PREVENTION OF SPATIAL DATA COMPRESSION QUALITY DEGRADATION

Bart Guetti

Tele Atlas North America
bart.guetti@teleatlas.com

KEY WORDS: Compression, indexing, Huffman, blocking, mobile

ABSTRACT

Data indexing and compression procedures have enabled several significant advances in the application of spatial data. Data coverage, portability, transfer, multi-scale representation (pyramids), queries, and encryption have all become more effective with the data reduction ratios attained with today's compression technology. Applications that were previously prohibitively expensive with Text data now become feasible with Shape, RLE, Huffman, and Reduced Huffman compressors.

However the compression techniques applied are not without risk. Data loss, corruption, chunking, and outliers, are all examples of problems that can arise, or are aggravated by, higher order compression processes.

To minimize the opportunities for data quality decline in the compression process, a well defined data conversion protocol has been established. The protocol includes gaining hands on understanding of data compression tools; the use of best software engineering practices for developing software using those tools; strictly controlled software deployment and execution; and testing the resulting product using a combination of spatial analysis and spatial queries.

This paper will present data compression fundamentals; their application to spatial data; examples of successes, and failures in their use; best development and implementation practices taken to protect data quality; and a test regimen developed to detect the failures. The context of this paper will be those experiences gained in the development of a production system for a high performance display, query, geocoding and routing data product in ESRI's Smart Data Compression (SDC) format.

1.0 Introduction

Compressed data is a double-edged sword. It provides great promise and opportunity for creating geographically intelligent applications and devices that were heretofore prohibitive due to space and performance constraints. However data compression can also result in unexpected data loss and corruption. To overcome these constraints and evaluate the risks, Tele Atlas undertook a project to create a data product that utilized high compression technology. The project involved the use of ESRI's Data Development Kit (DDK) to create Tele Atlas desktop, web service, and mobile capable StreetMap Premium for North America.

2.0 Problems

2.1 Data storage

A typical high-resolution street network with attribution sufficient to support routing and geocoding for a nation of 40 million street edges occupies the space listed in Figure 1.

Format	Size
SHP	25 GB
Text	20 GB
ArcSDE	10 GB

Figure 1
File size by format

Storage adequate to hold these amounts of data has been available in server environments for the past several years, and became available on PC clients within the past 4 years. Storage adequate for the text, SHP and ArcSDE formats is still not available on DVD, in-car navigation units, PDA (pocket pc) or mobile devices. As a result of this, tradeoffs have to be made in one or more of the following data parameters to enable storage of street network data on mobile devices:

- Resolution
- Attribution
- Coverage
- Topology

2.2 Application Requirements

While the tradeoffs in data parameters were acceptable when applications of the data were oriented towards consumer or casual users, mobile devices are increasingly being tasked with performing critical operations by field oriented organizations. Utilities, transportation, and telecommunications are just a few of the business types that are deploying mobile technology to deliver services and goods. These enterprises are also increasingly using these devices to capture, and even edit, spatial data. The practice of omitting data to enable the use of data in mobile devices is unacceptable for these newer, more demanding applications.

2.3 Processor Speed

Due to heat issues and power requirements, mobile devices generally run slower processors. This further complicates the situation for users of mobile technology and storage limited devices. As a result, these slower processors currently need data related improvements to attain the speeds that mobile applications require.

3.0 Compression Fundamentals

3.1 Background

Data compression has been used for decades with Braille, Morse code (Salomon, 2006) and Vocoder (Sayood, 2005) being early examples of encoding techniques developed to reduce data volume. The two major categories of modern compression are:

- Lossless
- Lossy

Lossy compression is used in audio and visual applications where the loss of certain bits of information will not be noticed. Unpredictable loss of data is not acceptable in the high performance geocoding and routing domain and as a result the remainder of this paper concerns lossless data compression.

The two major compression techniques employed in both categories above are:

- removing redundancy
- removing noise

Removing redundancy involves substituting symbols or codes that indicate the presence and/or frequency of repetitive bits of data whereas noise removal involves the elimination of data that adds no information value.

