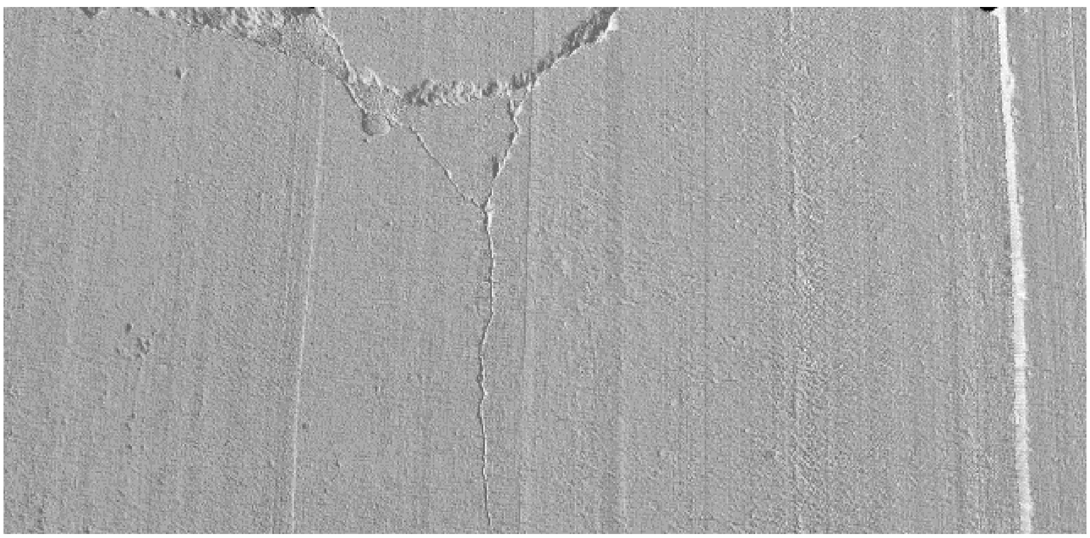
U.S. Department of Transportation

Federal Highway Administration

Development of Standard Data Format for 2-Dimensional and 3-Dimensional (2D/3D) Pavement Image Data used to determine Pavement Surface Condition and Profiles

Task 4 - Develop Metadata and Proposed Standards



Office of Technical Services
FHWA Resource Center
Pavement & Materials
Technical Services Team

December 2016

Notice

Quality Assurance Statement

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Development of Standard Data Format for 2-Dimensional and 3-Dimensional (2D/3D) Pavement Image Data that is used to determine Pavement Surface Condition and Profiles		**5. Report Date** 12-20-2016 **6. Performing Organization Code**
7. Author(s) **Wang Kelvin C. P., Qiang "Joshua" Li, and Cheng Chen**		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Oklahoma State University (OSU) 207 Engineering South Stillwater OK 74078		**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** **DTFH6115P00103**
12. Sponsoring Agency Name and Address Federal Highway Administration Office of Technical Services - FHWA Resource Center Howard Street, Suite 4000 Baltimore, MD 21201		**13. Type of Report and Period Covered** **14. Sponsoring Agency Code**
15. Supplementary Notes Mergenmeier, Andrew, FHWA Contract COR		

16. Abstract

This project was funded by the Transportation Pooled Fund study, TPF-5(299). This report documents the work performed by the research team for Task 4 of this project. Based on a comprehensive literature review regarding common pavement image, condition, and distress data formats, a survey of the current practices of the highway agencies, data collection vendors, and technology suppliers (Task 2), and the assessment of existing data items collected and data formats of pavement image data (Task 3), a draft standard data format to determine pavement surface condition and profiles is developed in this Task to meet transportation agencies' different data requirements. As efficient compression algorithms are key components for the proposed data format standard, this report also evaluates five compression methods recommended by several State highway agencies and industry vendors, namely JPEG2000, JPEG XT, JPEG XR, 16-bit TIFF, and the customized compression method by the team, on their suitability for compressing 16-bit 3D pavement images. In addition, desired new developments are recommended for further research to facilitate the adoption of the proposed file data format.

17. Key Words	18. Distribution Statement
Standard Data Format; 2D/3D; Pavement Image Data	No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	27	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS				
SYMBOL	**WHEN YOU KNOW**	**MULTIPLY BY**	**TO FIND**	**SYMBOL**
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS				
SYMBOL	**WHEN YOU KNOW**	**MULTIPLY BY**	**TO FIND**	**SYMBOL**
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

TABLE OF CONTENTS

Subject *Page*

TABLE OF CONTENTS

Subject *Page*

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This report documents the work performed by the research team for Task 4 of the project. Based on a comprehensive literature review regarding common pavement image, condition, distress data formats, and a survey and review of the current practices of the participating highway agencies of the Transportation Pooled-Fund Study TPF-5(299), data collection vendors, and technology suppliers (Task 2), and the assessment of existing data items collected and data formats for pavement image data (Task 3), a draft standard data format to determine pavement surface condition and profiles is developed in this Task (Task 4) to meet transportation agencies' different data requirements.

In addition, considering pavement images occupy large amounts of storage space, and demand efficient compression algorithms to store and transmit them, useful compression algorithms become one of the key components for the proposed data format standard, image archiving and data management. For 2D 8-bit intensity image data, JPEG or JPEG 2000 standard is exclusively used by highway agencies and industry vendors. For 3D range data, 16-bit depth data is commonly used because 8-bit depth dynamic range may not be adequate for pavements with various features and under different conditions. However, the commonly used compression algorithms, such as GIF, JPEG, and PNG, cannot be directly used for the compression of 16-bit single channel data. In recent years, various 16-bit based compression algorithms have been developed for different applications other than pavement engineering. Therefore, it is important to review and evaluate those algorithms for compressing 3D pavement images.

Particularly, the report is organized as follows:

- Chapter 1 provides a brief introduction of this report;
- Chapter 2 presents the proposed pavement image metadata/data format to determine pavement surface condition and profiles;
- Chapter 3 evaluates five compression methods recommended by several State highway agencies and industry vendors, namely JPEG2000, JPEG XT, JPEG XR, 16-bit TIFF, and the customized compression method by the team, on their suitability for compressing 16-bit depth pavement images;
- Chapter 4 recommends guidelines of desired future developments to facilitate the adoption of the proposed file data format; and,
- The draft metadata/data format to determine pavement surface condition and profiles is prepared in accordance with the AASHTO standard format, which is attached in Appendix A of this report.

2. PAVEMENT IMAGE METADATA FORMAT AND STANDARD

Data Format Requirements

The requirements for the standard pavement image data format have been developed based on input of both highway agencies and equipment suppliers, but also on the experience from other industries. In order to develop a widely accepted open standard for both highway agencies and equipment suppliers, the following challenges for pavement image data should be properly addressed:

- A desired data format shall be suitable for storing 2D data, 3D data, or hybrid 2D/3D image data. Considering that the dynamic range of 3D data is generally larger than that of 2D data, existing 2D data formats are not suitable for 3D data storage.
- The desired data format shall be suitable for efficient compression for data storage. Without compression, one lane-mile of pavement (4-meter in width) data needs require well over 10 GB storage space at 1mm resolution for both 2D and 3D.
- Considering the fact that several vendor-specific proprietary data compression methods have been used for some time, the recommended data format does not exclude the usage of the proprietary data compression methods, if the vendors would provide the decoding software so that users can take advantage of these proprietary data compression methods, while the proprietary information can be protected.

Based on review and assessment results in Task 2 and Task 3, the following requirements are developed to guide the design of the new standard data format for pavement image data:

- Simplicity – A single unified data format should be developed for both 2D image data and 3D range data;
- Interoperability – The data sets from any vendor who outputs pavement data into the standard format can be read and viewed with a single set of standard software writer/reader components;
- Speed – The data format should be designed to store and retrieve pavement data in an efficient way so that both encoding and decoding of the data are conducted with minimum delay with a mid-range computer system;
- Compatibility – The data format does not depend on any special operation system and programming languages; it should be compatible with all types of computers, operating systems, and programming languages;
- Low barrier for adoption – The developed format is not based on any patented technologies. The cost of adopting the new data format by vendors and users is kept to a minimum;
- Extensibility – The developed data format has a flexible format so that new capabilities and technologies can be integrated in the future without substantially revising the format.

