

Effective Data Representation and Compression in Ground Data Systems

David A. Maluf
david.a.maluf@nasa.gov

Peter B. Tran
peter.b.tran@nasa.gov
NASA Ames Research Center
Intelligent Systems Division
Mail Stop 269-4
Moffett Field, CA 94035

David Tran
Stanford University
Stanford, CA 94305
davetran@stanford.edu

Abstract—Storing vast amounts of multidimensional telemetry data presents a challenge. Telemetry data being relayed from sensors to the ground station comes in the form of text, images, audio, and various other formats. Compressing this data would optimize bandwidth usage during transmission and reduce storage resources needed at the ground level. However, the multitude of heterogeneous data types present in telemetry data and the need for data precision makes compression quite difficult. The application of a single compression technique for all data types usually yields ineffective results. We will present a telemetry data compression algorithm that utilizes Discrete Fourier Transforms (DFTs) along with different compression algorithms for different data types, including Lempel-Ziv-Welch (LZW) and Flate (which combines LZW with adaptive Huffman coding) for textual and numerical data and JPEG coding for images. Although these algorithms do not yield the greatest compression ratios, the Portable Document Format (PDF) standard supports decoding of all of them, which allows us to write our encoded data streams directly to a PDF file. This approach alleviates the need for traditional database storage systems. It also standardizes and simplifies the data retrieval, decoding, and viewing process. This work results in packets-oriented telemetry data encapsulated with multiple compression stream algorithms, which can be decoded, rendered and viewed by any standard PDF viewer. This paper presents the aforementioned algorithms and its development status as applicable proof-of-concept prototypes.^{1,2}

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. OBJECTIVES.....	2
3. SAMPLING AND TRANSFORM	3
4. MULTIPLE COMPRESSION TECHNIQUES	3

5. LEVERAGING OFF PDF FUNCTIONALITY	4
6. CONCLUSIONS.....	6
REFERENCES.....	7
BIOGRAPHIES.....	7

1. INTRODUCTION

We are currently developing a high performing ground data system utilizing a telemetry data compression technique for future space exploration and satellite applications. This procedure meets both the demands of future high-speed networks and telemetry data containing a very large number of parameters. It applies to multidimensional telemetry downloaded across many media types. The algorithm combines a number of lossless compression algorithms such as LZW and Huffman coding. We use single dimensional DFTs to transform single dimensional streamed data from their respective sensors (e.g. latitude, longitude, temperature, pressure, etc.), optimizing on their periodicity.

We apply the basic principle of applying different compression algorithms to different data types to try to achieve a balance between compression ratio and data precision. Similar approaches have gained popularity in document-oriented data, such as Adobe PDF format [5], where different data streams are compressed differently. Telemetry packets are routed and stored at the kilobyte and megabyte scale, alleviating the need of traditional database storage requirements. Paging through the telemetry would require at most two packets for a continuity of the data stream. The compression scheme performs well on a suite of test telemetry data acquired from spacecraft instruments. It applies to two dimensional data images. Continuous data are truncated and optimized either towards arbitrary packet size or in signal resets. Ground system implementations are currently in the development phase.

¹ 1-4244-1488-1/08/\$25.00 ©2008 IEEE.

² IEEEAC paper #1359, Version 2, Updated November 8, 2007

The scope of this paper applies to packet-oriented telemetry data. Packet telemetry sends measurements at particular frequencies in bursts, whereas frame-based telemetry accumulates many measurements over time [1]. Frame-based transmission follows a fixed structure to protect against transmission errors [2]. However, with improved transmission technologies, frames become antiquated because packets offer much more flexibility both in the structure of the data as well as for transmission purposes. Packet data structure should adhere to the Space Packet Protocol standard established by the Consultative Committee for Space Data Systems (CCSDS) in 2003 [9]. Our compression technique focuses on the benefits at the ground level in terms of data storage and viewing and

rendering of the data, but can be applied either at the data source or by the on-board data system before transmission through the space-to-ground link to reduce bandwidth usage and expedite data transmission time as recommended by the CCSDS [9]. Although lossy compression techniques can achieve high compression ratios, they also sacrifice data precision, which is usually crucial in telemetry data processing. Our algorithm adheres to the CCSDS packet telemetry standard, which recommends lossless encoding to ensure that the decompressed data is an exact replica of the original [8]. Figure 1 shows the various stages of the packet telemetry data system as defined in [8] with implementation notes about our algorithm.