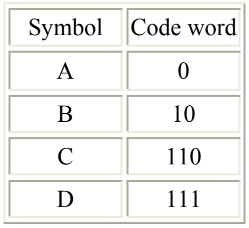
3.2 Categories

The major categories of compressors used in lossless data compression today are:

- Huffman
- RLE Huffman
- Reduced Huffman
- Universal shape

3.2.1 Huffman

Huffman compression is a lossless compression algorithm that is ideal for compressing text or program files as it uses codes of varying lengths to replace symbols based upon the probability of the symbol appearing in the data. Symbols with higher probabilities of incidence are coded with shorter code words, while symbols with lower probabilities are coded with longer code words. Huffman codes belong to a family of codes with a variable codeword length. That means that individual symbols which make a message are represented (encoded) with bit sequences that have distinct length. This characteristic of the code words helps to decrease the amount of redundancy in message data i.e. it makes data compression possible. For example, symbols A, B, C and D are represented with following code words:



Symbol	Code word
A	0
B	10
C	110
D	111

Figure 2
Example of code words for symbols

Symbols A and B have distinct lengths of their code words (“0” and “10”). At the first look it seems that the code words are not uniquely decodable. But, the power of Huffman codes is in the fact that all code words are uniquely decodable. So, the sequence of bits: ***01101110100*** is uniquely decodable as `***ACDABA***`.

3.2.2-RLE

Run length encoding (RLE) stores counts that indicate repeating values in sequential records. It is effective in spatial applications where proximity correlates to values (spatial autocorrelation). RLE Huffman combines the RLE counts with the Huffman codes to attain improved compression rates over the individual approaches.

3.2.3-Reduced Huffman

This compressor is used for numeric attribute fields when there is no information about distribution of values in the field. It is an optimization of the Huffman algorithm where Huffman codes are used to store lengths of values.

3.2.4 Universal shape

Universal shape data compression compresses the change in coordinate values (deltas) from a base value using Reduced Huffman encoding.

Figure 3 provides a quick reference of the best compressor to use based upon data type; spatial and row order distribution; and the number of unique values in the field to be compressed. NA indicates that this variable is not a significant factor in compressor selection.

Data Type	Distribution	Unique Values	Compressor
Number	Random	NA	Reduced Huffman
Number	Clustered	NA	RLE Reduced Huffman
String	Random	Low	Huffman
String	Clustered	NA	RLE Huffman
Table	Random	High	Huffman
Table	Random	Low	Huffman
Table	Clustered	Low	RLE Huffman
Geometry	NA	NA	Universal shape

Figure 3
Compressor selection

4.0 Implementation

4.1 Space

Compression ratios of at least 10:1 (Figure 4) over the input sources were consistently attained in the creation of features in the SDC format. For example the size of the street network in ESRI's ArcSDE format was 10 GB whereas the size of the final street network in compressed SDC format was 1 GB. Whereas the ArcSDE format requires the user to store the data in a relational database, such as Oracle or SQL Server, SDC is a file based format, requiring no relational database. When compared to the other file based formats, SDC has a compression ratio advantage of 20:1 over text representation and a 25:1 over ESRI's Shapefile format. The other breakthrough over the Shapefile format is the elimination of the 2GB file size limitation. This enables data users to now create nationwide, file based, layers.

Format	Compression Ratio
SHP	25:1
Text	20:1
ArcSDE	10:1

Figure 4
Compression ratios attained by format

4.2
Performance

Through a combination of more efficient data storage and indexing, compressed data access is faster resulting in improved drawing and querying speeds and greatly improved routing and geocoding speeds when compared to input sources.

