“Justification of the use of CCSDS 122.0-B-2 Compression Standard in Space Missions over other Compression Methods through the assessment of Image Quality”

CS39440 Major Project

Author: Charlie Curtis (chc73@aber.ac.uk)

Supervisor: Dr Helen Miles

3rd March 2022

Version 1.0

This report is submitted as partial fulfilment of a BSc degree in Computer Science (G400)

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB
Wales, UK

Declaration of originality

I confirm that:

- This submission is my own work, except where clearly indicated.
- I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
- I have read the regulations on Unacceptable Academic Practice from the University's Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
- In submitting this work, I understand and agree to abide by the University's regulations governing these issues.

Name: Charlie Curtis

Date: 04/05/2022

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name: Charlie Curtis

Date: 04/05/2022

Acknowledgements

I am grateful to my supervisor Dr Helen Miles.

Abstract

This project is concerned with comparing two separate standards of image compression algorithms in the context of space missions. Using statistics and existing Image Quality Assessment metrics, this paper assesses the viability of each with the aim of determining why one has been selected for use on the European Space Agencies' Rosalind Franklin Rover, which is due to launch as part of their ExoMars mission to Mars. Having processed a series of sample images through the CCSDS 122.0-B-2 and JPEG 2000 compression algorithms, it was found that the former was designed with the concern of resource management and optimisation, whereas the latter, more designed to have a higher perceptual quality. The project utilised Mean Squared Error, Peak Signal to Noise Ratio, and the Structural Similarity Index image quality assessment metrics to help justify these findings. The research leaves room for further exploration into this topic and the surrounding discipline and was conducted as part of a BSc degree in Computer Science.

Contents

1 Table of Contents

2 Background, Analysis & Process

2.1 Background

The European Space Agency (ESA) will soon launch the second part of their ExoMars mission. Rosalind Franklin is the rover that shall travel to the surface of Mars with the goal of attempting to establish if there is or ever has been life on Mars.

One of the rovers many instruments is PanCam, a camera system tasked with capturing how the planet looks and preparing these images to be sent back to Earth. Part of this preparation is the compression of the images to reduce their storage space and allow for a more streamlined and hopefully less error prone transfer back to Earth. The ESA has chosen to use the CCSDS 122.0-B-2 compression standard, the purpose of this report is to establish a justification for this, and reasons why other compression methods such as JPEG 2000 (J2K) are not used. This is done through the encoding of sample images taken from the Aberystwyth University PanCam Emulator (AUPE), through both compression algorithms, after which the results are compared against each other.

A sensible approach to begin seemed to be to conduct some preliminary research into image compression. This included not only image compression algorithms used in a variety of disciplines including medicine, graphic design, and military applications, but also, the theory behind image compression, i.e., lossy versus lossless compression, techniques used in image compression, including - but not limited to; quantization, discrete cosine transform (DCT), fractal compression – and performance comparisons of some compression algorithms.

The goal of this was to try and establish what sort of numerical operations these algorithms were performing, and through this, trying to build an understanding of how the quality assessment algorithms - that would be used as part of the overall analysis of the compression algorithms - might begin to calculate numerical quality in a general sense. This was very useful for understanding metrics such as Mean Square Error (MSE), Root Mean Square Error (RMSE), and Peak Signal to Noise Ratio (PSNR).

During this preliminary research, a grouping system of Image Quality Assessment (IQA) methods was identified. Methods are grouped into Full Reference (FR), Reduced Reference (RR), and No Reference (NR) IQA methods. They refer to if a ‘perfect’ quality - original image is used as a reference as part of the algorithm to compare to the compressed version. For this project, FR seemed the most appropriate since there *are* original images to be accessed and so if the more comprehensive algorithms can be exercised as part of testing of PanCam and as part of this project, the greater assurance we can have as to our understanding of the level of compression and quality of the images being sent back by Rosalind Franklin.

One motivation for this project was having the opportunity to produce something that could potentially have real world implications towards the operational effectiveness of the European Space Agencies, Rosalind Franklin rover. Moreover,

developing a tailored script (written in Python) that's sole purpose is to facilitate the gathering, processing, and analysis of the numerical aspects of Image Quality Assessment algorithms.

2.2 Analysis

The projects focus is on Image Quality Assessment algorithms. To that end, regardless of the direction of the project, these algorithms would have to play a significant role in any conclusions or findings. This led to the belief that a comparative study was the best approach, since the results that objective Image Quality Assessment algorithms provide are numerical values. This leaves great scope for statistical analysis. With this in mind, the logical progression was to compare the already selected compression algorithm with another - that was not selected to be a part of PanCam - to try and establish the reasons that the aforementioned algorithm was chosen over the other. This led to the construction of the overall research question for this project:

Given the quality of compressed images, what reasons are there for the use of the CCSDS 122.0-B-2 compression standard on PanCam, and why are other image compression algorithms not used in the context of interplanetary communication?

This in turn lead to more specific questions regarding the technicalities of the project and its process, such as; Which Image Quality Assessment algorithms would be best suited to a project of this nature? What sort of analysis and statistical metrics should be applied to any collected data? What would be the best method to approach this question?

The chosen research question also has the advantage of leaving the door open for further research or variations of this project, for instance, delving further into the question of which Image Quality Assessment algorithms are most suitable as metrics for assessing image quality in the context of space mission or conducting the same or similar research with other possibly more advanced image compression algorithms other than J2K.

2.3 Process

This project is a comparison using objective measures. Mathematical processes conducted by the computer and then interpreted by the user. Due to this fact, the data collected was always going to Quantitative data. A positivism-based [1] research project, collecting primary data for processing and to develop conclusions from.

An inductive process, aiming to answer the overall research question first proposed at the beginning of the project. [2] Given that the compression algorithm has already been decided upon for the rover, and that trying to produce a conclusive set of results would be beyond the scope of this project, the project took an approach indicative of an exploratory project, trying to establish the reasons for the selection of a given compression algorithm.

