5D-ODETLAP: A NOVEL FIVE-DIMENSIONAL COMPRESSION METHOD ON TIME-VARYING MULTIVARIABLE GEOSPATIAL DATASET

You Li, Tsz-Yam Lau, Peter Fox and W. Randolph Franklin

Computer Science Department, Computer Graphics Research Group
Rensselaer Polytechnic Institute
110, Eighth Street, Troy, NY, USA
liy13@cs.rpi.edu, laut@cs.rpi.edu, pfox@cs.rpi.edu, wrf@ecse.rpi.edu
http://www.cs.rpi.edu/research/groups/graphics/

Commission IV/4

KEY WORDS: 5D Compression, ODETLAP, 3D-SPIHT, WOA 2005, PDE Solver

ABSTRACT:

A five dimensional (5D) geospatial dataset consists of several multivariable 4D datasets, which are sequences of time-varying volumetric 3D geographical datasets. These datasets are typically very large in size and demand a great amount of resources for storage and transmission. In this paper, we present a lossy compression technique for 5D geospatial data as a whole, instead of applying 3D compression method on each 3D slice of the 5D dataset. Our lossy compression technique efficiently exploits spatial and temporal similarities between 2D data slices and 3D volumes in 4D oceanographic datasets. 5D-ODETLAP, which is an extension of, but essentially different from, the Laplacian partial differential equation, solves a sparse overdetermined system of equations to compute data at each point in (x,y,z,t,v) space from the data given at a representative set of points. 5D-ODETLAP is not restricted to certain types of datasets. For different datasets, it has the flexibility to approximate each one according to their respective data distributions by using suitable parameters. The final approximation is further compressed using Run Length Encoding. We use different datasets and metrics to test 5D-ODETLAP, and performance evaluations have shown that the proposed compression technique outperforms current 3D-SPIHT method on our selected datasets, from the World Ocean Atlas 2005. Having about the same mean percentage error, 5D-ODETLAP's compression result produces much smaller maximum error than 3D-SPIHT. A user-defined mean or maximum error can be set to obtain desired compression in the proposed method, while not in 3D-SPIHT.

1 INTRODUCTION

Current advances in data-acquiring technology in geospatial fields have greatly facilitated the research in Geology and other interdisciplinary studies. For example, the National Oceanographic Data Center (NODC) and National Geophysical Data Center (NGDC), which are operated by the National Oceanic and Atmospheric Administration (NOAA) of the U.S. Department of Commerce, serve as national repositories and dissemination facilities for global geospatial data. They provide a record of earth's changing environment, and support numerous research and operational applications. Specifically, they keep records of various geospatial data including temperature, salinity, nitrate and silicate of oceans(Garcia et al., 2006, Locarnini et al., 2006, Antonov et al., 2006). These data are often collected and stored in high dimension, usually in four dimension (4D) or five dimension (5D).

A 5D dataset consists of several 4D datasets, which keeps records of different environmental variables in the same area at a certain period of time. Then, these 4D geospatial datasets describe temporal changes of geological variables, such as temperature of ocean water, as a sequence of three-dimensional (3D) volumes. Similarly, a 3D geospatial dataset is a sequence of two dimensional (2D) datasets in the same 2D area. As the dimension goes higher, the amount of data will consequently increase, which makes storage and transmission of these data more difficult than ever before even at today's internet speed. Without effective way of utilizing these data, it's a waste of time and effort to collect and store them.

Current improvements on data storage and communication methods are minimizing the cost of storing and transmitting large amounts of geospatial data. Nevertheless, as the research goes on in this field, the decrease rate in storage and bandwidth costs will not be able to surpass the rate of growth in high dimensional geospatial data. Furthermore, the emergence of real-time rendering of 3D earth environment such, as Google Earth, makes high dimensional geospatial data compression a significant area of research. Unfortunately, not much effort in this field has been made, and work still needs to be done. Therefore, it's still a challenge to design a progressive and effective compression scheme for efficient storage and transmission of rising high dimensional geospatial data.