File Structure

Pavement imaging technology will continue to evolve with new capabilities. Therefore, it is important for the pavement image file format to support both backward and forward compatibilities of the related software, for instance, a pavement image viewer.

- Backward compatibility – Ability of a software system, such as a pavement image viewer, to read and display a file created with an older generation of technology, such as an area-scan digital system, as long as the format is per the new standard.
- Forward compatibility – Ability of a software system to read a file that conforms to the new format but with higher specification than currently available. For instance, 3D pavement data at 0.25mm resolution.

To achieve backward and forward compatibilities, pavement 2D/3D data file shall include two parts. The first part is the core section including File Header, 2D Image Data, and 3D Range Data. The data format definitions of the first part shall not be changed as the data format standard evolves. The second part primarily contains the optional user defined metadata section, which can be extended and modified as needed by vendors or highway agencies to accommodate individual data collection practices and equipment. The offsets in the file header serve as the guides to locate various portions of a file. All offsets are relative to the beginning of the file. The offset values can be obtained only after 2D/3D data are compressed. The four sections are stored sequentially (Figure 2.1). Either storage space for 2D data or 3D data is variable depending on the compressed data size. Each of these portions in the file is described in the following sections. The data types and related descriptors required in the file are included.

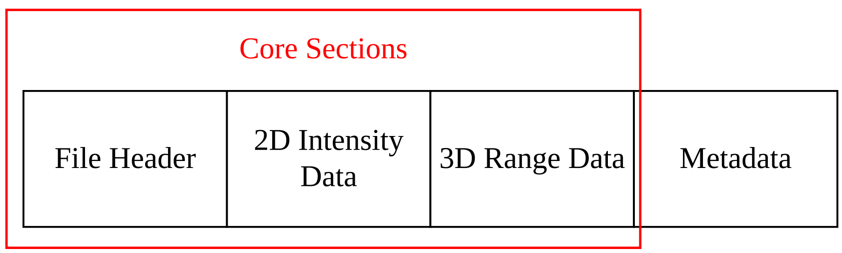


Figure 2.1 Layout of the File Structure

File Header

The file header describes the core properties of the 2D/3D data stored in the file. Each property shall be denoted by a variable and each variable shall have a predefined data type with fixed byte length. The header of a pavement image file shall be encoded in binary format. An example of file header is shown in Table 2.1. These properties can be classified into two categories:

- Required variables for data reading, which include file format version, image size, resolution, bit depth, compression algorithm name, and storage offset that are critical for successfully reading the 2D/3D data matrices for the purposes of data analysis or visualization.
- Required variables for fast archiving and retrieving, which include location and time data that are critical for global database-based pavement data management and local section-based pavement data view.

Table 2.1 File Header

Variable Name	Data Type	Data Details
Version	4-byte String	Version number of the file format
SW version	8-byte String	Identifier of the software that produced the file
State Name	2-byte String	FIPS State Code
Route Name	12-byte String	Name of the highway HPMS standard
Direction	2-byte String	Direction of travel
Lane identification	2-byte String	Lane index
File Serial Number	Int32	File serial number in continuous data collection
GPS Longitude	Float 32bit	GPS longitude value IEEE 754 binary32
GPS Latitude	Float 32bit	GPS latitude value: IEEE 754 binary32
DMI Pulse	Int32	DMI pulse counting index
Date	8-byte String	Date data was collected—(yyyymmdd)
Time	6-byte String	Time data was collected—(hhmmss)
Event Mark ID	Int32	Event marker(s) by data collection crew
2D Compression Method	4-byte String	Identifies compression algorithms, such as 1: PNG; 2: JPEG; 3:JPEG2000.
2D Resolution Longitude Direction	Float 32bit mm	Distance between two data rows in longitude direction in millimeters.
2D Resolution Transverse Direction	Float 32bit mm	Distance between two data columns in transverse direction in millimeters
2D Width	Int32	Pixel numbers in transverse direction
2D length	Int32	Pixel numbers in longitude direction
2D Data Bit Depth	Int32	The bit depth for each data point
2D Data Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the 2D data
2D Compression Quality	Float 32bit	Compression quality level
3D Compression Method	8 byte String	Identifies compression algorithms, such as 1: PNG; 2: JPEG; 3:JPEG2000
3D Resolution Longitude Direction	Float 32bit mm	Distance between two data rows in longitude direction in millimeters
3D Resolution Transverse Direction	Float 32bit mm	Distance between two data columns in transverse direction in millimeters
3D Resolution Elevation Direction	Float 32bit mm	Units for range data value in millimeters
3D Width	Int32	Pixel numbers in transverse direction
3D length	Int32	Pixel numbers in longitude direction
3D Data Bit Depth	Int32	The bit depth for each data point
3D Data Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the 3D data
3D Compression Quality	Index Float 32bit	Compression quality level
Metadata Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the metadata
Speed	Float 32bit mm/s	Average vehicle speed associated with data
Time stamp	Long long int	Milliseconds since UNIX Epoch: Jan 1, 1970 00:00:00
Vehicle Number	8 byte String	Vehicle identification
Operator Name	32 byte string	Operator identification
Reserved Item1	16 byte string	Reserved for future usage or additional vendor specific information
Reserved Item2	Int32	Reserved for future usage or additional vendor specific information
Reserved Item3	Float 32bit	Reserved for future usage or additional vendor specific information
Reserved Item4	8 byte String	Reserved for future usage or additional vendor specific information

Data Sections

The data sections store blocks of binary data, which are generated by compression algorithms. Vendors may use currently accessible compression algorithms or customized proprietary algorithms for data compression. The algorithm used in the proprietary software should be provided in the form of either dynamic-link library (DLL) format or source codes.

More detailed comparisons and evaluation of various compression algorithms for 8-bit and 16-bit image data will be discussed in the subsequent section.

User defined Metadata Section

In the data format standard, the first value in the metadata portion shall provide the number of metadata entries (MDE) as shown in Table 2.2. Table 2.3 shows the partial list of information to construct a MDE. The specific metadata data entries and their tags shall be designed based upon the future or specific needs of users.

Table 2.2 Metadata Example

Variable Name	Data Type	Data
Number of MDEs	Int32	Number of MDEs

Table 2.3 Metadata Entries (Partial Listing)

Variable Name	Data Type	Data
Array size	Int32	"-1" if not an array. "0" if array is empty. Numbers greater than 0 specify the number of elements in the array
Count	Int32	For data types "String" and "Array (String)", count = the number of bytes in the string. For other data types, count = 1.
Data type of MDE	Int32	Data type index of MDE
MDE	varies	Information associated with tag of MDE
Name	String	Name of the metadata
Name length	Int32	For metadata entries listed in Table 5, this is 0. For user-defined entries, this value is the length of the name.
Tag of MDE	Int32	Metadata tag

Draft Data Format Standard

The draft metadata/data format to determine pavement surface condition and profiles has been prepared in accordance with the AASHTO standard format, which is attached in Appendix A of this report.

3. EVALUATION OF COMPRESSION ALGORITHMS

Introduction

In Task 2, various 2D/3D data formats have been reviewed. Among them, 2D/3D data is stored in either compressed or un-compressed forms. Considering pavement images occupy large amounts of storage space, efficient compression algorithms become one of the key components for the proposed data format standard.

For 2D images, TIFF, PNG, JPEG and JPEG2000 are commonly used compressed data formats. According to the survey results, the commonly used JPEG should be used for 2D pavement intensity images compression because of its efficiency and compatibility to various file/web browsers. There is no need to further investigate available compression methods for 2D pavement intensity images. This evaluation work only focuses on 16-bit depth 3D range data compression.

There are two types of compression methods: lossy and lossless. Lossy compression creates smaller files by discarding (losing) some information from the original image. Lossless compression does not discard any information from the original file, but it generates larger file sizes. Among numerous compression methods, candidate compression methods should be selected for the evaluation according to two criteria: (1) capable to handle 16-bit depth data and (2) available for public access. After surveying several State highway agencies and industry vendors, JPEG2000, JPEG XT, JPEG XR, 16-bit TIFF, and the customized compression method by the team are selected for evaluation. The GeoTIFF is recommended by one vendor, but is not selected for evaluation because it is an extension of TIFF and uses the same compression method as TIFF (Ritter 2000).