4.3
Function

4.3.1 Pyramids

The goal of creating and storing data for multiple representations has been an elusive one partially due to space issues and associated rendering speeds. Through a combination of advances in RDBMS technology and compression approaches, the creation and storage of multiple levels of data granularity is now feasible and practical (Figure 5). SDC files enable data producers to store several generalization levels in a single file. And applications that read SDC can reveal them at user defined thresholds. Figure 6 shows the generalization settings used to create the pyramid and the fields used to store the generalized output.

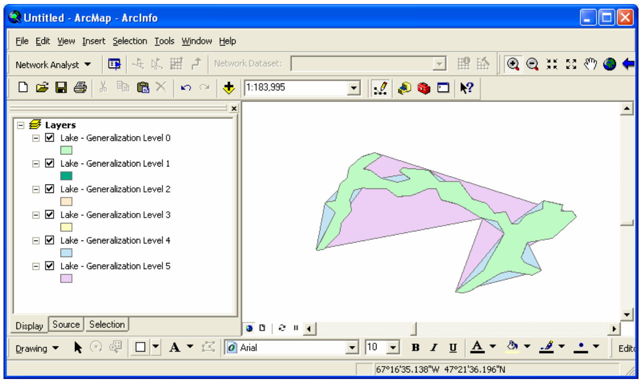
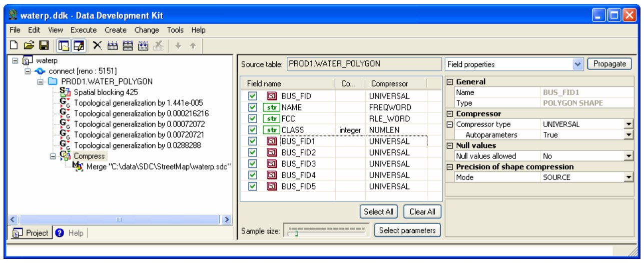

Untitled - ArcMap - ArcInfo
File Edit View Insert Selection Tools Window Help
Network Analyst
Network Dataset:
1:183,995
Layers
☑ Lake - Generalization Level 0
☑ Lake - Generalization Level 1
☑ Lake - Generalization Level 2
☑ Lake - Generalization Level 3
☑ Lake - Generalization Level 4
☑ Lake - Generalization Level 5
Display
Source
Selection
Drawing
Arial
10
67°16'35.138"W 47°21'36.196"N

Figure 5
Pyramid of generalization levels


waterp.ddk - Data Development Kit
File Edit View Execute Create Change Tools Help
Select All
Clear All
Select parameters
Propagate
Project
Help

waterp
connect [reno : 5151]
PROD1.WATER_POLYGON
Spatial blocking 425
Topological generalization by 1.441e-005
Topological generalization by 0.000216216
Topological generalization by 0.00072072
Topological generalization by 0.00720721
Topological generalization by 0.0288288
Compress
Merge "C:\data\SDC\StreetMap\waterp.sdc"

Source table: PROD1.WATER_POLYGON

	Field name	Co...	Compressor
☑	BUS_FID		UNIVERSAL
☑	NAME		FREQWORD
☑	FCC		RLE_WORD
☑	CLASS	integer	NUMLEN
☑	BUS_FID1		UNIVERSAL
☑	BUS_FID2		UNIVERSAL
☑	BUS_FID3		UNIVERSAL
☑	BUS_FID4		UNIVERSAL
☑	BUS_FID5		UNIVERSAL

Sample size:

Field properties

General	
Name	BUS_FID1
Type	POLYGON SHAPE
Compressor	
Compressor type	UNIVERSAL
Autoparameters	True
Null values	
Null values allowed	No
Precision of shape compression	
Mode	SOURCE

Figure 6
Generalization settings

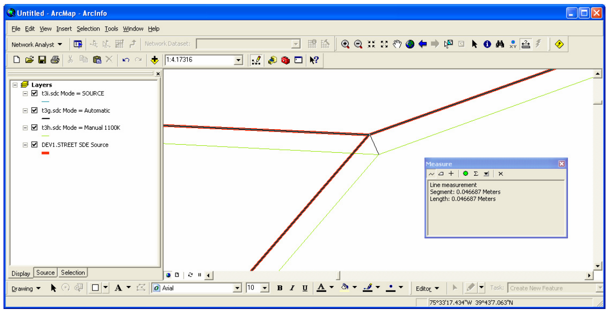
4.3.2 Single DVD

Data sets that formerly required multiple CD's or DVD's now fit on a single disc, thereby enabling developers to create products that eliminate the need to copy the product to a computer's hard drive. Users can now use the product directly from the disc. The entire SMP North America occupies 3.7 GB which easily fits on a 4.7 GB single layer DVD.