These are very blanket terms that best describe a process that was dynamic and constantly changing based on complications encountered at various stages and the capricious availability of pre-existing libraries and/or implementations of Image Quality Assessment algorithms.

Additionally, regarding the production of a technical submission. The project was research focused and so, a technical submission was not going to be of the same nature or standard of complexity as the likes of a project involving the development of some bespoke software, because of this, following an Agile Development approach was challenging. Nonetheless, a SCRUM-style of approach was loosely adopted and maintained over the lifecycle of the project, with a 'sprint' being a weeklong process, and the planning for said sprint being conducted at the end of the previous sprint.

Finally, in the forthcoming sections the report will address the following subjects to aid in building an understanding of the process, and algorithms involved.

In the next section, a discussion of the algorithms used for both the compression of the images, as well as those used for the assessment of image quality, to establish how they work, and therefore why these particular algorithms have been chosen for the process. Upon conclusion of this discussion, section four will discuss the script that was developed alongside the project and look into some of the key functions it performs. Then in section five, a description of the method that was adopted, not only for the collection of the data the project relies upon, but also an introduction to the process for the analysis of this data that is necessary for the development of a based, well-founded conclusion. This will be followed by the presentation of the results in section six and then a discussion of these results, what they mean, and what can be learnt from them in the analysis section – section seven. The report is to be finalised in section eight, with a critical evaluation of the process and administration surrounding the project as a whole - from beginning - to end.

3 Use of Algorithms

This section will explore the various algorithms that are used as part of this project including algorithms for both the compression of sample images; namely the CCSDS standard and JPEG 2000, as well as those used in the Image Quality Assessment process – Mean Squared Error, Peak Signal to Noise Ratio, and the Structural Similarity Index. The compression algorithms are as follows:

3.1 CCSDS 122.0-B-1

Bit Plane Encoder (BPE) is a command line implementation of the CCSDS 122.0-B-1 algorithm. The algorithm is based on wavelet transform and bit plane scanning. [3]

Bit plane encoding operations involve the conversion of image pixel values into their binary form, and then slicing. The process by which the binary values are separated by the most a least significant bits and then different bit planes constructed. Subsequently, the original image is reconstructed using the greatest value planes. This creates an image that has been lossy compressed using a significantly reduced amount of data as the refinement bits in the lower value planes are removed. [4] One of the main factors for the use of binary representation of pixel values, is because with current hardware standards this is particularly convenient. [5]

Wavelet transform is an adaptation of the Fourier Transform, that considers a wavelet of frequency, in this case, pixel intensity. The function operates in a localised region of the image known as a 'window' and based on the pixel intensity and window function (function used to select the area of the image for encoding) essentially quantizes the pixel values. [6][7]

On PanCam – this CCSDS algorithm is used by default at a ratio of 1, which achieves an x8 compression, and for quick look images at a ratio of 0.1; x80 compression.

3.2 JPEG 2000 (J2K)

J2K is a compound compression algorithm curated from a series of image compression procedures and similarly, is based on wavelet transforms. [8] J2K is an accessible, and popular compression algorithm which is why it was selected as part of this project. Moreover, when trying to discern reasons for CCSDS 122.0-B-1, comparing it to a relatively more standard compression algorithm seemed like a good step, as it helps to develop a picture of the magnitude of difference between an inter-disciplinary 'basic' method, and a bespoke solution for the demanding environment that is space and missions in space.

For this project – OpenJPEG was employed. [9] This is again, a command line implementation, alike to BPE, with parameters to change the compression ratio up to x9000 compression.

Given the parameters for compression outlined by the team behind PanCam, for this project. OpenJPEG was used with compression rates of x8 and x80, this is to bring it in line with what is to be used on Rosalind Franklin, and so produce an analysis and discussion/conclusion that is valid.

3.3 Mean Squared Error (MSE)

Mean Squared Error is the first of three Image Quality Assessment metrics that were used in this project. MSE measures the difference between a predicted set of data, and an actual, measured set. It is the second moment of error [10]. In this scenario, the predicted is the original 'perfect' quality image, and the actual is the compressed version, this makes MSE a full reference objective method. MSE is defined by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

Where x is the original image, and y the distorted image.

MSE has a clear physical meaning and can be applied in multiple ways to linear algebra, this makes it one the most used metrics in image quality assessment [11].

Because it is a measure of error, values closer to 0 mean a greater quality, and thus, less information loss during compression.

3.4 Peak Signal to Noise Ratio (PSNR)

Peak Signal to Noise Ratio is the ratio between the maximum possible signal value (pixel value) and the power of distorting noise where the signal is the perfect reference image, and the noise the error yielded by compression [10].

PSNR can be defined by:

$$PSNR = 10log_{10}\frac{L^2}{MSE}$$

Where L is the maximum possible signal value; In the case of an 8-bit (unsigned) image this would be $2^8 - 1 = 255$. The sample images however used in this project are all 16-bit, and as such $L = 65535$.

PSNR is measured in decibels, as such, a PSNR of 15 dB is a better score than a PSNR of 3dB because the signal level is 15 dB higher than any given level of noise, and so the result is a compressed image higher in quality and closer in likeness to the original image.

PSNR is a valuable metric as it will help to identify the relationship between noise and perfection, in a given image.

3.5 Structural Similarity Index (SSIM)

SSIM is a perception-based method [10]. The only of the three IQA algorithms to be based on the Human Visual System. Algorithms that replicate the Human Visual System consider different aspects of the image rather than the image as a whole, aspects such as brightness, contrast, and texture [12].

SSIM specifically – considers luminance, contrast, and structure [10][11]. SSIM can be defined as [13]:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where:

μ_x is the average of x
μ_y is the average of y
σ_x^2 is the variance of x
σ_y^2 is the variance of y
$C_1 = (k_1 L)^2 \; and \; C_2 = \; (k_2 L)^2$

Where:
L is pixel range (for 16-bit this is 65535)
k_1 = 0.01 and k_2 = 0.03 (these are default values)

This overall equation may be further broken down into component formulas for luminance, contrast, and structure. SSIM operates by assessing an $N \times N$ area of each image.