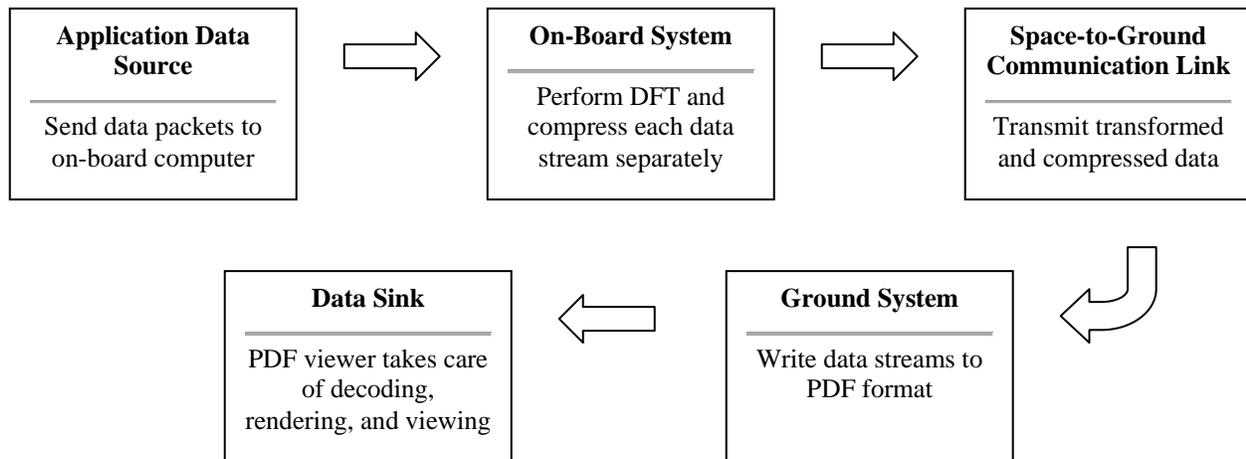


Figure 1: Application of compression technique at different stages of the Packet Telemetry Data System
*adapted from [8], pg. 2-1

Of course, a few telemetry data compression methods already exist; however, none of these techniques can be effectively applied to telemetry data consisting of multiple data types. To preserve data accuracy, most of these techniques use lossless encoding algorithms. The exception to this rule is that Joint Photographic Experts Group (JPEG) compression, which is lossy, is usually used for images because some lossiness is acceptable for images. The most popular of these techniques is Golomb-Rice coding, a form of universal encoding used in the Voyager program [3]. CCSDS recommends Rice coding over standard Huffman, Lempel-Ziv, arithmetic, and other forms of lossless coding for packet compression because it has achieved better compression ratios for a suite of test images [8]. In another example, Staudinger, et al. [6] implement an algorithm that uses Huffman Truncated Run Length Encoding and gzip [10]. Telemetry data from the Mars Pathfinder consisted predominantly of images and therefore used JPEG compression [1]. Although these methods can sometimes achieve high compression rates, no formalized standard has been developed for decompression and rendering/viewing the original telemetry data. Moreover, all of the aforementioned algorithms apply one compression technique to all data types. To date, no specification exists

to apply different coding techniques to different data types for multimedia telemetry data.

We propose a document-oriented telemetry data compression technique that uses different compression algorithms for different data types. It consists of compressing and decompressing data in document-like packets and supports conventional LZW and Flate coding algorithms [11]. Adhering to compression algorithms supported by the Portable Document Format (PDF) convention simplifies and standardizes the presentation of the telemetry data. This process ensures that users viewing the data on different machines will see the same information because the rendering of the telemetry data in PDF document viewers is pre-specified.