5.0 Surprises

5.1 Feature movement

During the geometry compression phase, users have the option to manually set output precision. The advantage of lowering the precision is a reduction in the size of the output. However the disadvantage is the movement of features relative to their location in the source data (Figure 7).



Layers
☑ t3i.sdc Mode = SOURCE
☑ t3g.sdc Mode = Automatic
☑ t3h.sdc Mode = Manual 1100K
☑ DEV1.STREET SDE Source

Measure
Line measurement
Segment: 0.046687 Meters
Length: 0.046687 Meters

75°33'17.434"W 39°43'7.063"N

Figure 7
Feature movement at various precisions

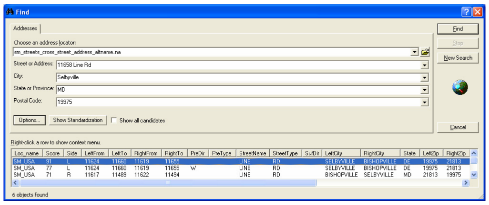
A significant reduction in file size can be attained however with a minimal reduction in precision. A file can be reduced nearly 25% in size with less than 1 meter loss in precision. Figure 8 shows the file size changes associated with reductions in precision. For applications that will not tolerate precision loss, the SOURCE data precision compressor option was selected which eliminated any feature movement.

Precision	File size
.1 Meter	1,386,434
1 Meter	1,054,116
10 Meter	670,886

Figure 8
File size reduction by precision

5.2 Corruption

Logical compression is an option to the Huffman compressors and enables users to combine multiple fields into a single field. This works fine if the values in the fields are the same. However when they are different, using the compression strategy available at the time, one overwrote the other resulting in confusing results. In Figure 9, an attempt to geocode to an address that had a different postal code on the opposite side of the street results in a lowered match score (91/100). This occurred because the POSTAL_L and POSTAL_R were compressed using the logical option and the right sided value overwrote the left sided value.



Choose an address locator: sm_streets_cross_street_address_altname.na
Street or Address: 11658 Line Rd
City: Selbyville
State or Province: MD
Postal Code: 19975
☐ Show all candidates

Right-click a row to show context menu.

Loc_name	Score	Side	LeftFrom	LeftTo	RightFrom	RightTo	PreDir	PreType	StreetName	StreetType	SufDir	LeftCity	RightCity	State	LeftZip	RightZip
SM_USA	91	L	11624	11660	11619	11655			LINE	RD		SELBYVILLE	BISHOPVILLE	DE	19975	21813
SM_USA	77	L	11624	11660	11619	11655	W		LINE	RD		SELBYVILLE	BISHOPVILLE	DE	19975	21813
SM_USA	71	R	11617	11489	11622	11494			LINE	RD		BISHOPVILLE	SELBYVILLE	MD	21813	19975

6 objects found

Figure 9
Logical compressor geocoding penalty

This error was traced to an unexpected scenario in the data which the logical compression option can now handle correctly.

5.3 Blocking

Blocking is the indexing and sorting of data based upon spatial distribution and/or spatial attribute value. It is a powerful tool for spatial data organization and results in blocks that are roughly equivalent in size. This predictability in block size makes for consistent and modifiable data access results. By establishing a maximum number of features per block users have a fair amount of control over the spatial indexing. Quad-tree and Oct-tree tessellation is available and enables users to tune their indexing to match their application requirements. The basic tradeoff is access speed vs. file size with a smaller number of features per block resulting in more blocks, and a larger file size, but faster access.