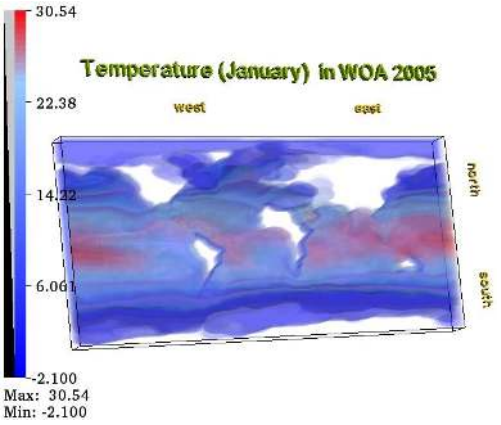

Figure 1: $180 \times 360 \times 24$ Monthly Temperature (January) Data in WOA05

For example, monthly temperature(Locarnini et al., 2006) in World Ocean Atlas 2005 is of size $180 \times 360 \times 24 \times 12$, which was compressed with gzip. Figure 1 visualizes January's data derived

from this monthly temperature data.

In this paper, we use a 5D Over-determined Laplace Partial Differential Equation (5D-ODETLAP) to progressively compress 5D marine data. Firstly, for a sequence of 4D data in 5D space, we construct over-determined systems using a specially designed point selection method. Then we solve these systems with an over-determined PDE for a smooth 5D approximation. This approximation is likely to have large errors due to a limited number of selected points. But we can improve this approximation by adding points which have the largest error with respect to the original 5D marine data. After this, we run 5D-ODETLAP again on the augmented representation for each 4D data to obtain a better 5D approximation. These two steps are run alternatively until we reach a stopping criteria, which is often a user-specified maximum error.

2 PRIOR ART

Current compression methods have two categories: lossy compression and lossless compression. While lossless techniques allow exact reconstruction of the original data, they usually can't achieve high compression ratios. Some data, like 3D medical images, are always stored in lossless format because of a possible false diagnostic and its legal implications. But for geospatial data, we can still conduct data analysis and obtain satisfying information from it as long as we keep the compression error relatively small.

Various compression schemes have been proposed for 2D, 3D and 4D gridded data. Most of those schemes are focused on 2D images and 3D image sequence data, especially in multidimensional medical images (Menegaz and Thiran, 2002, Kim and Pearlman, 1999). These 3D compression methods either compress image slices independently (ignoring the correlation in the third dimension), or compress the whole 3D volume using 3D wavelets transform, such as 3D-SPIHT (Kim and Pearlman, 1997), with which we will compare our method later in this paper. In the former case, JPEG2000 (Skodras et al., 2001) and JPEG-LS (Weinberger et al., 2000) are the most popular ones.

For 4D data, including videos and time-varying 3D geospatial volumes, various methods have been proposed to compress them, including 4D wavelets (Yang et al., 2006), run length encoding (RLE) (Anagnostou et al., 2000) and discrete cosine transform (DCT) (Lum et al., 2001). Similar to 3D compression, a 4D dataset can also be treated as a sequence of 3D volumes; thus those 3D compression methods can be applied. But there are methods which exploit the temporal redundancy between volumes and usually outperform their 3D counterparts. These include video compression methods using motion compensation technique (Sanchez et al., n.d.) and 4D-SPIHT, a wavelet based method. These methods using different schemes to compress the temporal dimension, utilize the data correlation between volumes; thus they have a higher compression ratio.

Unfortunately, there aren't many prior works done in compressing 5D dataset mainly because the difficulty in compression increases significantly from 4D to 5D, and 4D compression methods can also be applied on 5D data individually.

3 5D-ODETLAP

3.1 Definition

5D-ODETLAP, or Five Dimensional Over-Determined Laplacian Partial Differential Equation, is an extension of the Laplacian PDE $\frac{\delta^2 z}{\delta x^2} + \frac{\delta^2 z}{\delta y^2} = 0$ to an overdetermined linear system (Stookey et al., 2008, Xie et al., 2007). In this overdetermined linear system, every point, known or unknown, has an equation setting its value as the average of its 4, 5, 6, 7 or 8 neighbors in four dimensional space. The equation is:

$$\begin{aligned} u_{i,j,k,t} = (u_{i-1,j,k,t} + u_{i+1,j,k,t} + u_{i,j-1,k,t} \\ +u_{i,j+1,k,t} + u_{i,j,k-1,t} + u_{i,j,k+1,t} \\ +u_{i,j,k,t-1} + u_{i,j,k,t+1})/8 \end{aligned} \quad (1)$$

for every point, which means the 4D volume satisfies 4D Laplacian PDE,

$$\frac{\delta^2 u}{\delta x^2} + \frac{\delta^2 u}{\delta y^2} + \frac{\delta^2 u}{\delta z^2} + \frac{\delta^2 u}{\delta t^2} = 0 \quad (2)$$

Unfortunately, this simple 4D Laplacian PDE will only have one solution, which is probably not the optimal one for different data distribution. On the other hand, the solution of laplace equation doesn't have a relative maximum or minimum in the interior of the solution domain, which is defined as the maximum principle(Sewell, 2005).

