEG and H.26x) to address the problem of lossless compression of weather radar data. Further improvement of the method might be possible by refining the techniques for motion estimation and compression of motion parameters and motion compensation residues. In addition, fast block-matching methods for lossless motion estimation (e.g., Cai and Pan (2010)) could be considered for significantly reducing the computational complexity of the inter-file compression method.

While providing a useful lossless data compression option for the current efficient file transfer problem, the proposed differential compression technique may also benefit applications in efficient data archiving, where strictly lossless data compression is often a requirement.

7. ACKNOWLEGEMENTS

The research was supported by the Office of Naval Research Summer Faculty Research Program, and NRL Grant N00173091G005.

8. REFERENCES

Barnes, S. L., 1980: Report on a meeting to establish a common Doppler radar data exchange format. *Bull. Amer. Meteor. Soc.*, **61**, 1401–1404.

Cai J., and W. D. Pan, 2010: Fast exhaustive-search motion estimation based on accelerated multilevel successive elimination algorithm with multiple passes. *Proceedings of 35th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (to appear) Dallas, TX. March 2010.

Golomb, S.W., 1966: Run-length encodings. *IEEE Transactions on Information Theory*, **12**(3): 399 – 401.

Harasti, P. R., Pan, W. D., M. Frost, Q. Zhao, J. Cook, L. J. Wagner, T. Maese, and R. Owens, 2009: On the development of a weather radar data assimilation system for the US Navy. *Proceedings of the 34th Conference on Radar Meteorology*, Williamsburg, VA 5-9 October, 2009, Amer. Meteor. Soc., 5 pp.

Harasti, P. R., J. Cook, Zhao, Q., L. J. Wagner, C. Kessinger, D. Megenhardt, J. Pinto, and B. Hendrickson, 2006: Extracting weather information from the SPS-48E for US Navy NOWCAST. *Proceedings of the Fourth European Conference on Radar in Meteorology and Hydrology,* Barcelona, Spain, 18-22 September 2006, 314-317.

Maese, T., J. Hunziker, H. Owen, M. Harven, L. Wagner, R. Wilcox, K. Koehler and CDR Gerald Cavalieri, 2007: Harzardous weather detection and display capability for US Navy Ships. *23rd Conference on Interactive Information and Processing Systems (IIPS)*, San Antonio, Texas, 14-18 January 2007, Amer. Meteor. Soc., 7pp.

Pan, W. D., P. R. Harasti, M. Frost, Q. Zhao, J. Cook, T. Maese, and L. J. Wagner, 2009: Efficient reduction and compression of weather radar data in universal format. *25th Conference on International Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, *AMS 89th Annual Meeting*, Phoenix, Arizona, 10-16 January.

Richardson, I. E. G., 2003: H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia, Wiley.

Rice, R. F. and R. Plaunt, 1971: Adaptive Variable-Length Coding for Efficient Compression of Spacecraft Television Data, *IEEE Transactions on Communications,* **16**(9), 889 -- 897.

Zhao, Q., J. Cook, Q. Xu and P. R. Harasti, 2006: Using radar wind observations to improve Mesoscale Numerical Weather Prediction, *Wea. Forecasting*, **21**, 502-522.

Zhao, Q., J. Cook, Q. Xu and P. R. Harasti, 2008: Improving Short-Term Storm Predictions by Assimilating both Radar Radial-Wind and Reflectivity Observations, *Wea. Forecasting*, **23**, 373-391.