5.3.1 Chunking

A by product of the control of the maximum number of features per block is the creation of super blocks. Super blocks are larger blocks to hold the features that did not “fit” into the blocks. They did not fit because they were either too large, too many or a combination of both. This tends to occur when there is a mix of many long segments in an area with many short segments. The results can be super blocks with an excessive number of segments that can degrade access times. Figure 10 is a representation of a geocoding segment distribution that resulted in the creation of a poor performing super block, containing 90K segments. This was corrected by increasing the maximum number of features per block, thereby reducing the number of features that ended up in the super block. Another, less than ideal, solution is the splitting of the longer edges to reduce the extent of the super block.

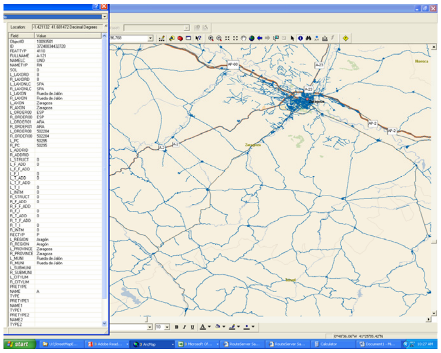
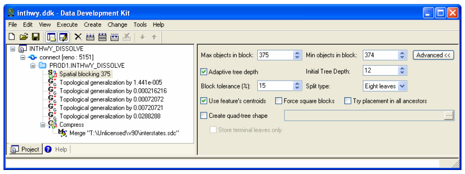


Figure 10
Super block scenario

5.3.2 Sweet spot

Occasionally a spatial distribution scenario resulted in blocking failures using anything but one particular maximum/minimum combination, aka a sweet-spot. These were very difficult to deal with as they involved a heuristic approach to finding that value. The use of a spatial distribution analysis tool is being considered to help determine where these scenarios exist and what blocking settings to use. Figure 11 shows the parameters for dealing with one of those scenarios.


inthwy.ddk - Data Development Kit
File Edit View Execute Create Change Tools Help
Project
Help
Advanced <<

INTHWY_DISSOLVE
connect [reno : 5151]
PROD1.INTHWY_DISSOLVE
Spatial blocking 375
Topological generalization by 1.441e-005
Topological generalization by 0.000216216
Topological generalization by 0.00072072
Topological generalization by 0.00720721
Topological generalization by 0.0288288
Compress
Merge "T:\Unlicensed\v90\interstates.sdc"

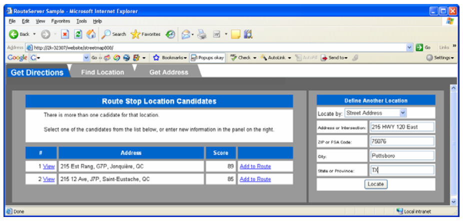
Max objects in block: 375
Min objects in block: 374
☑ Adaptive tree depth
Initial Tree Depth: 12
Block tolerance (%): 15
Split type: Eight leaves
☑ Use feature's centroids
□ Force square blocks
□ Try placement in all ancestors
□ Create quad-tree shape
□ Store terminal leaves only

Figure 11
Heuristically achieved spatial blocking settings

Until the analytical tool is developed, the software developers have calculated a pre-determined set of values for users to reference.

5.4 Outliers

Like any statistically based process there are scenarios that deviate from the norm. In the scenario in Figure 12, an address 2000 KM away from the actual address, and in a different country, was selected when geocoding. The only item in common between the two address was the address number, 215. This outlier is still being examined for cause. One possible explanation is a symbol substitution error in the Huffman encoding.



Address http://2k-32307/website/streetmap800/

Route Stop Location Candidates

There is more than one cadidate for that location.

Select one of the candidates from the list below, or enter new information in the panel on the right.