In order to generate local maximum/minimum values, we first apply the Equation 1 to every non-border point and then add one equation for each known point in a set S:

$$u_{i,j,k,t} = h_{i,j,k,t} \quad (3)$$

where $h_{i,j,k,t}$ stands for the known value of points in S and $u_{i,j,k,t}$ is the "computed" value as in Equation 1.

So now we have more equations than points in the data, and this means the linear system is over-determined. A least-square solution to this system will be computed, and since it may not be consistent, we obtain an approximate solution instead of an exact solution (which is impossible) by keeping the error as small as possible. Equation 1 sets the value at each point to be the average of its neighbors, which makes the data distribution in 4D more "smooth" and continuous. In the meantime, Equation 3 keeps the value at each known point equal to its known value. So for every known point, we can choose the relative importance of data continuity versus accuracy through a continuity parameter R added to Equation 3 when solving this over-determined system(Franklin, 2000).

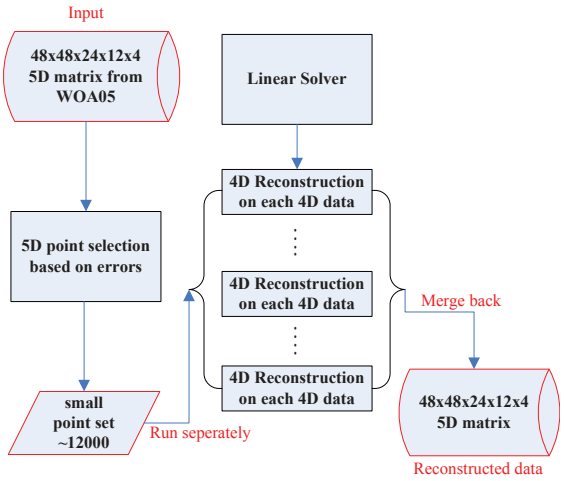

Figure 2: 5D-ODETLAP Algorithm Outline

In our algorithm, factor R is used to weight Equation 1 relative to equation 3 for all known points. If R is small, the system will approximate a determined solution and data accuracy will be main-

tained. On the other hand, if R is very large, the system will produce a data distribution with no divergence, completely ignoring all the known values. The reconstructed 4D volume doesn't necessarily have the same value on those known points, but since we already know those values, we can always keep them and replace the reconstructed values in those known points' positions. So now we have an approximation with exact values on those known points and reconstructed values for all unknown points. Since a 5D dataset essentially consists of a sequence of 4D volumes, we apply the above approximation on each 4D volume dataset and then refine this approximation in 5D space to minimize the compression error.

3.2 Algorithm Outline

Input: $5D - GeospatialData : V$
Output: $PointSet : S$
$S = InitialSelection(V)$;
foreach *4D volumes s in S* **do**
  $vReconstructed = 4DReconstruction(s)$;
  add $vReconstructed$ in set $VReconstructed$;
end
while $MeanError > Max\_MeanError$ **do**
  $S = S \cup Refine(V, VReconstructed)$;
  foreach *4D volumes s in S* **do**
    $vReconstructed = 4DReconstruction(s)$;
    add $vReconstructed$ in set $VReconstructed$;
  end
end
return S

Algorithm 1: 5D-ODETLAP algorithm pseudo code

An algorithm outline of 5D-ODETLAP is given in Figure 2 along with the pseudo code in Algorithm 1. First, initial point selection is conducted to produce point set S. We use random selection in practice. After that, an initial approximation of 5D data is computed using Equations 1 and 3.

This approximation is done by individually applying 4D reconstruction on each 4D dataset within the 5D dataset and then merge them back together to form a 5D approximation. Then we require users to set a stopping condition based on an error metric. In practice, we use the average mean percentage error of all 4D datasets within the 5D dataset as the stopping condition, because each 4D dataset has different data range and represents completely different practical meaning in GIS fields. If this condition is not satisfied, we calculate all the percentage errors on all reconstructed points in the 5D dataset.