SSIM has be included in this project because of its operational likeness to the Human Visual System and as such, provides an alternative, less mathematical, perception-based approach to Image Quality Assessment.

4 Development of Script

The script that was worked on and developed as part of this project was originally started in the interest of 'insurance.' A fall back in the event libraries or command line tools for various functions and requirements of the overall process weren't available. Of course, these were eventually available in the form of Bit Plane Encoder and OpenJPEG as previously mentioned, but also additional tools that will be touched upon in later sections such as ImageMagick [14]. Initially the intention of the script was to run the objective Image Quality Assessment algorithms but has since been adapted to also generate absolute error maps, convert images between 8 and 16 bits, and perform the statistical analysis of data. This is achieved through the use of various libraries along with some basic arithmetic and data type manipulation.

The script can be found at the technical submission point for this project as well as on this projects GitHub repository found here:

https://github.com/charlieocurtis/cs39440-project

The following section will discuss the use of the libraries in question, as well as some of the major functions in the script that were used extensively throughout the process.

To begin, a list of the libraries:

- Python Image Library – PILLOW (PIL) [15]

- Scikit-Learn – sklearn [16]
- SciKit-Image – skimage [17]
- Numpy [18]
- SciPy – specifically stats module [19]
- MatPlotLib – pyplot module [20]
- Pandas [21]

PIL was necessary to deal with the loading and saving of the images in a streamlined fashion. It also allowed for identification of the 'mode' of the image, (whether or not the image is 8 or 16bits.) Sklearn contained a prewritten Mean Squared Error calculation and skimage an Structural Similarity Index calculation, this meant that these two metrics didn't need to be hardcoded.

Numpy was essential for collecting image pixel values, storing, and then handing these over to the relevant functions that take an array-like object. Moreover, arithmetic was made significantly easier given the nature of NumPy arrays to iterate through themselves, without the need for an additional loop, this benefits readability massively.

Finally, SciPy, MatPlotLib and Pandas were all required for the calculation and presentation of the resultant statistics that would be used to inform the analysis section (section 7) of this report. Any generated figures were produced within MatPlotLib, and statistical calculations such as Pearson's correlation coefficient and two-sample t-tests were all done through SciPy. Pandas was required to facilitate these calculations.

In relation to the development of functions. For the majority of the script's lifecycle, it was in an experimental/developmental stage, meaning some functions were added and then later removed, some incorporated into other functions, etc. This inevitably means that some functions – particularly at the start of the process – were and have been used more than others. Moreover, some functions were simply written to aid in the debugging process and had no bearing on the processing of any data, so were removed from the final version of the script.

There are two main functions that played pivotal roles in the process (discounting those functions that compute the various IQA metrics) those are; *collect_greyscale_pixels()* and *load_raw_image()*. In the following passage, the necessity and process of developing these essential functions will be discussed, beginning with *collect_greyscale_pixels()*.

The purpose of this function is to collect the pixel values from a grey scale image. If the passed image is not grey scale, it will be transformed as such. This is because the images that get passed to the compression algorithms must be in grey form to be run through the algorithm since they can only process single channel images, as a result any RGBA/RGB images passed, are converted.

This conversion is achieved by collecting the images 'mode' (i.e., the colour nature and pixel density) of each image being included in the process. The modes that are mentioned as part of the implementation are 'L', 'I', and 'I;16'.
Referring to the PIL documentation [22] it can be noted that:

- 'I' is 32-bit signed integer pixels
- 'L' is 8-bit black and white images
- 'I;16' is 16-bit unsigned pixel integers

In the event the image is 8 bit and already black and white (mode 'L'.) Then all that happens is the pixel values are read into a NumPy array, and the shape of the array adjusted to create a 2-dimensional array, that equates to the resolution of the image that was originally passed to the function.

When the image mode is returned to be 'I' this usually means one of the 16-bit grey scale sample images has been passed, and so the image is converted to 'I;16' before being read into the NumPy array and the arrays shape changed as aforementioned. This conversion means that the dynamic range of a given pixel is reduced and moreover, makes conversion to 8-bit that much easier, which when calculating absolute error maps for interesting cases, is welcome. Moreover, it aids in reproducibility.

The main reason for the importance of this function, is that the functions to calculate Mean Squared Error, Peak Signal to Noise Ratio, and Structural Similarity Index all require an array-like object to be passed as a parameter. This means that the image cannot simply be loaded in PIL and then passed into the function, the individual pixel values must be collected into an array. Only then can the various Image Quality Assessment metrics be computed. Moreover, these metrics require the images to be of the same pixel-depth, hence the need for conversion, pre-collection.

The final else statement within the function is a legacy statement from the debugging process, however, it was decided to leave it in, in the event that if the script is run with any other images, the basic pixel collection can be applied and stop any errors being thrown.

Finally, the *load_raw_image()* function. Because Bit Plane Encoder requires to be provided with raw image data, this means the original sample images have to be converted into a DAT format. The decode delegate for this format in ImageMagick, is not included in the basic install, as a result, this function was required, in order to reproduce a PNG image from the compressed DAT files. Utilising the NumPy *fromfile()* [23] function to generate an array from the raw data file.

Initially this is a 1-dimensional array, however, reshaping the array to be the square root of the array size, creates the final 2-dimensional array that has the same dimensions as the original image.

Without the function, the compressed DAT files coming out of Bit Plane Encoder, would not be able to be converted back into a PNG image, this function was applied to every image that was tested, and without it the final Image Quality Assessment metrics and statistical analysis could not be conducted as there would be no compressed images to utilise.

In all, the script went from being an answer to a potential worst-case scenario, to an integral part of the project and method, going through many different stages with varying levels of capability, before reaching its final version.