Existing Compression Algorithms

<u>JPEG2000</u>

The JPEG standard was approved a quarter of a century ago. JPEG 2000 was developed as an important successor for JPEG. The aim of JPEG 2000 is not only improving compression performance over JPEG but also adding (or improving) features such as scalability and editability. The JPEG 2000 offers numerous advantages over the JPEG standard. One main advantage is that JPEG 2000 offers both lossy and lossless compression in the same file stream, while JPEG usually only utilizes lossy compression. In addition, several well developed libraries are available online for JPEG2000. However, JPEG 2000 is many times slower than JPEG computationally and therefore not a good candidate to replace JPEG in some areas, but it is capable to deal with 16-bit data. It should be noted that the JPEG2000 and JPEG are not compatible.

<u>JPEG XT</u>

The JPEG format fails to deal with higher bit depths (9 to 16 bits), high-dynamic-range imaging, and lossless compression. To overcome the limitations, the JPEG Committee is developing a new coding standard called JPEG XT that is backward compatible to the JPEG compression. The JPEG committee has carried out a large number of experiments, using both subjective and objective methodologies, to assess the capability of the JPEG XT (Richter 2016). Three profiles A, B, and C are defined in JPEG XT Part 7 for floating point coding. To implement this compression method for 3D pavement images, only profile C is used because it allows implementations operating entirely with integers until the final step of the compression where the exponential generates floating-point output.

JPEG XR

As a JPEG alternative, Microsoft released JPEG-XR in 2006 (ITU 2012). JPEG-XR can produce smaller files than JPEG and is designed to handle 16-bit data. The only browser to support JPEG-XR thus far is the Microsoft's Internet Explorer. Online source code is available at http://jpeg.org/jpegxr/index.html, which has not been updated for years, and no technical support is currently available. The source code has been tested by the Oklahoma State University (OSU) team without any success in compiling, whose error report is shown in Figure 3.1. Therefore, this algorithm is excluded for the evaluation as it is not used anymore in the computer industry for practical applications.

Error List

7 Errors | 102 Warnings | 0 Messages | Search Error List

	Description	File	Line	Column	Project
3	error C2065: 'my' : undeclared identifier	w_strip.c	1256	1	DLL
96	error C2065: 'save_count' : undeclared identifier	r_parse.c	657	1	DLL
98	error C2440: 'return' : cannot convert from 'int' to 'jxrc_t_pixelFormat'	cr_parse.c	744	1	DLL
102	error C2440: 'return' : cannot convert from 'int' to 'jxr_output_clr_fmt_t'	api.c	456	1	DLL
103	error C2440: '=' : cannot convert from 'void *' to 'unsigned int *'	api.c	726	1	DLL
104	error C2440: '=' : cannot convert from 'void *' to 'unsigned int *'	api.c	755	1	DLL
106	error C1083: Cannot open include file: 'unistd.h': No such file or directory	qp_lexor.c	612	1	APP

Figure 3.1 Compiling Error of the JPEG-XR Source Code

16-bit TIFF

The TIFF (Tag Image File Format) is one of the most common graphic formats for exchanging raster graphics images between application programs. This lossless compression method is capable to compress 16-bit data. Moreover, several free and/or open source tools are available, such as ImageMagick (ImageMagick 2016), GraphicsMagick (GraphicsMagick 2016) and NetPBM (NetPBM 2016).

Propriety Compression Method by the OSU Team

According to the survey results, several venders have developed their proprietary 3D data compression algorithms. The proprietary algorithms are used largely to protect intellectual properties of the vendors and to address the needs of the unique characteristics of pavement image data.

Since these proprietary compression algorithms are not publicly accessible, in this study only two compression methods (lossy and lossless) developed by the OSU team are evaluated for demonstration purposes. The first OSU method is a lossy method (OSU Method 1), which has been specially designed for pavement image data (Zhang and Wang 2016). The second OSU method (OSU Method 2) is based on the same strategy as the first approach, which divides the 16-bit data into two 8-bit (the high 8-bit and the low 8-bit) data chunks. Considering the different dynamic ranges of the two data chunks as most detailed pavement data are included in the low-8-bit data chunk, the high 8-bit data is compressed by GIF (GIF 2016) compression method and the low 8-bit data is compressed by PNG (PNG 2016) compression method. The OSU methods are then compared with the general purpose compression algorithms, such as JPEG2000, JPEG XT, and 16-bit TIFF herein.

Comparisons and Evaluation of the Compression Algorithms

Compression Quality Metric

Compression efficiency for lossy compression is typically measured using Peak Signal-to-Noise Ratio (PSNR), the ratio between the maximum possible power of a signal and the power of corrupted noise that affects the fidelity of its representation. Because many signals including 3D pavement data have a wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. A higher PSNR indicates that the compression method yields higher-quality data reconstruction. The formula for PSNR, in dB, is shown in Equation 3.1, where the peak value b, is taken from the range of the image data type (e.g. b is 8 for uint8 images). The MSE is the mean square error between the original and compressed images (Matlab 2016).

$$PSNR(dB) = -20\ \log(\frac{MSE}{2^b-1}) \quad (3.1)$$

Besides PSNR being a compression efficiency measure, compression ratio and speed are also widely used to evaluate an image compression algorithm. Compression methods with low compression ratio (which leads to large file sizes) and low processing speed may not be suitable for pavement data collection.

For JPEG and JPEG XT, the degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. A set of quantization matrices indexed by a quality factor from the set {1, 2, . . . , 100} is commonly used to adjust the degree of compression (Pandit 2013) for lossy compression. This quality factor is not available for the 16-bit TIFF and the propriety compression methods developed by the OSU team. The JPEG 2000 does not provide a quality factor, instead, the degree of compression can be indirectly adjusted by setting up a pre-defined compression ratio.

Testing Environments and Images

The evaluation of the compression algorithms is conducted on a notebook computer with i7-4810MQ CPU and 16G RAM. The sources of the compression packages are provided in Table 3.1.

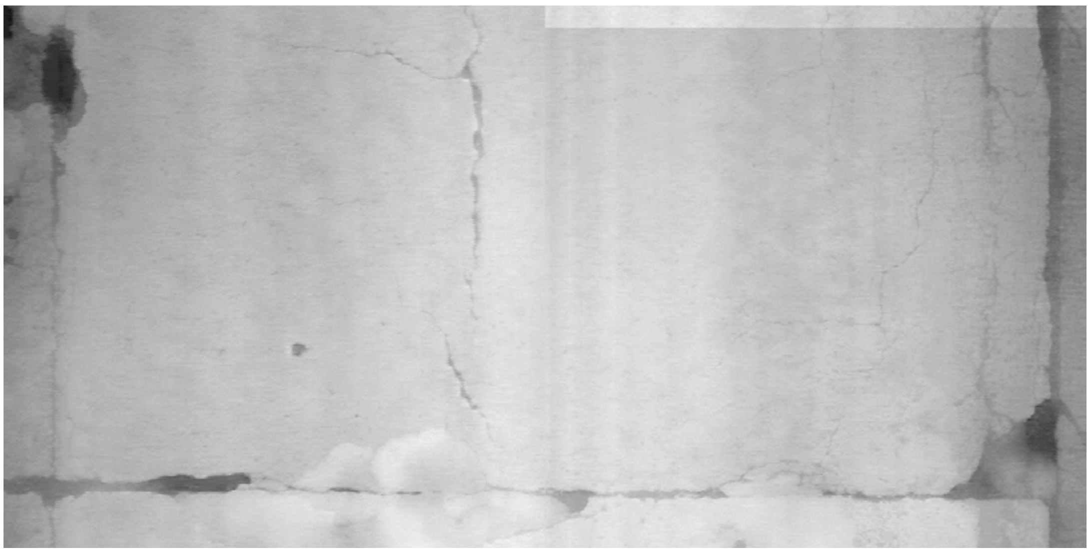
Table 3.1 Source Codes

Method	Source Codes
JPEG2000	https://github.com/uclouvain/openjpeg
JPEG XT	https://jpeg.org/jpegxt/software.html
JPEG XR	https://jpeg.org/jpegxr/software.html- www.itscj.ipsj.or.jp/sc29/open/29view/29n10430c.htm
Lossless 16-bit TIFF	ftp://ftp.remotesensing.org/pub/libtiff
Propriety method	Developed by OSU team (for demo purpose) (Zhang and Wang 2016)