#	Address	Score	
1 View	215 Est Rang, G7P, Jonquière, QC	89	Add to Route
2 View	215 12 Ave, J7P, Saint-Eustache, QC	85	Add to Route

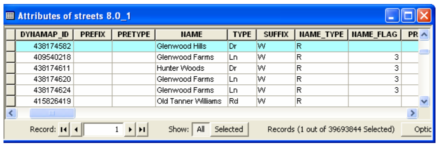
Define Another Location

Locate by: Street Address
Address or Intersection: 215 HWY 120 East
ZIP or FSA Code: 75076
City: Pottsboro
State or Province: TX

Figure 12
Geocoding outlier

5.5 Dropped values

When compressing numeric data, users have the option of compressing the data as a character string. In this case, the choice of string to compress STREET.NAME_FLAG resulted in values other than 3 omitted from the output, Figure 13. This error was traced to an incorrect default setting for handling NULLs within the DDK when using the compress as string option.

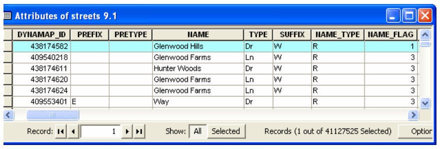


DYNAMAP_ID	PREFIX	PRETYPE	NAME	TYPE	SUFFIX	NAME_TYPE	NAME_FLAG	PR
438174582			Glenwood Hills	Dr	W	R		
409540218			Glenwood Farms	Ln	W	R	3	
438174611			Hunter Woods	Dr	W	R	3	
438174620			Glenwood Farms	Ln	W	R	3	
438174624			Glenwood Farms	Ln	W	R	3	
415826419			Old Tanner Williams	Rd	W	R		

Record: 1 Show: All Selected Records (1 out of 39693844 Selected)

Figure 13
Incorrect compress as string setting

This error was corrected by disabling the compress as string option.



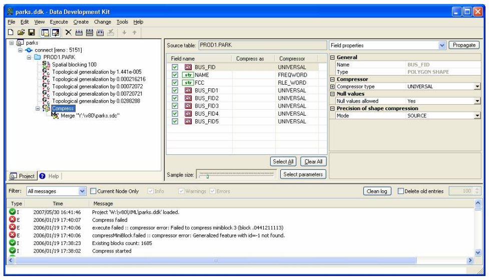
DYNAMAP_ID	PREFIX	PRETYPE	NAME	TYPE	SUFFIX	NAME_TYPE	NAME_FLAG
438174582			Glenwood Hills	Dr	W	R	1
409540218			Glenwood Farms	Ln	W	R	3
438174611			Hunter Woods	Dr	W	R	3
438174620			Glenwood Farms	Ln	W	R	3
438174624			Glenwood Farms	Ln	W	R	3
409553401	E		Way	Dr		R	3

Record: 1 Show: All Selected Records (1 out of 41127525 Selected)

Figure 14
Correct compress as string setting

5.6 Environment issues

The compression of the generalized geometry is a resource intensive process, consuming considerable processor and disk resources on both the client and the server. Some datasets resulted in unexplainable compression failures, Figure 15. The root cause of this category of failure is still outstanding. However it does appear to be an RDBMS environment issue as a change from Oracle to SQL Server as the host RDBMS cleared up the problem.


parks.ddk - Data Development Kit
File Edit View Execute Create Change Tools Help
Project
Help
Propagate
Select All
Clear All
Select parameters
Clean log

parks
connect [reno : 5151]
PROD1.PARK
Spatial blocking 100
Topological generalization by 1.441e-005
Topological generalization by 0.000216216
Topological generalization by 0.00072072
Topological generalization by 0.00720721
Topological generalization by 0.0288288
Compress
Merge "Y:\v80\parks.sdc"