5 Method

The method can be divided into three distinct sections, listed below in the order within which they were conducted:

1. Sample preparation
2. Processing
3. Analysis

Each section will be addressed in turn whilst also making clear how they fit into the overall method, prior to this however, a brief discussion regarding the sample images that are used in the project.

Given there is an evaluation of real time involved with this project it is necessary to state the specifications of the machines this experiment was run on:

Compression and Decompression inside Bit Plane Encoder was done using an ASUS Zenbook with 16GB of RAM and an Intel I5 quad core processor. This system operated a Linux environment and processing was executed through a bash shell. All other necessary computation including, OpenJPEG compression, PNG to DAT conversion and functions executed through the developed Python script, were executed on a 2018 MacBook Pro, again with an Intel I5 processor and 8GB of RAM.

5.1 Introduction to Sample Images

Sample images for this project were provided by the Aberystwyth University PanCam Emulator (AUPE.) The set of images consisted of nine different folders, each containing eight different images. This gave a total of 72 sample images. Of the eight images in each folder, two were multichannel/colour composites. On the rover the camera taking these images can have various coloured filters applied in order to facilitate the capture of different information in a given image. Because of the coloured nature of these images, they require a separate CCSDS compression algorithm in order to be processed, because of this, these multi-channel images were excluded from the analysis of data in the project. This was due to the fact that the comparison of two separate CCSDS algorithms in tandem, against a third, would have been outside the scope of the project, they were processed as far as they could be

with each algorithm for the purpose of a more comprehensive documentation of the process, but their results were discounted from the final stage of the method.

Moreover, for reasons that continue to be unclear, there was six more that were not able to be passed through various stages of the method and so results could not be collected for them.

These complications left 48 images available for processing: 66% of the original sample size.

In their original, pre-compressed, 'perfect' form, the images (that did end up being considered in the final analysis) were 1024x1024 pixels in resolution, with a 16-bit pixel depth and with a Gray colour model (72 pixels per inch DPI.) The images came as PNGs with a naming system based on which camera took the image, and the subject of the images.
The images maintained a theme as to their contents. They showed either 'items' or 'rocks' on different backdrops of a desk, sandpit, and a brick-paved floor. These changes in backdrop would prove to yield some interesting results. The 'items' consisted of books, a reel of string, and an ethernet cable, i.e., unnatural, man-made objects. The 'rocks' were as what would be expected.

5.2 Sample Preparation

PanCam, and therefore AUPE, have three cameras that this project is concerned with; two Wide Angle Cameras (WACs) and one High Resolution Camera (HRC.)

The sample images come as PNG images. Bit Plane Encoder can only process images in a raw format, as such the images had to be first converted into DAT format for the purposes of compression through Bit Plane Encoder. With OpenJPEG however, this was not necessary.

This conversion was simply done through ImageMagick, this is a command line tool for processing, editing, and formatting images. This tool was used, and proved particularly useful, in this conversion process. See the command below:

```
magick convert P00_T00_L01_00.png gray: P00_T00_L01_00.dat
```

This example of a magick command is representative of those that were used in the project. In the scenario of the example an image entitled "P00_T00_L01_00.png" is converted to a DAT format.

This command was executed semi autonomously utilising bash commands, it was executed in each directory containing sample images, in order to convert all into binary format. Throughout the overall method, a naming convention was adopted that made interpreting the results of executing these bash scripts much more intuitive and fluid.

Images simply converted to a different format merely had the file extension changed – whether that be .png, .dat, or .j2k. For the images parsed to Bit Plane Encoder, this was not enough however, as BPE simply compresses the file and returns it. To overcome this, when compressed, the images had '_compressed_' appended, followed by the ratio of compression for instance - '1', all of this was appended prior to the file extension. Once decoded, '_compressed' was replaced by '_decompressed_' again with the compression ratio, and again, prior to the file extension. This made managing all the files significantly easier and developing a spreadsheet of results similarly as perceptive.

Once the original sample PNG images were converted into their respective binary versions. They could then be passed to the Bit Plane Encoder for encoding.

With respect to the J2K compression, this conversion step was not required. The original images could simply be used as they were and OpenJPEG would handle any conversion, and subsequent compression of the images.

5.3 Processing

Image processing was the longest stage in the process. This is largely due to each image having to be passed through each algorithm twice. Once, in order to achieve an x8 compression, and twice for an x80 compression rate. The reason for this is that these two values are the extremes according to the PanCam documentation. These compressions are used as a default, and for 'quick look' images, respectively.

When encoding with Bit Plane Encoder, the trend of being conducted semi autonomously through bash commands was adhered to. Given the nature of the files that were being run through BPE (raw binary file) the tool requires that some parameters be specified when the compression command is run. In this instance this information was the width and height of the image, pixel depth, and the compression ratio (for this project limited to either 1 or 0.1.) In this scenario all these values are the same from image to image which made the time taken to process, significantly less of a hindrance.

In addition, for the encoding and decoding of images using Bit Plane Encoder, the bash 'time' key word was used to effect, in order to report on the real time taken to compress and decompress images. This would prove valuable in the overall analysis of the process and results.

The following are the compression (for a ratio of 1 and 0.1) and decompression commands used within Bit Plane Encoder respectively, to achieve the generation of compressed version (at both levels) of the original images:

```
time ./bpe -e P00_T00_L01_00.dat -o P00_T00_L01_00_compressed_1.dat -w 1024
-h 1024 -b 16 -r 1
```

```
time ./bpe -e P00_T00_L01_00.dat -o P00_T00_L01_00_compressed_1.dat -w 1024
-h 1024 -b 16 -r 0.1
```

```
time ./bpe -d P00_T00_L01_00_compressed_1.dat -o
p00_T00_L01_00_decompressed_1.dat
```

Note that during decompression the extra parameters are not required as these are collected from the compressed data stream.