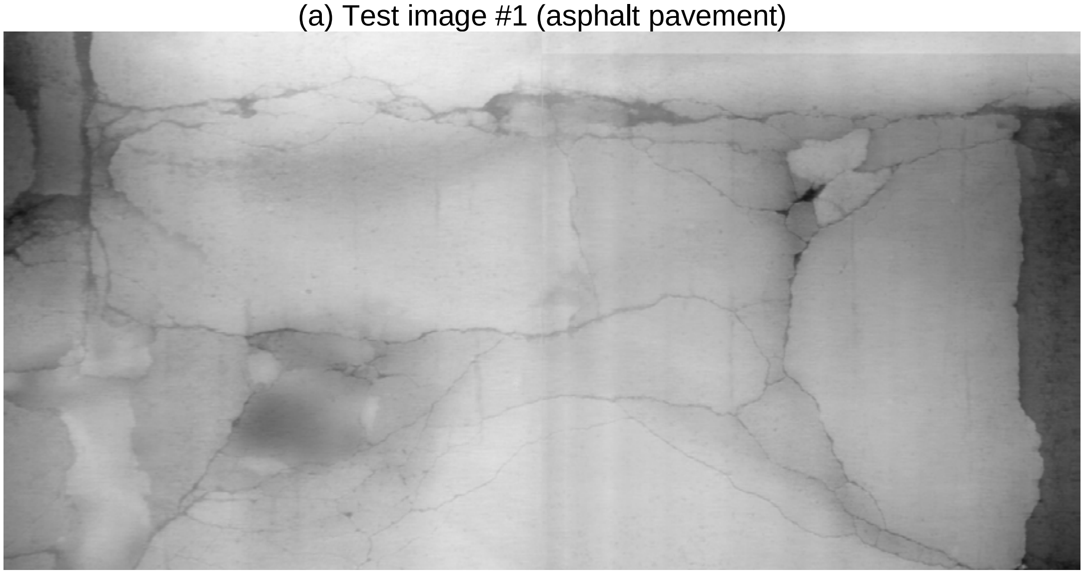
Four 3D 16-bit range images: two for asphalt and two for concrete pavements, are selected for the evaluation of the compression algorithm performance. All images have the dimension of 4,096mm in width and 2,048mm in length with 1mm resolution as shown in Figure 3.2. In order to view these images in Windows, the images are normalized to 0 to 255 scale (8-bit) as 16-bit data cannot be properly displayed, and saved into bmp format for visualization purpose. Table 3.2 shows the storage sizes of the four raw uncompressed image data.

Table 3.2 Test Image Data

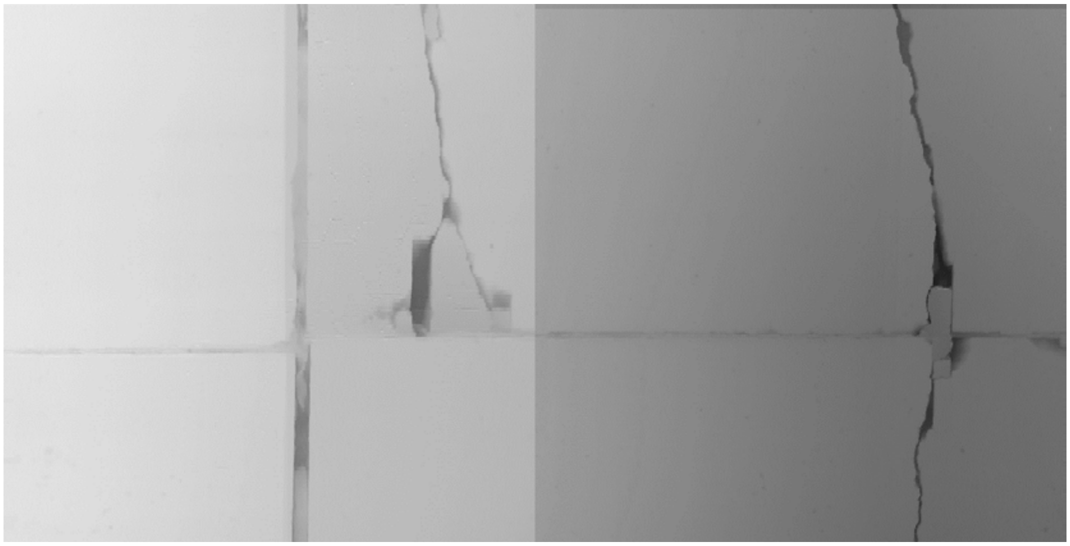
Image	Data Size (kb)	Width	Height	Pavement Surface	Road Name
1	32770	4096	2048	AC	US-51 EB, Stillwater OK
2	32770	4096	2048	AC	US-51 EB, Stillwater OK
3	32413	4052	2048	PCC	I-44 EB, Oklahoma City
4	32414	4052	2048	PCC	I-44 EB, Oklahoma City



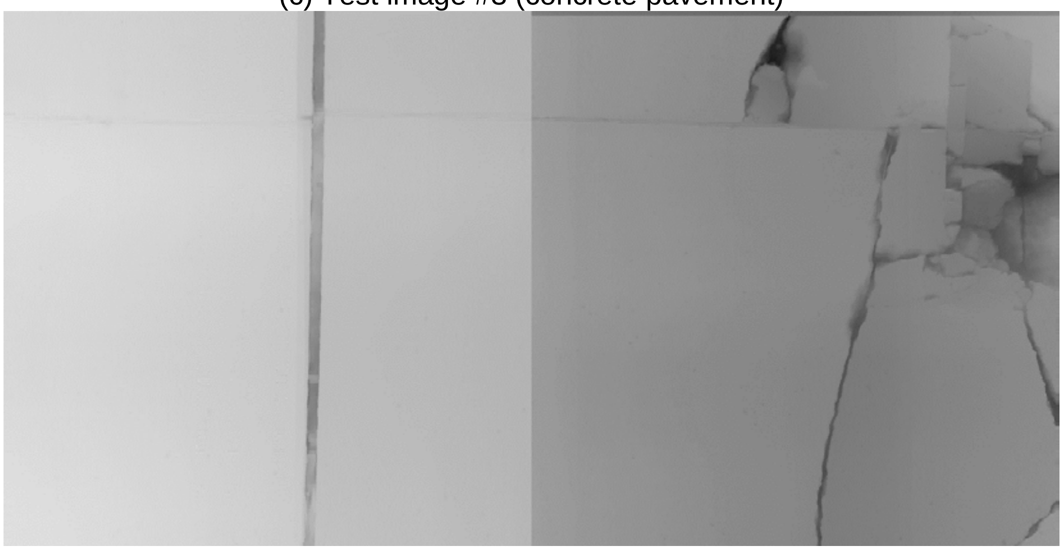
(a) Test image #1 (asphalt pavement)



(b) Test image #2 (asphalt pavement)



(c) Test image #3 (concrete pavement)



(d) Test image #4 (concrete pavement)
Figure 3.2 3D Range Testing Images (Normalized to 0~255)
(From WayLink PaveVision3D)

Evaluation Results

Among the four testing algorithms, TIFF is lossless, the OSU methods are proprietorial, and JPEG XT and JPEG2000 are lossy algorithms. The JPEG XT allows users to set different quality factors, while JPEG 2000 does not provide such option. In order to make a fair comparison between JPEG XT and JPEG 2000, the same compression ratio is used for each comparison. Four compression quality factors, 100, 95, 90, and 85, are studied and compared in the report.