2. OBJECTIVES

This paper introduces a telemetry data compression technique that combines Discrete Fourier Transforms with lossless compression algorithms such as Huffman and LZW in multiple compression streams. Adobe PDF specifications (version 1.2 or higher) [5] supports decoding of these

algorithms, so any standard PDF document viewer will be able to decode and render the original data. The specific objectives of this algorithm are:

- Achieve better compression ratios while maintaining the necessary data precision by applying different encoding algorithms to different data streams.
- Optimize transmission bandwidth and data storage by representing telemetry data in the frequency domain obtained by performing single dimensional DFTs on the individual data streams.
- Further reduce the necessary storage resources required by using lossless coding techniques such as LZW and Huffman encoding and lossy techniques for images using the JPEG standard.
- Allow for efficient decoding and viewing of the original data by adhering to Adobe PDF standards, enabling the user to render and view the data in any PDF document viewer.

3. SAMPLING AND TRANSFORM

The on-board data system processes streaming data from the application data source and truncates the continuous data into discrete samples either based on signal resets or to achieve a certain packet size. The sampling interval can be user-defined to fit the needs of the telemetry data being measured. Previous works on telemetry data compression justify that sampling continuous telemetry data, when done appropriately, can maintain the accuracy of the original data because it tends to be over-sampled [6]. The selected sampling rate should not be so low as to introduce lossiness to the data. However, the sampling rate must at least meet the minimum sampling rate as defined by the Nyquist-Shannon Theorem: the Nyquist frequency (half of the sampling frequency) must exceed the bandwidth. In other words, the minimum sampling frequency should be roughly twice the signal bandwidth to ensure that no Fourier components are lost due to sampling [14]. Because our algorithm focuses on the ground implementation and be paired with any sampling algorithm, the process of choosing an appropriate sampling technique, which deserves its own discussion, falls outside the scope of this paper. One such implementation is discussed in [6], which introduces an adaptive sampling technique that changes the sampling rate to try to reduce autocorrelations (redundancies) found in the data. Henceforth, we will assume that an appropriate sampling rate for the data has been chosen.

After we sample the data, we perform a DFT on it, which will correct for any remaining over-sampling resulting in significant autocorrelations in the data. Performing one-dimensional DFTs on packets of telemetry data takes advantage of the periodicity of the stream and eliminates

redundancy in the data. Multidimensional telemetry data can be decomposed into separate data streams, which can be analyzed and compressed using different algorithms. Moreover, single dimensional data streams can be transformed to the frequency domain with greater computational efficiency by eliminating the need for multi-dimensional DFTs. This technique can be applied to two-dimensional $M \times N$ images represented as a $1 \times MN$ matrix as long as we separately save its dimensions separately. The use of the Fast Fourier Transform further optimizes the computation, allowing for $N \log_2 N$ complexity [4].

Two packets are required to be able to reconstruct the original data: the signal data containing the DFT coefficients, and secondly, the phase data, which is not stored in the frequency domain of the DFT. If an appropriate sampling interval was chosen, the original data can be reconstructed accurately because no data is lost in the transformation or lossless encoding processes. This level of precision is absolutely essential to telemetry data compression because we often do not know what values a telemeter might measure and cannot error-check based on a possible range of values. Before sending the packets through the space-to-ground communications link, the on-board system separates the signal data in the DFT domain and the corresponding phase data, both as single dimensional matrices of values.

4. MULTIPLE COMPRESSION TECHNIQUES

After the DFT, depending on the data type (e.g text, numeric, image, audio, video), this technique applies different methods of compression to achieve optimal compression rates. For text, lossless encoding techniques such as LZW or FLATE are applied, and for images, JPEG encoding is used because lossiness can generally be tolerated and it is supported by the PDF specification. The compression process within JPEG encoding is analogous to our compression for supporting additional encoding techniques, such as those supported by the PDF standard (Run-length Encoding, CCITTFax, etc.), may be added later.

Although Huffman and LZW coding achieve less optimal compression rates than Golomb-Rice and other forms of lossless encoding, Adobe PDF supports decoding of both LZW and Flate, a compression technique that combines adaptive Huffman encoding and LZW [5]. No data is lost during this stage of the algorithm and decoding should be able to exactly replicate the original data, which will be in the DFT domain.