We then select k (k≥1) points with the largest percent error with a restriction called "Four Dimension Forbidden Zone" to optimize our selection. These selected points will be added into the existing point set S, and this extended set is used again to compute a more refined 5D approximation by applying 4D reconstruction on each 4D dataset and putting them back together. Again, this 5D approximation will be evaluated by the defined stopping criteria to see if condition is satisfied. These two steps run alternatively as the algorithm proceeds. A better approximation is obtained as the total size of point set S increases and the total error converges.

After we have a satisfying point set S, further compression is done by using Run Length Encoding to compress the 5D coordinates *(x,y,z,t,v)*, where v represents the variable of that 4D dataset. For details, please refer to the paper (Li et al., in-press).

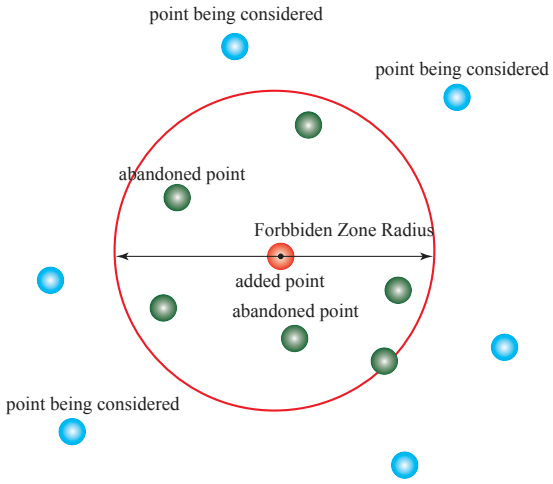

Figure 3: 4D Forbidden Zone check. The red sphere represents the forbidden zone of the red point in the center. The green point are abandoned. The blue ones are being considered.

3.3 Four Dimension Forbidden Zone

The naive way of selecting refined points would be to pick the first k(k≥1) points with largest percentage error. But this strategy needs further improvement in that those selected points are often clustered due to abrupt value change within a small 4D region. In our experiment, if one point with large error is selected, it's most likely that its immediate neighbors are erroneous as well and thus will be selected. So this introduces redundancy in some regions, which we need to minimize in order to achieve high compression ratio. The *Four Dimension Forbidden Zone* is the restriction we

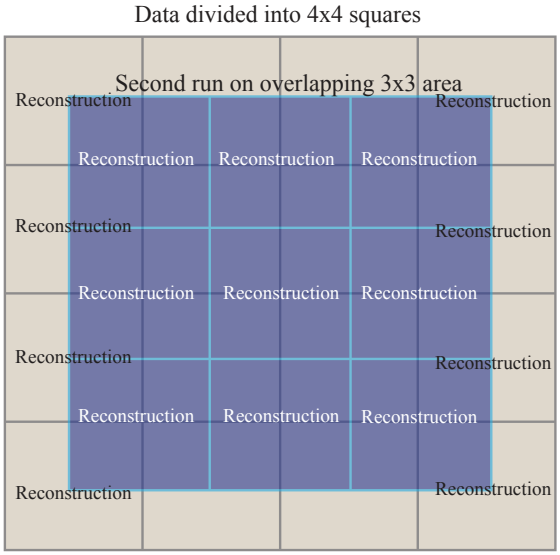

Figure 4: 2D illustration of *Dividing into Boxes* method. Reconstruction is done individually for each square in first run as marked by grey squares. The 50% overlapping blue squares also have approximation separately for each square and we take a weighted average of both values in the overlapping area (blue) to produce a good approximation.

put upon the process of adding new refined points: the candidate point's spatial local neighbors in 4D space will be checked to test if there is one or more existing refined points added in the same

iteration. If yes, this candidate point is abandoned and the point with the next biggest error is tested until we add a predefined number of points. Figure 3 is a 2D illustration of a forbidden zone. The red point has been added in this iteration, and the circle around it is the forbidden zone. The green points are in the circle so they are not included. All the blue points are outside the zone and thus have the possibility to be included in this iteration if they are not in other added points' forbidden zone.

3.4 Implementation

3.4.1 Speed-up Solving such a large linear system can be time-consuming and requires considerable memory. Since this linear system is overdetermined, the underlying solver uses sparse QR decomposition in our implementation in Matlab. It runs much slower than the Cholesky factorization, which solves only Symmetric Positive Definite linear systems.