Tables 3.3, 3.4 and 3.5 show the compression results of the four testing images for the 16-bit TIFF, the OSU Method 1(lossy) and the OSU Method 2 (lossless) respectively, while Table 3.6 summarizes the compression performance for JPEG XT and JPEG 2000 at four quality factors at 100, 95, 90, and 85. The following observations are obtained based on these comparisons:

- At the same compression ratio, except for test image #1, JPEG XT achieves better PSNR scores than JPEG 2000, and is more than ten times faster than JPEG 2000.
- At the similar compression ratio and PSNR score, the OSU Method 1 (lossy) is about ten times faster than JPEG XT and 16-bit TIFF. Moreover, while keeping a high PSNR score, the OSU Method 1 generates eight times smaller file size than the 16-bit TIFF. These data validate the need and feasibility of developing highly efficient custom compression methods for pavement image data, which is desired by several survey respondents as presented in the Task 2 report.
- For lossless compression, the compression rates of OSU Method 2 are about three times better than those of the 16-bit TIFF method on the four testing images. The experimental results demonstrate the efficiency of the strategy: dividing the 16-bit data into two 8-bit data chunks. However, the OSU Method 2 is about seven times slower than the 16-bit TIFF method, primarily due to the fact that the OSU Method 2 was specifically developed for the testing in the project and has not been optimized for efficiency and parallelism. Improvement of its computational efficiency is beyond the scope of this project. It is anticipated that at minimum improvement of a factor of 10 would be made in compression speed for the OSU Method 2.
- For both JPEG XT and JEPG 2000, the data sizes are approximately five times smaller if the compression quality factor decreases from 100 to 85, while suffering a 5 percent reduction in image quality performance. For instance, for test image #1, the compression ratio increases from 24:1 to 107:1 for JPEG XT compression, while the PSNR score is reduced by 5 percent from 93 to 89 if the compression quality factor changes from 100 to 85.

Table 3.3 Compression Results: 16-bit TIFF Algorithm

Image No.	Time (ms)	File Size after Compression (kb)	Compression Ratio	PSNR after Decoding
1	304	5268	6.22 : 1	inf.
2	298	5049	6.49 : 1	inf.
3	245	6834	4.74 : 1	inf.
4	239	6491	4.99 : 1	inf.

Table 3.4 Compression Results: the OSU Method 1 (lossy)

Image No.	Time (ms)	File Size after Compression (kb)	Compression Ratio	PSNR after Decoding
1	33	602	54.44 : 1	90.67
2	31	603	54.34 : 1	90.72
3	34	748	43.33 : 1	92.85
4	33	743	43.63 : 1	92.57

Table 3.5 Compression Results: the OSU Method 2 (lossless)

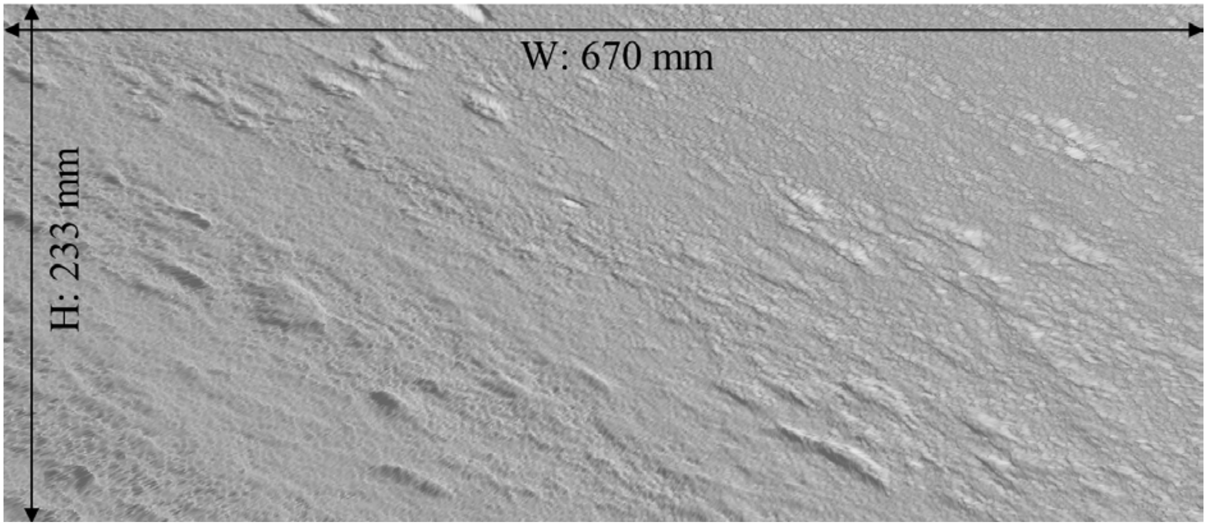
Image No.	Time (ms)	File Size after Compression (kb)	Compression Ratio	PSNR after Decoding
1	2292	2238	14.64 : 1	Inf.
2	2247	2325	14.09 : 1	Inf.
3	1175	2448	13.24 : 1	Inf.
4	1703	2513	12.90 : 1	Inf.

Table 3.6 Compression Results: JPEG XT and JPEG 2000 at Four Quality Factors

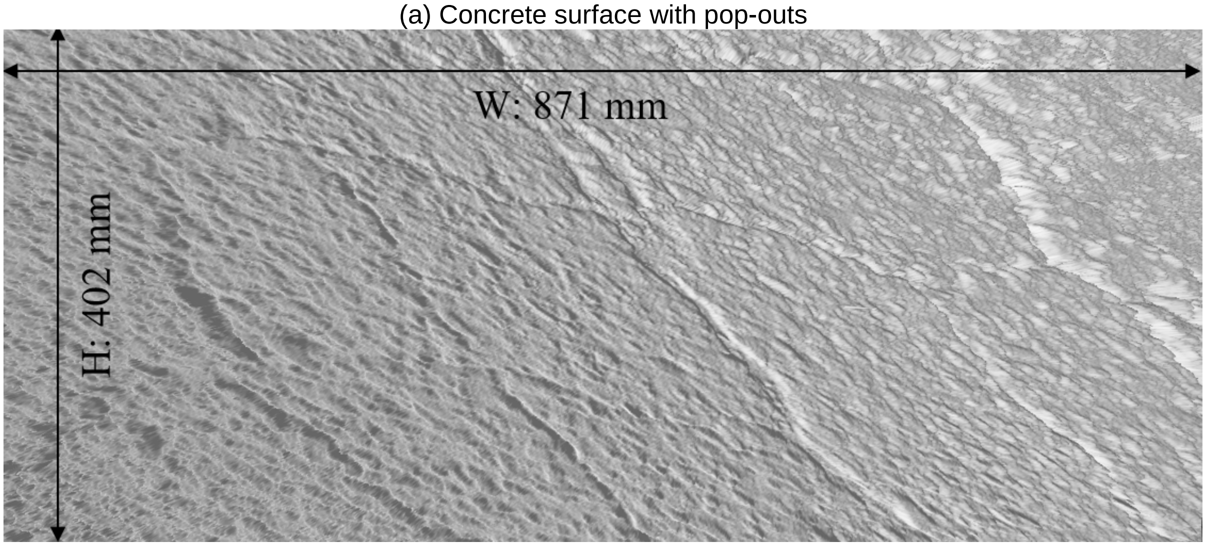
Quality factor	Method	Image No.	Time (ms)	File size after Compression (kb)	Compression Ratio	PSNR after Decoding
100	JPEG XT (Profile C)	1	336	1375	23.83 : 1	93.36
		2	343	1458	22.48 : 1	93.84
		3	305	771	42.04 : 1	86.72
		4	297	870	37.26 : 1	87.00
	JPEG 2000	1	4608	1375	23.83 : 1	101.41
		2	4742	1459	23.46 : 1	63.94
		3	4720	772	41.99 : 1	50.73
		4	4603	871	37.21 : 1	53.11
95	JPEG XT (Profile C)	1	287	710	46.15 : 1	91.19
		2	291	767	42.72 : 1	91.64
		3	259	382	84.85 : 1	86.03
		4	254	399	81.24 : 1	86.21
	JPEG 2000	1	4547	711	46.09 : 1	96.95
		2	4755	768	42.67 : 1	63.94
		3	4789	383	84.63 : 1	50.73
		4	4653	400	81.04 : 1	53.11
90	JPEG XT (Profile C)	1	262	411	79.73 : 1	89.81
		2	273	459	71.39 : 1	90.25
		3	247	264	122.78 : 1	85.55
		4	250	269	120.50 : 1	85.86
	JPEG2000	1	4470	412	79.54 : 1	94.01
		2	4661	460	71.24 : 1	63.94
		3	4639	269	120.49 : 1	50.76
		4	4666	270	120.05 : 1	53.11
85	JPEG XT (Profile C)	1	261	313	104.70 : 1	89.08
		2	259	352	92.08 : 1	89.47
		3	249	235	139.45 : 1	85.26
		4	263	236	137.35 : 1	85.63
	JPEG 2000	1	4423	314	104.36 : 1	91.03
		2	4506	353	92.83 : 1	63.94
		3	4663	236	137.34 : 1	50.74
		4	4607	237	136.77 : 1	53.11