Once this process is complete, the filenames finishing with 'decompressed_X.dat' where X is either compression ratio, are now ready to be read, saved as PNG images, and then data collected from them such as file size and compression/decompression time, as well as the scores from the various Image Quality Assessment metrics.

The process with J2K was significantly easier. Since OpenJPEG can handle PNG image files, this means that no manual conversion is required, the image can be passed to the tool as it is and processed from there. For compression rates of x8 and x80 respectively:

```
opj_compress -i P00_T00_L01_00.png -o P00_T00_L01_00.j2k -r 8
```

```
opj_compress -i P00_T00_L01_00.png -o P00_T00_L01_00.j2k -r 80
```

A much more streamlined process, generating compressed images in a significantly shorter time. This along with the semi autonomy provided by bash commands made this process by far the most accessible.

At this stage, data is ready to be collected on the J2K compression process, however, as previously discussed there is still a step remaining in order to turn the raw data files generated from the BPE, into useable PNG images from which data on the BPE process can be collected.

This final step involves utilising the developed script, that was created alongside the project, to collect the raw data and generate PNG images. The steps involved in this are explained in section 4, however, once this process was complete, data from the Bit Plane Encoder compression process too, was able to be collated.

The data collected was as may be expected from such a task; the times outputted from the compression tasks were recorded, along with the original and compressed file sizes.
The last step was to run each image through the MSE, PSNR, and SSIM functions and record these results. This was done, for both compression levels, and for both algorithms. This led to the construction of Tables 1 and 2, in the next section.

5.4 Analysis Background

The analysis will be addressed in section seven of this report. However, it is important to note that prior to data collection, most of the statistical metrics that were to be used had already been decided upon. This is because the nature of the tests being conducted - as part of the analysis - helped to inform what sort of data would be valuable and beneficial to the project.

Analysis was done through a mixture of statistical function implementations in the developed script for this project, and tools within available spreadsheet editing software.

6 Results

Upon the processing and collection of the necessary data, Tables 1 and 2 were produced.

Table 1 is the relevant data that was collected from compression, decompression, and processing of images through Bit Plane Encoder, at both ratios of compression (1 and 0.1.) The data in Table 2, is the equivalent data that was collected for the compression of images through OpenJPEG. For both tables, file names in red, are files for which information is missing, this is because they weren't processed for reasons mentioned previously. Whereas cells that are empty and highlighted red, are indicative of the raw data that could not be collected i.e., the *actual* value(s) that is missing from that file in order for it to be considered 'incomplete.' Located, in the final row of each table are the mean averages for their respective columns.