The *Normal Equations* is introduced in (Li et al., in-press). It transforms the overdetermined linear system $Ax = b$ into an equivalent system:

$$A^T Ax = A^T b \tag{4}$$

Now we can use Cholesky factorization to solve it because $A^T A$ is Symmetric Positive Definite. By applying this method on 5D-ODETLAP, the running time of our algorithm significantly decreases. For a performance comparison between original solver and transformed solver using the *Normal Equations* method and technical details, refer to (Li et al., in-press).

Solving this large linear system still requires much memory and the running time is not fast enough to be applied on large real world data. So we use a divide-and-conquer strategy, *Dividing into Boxes* to enable 5D-ODETLAP to be applied on large dataset. We also used similar approach, except that now the sub-box is a 4D box instead of 3D. A 2D illustration is presented in Figure 4.

	Data1			
Method	Random1	Random2	Grid	Border
Points	10527	10227	10356	10344
Mean Err(%)	2.240	2.302	3.040	2.460
Max Err(%)	36.33	46.67	58.76	49.60
	Data2			
Method	Random1	Random2	Grid	Border
Points	14040	14058	13968	14088
Mean Err(%)	2.341	2.354	3.610	3.066
Max Err(%)	59.360	59.636	58.358	59.900

Table 1: This table shows the consequent initial approximation of using different initial point selection methods on two datasets. Data1 has 99.82% filled points and Data2 has only 58.01% filled points. Random1 and Random2 are two runs on the same dataset using Matlab's random method.

3.4.2 Initial Point Selection Different initial point selection strategies may result in large differences in the quality of consequent approximation. Our above *Dividing into Boxes* strategy involves doing reconstruction in smaller 4D boxes and then merging them back together to form a complete 4D reconstruction. So it's inevitable that on the borders of these small 4D boxes, we don't have enough information from the data to produce a good approximation even though we alleviate this problem by taking weighted average of two different recontracted values at one point(Li et al., in-press). To have a satisfying initial approximation, we created a *Border Selection* initial point selection method, which gives priority to points on borders of smaller sub-boxes, and then use regular grid selection inside each sub-boxes to get enough number of point initially. Then we compared this method with the regular grid selection and the random selection method and we kept the number of initially selected point to be approximately the same.

We can see from Table 1 that the random selection method actually has a surprisingly smaller mean and max percentage error on both datasets. Although two runs of random selection have slightly different results, the number of points selected by all methods are approximately the same and random methods in general are better than regular grid and *Border Selection* methods. Note that since each 4D dataset in the 5D datset has different data range and different practical meaning, we take the average of all 4D datasets' mean percentage error as the mean percentage error for the whole 5D dataset. Similarly, we also take maximum of all 4D datasets' maximum percentage error as the maximum percentage error for the whole 5D dataset. This metric is used in the entire paper.

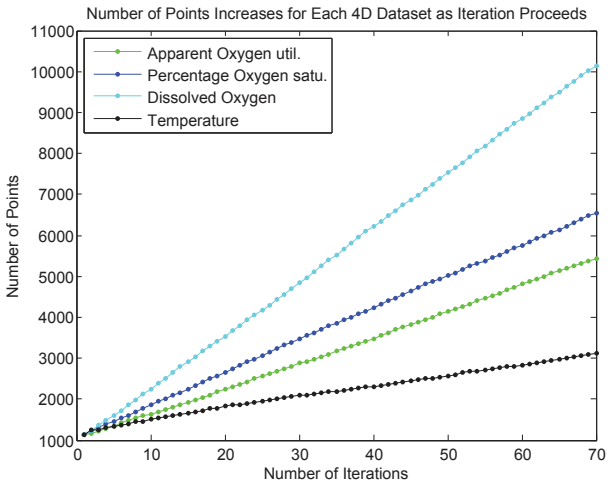

Figure 5: This figure shows that as the iterative process proceeds, the number of points added to each 4D dataset within the 5D data. Each 4D data is of size $48 \times 48 \times 24 \times 12$ and the test 5D data is of size $48 \times 48 \times 24 \times 12 \times 4$.

4 RESULT AND COMPARISON

4.1 Experiment on WOA05

Our experiment of 5D-ODETLAP is based on a real world marine 5D dataset–World Ocean Atlas 2005 (WOA05), which is provided by NODC (National Oceanographic Data Center). WOA05 is a set of objectively analyzed (1 degree grid) climatological fields of in situ temperature, salinity, dissolved oxygen, Apparent Oxygen Utilization (AOU), percent oxygen saturation, phosphate, silicate, and nitrate at standard depth levels for annual, seasonal, and monthly compositing periods for the World Ocean. In our experiment, we derive four $48 \times 48 \times 24 \times 12$ 4D datasets for fields including Temperature, Apparent oxygen utilization, Percentage oxygen saturation and Dissolved oxygen. So this derived test dataset has dimension of $48 \times 48 \times 24 \times 12$.