In addition, visual evaluations have also been conducted on the lossy compression methods. The evaluation provides intuitive visual comparisons of the images at different compression quality factors. Two zoomed-in images (Figure 3.3) are tested and the JPEG XT results are shown in Figure 3.4. It can be observed that when the compression quality factor decreases, texture features and detailed information on the pavement surfaces have been washed out, and the hairline cracks are fading gradually or totally disappear on the images with high compression ratios, even though there is only a 5 percent deduction in the image quality factor (Table 3.6).

Therefore, a proper lossy quality factor should be selected with caution for different applications. For example, for pavement cracking analysis, compressed image data at a lower quality factor (such as 85) may be satisfactory. However, for pavement texture analysis, significant amount of surface texture may be lost on the compressed images, and thus image data should be compressed at a higher quality factor (such as 95 or 100) or even lossless approaches are desired. Further research is recommended on how to select proper compression quality factors for different pavement image applications, which is beyond the scope of this project.

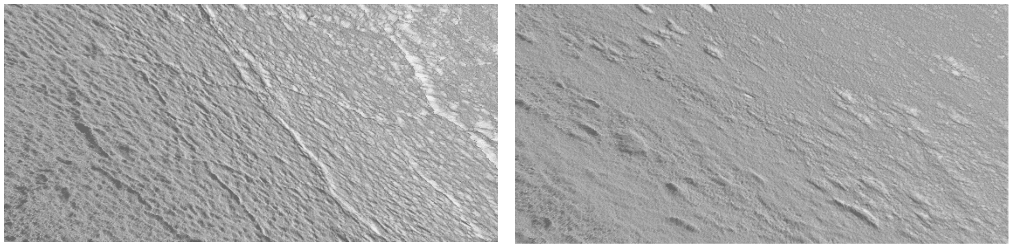


(a) Concrete surface with pop-outs

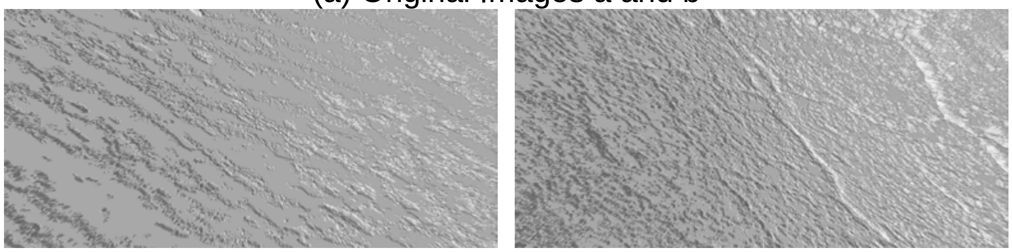


(b) Asphalt pavement with alligator cracking

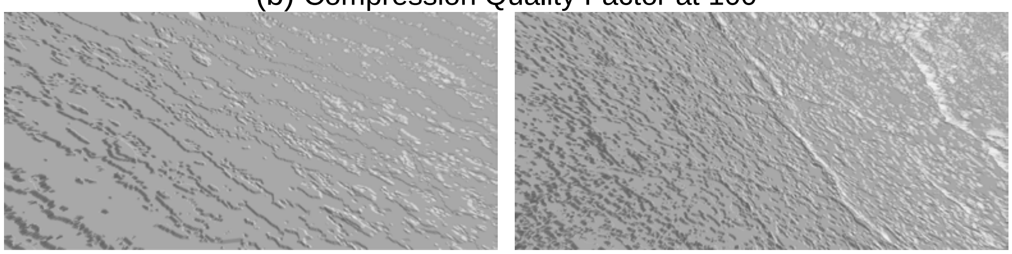
Figure 3.3 Original Images for Visual Evaluation, from WayLink PaveVision3D (Different Surface from Figure 3.2)



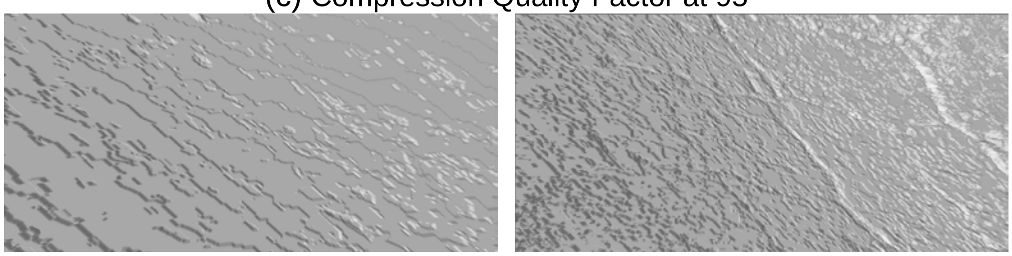
(a) Original Images a and b



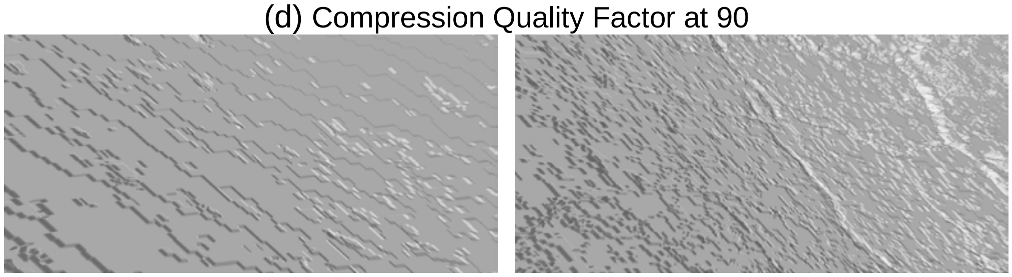
(b) Compression Quality Factor at 100



(c) Compression Quality Factor at 95



(d) Compression Quality Factor at 90



(e) Compression Quality Factor at 85

Figure 3.4 JPEG XT Visual Evaluation Results at Different Compression Quality Factors

4. GUIDELINES FOR FUTURE DEVELOPMENTS

Pavement Viewer Software

It is of great importance for users to be able to visually examine both the "intensity" and "range" pavement image data for various data operations at the highest possible quality, such as data comparisons, semi-automated rating, calibration, and validation. However, computer screens are specifically designed devices for displaying 2D images, but not for 3D data. In recent years, various 3D data view software interfaces have been developed for different formats of 3D data sets. As a matter of fact, approximately all commonly used 3D data formats have their associated viewing software tools. For example, the free LandXML viewer, developed by Carlson Software (Ziering et al., 2007), can not only view the 3D data sets, but also generate 3D image thumbnails automatically in Windows File Explorer view before opening a file.