			Ratio 1 (8x compression)							Ratio 0.1 (80x compression)						
			Compression		De-Compressed					Compressed		De-Compressed				
Folder	Filename	Original Size (bytes)	Size (bytes)	Real Time (s)	Size (bytes)	Real Time (s)	MSE (4d.p)	PSNR (4d.p)	SSIM (4d.p)	Size (bytes)	Real Time (s)	Size (bytes)	Real Time (s)	MSE (4d.p)	PSNR (4d.p)	SSIM (4d.p)
ACE_outside_2xpct	P00_T00_L01_00	1596702	131072	0.083	1890400	0.069	47.5256	2.0269	0.7647	13056	0.048	1564431	0.036	703.8074	0.1369	0.1621
	P00_T00_L02_00	1748434	131072	0.075	1885651	0.071	44.0643	2.1861	0.7808	13056	0.04	1549896	0.038	623.2159	0.1546	0.1633
	P00_T00_L03_00	1669407	131072	0.08	1871990	0.072	36.8809	2.6119	0.8238	13056	0.038	1544806	0.035	492.0428	0.1958	0.186
	P00_T00_R01_00	1709277	131072	0.076	1884581	0.067	43.6647	2.2061	0.7664	13056	0.039	1559886	0.036	512.3166	0.188	0.1573
	P00_T00_R02_00	1696972	131072	0.074	1880622	0.067	40.6435	2.3701	0.7761	13056	0.039	1548843	0.038	426.2502	0.2259	0.1589
	P00_T00_R03_00	1671523	131072	0.081	1871933	0.069	37.0078	2.6029	0.8184	13056	0.038	1551906	0.036	477.3634	0.2018	0.1713
	L0123-Composite	2749794														
	R0123-Composite	2698344														
ACE_outside_brickpattern	P00_T00_L01_00	1609526	131072	0.083	1862934	0.068	24.5465	3.9244	0.7991	13056	0.048	1580847	0.037	606.4811	0.1588	0.2289
	P00_T00_L02_00	1579151	131072	0.077	1853720	0.062	22.2953	4.3206	0.8032	13056	0.039	1566275	0.039	524.6048	0.1836	0.2278
	P00_T00_L03_00	1550523	131072	0.077	1850745	0.064	17.6755	5.4499	0.8137	13056	0.038	1566458	0.034	337.2181	0.2857	0.2359
	P00_T00_R01_00	1598862	131072	0.077	1874290	0.061	33.3834	2.8856	0.7882	13056	0.039	1591406	0.036	581.3768	0.1657	0.1923
	P00_T00_R02_00	1564685	131072	0.079	1860433	0.063	28.3424	3.3988	0.7957	13056	0.04	1574545	0.039	475.4746	0.2026	0.1989
	P00_T00_R03_00	1543539	131072	0.078	1859624	0.064	23.3834	4.1196	0.8066	13056	0.039	1575381	0.035	373.6912	0.2578	0.2109
	L0123-Composite	2765778														
	R0123-Composite	2777710														
ACE_sandpit_2xpct	P00_T00_L01_00	720540														
	P00_T00_L02_00	806461														
	P00_T00_L03_00	822020														
	P00_T00_R01_00	700044														
	P00_T00_R02_00	783363														
	P00_T00_R03_00	827179														
	L0123-Composite	2641724														
	R0123-Composite	2632528														
acli_desk_items	comptest_objects1-desk_LWAC01_T00_P00	940728	131072	0.087	1663694	0.069	2.9155	33.0408	0.6514	13056	0.048	1215438	0.038	42.6826	2.2569	0.1642
	comptest_objects1-desk_LWAC02_T00_P00	952312	131072	0.077	1669272	0.071	3.1279	30.7968	0.6494	13056	0.039	1229513	0.038	44.0515	2.1867	0.1645
	comptest_objects1-desk_LWAC03_T00_P00	967726	131072	0.078	1678365	0.068	3.2262	29.8585	0.652	13056	0.038	1272686	0.034	76.08	1.2662	0.1759
	comptest_objects1-desk_RWAC01_T00_P00	984918	131072	0.077	1695376	0.068	2.4918	38.6584	0.7511	13056	0.039	1308490	0.036	77.0992	1.2494	0.3876
	comptest_objects1-desk_RWAC02_T00_P00	977744	131072	0.077	1692650	0.066	2.5135	38.3245	0.7503	13056	0.038	1298944	0.039	41.816	2.3037	0.3847
	comptest_objects1-desk_RWAC03_T00_P00	955770	131072	0.074	1685470	0.067	2.2419	42.9661	0.7359	13056	0.038	1286472	0.042	33.8695	2.8441	0.3602
	L0123-Composite	2143196														
	R0123-Composite	2263410														
acli_desk_rocks	comptest_objects3-desk3_LWAC01_T00_P00	1018060	131072	0.081	1734660	0.066	2.4301	39.6401	0.7796	13056	0.042	1287969	0.038	21.4324	4.4946	0.5037
	comptest_objects3-desk3_LWAC02_T00_P00	1019407	131072	0.071	1734776	0.068	2.5996	37.0534	0.7737	13056	0.038	1278256	0.038	21.5849	4.4628	0.4813
	comptest_objects3-desk3_LWAC03_T00_P00	1017347	131072	0.07	1735431	0.065	1.7938	53.6999	0.8249	13056	0.038	1282170	0.034	20.5787	4.681	0.6303
	comptest_objects3-desk3_RWAC01_T00_P00	1008835	131072	0.073	1728494	0.067	1.8994	50.7164	0.7849	13056	0.039	1258803	0.035	20.126	4.7863	0.5013
	comptest_objects3-desk3_RWAC02_T00_P00	1010680	131072	0.072	1728644	0.065	2.3423	41.1256	0.7155	13056	0.039	1258026	0.04	20.5626	4.6847	0.3798
	comptest_objects3-desk3_RWAC03_T00_P00	1012794	131072	0.07	1731482	0.065	0.9769	98.605	0.8415	13056	0.039	1265146	0.035	17.7305	5.4329	0.6967