Table 2 shows information about the raw 5D data from WOA05. Besides smooth variable R and initial point selection strategy, other parameters can also affect the effectiveness of our method. These are points added at each iteration and size of forbidden zone. Aiming at achieving high compression ratio while keeping the errors beneath a tolerant level, we approach the optimal parameters as close as possible for the test 5D dataset.

	Fixed Mean Percentage Error	
Method	5D-ODETLAP	3D-SPIHT
Mean Err(%)	1.6803 (1.6408, 1.6215, 1.8281, 1.6310)	1.6788 (1.6399, 1.6230, 1.8258, 1.6263)
Max Err(%)	29.2318 (6.7723, 29.2318, 8.7549, 6.2170)	32.7897 (31.0172, 26.3545, 29.1149, 32.7897)
Compressed Size(byte)	96195	192204
Compression Ratio	110.37 : 1	55.24 : 1
	Fixed Maximum Percentage Error	
Method	5D-ODETLAP	3D-SPIHT
Max Err(%)	29.2318 (6.7723, 29.2318, 8.7549, 6.2170)	32.3055 (8.2964, 32.3055, 9.9796, 6.7117)
Mean Err(%)	1.6803 (1.6408, 1.6215, 1.8281, 1.6310)	0.9637 (0.6104, 1.7630, 0.8855, 0.5962)
Compressed Size(Kb)	96195	389495
Compression Ratio	110.37 : 1	27.26 : 1

Table 3: This table shows the performance comparison between 5D-ODETLAP and 3D-SPIHT on the derived dataset, which is the same with Data2 dataset in Table 1. The upper part of this table shows the results with approximately same mean percentage error of both methods; lower part shows results with approximately same maximum percentage error. Random initial selection is used for 5D-ODETLAP. The four values within brackets in the rows "Mean Err" and "Max Err" represent respectively values of each 4D dataset, including apparent oxygen utilization, percentage oxygen saturation, dissolved oxygen and temperature. Mean error is the average of all four 4D datasets' mean percentage error and similarly, Max error is the maximum of all four 4D data's maximum percentage error. Each point uses 4 bytes to store its value in single precision; thus total size of each dataset is 10.13Mb.

Variable	Unit	Data Range	Size(Mb)
Temperature	$^\circ$	[-2.10, 33.23]	71.20
Dissolved oxygen	ml^{-1}	[0, 11.80]	71.20
Apparent oxygen utilization	ml^{-1}	[-4.11, 8.42]	71.20
Percent oxygen saturation	%	[0, 158.57]	71.20

Table 2: Before compression, these are monthly objectively analyzed climatology datasets on 24 standard depth levels on four variables. Each point uses 4 byte to store its value in single precision; thus total size of each 4D dataset is 71.20Mb, which forms a 5D data of 284.77Mb in total.

Figure 5 shows the number of points increases in each 4D dataset as we add 300 points with the largest errors in 5D space. The derived test data have size of $48 \times 48 \times 24 \times 12 \times 4$. We demonstrate the iterative process of 5D-ODETLAP in each 4D dataset. It can be seen from this figure that during each iteration, the points added in each 4D dataset are not even. Because each 4D data has different data distribution, it's possible that the approximation of one 4D dataset is more likely to have more erroneous points than others. Figure 5 clearly illustrates this possibility. Consequently, since the number of points added at each 4D dataset are different, they tend to have different mean and max percentage errors. Figure 6 and 7 show that as the points are added, each 4D datasets' mean and maximum percentage error decrease at different speed, and priority is given to the datasets with most large errors when adding points.

4.2 Compression Comparison with 3D-SPIHT

We have illustrated the iterative process of compressing a $48 \times 48 \times 24 \times 12 \times 4$ 5D dataset above. In our comparison with the 3D-SPIHT method, we use the same derived dataset with each point caring a single-precision value, stored in 4 bytes each and resulting in a total size of 10.13Mb. We also apply 3D-SPIHT on the same dataset in our experiment in order to provide an objective comparison of the compression performance between two algorithms. The idea is to compress every 3D blocks of this 5D data and sum up the size of all compressed 3D data. Since the first three dimension of this 5D data is $48 \times 48 \times 24$, the length of each dimension is not an integer power of 2. In order to apply 3D-SPIHT on this 3D block, zero padding is used to extend 3D block from $48 \times 48 \times 24$ to $64 \times 64 \times 32$.