Developing a viewer software will aid the implementation of the new data format developed from this project, which is beneficial in data sharing between agencies and vendors, data reporting, developing new software and analysis solutions, and setting national, state, and local performance goals. Specific features and capabilities of the anticipated viewer software should be defined and developed through a separately funded project in the future. Herein only technical guidelines are recommended for designing such a viewer software:

- The viewer software should be flexible for various data viewing purposes. Two levels of scopes should be included for the future development of the pavement viewer software.
    - Single File Viewer - given a pavement image file, this viewer allows users to display the 2D/3D images of interest;
    - Route Viewer - given a location that is of interest to users, this viewer is able to display this location from several meters to several hundred meters in length for a single-lane pavement. Route Viewer requires effective pavement data archiving and retrieving. For this purpose, the file header should be designed for effective information indexing and random data access. For example, file header can be read into memory independently without the need of reading the whole data file, which can be huge in size; therefore the time (data, time) and location (GPS, DMI, lane and direction) information in file header can be obtained and used for fast pavement data retrieving.
- Proper visualization methods, for instance, OpenGL or Direct3D techniques, should be used to take fully advantages of available hardware resources for fast 3D and/or 2D visualization, including rotation in three directions, and zooming-in and out features.
- The viewer software should support simple manual analysis, such as measure crack length, depth and width. Measurement tools and human–machine interface should be provided for this purpose. In addition, the viewer should have the capability to export manual analysis results to commonly used data formats, such as text and Excel csv files.
- The viewer software should provide warning messages for users to troubleshoot and diagnose if there are errors in the image data format.

- The viewer software should be capable of exporting pavement images (2D and/or 3D) into a common used image format so that they can be viewed in standard image viewers (i.e. Windows Paint, Photo Gallery, et al).
- Fundamental image data analyses (such as histogram analysis) and data visualization should be available in the viewer software, similar to profiling data analysis tools available in ProVal.
- 2D and 3D data characteristics in terms of precision, accuracy, and tolerances should be obtained from the viewer software, so that users can determine the appropriateness of the data for applications such as quality control/quality assurance activities.
- Lastly, since image data collection technology and the data formats will continue to evolve, the image viewer software shall have both backward and forward compatibilities to visualize files saved in older and successor data formats.

Web-Based Validation Software

In Task 2, the research team reviewed several relevant file formats for the storage of pavement image data, many of which include a supporting webpage to facilitate the adoption of the file format, such as LandXML (http://www.landxml.org/), GeoTIFF (https://trac.osgeo.org/geotiff/), OpenCRG (http://www.opencrg.org/), Universal 3D (U3D) Format (http://www.pdf3d.com/u3d/), the E57 format (http://www.libe57.org/), the "Elementary Data" format (http://www.w3.org/TR/REC-xml), and the HiSPEQ Project data format (https://hispeq.com/). The webpages generally contain a software development kit, a validator software, and a publically available data viewer.

It is strongly recommended to develop a website for future implementation of the proposed image data formats through another research project, in which example data sets conforming to the proposed data format standards are provided, and a web-based validation software is developed for users to check whether the image data files are correctly prepared according to the syntax of the standard data formats developed in this project.

A commercial cloud service, such as Microsoft Azure, can be used as the platform to have data checking capability available to anyone at any time with a commonly used browser in any operating system, such as Windows, iOS, or Android.

5. SUMMARY AND RECOMMENDATIONS

The study demonstrates that it is possible to conduct highly compressed and lossless operation on complex and high-resolution pavement 3D surface data through the OSU Method 2. With a mid-range computer made in 2014, the encoding procedure cannot be directly implemented in data collection in any data vehicle yet; however, such an implementation would be possible in the next couple of years due to improvements of CPU and GPU performance, and relevant software improvements. It is recommended that the lossless compression method (OSU Method 2) be used for future implementation as part of the standard. Recommended future research work includes (1) improvement of computation efficiency of the OSU Method 2, and (2) the development of a software viewer integrated with the new compression algorithm for compliance validation with the recommended standard data format, and possible future data analysis applications.

REFERENCES

Christopoulos C., A. Skodras and T. Ebrahimi. The JPEG2000 Still Image Coding System: An Overview. *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103-1127, Nov 2000.

GraphicsMagick. GraphicsMagick Image Processing System. http://www.graphicsmagick.org/index.html, accessed on October 7, 2016.

ImageMagick. The ImageMagick Graphics Library: A Gentle Introduction to Magick++. http://www.imagemagick.org/Magick++/tutorial/Magick++_tutorial.pdf, accessed on October 7, 2016.

International Telecommunications Union (ITU). Recommendation ITU-T Rec. T.832 ISO/IEC 29199-2, Information Technology: JPEG XR Image Coding System Image Coding Specification. ITU, Switzerland, 2012.

MatLab. Peak Signal-to-Noise Ratio (PSNR). https://www.mathworks.com/help/images/ref/psnr.html, accessed on October 7, 2016.

NetPBM. About Netpbm. http://netpbm.sourceforge.net/, accessed on October 7, 2016.

GIF, Graphics Interchange Format, Version 89a http://www.digitalpreservation.gov/formats/fdd/fdd000133.shtml, accessed on October 22, 2016

PNG, Portable Network Graphics, An Open, Extensible Image Format with Lossless Compression http://www.libpng.org/pub/png/, accessed on October 22, 2016

Pandit Rahul, Nindhiya Khosla, Gurjeet Singh and Hiteshwari Sharma. Image Compression and Quality Factor in case of JPEG Image format. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 7, July 2013

Richter T., A. Artusi and T. Ebrahimi. JPEG XT: A New Family of JPEG Backward-Compatible Standards. *IEEE MultiMedia*, vol. 23, no. 3, pp. 80-88, July-Sept. 2016.

Ritter, N. and Ruth, M. GeoTIFF Format Specification: GeoTIFF Revision 1.0. Available online at: http://trac.osgeo.org/geotiff/, accessed on March, 24 2016.

Sustainable Road Management in Texas: Network-Level Flexible Pavement Structural Condition Analysis Using Data-Mining Techniques Chi, S., Murphy, M., and Zhang, Z. Journal of Computing in Civil Engineering 2014 28:1, 156-165

Zhang A. and Wang K.C.P.. The Fast Prefix Coding Algorithm (FPCA) for 3D Pavement Surface Data Compression. *Computer-Aided Civil and Infrastructure Engineering* (In press), 2016

Ziering Eric, Frances Harrison, and Paul Scarponcini. TransXML: XML Schemas for Exchange of Transportation Data. National Cooperative Highway Research (NCHRP) Program. Report 576. Transportation Research Board (TRB), Washington D.C.C, 2007.

APPENDIX A DRAFT METADATA FORMAT TO DETERMINE PAVEMENT SURFACE CONDITION AND PROFILES

Standard Specification for

File Format of 2-Dimensional and 3 Dimensional (2D/3D) Pavement Image Data

AASHTO Designation: MP NN-NN

THE VOICE OF TRANSPORTATION

American Association of State Highway and Transportation Officials
444 North Capitol Street N.W., Suite 249
Washington, D.C. 20001

Standard Specification for

File Format of 2-Dimensional and 3-Dimensional (2D/3D) Pavement Image Data

AASHTO Designation: MP NN-NN

AASHTO THE VOICE OF TRANSPORTATION

1. SCOPE

1.1 This specification describes the standard specification for file format of 2-dimensional and 3-dimensional (2D/3D) pavement image data that is used to determine pavement surface condition and profiles.

1.2 This specification describes the variables and sizes of the data items that are stored in the file. The file is in binary format, and are fully documented in this specification.

1.3 This specification is designed to be independent of hardware platforms, computer languages, and Operating System (OS).

1.4 This standard does not support to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.

2. REFERENCED DOCUMENTS

2.1 ASTM Standards:
- E2560-13 Standard Specification for Data Format for Pavement Profile.
- E867 Terminology Relating to Vehicle-Pavement Systems.

2.2 IEEE Standards:
- IEEE 754–2008 (2008) Floating-Point Arithmetic

3. TERMINOLOGY

3.1 *Definitions:*

3.1.1 Terminology used in this specification conforms to the definitions included in Terminology E867.

3.2 *Definitions of Terms Specific to This Standard:*

3.2.1 Signed—integer capable of representing negative values.

3.2.2 Unsigned—integer only capable of representing nonnegative values.

3.2.3 Int8—data type for an 8-bit, unsigned integer.

3.2.4 Int32—data type for a 32-bit, signed integer.

3.2.5 Single—data type for a 32-bit, signed real number, such as, single precision IEEE floating point.

3.2.6 String—data type for a variable-length ASCII string. No null character is included at the end of the string. A separate field defines the length of the string.

3.2.7 4-byte String—an ASCII string of 4 characters in length. No null character is included at the end of the string.