	L0123-Composite	2559974														
	R0123-Composite	2316399														
acli_floor_items	comptest_objects2-all50_LWAC01_T00_P00	989142	131072	0.083	1701693	0.066	3.6043	26.7264	0.7586	13056	0.044	1417040	0.037	90.6644	1.0625	0.2537
	comptest_objects2-all50_LWAC02_T00_P00	984329	131072	0.078	1700470	0.068	3.6526	26.3732	0.7581	13056	0.038	1413123	0.039	78.1891	1.232	0.2307
	comptest_objects2-all50_LWAC03_T00_P00	985001	131072	0.079	1701062	0.066	3.4399	28.0033	0.7697	13056	0.038	1427704	0.034	78.2929	1.2304	0.2495
	comptest_objects2-all50_RWAC01_T00_P00	1031973	131072	0.079	1735483	0.066	4.8345	19.9253	0.7783	13056	0.041	1449578	0.035	49.9154	1.9299	0.2833
	comptest_objects2-all50_RWAC02_T00_P00	1020464	131072	0.078	1727713	0.066	4.8863	19.7141	0.7684	13056	0.041	1444673	0.04	59.4096	1.6214	0.2319
	comptest_objects2-all50_RWAC03_T00_P00	986019	131072	0.078	1709892	0.068	3.7441	25.7284	0.769	13056	0.037	1423469	0.036	67.0528	1.4366	0.2159
	L0123-Composite	2312723														
	R0123-Composite	2456170														
acli_floor_rocks	comptest_objects4-all50_LWAC01_T00_P00	1003479	131072	0.076	1721353	0.068	3.7316	25.8143	0.8119	13056	0.044	1432406	0.036	57.3277	1.6803	0.304
	comptest_objects4-all50_LWAC02_T00_P00	1023566	131072	0.075	1736199	0.065	3.9166	24.5954	0.8155	13056	0.038	1454879	0.036	28.7409	3.3516	0.3384
	comptest_objects4-all50_LWAC03_T00_P00	1024419	131072	0.074	1736778	0.067	3.8287	25.1593	0.8195	13056	0.037	1464165	0.033	26.8231	3.5913	0.3396
	comptest_objects4-all50_RWAC01_T00_P00	1010674	131072	0.074	1728584	0.065	3.7268	25.8475	0.8018	13056	0.038	1441370	0.033	31.5099	3.0571	0.3056
	comptest_objects4-all50_RWAC02_T00_P00	1029831	131072	0.075	1738408	0.062	4.0395	23.8472	0.8092	13056	0.037	1462535	0.037	32.2076	2.9909	0.3236
	comptest_objects4-all50_RWAC03_T00_P00	1030803	131072	0.076	1739215	0.063	3.9729	24.2464	0.8181	13056	0.037	1471539	0.033	30.6044	3.1476	0.3379
	L0123-Composite	2536059														
	R0123-Composite	2527178														
acli_sandpit_items	comptest_objects4-sandpit_LWAC01_T00_P00	1107270	131072	0.087	1785962	0.067	14.1326	6.8161	0.7206	13056	0.042	1411984	0.035	170.3473	0.5655	0.1144
	comptest_objects4-sandpit_LWAC02_T00_P00	1203363	131072	0.078	1853484	0.068	33.0005	2.919	0.7522	13056	0.038	1487468	0.033	197.1908	0.4885	0.1172
	comptest_objects4-sandpit_LWAC03_T00_P00	1143708	131072	0.076	1820032	0.066	18.6493	5.1653	0.7879	13056	0.038	1473770	0.033	146.2049	0.6589	0.1357
	comptest_objects4-sandpit_RWAC01_T00_P00	1096169	131072	0.082	1773311	0.068	13.0168	7.4004	0.7075	13056	0.038	1396644	0.032	140.7265	0.6845	0.1192
	comptest_objects4-sandpit_RWAC02_T00_P00	1173677	131072	0.075	1829447	0.066	24.8988	3.8688	0.7378	13056	0.038	1468261	0.032	162.3644	0.5933	0.1265
	comptest_objects4-sandpit_RWAC03_T00_P00	1161168	131072	0.075	1826858	0.067	20.6842	4.6572	0.7744	13056	0.038	1481166	0.032	151.0923	0.6376	0.1411
	acli_sandpit_items_pc_L-Composite	2497257														
	acli_sandpit_items_pc_R-Composite	2479493														
acli_sandpit_rocks	comptest_objects2-sandpit2_LWAC01_T00_P00	1262463	131072	0.095	1892452	0.065	51.9449	1.8545	0.7211	13056	0.053	1487859	0.037	154.207	0.6247	0.0871
	comptest_objects2-sandpit2_LWAC02_T00_P00	1252404	131072	0.082	1891182	0.063	35.8206	2.6892	0.7189	13056	0.044	1492980	0.033	112.348	0.8574	0.0855
	comptest_objects2-sandpit2_LWAC03_T00_P00	1216406	131072	0.095	1866802	0.063	20.6368	4.6679	0.7222	13056	0.042	1493527	0.032	65.1574	1.4784	0.1078
	comptest_objects2-sandpit2_RWAC01_T00_P00	1161606	131072	0.085	1824241	0.064	23.0325	4.1823	0.7261	13056	0.043	1439582	0.032	101.6144	0.9479	0.1039
	comptest_objects2-sandpit2_RWAC02_T00_P00	1230235	131072	0.089	1875626	0.063	36.2103	2.6603	0.7484	13056	0.045	1490405	0.032	123.8223	0.7779	0.1079
	comptest_objects2-sandpit2_RWAC03_T00_P00	1201786	131072	0.097	1855134	0.066	20.4203	4.7173	0.7329	13056	0.042	1495482	0.032	87.7999	1.0971	0.1389
	L0123-Composite	2601832														
	R0123-Composite	2654095														
	Average	1500113.167	131072	0.078916667	1785346.208	0.066208333	16.41245	19.17015625	0.766145833	13056	0.040229167	1436212.54	0.035625	183.647288	1.640620833	0.248397917