In Figure 2, we show the application of the transforms and compression algorithms at the on-board system before we transmit the data through the space link. Positional and velocity information in the form of textual data is followed by image data in the stream coming to the on-board system from the application data source. From the textual data, we

apply a DFT then LZW or Flate coding. Analogously, we apply a Discrete Cosine Transform (DCT) for the JPEG and encoding using Huffman; however, we must first convert the raw image data from the RGB color space to the YCbCr subspace and perform chrominance sub-sampling; this step

takes advantage of the human eye's greater relative sensitivity to luminance than chrominance. After we apply a DCT, we quantize the transform coefficients before applying Huffman coding.

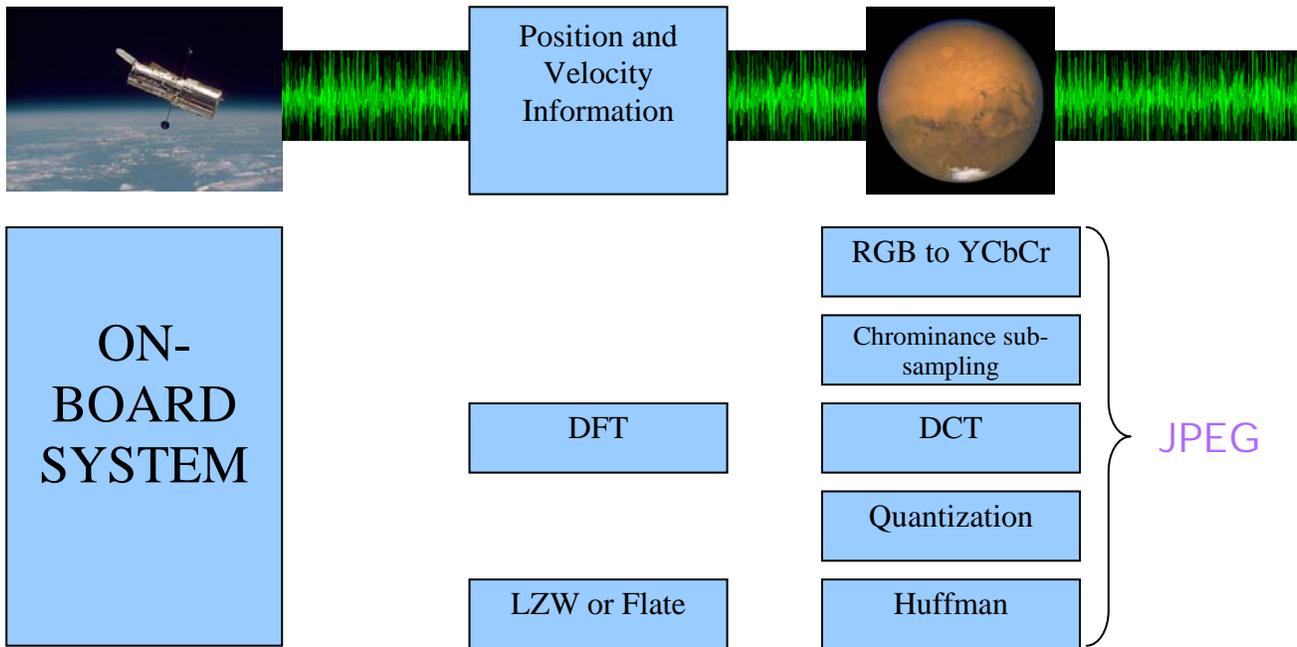


Figure 2: Application of different transform and compression algorithms to textual/numeric data (“Position and Velocity Information” on the left) and an image (on the right) within a data stream coming from the application data source to the on-board system.

By the end of all of the steps detailed in Figure 2, our data stream will consist of encoded text following by a JPEG image, both ready to be passed through the space link and then written directly to a PDF-like document at the ground systems level. Along with this data stream, we must either send along another stream containing information that the PDF Decoder filters will need to decompress the data, such as which compression or transforms we used and the size of the streamed data.

5. LEVERAGING OFF PDF FUNCTIONALITY

Adhering to the PDF standard allows the users to render and view the decompressed telemetry data in common PDF document viewers. Moreover, this format allows easy access, transmission, and storage of the telemetry data, eliminating the need for intermediate databases by writing the streamed telemetry data directly to PDF format. The PDF format documents containing the telemetry data can then be easily and securely stored in a local file system for

intuitive user rendering using a simple PDF-compatible viewer.