3.2.8 8-byte String—an ASCII string of 8 characters in length. No null character is included at the end of the string.

3.2.9 Array (numeric data type)—sequence of data of the specified numeric data type. Only the values are stored, no information about the array is stored.

3.2.10 Array (String)—ASCII strings separated by a tab. There is no tab after the last string.

3.2.11 Backward compatibility—ability of a software system, such as a pavement image viewer, to read a file in an earlier version of the format standard.

3.2.12 Forward compatibility—ability of a software system such as a pavement image viewer to read a file in a future version of the format standard.

3.2.13 Offset— a byte-based index measuring the distance to the beginning of a data file. The offsets in the file header serve as the guides to locate the various data sections of the file.

4. IMAGE DATA SPECIFICATIONS

4.1 File Structure

4.1.1 The file structure is divided into five sections: (1) File Header, (2) 2D Image Data (intensity), (3) 3D Range Data, (4) User Defined Metadata, and (5) a File Trailer. The five sections are stored sequentially (Figure A-1). Either storage space for 2D data or 3D data is variable depending on the compressed data size.

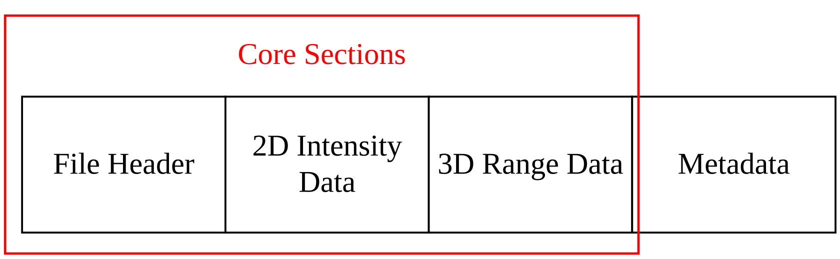


Figure A-1—Layout of the File Structure

4.1.2 Pavement imaging technology will continue to evolve with new capabilities. Therefore, it is important for the pavement image file format to support both backward and forward compatibilities of the related software, for instance, a pavement image viewer.

4.1.3 To achieve backward and forward compatibilities, pavement 2D/3D data file shall include two parts. The first part is the core section including File Header, 2D Image Data, and 3D Range Data. The data format definitions of the first part shall not be changed as the data format standard evolves. The second part primarily contains the optional user defined metadata section, which can be extended and modified as needed by vendors or highway agencies to accommodate individual data collection practices and equipment. The offsets in the file header serve as the guides to locate various portions of the file. All offsets are relative to the beginning of the file. The offset values can be obtained only after 2D/3D data are compressed.

4.1.4 Each of these portions of the file is described in the following sections. The data types and related descriptors required in the file are included. The data shall be written to the file sequentially, with the offsets listed in the file header as guides to find various portions of the file.

4.2. File Header

4.2.1 The file header describes the core properties of the 2D/3D data stored in the file. Each property shall be denoted by a variable and each variable shall have a predefined data type with fixed byte length. An example of file header is shown in Table A-1. These properties can be classified into two categories:

4.2.1.1 Required variables for data reading. These include file format version, image size, resolution, bit depth, compression algorithm name, and storage offset that are critical for successfully reading the 2D/3D data matrices for the purposes of data analysis or visualization.

4.2.1.2 Required variables for fast archiving and retrieving. These include location and time data that are critical for global database-based pavement data management and local section-based pavement data view.

4.2.2 The header of a pavement image file shall be encoded in binary format.

Table A-1—File Header

Variable Name	Data Type	Data Details
Version	4-byte String	Version number of the file format
SW version	8-byte String	Identifier of the software that produced the file
State Name	2-byte String	FIPS State Code
Route Name	12-byte String	Name of the highway HPMS standard
Direction	2-byte String	Direction of travel
Lane identification	2-byte String	Lane index
File Serial Number	Int32	File serial number in continuous data collection
GPS Longitude	Float 32bit	GPS longitude value IEEE 754 binary32
GPS Latitude	Float 32bit	GPS latitude value: IEEE 754 binary32
DMI Pulse	Int32	DMI pulse counting index
Date	8-byte String	Date data was collected—(yyyymmdd)
Time	6-byte String	Time data was collected—(hhmmss)
Event Mark ID	Int32	Event marker(s) by data collection crew
2D Compression Method	4-byte String	Identifies compression algorithms, such as 1: PNG; 2: JPEG; 3:JPEG2000.
2D Resolution Longitude Direction	Float 32bit mm	Distance between two data rows in longitude direction in millimeters.
2D Resolution Transverse Direction	Float 32bit mm	Distance between two data columns in transverse direction in millimeters
2D Width	Int32	Pixel numbers in transverse direction
2D length	Int32	Pixel numbers in longitude direction
2D Data Bit Depth	Int32	The bit depth for each data point
2D Data Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the 2D data
2D Compression Quality	Float 32bit	Compression quality level
3D Compression Method	8 byte String	Identifies compression algorithms, such as 1: PNG; 2: JPEG; 3:JPEG2000
3D Resolution Longitude Direction	Float 32bit mm	Distance between two data rows in longitude direction in millimeters
3D Resolution Transverse Direction	Float 32bit mm	Distance between two data columns in transverse direction in millimeters
3D Resolution Elevation Direction	Float 32bit mm	Units for range data value in millimeters
3D Width	Int32	Pixel numbers in transverse direction
3D length	Int32	Pixel numbers in longitude direction
3D Data Bit Depth	Int32	The bit depth for each data point
3D Data Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the 3D data
3D Compression Quality	Index Float 32bit	Compression quality level
Metadata Offset	Int32	Offset in bytes from the beginning of the file to the beginning of the metadata

Variable Name	Data Type	Data Details
Speed	Float 32bit mm/s	Average vehicle speed associated with data
Time stamp	Long long int	Milliseconds since UNIX Epoch: Jan 1, 1970 00:00:00
Vehicle Number	8 byte String	Vehicle identification
Operator Name	32 byte string	Operator identification
Reserved Item1	16 byte string	Reserved for future usage or additional vendor specific information
Reserved Item2	Int32	Reserved for future usage or additional vendor specific information
Reserved Item3	Float 32bit	Reserved for future usage or additional vendor specific information
Reserved Item4	8 byte String	Reserved for future usage or additional vendor specific information

4.3 Data Sections

4.3.1 The data sections store blocks of binary data, which are generated by compression algorithms.

Note — Vendors may use their proprietary algorithms for data compression. In this case, the algorithm used in the proprietary software should be provided in the form of either Dynamic Link Library (DLL) or source codes.

4.4 User defined Metadata Section

4.4.1 In the data format standard, the first value in the metadata portion shall provide the number of metadata entries (MDE) as shown in Table A-2. Table A-3 shows the partial list of information to construct a MDE. The specific metadata data entries and their tags shall be designed based upon the future needs or specific needs of users.

Table A-2—Metadata Example

Variable Name	Data Type	Data
Number of MDEs	Int32	Number of MDEs

Table A-3—Metadata Entries (Partial Listing)

Variable Name	Data Type	Data
Tag of MDE	Int32	Metadata tag
Data type of MDE	Int32	Data type index of MDE
Array size	Int32	"-1" if not an array. "0" if array is empty. Numbers greater than 0 specify the number of elements in the array
Count	Int32	For data types "String" and "Array (String)", count = the number of bytes in the string. For other data types, count = 1.
Name length	Int32	For metadata entries listed in Table 5, this is 0. For user-defined entries, this value is the length of the name.
Name	String	Name of the metadata
MDE	varies	Information associated with tag of MDE

4.5 File Trailer or Checksum

4.5.1 The file trailer is used to signal the end of the file (Table A-4). Alternatively, the checksum technique can be used to assure the completeness of the data file.

Table A-4—File Trailer

Variable Name	Data Type	Data	Default Value
End of file	4-byte String	Indicates the end of the file	"@@@@"

5. KEYWORDS

5.1 Standard Data Format; Pavement Images; Pavement Profile; Pavement Surface Condition; 2-Dimentional (2D) Intensity Data; 3-Dimentional (3D) Range Data