Source table: PROD1.PARK

Field name	Compress as	Compressor
☑ BUS_FID		UNIVERSAL
☑ str NAME		FREQWORD
☑ str FCC		RLE_WORD
☑ BUS_FID1		UNIVERSAL
☑ BUS_FID2		UNIVERSAL
☑ BUS_FID3		UNIVERSAL
☑ BUS_FID4		UNIVERSAL
☑ BUS_FID5		UNIVERSAL

Sample size:

Field properties
General
Name BUS_FID
Type POLYGON SHAPE
Compressor
Compressor type UNIVERSAL
Null values
Null values allowed Yes
Precision of shape compression
Mode SOURCE

Filter: All messages □ Current Node Only ☑ Info ☑ Warnings ☑ Errors □ Delete old entries 100

```
Type  Time                 Message
I     2007/05/30 16:41:46  Project 'W:\v80\XML\parks.ddk' loaded.
E     2006/01/19 17:40:07  Compress failed
E     2006/01/19 17:40:06  execute failed :: compressor error: Failed to compress miniblock 3 (block .0441211113)
E     2006/01/19 17:40:06  compressMiniBlock failed :: compressor error: Generalized feature with id=-1 not found.
I     2006/01/19 17:38:23  Existing blocks count: 1685
I     2006/01/19 17:38:02  Compress started
```

Figure 15
Environment related compression failures

6.0 Best practices

As evidenced in the previous discussion, there are a large number of variables involved in conducting a data compression project, and an equally large number of opportunities for mistakes to be made. As a result best software engineering practices were observed in developing a data compression workflow and production system. All requirements are assigned a number and tracked throughout the life of the requirement. Conversion and production issues are also tracked to insure that they are not overlooked or re-introduced. Production system software is managed for version in a code repository and is released for quarterly production.

7.0 Testing

The resulting quarterly product contains 3.7 GB of data, in 180 files and 5 directories. Thorough testing this amount of data is significant undertaking requiring a rigorous test protocol to insure that the compression process did not result in loss of source data quality. The primary testing procedures include:

- Regression
- Issue
- Routing
- Geocoding

This testing is a challenge, yet it also presents an opportunity. The scope of this product enables large area testing that was heretofore difficult because of the data management involved. Having continental layers of data available in single files enables macro and micro level testing without the overhead of assembling a large number of files.

8.0 Further Investigation

Spatial data is a unique category of information. For some applications high accuracy is critical, whereas for others more general location is adequate. It could be argued that no other field of data usage has such a range of precision requirements.

To fulfill the needs of this user community we as developers of spatial data have an opportunity to help refine the accuracy requirements of the variety of disciplines that consume our offerings. Associated with this opportunity is the responsibility to determine ways to more efficiently create, store and use information. Areas where we should focus our energies:

- Higher compression orders
- More user friendly compression tools
- Better understanding of compressor behavior
- Lower data compression costs

9.0 Conclusions

Spatial data compression is an application enabling technology that has passed the bleeding edge of the innovation lifecycle and is now entering the mainstream. To successfully harness its full potential, users of the technology should have a basic understanding of its principles, and a full appreciation of its risks. The errors that can be introduced are insidious and many, but with adequate development, implementation and testing safeguards in place, users can create lightweight, yet powerful, data offerings.

Also, applications that consume spatial data have a variety of accuracy requirements. The author has demonstrated that by better understanding the compression tools that are available to data developers, data compression can be used to create products for this variety of application accuracy requirements, and in the process, avoid undesirable results.

References

2003, *Understanding SDC Compression, DDKP Approach*, ESRI Press

Sayood, Kahlid, 2005. *Introduction to Data Compression,* Third edition, Morgan Kaufman Series in Multimedia Information and Systems, Elsevier, Oxford, UK, p. 3

Salomon, David, 2006. *Data Compression: The Complete Reference,* Springer-Verlag, New York, NY, p. 15

Acknowledgements

The author would like to thank Alexey Andruschenko of Software Technologies, Taganrog, Russia, DataEast, Novosibirsk, Russia and Jeff Rogers of ESRI, Redlands, California for their enlightenment of the author in data compression and innovative spatial indexing.