PDF documents can be broken down into four principal components: the actual objects storing data, the document structure, the file structure, and the content stream [5]. The Adobe PDF standard supports eight object types: Boolean values, numbers, strings, names, arrays, dictionaries, streams, and null objects. Numerical and textual data can simply be written using numeric and string objects as specified in the PDF standard [5]. Any repeated measurement of the same data can be stored in arrays or arrays within arrays to be rendered as tables in the final PDF document.

The second to last type of object mentioned above, streams, are just a special type of graphic object, comprised of a sequence of zero or more bytes of information accompanied by a dictionary. Our algorithm uses object streams to represent the majority of the telemetry data streams because unlike strings, object streams do not require any length specification and can be encoded using any of the PDF-supported compression algorithms. For example, if

telemetry data from a sensor were to relay a data stream with some unknown amount of text and numeric data followed by a series of images, our algorithm would do the following:

1) If we are just starting a PDF document, write the header, which includes any necessary metadata for later use, and write the body marker to signal the beginning of the content. Write our object marker.

2) Write the dictionary preceding the stream object, specifying its length, the appropriate decoding Filter (and the decoded length of the original data.

3) Upon receiving the text data, write the marker and commence encoding the text data using the LZW algorithm, writing to a new object stream until we encounter some kind of signal reset or a marker signaling the beginning of a new data type.

4) Write an *end stream* marker signaling the end of this encoded data (See Figure 3).

```

EXAMPLE.PDF
%Textual/Numeric Data Compressed using LZW encoding
1 0 OBJ
  <<    /Length 500
        /Filter [/ASCII85Decode /LZW Decode]
  >>
  STREAM
    J..)6T`?p&<!J9%_ [umg"B7/Z7KNXbN'S+, *Q/&"OLT'FLIDK#!n`$" <Atdi`¥V
    n%b%) &'cA*VnK¥CJY (sF>c!Jnl@RM]WM;jjH6Gnc75idkL5]+cPZKEBPwDR>FF(
    kj1_R%W_d&/jS!;iuad7h?[L-F$+]]0A3Ck*$I0KZ?;<)CJtqi65XbVc3¥n5ua:
    Q/=0$W<#N3U;H,MQKqfg1?:lUpR;6oN[C2E4Znr8Udn.'p+?#X+1>0Kuk$bCDF/
    (3fL5]Oq)^kJZ!C2H1'TO]Rl?Q:&'<5&iP!$Rq;BXRecDN[IJB`,)o8XJOSJ9sD
    S]hQ;Rj@!ND)bD_q&C¥g:inYC%)&u#:u,M6Bm%IY!Kb1+":aAa'S`ViJgllb8<W
    9k6Yl¥¥0McJQkDeLwdPN?9A'jX*al>iG1p&i;eVoK&juJHs9%;Xomop"5KatWRT
    "JQ#qYuL,JD?M$0QP)lKn06l1apKDC@¥qJ4B!!(5m+j.7F790m(Vj8818Q:_CZ(
    Gm1%X¥N1&u!FKHMB~>
  END STREAM
ENDOBJ
%Image compressed using JPEG compression
2 0 OBJ
  <<
    /Type Xobject
    /Subtype Image
    /Width 100
    /Height 100
    /Length 30000
    /Filter [/DCTDecode]
    /DecodeParms ColorTransform 1
  >>
  STREAM
    ..... encoded bytes for a 100x100 image, 3 components.....
  END STREAM
ENDOBJ

```

Figure 3: Example of an PDF document containing two object streams, one representing encoded textual and numeric data followed by a 100x100 JPEG encoded image. *Adapted from [5], pp. 68, 84-85

After we close this object stream, we can start writing the images in the data stream using the JPEG encoding standard. In Figure 3, the data that we write inside the stream for both the textual and image data is the compressed and transformed data we passed from the ground station through the space link. For images, in addition to simply writing the JPEG to the PDF file, we can write pertinent

elements included in the Exchangeable Image File Format (EXIF) data [12] to the PDF file as well.