We can see from Table 3 that 5D-ODETLAP outperforms 3D-

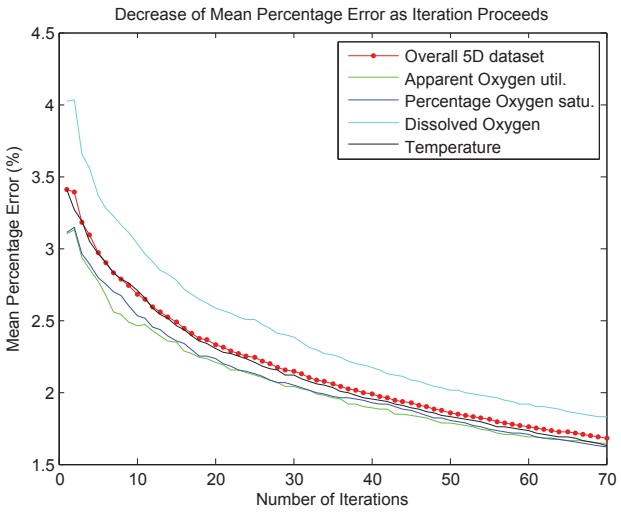

Figure 6: This figure demonstrates the decrease of mean percentage error of each 4D dataset within the 5D dataset and the overall 5D dataset, which is the average mean percentage error of all 4D datasets. Test data is the same as that in Figure 5.

SPIHT in general. Firstly, with approximately same mean percentage error in overall 5D dataset and every individual 4D dataset, 5D-ODETLAP has a high compression ratio of 110:1 as shown in the upper part of Table 3. But the compression ratio of 3D-SPIHT in this case is only 55:1, which is about half as much as that of proposed method. Furthermore, the maximum percentage errors of 5D-ODETLAP in overall 5D dataset and every individual 4D dataset is a lot smaller than the ones from 3D-SPIHT. In addition, our method provides users the flexibility to set a desired maximum or mean percentage error before compression, while 3D-SPIHT can't.

Secondly, in the lower part of Table 3, if we force the maximum percentage errors of 3D-SPIHT in overall 5D dataset and every individual 4D dataset to be approximately the same as, or even a little worse than, the ones of 5D-ODETLAP, we can see that 5D-ODETLAP has a compression ratio of 110:1, which is almost four times as much as that of 3D-SPIHT. Since we are proposing a lossy compression method, the maximum error is important because we need to guarantee, in many application, that the compression's error is under certain limit. So if we take this into consideration, 5D-ODETLAP is much more better than 3D-SPIHT.

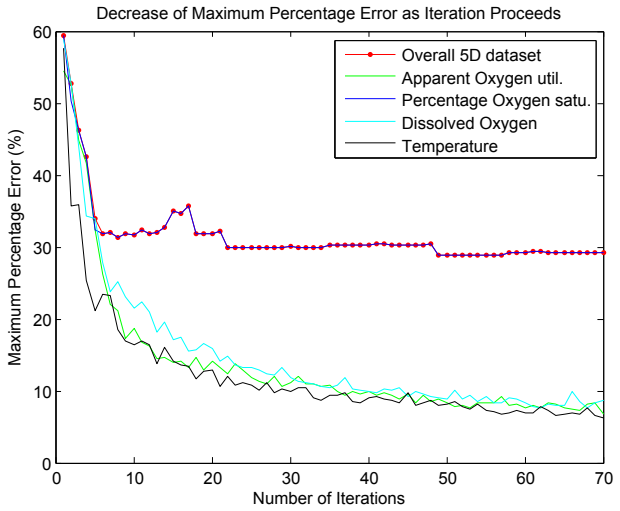

Figure 7: This figure demonstrates the decrease of maximum percentage error of each 4D dataset within the 5D dataset and the overall 5D dataset, which is the maximum maximum percentage error of all 4D datasets. Test data is the same as that in Figure 5.

5 CONCLUSION AND FUTURE WORK

Our recent work in 5D Time-varying Multivariable Geospatial Dataset Compression has been demonstrated in this paper. Our technique efficiently exploits spatial and temporal redundancies in 5D geospatial data to achieve high compression ratio. Performance evaluation shows that the proposed method achieves great compression ratio on our test data–WOA05.

We have limited 5D-ODETLAP in the application of compressing 5D geospatial data in this paper. However, the potential of 5D-ODETLAP is far beyond this field. 5D-ODETLAP is certainly not restricted in geology-related fields, and it provides a framework for researchers to explore its usability in other fields which necessitate true 5D compression on large datasets. With proper parameter settings, 5D-ODETLAP has the ability to approximate various kinds of 5D data. But this still needs much more work to adjust 5D-ODETLAP for those specific purposes.

Now 5D-ODETLAP hasn't taken into account the correlation between each 4D dataset within the 5D dataset. It only evaluates the approximation in 5D without taking advantage of possible variable correlations. So the next step is to incorporate the correlation analysis into 5D-ODETLAP and exploits the redundancy in 5D space further more to achieve a even higher compression ratio.

6 ACKNOWLEDGEMENTS

This research was partially supported by NSF grant CMMI-0835762. We also thank Zhongyi Xie for his valuable advice.

REFERENCES

Anagnostou, K., Atherton, T. and Waterfall, A., 2000. 4D volume rendering with the Shear Warp factorisation. In: Proceedings of the 2000 IEEE symposium on Volume visualization, ACM, p. 137.

Antonov, I., J., Locarnini, R. A., Boyer, T. P., Mishonov, A. V. and Garcia, H. E., 2006. World ocean atlas 2005, volume 2: Salinity. p. 182.

Franklin, W. R., 2000. Applications of geometry. In: K. H. Rosen (ed.), Handbook of Discrete and Combinatorial Mathematics, CRC Press, chapter 13.8, pp. 867–888.

Garcia, E., H., Locarnini, R. A., Boyer, T. P. and Antonov, J. I., 2006. World ocean atlas 2005, volume 4: Nutrients (phosphate, nitrate, silicate). p. 396.

Kim, B.-J. and Pearlman, W., 1997. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In: Data Compression Conference, 1997. DCC '97. Proceedings, pp. 251 –260.

Kim, Y. and Pearlman, W., 1999. Lossless volumetric medical image compression. Proc. of SPIE, Applications of Digital Image Processing 3808, pp. 305–312.

Li, Y., Lau, T.-Y., Stuetzle, C.-S., Fox, P. and Franklin, W., in-press. 3D Oceanographic Data Compression Using 3D-ODETLAP. 18th ACM SIGSPATIAL international conference on Advances in geographic information systems.

Locarnini, R. A., Mishonov, A. V., Antonov, J. I., Boyer, T. P. and Garcia, H. E., 2006. World ocean atlas 2005, volume 1: Temperature. p. 182.

Lum, E. B., Ma, K. L. and Clyne, J., 2001. Texture hardware assisted rendering of time-varying volume data. In: VIS '01: Proceedings of the conference on Visualization '01, IEEE Computer Society, Washington, DC, USA, pp. 263–270.

Menegaz, G. and Thiran, J., 2002. Lossy to lossless object-based coding of 3-D MRI data. IEEE Transactions on Image Processing 11(9), pp. 1053.

Sanchez, V., Nasiopoulos, P. and Abugharbieh, R., n.d. Lossless compression of 4D medical images using H. 264/AVC.

Sewell, G., 2005. The numerical solution of ordinary and partial differential equations. Wiley-Interscience.

Skodras, A., Christopoulos, C. and Ebrahimi, T., 2001. The JPEG 2000 still image compression standard. IEEE signal processing magazine 18(5), pp. 36–58.

Stookey, J., Xie, Z., Cutler, B., Franklin, W., Tracy, D. and Andrade, M., 2008. Parallel ODETLAP for terrain compression and reconstruction. In: GIS '08: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems.

Weinberger, M., Seroussi, G. and Sapiro, G., 2000. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. IEEE Transactions on Image Processing 9(8), pp. 1309.

Xie, Z., Franklin, W. R., Cutler, B., Andrade, M., Inanc, M. and Tracy, D., 2007. Surface compression using over-determined Laplacian approximation. In: Proceedings of SPIE Vol. 6697 Advanced Signal Processing Algorithms, Architectures, and Implementations XVII.

Yang, W., Lu, Y., Wu, F., Cai, J., Ngan, K. and Li, S., 2006. 4-D wavelet-based multiview video coding. IEEE Transactions on Circuits and Systems for Video Technology 16(11), pp. 1385–1396.