The content stream consists of a sequence of well-defined graphics objects (glyph, images, etc), which, along with the document structure, determine the appearance of the PDF document. PDF documents keep this data separate from our

object data; sets of standard instructions about how to render different types of data can be specified at the ground level and written to the PDF document. Essentially, the content stream consists of a set of instructions that tell the PDF document viewer how to “paint” objects onto the pages [7]. This ensures a standardized rendering for multiple users viewing the same telemetry data. While the graphics objects and the document structure are related to the content stream, the latter also exists as a separate and distinct entity. Unlike the static, randomly-accessible references to text, images and other objects, the content stream must be read sequentially because it tells the viewer how to render the document. We essentially have two streams here: one stream contained our compressed data, which we write directly from our compressed data stream, and another detailing some kind of default way to render this data.

By directly writing to the PDF streams, we bypass the process of recovering the data from mass storage, decompressing it, and rendering for viewing, which can be

expensive both in terms of computation time and monetary cost. Instead, compressed data will be written directly to the PDF document, which will be responsible for all decompression, rendering, and viewing.

Because of the way a PDF file is organized, we can page through its contents to do a search much more efficiently than performing a linear keyword search as some current multimedia telemetry databases would require. Here we can make use of the document catalog to quickly narrow down the PDF file and only capture the relevant parts to search through for our desired query. Moreover, because every page is rendered independently, storing our data in PDF files will allow a user looking through the PDF files to jump to necessary pages in a non-sequential order. The data stream in the DFT frequency domain lends itself to more efficient searching than linearly searching through the uncompressed data in the time domain. Figure 4 demonstrates how much simpler our algorithm makes the telemetry data archival and retrieval process.

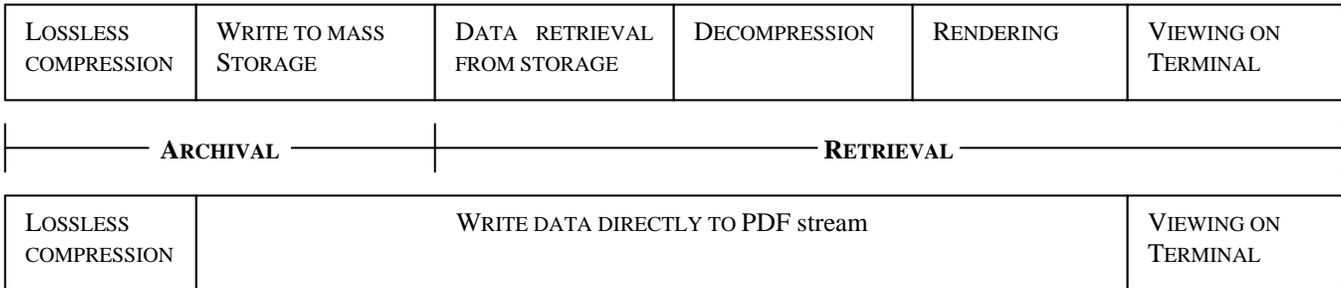


Figure 4: The telemetry data archival and retrieval process, first with current implementations, and below, with our implementation, which reduces the complexity of the data retrieval process greatly.

6. CONCLUSIONS

We have presented an algorithm that uses multiple compression streams to effectively compress multidimensional telemetry data. We adhere to the Adobe PDF standard to leverage off its rendering standards and built-in streams.

Our algorithm still needs to be tested for scalability on the level of thousands or millions of images and gigabytes and terabytes of textual and numerical data. The Adobe PDF format has not been specifically designed for storing documents on such a large scale, so adjustments may have to be made in the future to deal with any scalability issues that may arise.

Eventually, our algorithm will be able to support a database of PDF documents. Further work will look into improving the ability to query within these documents besides simply implementing the keyword searches already built into most PDF document viewers. One possible strategy would be to design our own custom PDF document reader, which can

search through the data within the DFT domain, which would enable much more efficient querying, novelty detection, and pattern matching. We could build an index of PDF documents, taking advantage of the document structure, writing any metadata important for searching later on as interchange information within the PDF file in addition to the actual textual and graphical data.

Testing must still be done to test for the scalability of PDF documents on the terabytes and gigabytes level. Moreover, our algorithm does not specify any method of error-checking during the encoding and transmission process, but this feature could be implemented in the future. Another feature that could be added in the future would be error recovery within our PDF storage system, which is a crucial part of maintaining a database. Another thing to consider in the future would be how to partition the storage system disk containing all the PDF documents once we start to get on the order of hundreds of gigabytes and terabytes of telemetry data.

Replacing existing ground-level mass storage systems with the PDF files resulting from this algorithm would reduce

costs greatly. Current telemetry data storage implementations use traditional database storage; the cost of this type of storage grows quickly as the amount of data we need to store increases. Moreover, within the old

framework, different rendering and viewing programs must be designed for each individual project, whereas within our algorithm, any standard PDF viewer will work.

REFERENCES

- [1] Sayood, Khalid, *Lossless Compression Handbook*, Elsevier Science, San Diego, CA, 2003.
- [2] Horan, Stephen, "Introduction to PCM Telemetering Systems", Second Edition. CRC Press, 2002.
- [3] Gray, R. M. "Fundamentals of Data Compression", International Conference on Information, Communications, and Signal Processing, Singapore, September 1997. IEEE Publication, New York.
- [4] Rao, K.R. and Yip, P.C., *The Transform and Data Compression Handbook*, CRC Press, 2001.
- [5] *PDF Reference: Adobe Portable Document Format Version 1.7*, Sixth Edition, Adobe Systems Incorporated, 2006.
- [6] Staudinger, P., et al., *Lossless Compression for archiving satellite telemetry data*, Aerospace Conference Proceedings, 2000 IEEE. Volume 2, 18-25 March 2000, pp. 299 – 304.
- [7] "CGPDFContentStream Reference", Apple. <<http://developer.apple.com/documentation/GraphicsImaging/Reference/CGPDFContentStream/Reference/reference.html>>.
- [8] *Lossless Data Compression*, Report Concerning Space Data Systems Standards, CCSDS 120.0-G-2. Green Book. Issue 2. Washington, D.C.: CCSDS, December 2006.
- [9] *Space Packet Protocol*, Recommendation for Space Data Systems Standards, CCSDS 133.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [10] Gailly, Jean-loup and Adler, Mark, "The gzip home page", 27 July 2003. <<http://www.gzip.org>>.
- [11] "How to compress your PDF files? Compression arithmetic for PDF files", VeryPDF.com, Inc. <<http://www.verypdf.com/pdfinfoeditor/compression.htm>>.
- [12] *Exchangeable image file format for digital still cameras: Exif Version 2.2*, Standard of Japan

Electronics and Information Technology Industries Association, April 2002.

- [13] Bourke, Paul, "Fast Fourier Transform", Western Australian Supercomputer Program, University of Western Australia, June 1993. <<http://local.wasp.uwa.edu.au/~pbourke/other/dft/>>.
- [14] Weisstein, Eric W. "Nyquist Frequency." From Mathworld—A Wolfram Web Resource. <<http://mathworld.wolfram.com/NyquistFrequency.html>>.

BIOGRAPHIES

David A. Maluf received his Ph.D. from McGill University in 1995 and his postdoctoral from Stanford University. He has been involved in Intelligent Information Integration and databases since. David was also Director of Software Development at Incyte. Before NASA, David founded and operated Science Gate as CTO. The company was successfully acquired. At NASA, David was the Project Manager for Knowledge Engineering under the Engineering for Complex Systems program. David was the CIO for the program. In conjunction with the FAA, David has been leading, from its inception, the development and operation of very large government information grid projects, connecting US government centers nation wide. David is the inventor on many NASA patents, including Netmark tool suites, which were commercialized leading to products such as NX and PMT. David is the recipient of many NASA Awards: Best Technology Commercialization, Turning Goals into Reality, and Space Act Awards.

Peter B. Tran is currently a Senior IT Software Architect for NASA Ames Research Center working on data integration and information management projects for the NASA's Constellation Program. Previously, Peter worked as a software consultant, architect, technical lead, and software engineer at several technology companies, including QSS Group, Inc., BEA Systems, XUMA, Computer Sciences Corporation, and Recom Technologies. Peter has a degree in electrical engineering and computer sciences from the University of California at Davis, and has taken graduate-level coursework at Stanford University majoring in Computer Science.

David Tran is currently an undergraduate majoring in Computer Science at Stanford University. He interned at NASA Ames Research Center during the summer of 2007.