Table 1 – Data collected from compression and decompression of images in BPE (CCSDS 122.0-B-2)

			8x Compression					80x Compression				
Folder	Filename	Original Size (bytes)	Size (bytes)	Real Time (s)	MSE (4 d.p)	PSNR (4 d.p)	SSIM (4 d.p)	Size (bytes)	Real Time (s)	MSE (4 d.p)	PSNR (4 d.p)	SSIM (4 d.p)
ACE_outside_2xpct	P00_T00_L01_00	1596702	262061	0.29	510.9419	0.1885	0.8003	26105	0.279	568.8209	0.1693	0.3836
	P00_T00_L02_00	1748434	261780	0.282	553.4122	0.1741	0.8374	26048	0.281	625.5598	0.1539	0.3922
	P00_T00_L03_00	1669407	262117	0.281	580.4763	0.1659	0.8608	26094	0.278	652.6061	0.1476	0.4324
	P00_T00_R01_00	1709277	262141	0.3	529.2761	0.182	0.8095	26107	0.288	584.1468	0.1649	0.3659
	P00_T00_R02_00	1696972	262028	0.291	588.3433	0.1637	0.8345	25861	0.278	651.6186	0.1478	0.3685
	P00_T00_R03_00	1671523	262101	0.284	552.4899	0.1744	0.8597	26082	0.274	623.6293	0.1545	0.4291
	L0123-Composite	2749794	393086	0.425				39240	0.417			
	R0123-Composite	2698344	393223	0.417				39335	0.398			
ACE_outside_brickpattern	P00_T00_L01_00	1609526	261914	0.294	782.3068	0.1231	0.67	25994	0.281	791.0699	0.1218	0.3591
	P00_T00_L02_00	1579151	261783	0.301	882.9809	0.1091	0.6806	26180	0.276	893.2758	0.1078	0.3578
	P00_T00_L03_00	1550523	261864	0.293	926.5665	0.1039	0.649	26160	0.275	929.9862	0.1036	0.3701
	P00_T00_R01_00	1598862	261934	0.299	638.5632	0.1509	0.7229	25990	0.279	671.1303	0.1435	0.3746
	P00_T00_R02_00	1564685	262154	0.29	762.9493	0.1263	0.7299	26220	0.281	791.8189	0.1217	0.3625
	P00_T00_R03_00	1543539	261996	0.286	836.9289	0.1151	0.6934	26015	0.273	858.4632	0.1122	0.3849
	L0123-Composite	2765778	393096	0.414				39316	0.407			
	R0123-Composite	2777710	393065	0.411				39246	0.392			
ACE_sandpit_2xpct	P00_T00_L01_00	720540										
	P00_T00_L02_00	806461										
	P00_T00_L03_00	822020										
	P00_T00_R01_00	700044										
	P00_T00_R02_00	783363										
	P00_T00_R03_00	827179										
	L0123-Composite	2641724										
	R0123-Composite	2632528										
acli_desk_items	comptest_objects1-desk_LWAC01_T00_P00	940728	261801	0.241	69.4112	1.3878	0.8491	26132	0.246	75.9991	1.2675	0.2787
	comptest_objects1-desk_LWAC02_T00_P00	952312	262067	0.245	71.796	1.3417	0.8636	26140	0.244	78.8079	1.2223	0.2919
	comptest_objects1-desk_LWAC03_T00_P00	967726	262099	0.249	70.8541	1.3595	0.847	26228	0.242	77.8654	1.2371	0.3119
	comptest_objects1-desk_RWAC01_T00_P00	984918	261889	0.253	463.0589	0.208	0.632	26229	0.246	464.7944	0.2073	0.2958
	comptest_objects1-desk_RWAC02_T00_P00	977744	262069	0.253	450.2127	0.2139	0.6565	26136	0.248	451.6977	0.2133	0.3019
	comptest_objects1-desk_RWAC03_T00_P00	955770	262011	0.245	391.7082	0.2459	0.6927	26216	0.242	392.8783	0.2452	0.3023
	L0123-Composite	2143196	392821	0.323				39158	0.309			
	R0123-Composite	2263410	393022	0.342				39190	0.33			
acli_desk_rocks	comptest_objects3-desk3_LWAC01_T00_P00	1018060	262125	0.256	227.2918	0.4238	0.4861	26168	0.248	226.0822	0.4261	0.212
	comptest_objects3-desk3_LWAC02_T00_P00	1019407	262067	0.252	212.237	0.4539	0.5057	26137	0.246	211.2615	0.4559	0.2056
	comptest_objects3-desk3_LWAC03_T00_P00	1017347	262133	0.242	264.7528	0.3638	0.3593	26082	0.246	261.5293	0.3683	0.2057
	comptest_objects3-desk3_RWAC01_T00_P00	1008835	262026	0.251	210.3174	0.458	0.4469	25967	0.243	208.5426	0.4619	0.1724
	comptest_objects3-desk3_RWAC02_T00_P00	1010680	262119	0.252	158.7907	0.6066	0.5366	26213	0.25	158.3709	0.6083	0.1747
	comptest_objects3-desk3_RWAC03_T00_P00	1012794	262117	0.25	262.3866	0.3671	0.2381	25911	0.242	257.8333	0.3736	0.1761
	L0123-Composite	2559974	393135	0.359				38961	0.346			
	R0123-Composite	2316399	393097	0.354				39316	0.336			
acli_floor_items	comptest_objects2-all50_LWAC01_T00_P00	989142	262086	0.242	289.1584	0.3331	0.8315	25998	0.247	295.072	0.3265	0.3262
	comptest_objects2-all50_LWAC02_T00_P00	984329	262044	0.249	138.8758	0.6936	0.8725	26075	0.248	145.632	0.6615	0.3539
	comptest_objects2-all50_LWAC03_T00_P00	985001	262047	0.253	148.3738	0.6492	0.8516	26049	0.247	154.1955	0.6247	0.3368
	comptest_objects2-all50_RWAC01_T00_P00	1031973	262003	0.25	397.5146	0.2423	0.7708	26055	0.248	404.3841	0.2382	0.3194
	comptest_objects2-all50_RWAC02_T00_P00	1020464	261913	0.256	140.8045	0.6841	0.8413	26203	0.251	150.0503	0.6419	0.3536
	comptest_objects2-all50_RWAC03_T00_P00	986019	261762	0.252	119.6589	0.805	0.8749	26200	0.246	126.7611	0.7599	0.3653
	L0123-Composite	2312723	392760	0.344				39272	0.333			
	R0123-Composite	2456170	393026	0.347				39311	0.345			
acli_floor_rocks	comptest_objects4-all50_LWAC01_T00_P00	1003479	262073	0.25	582.8003	0.1653	0.7763	26080	0.246	587.8168	0.1639	0.3809
	comptest_objects4-all50_LWAC02_T00_P00	1023566	261996	0.251	517.6206	0.1861	0.7266	26210	0.248	521.6467	0.1847	0.3905
	comptest_objects4-all50_LWAC03_T00_P00	1024419	261919	0.258	555.8336	0.1733	0.7173	26219	0.252	558.8236	0.1724	0.3911
	comptest_objects4-all50_RWAC01_T00_P00	1010674	262127	0.254	513.0189	0.1878	0.757	26131	0.251	517.3499	0.1862	0.3636
	comptest_objects4-all50_RWAC02_T00_P00	1029831	261721	0.254	465.0772	0.2071	0.7254	26222	0.246	468.331	0.2057	0.3724
	comptest_objects4-all50_RWAC03_T00_P00	1030803	262023	0.255	517.2761	0.1862	0.7167	26186	0.247	520.8058	0.1849	0.3767
	L0123-Composite	2536059	393025	0.348				39151	0.337			
	R0123-Composite	2527178	392897	0.361				39301	0.344			
acli_sandpit_items	comptest_objects4-sandpit_LWAC01_T00_P00	1107270	262061	0.259	118.3115	0.8142	0.8665	26055	0.252	142.5047	0.6759	0.2098
	comptest_objects4-sandpit_LWAC02_T00_P00	1203363	262104	0.27	124.49	0.7738	0.8769	26049	0.269	181.3345	0.5312	0.3277
	comptest_objects4-sandpit_LWAC03_T00_P00	1143708	262041	0.265	69.1874	1.3923	0.9088	26034	0.259	106.2584	0.9066	0.372
	comptest_objects4-sandpit_RWAC01_T00_P00	1096169	262115	0.255	153.1733	0.6289	0.8371	26159	0.254	172.9355	0.557	0.1883
	comptest_objects4-sandpit_RWAC02_T00_P00	1173677	261993	0.275	148.0344	0.6507	0.8562	26192	0.256	191.0486	0.5042	0.2967
	comptest_objects4-sandpit_RWAC03_T00_P00	1161168	262113	0.26	107.0372	0.8999	0.8728	25976	0.255	147.6328	0.6525	0.3569
	acli_sandpit_items_pc_L-Composite	2497257	393039	0.399				39159	0.39			
	acli_sandpit_items_pc_R-Composite	2479493	392985	0.398				39182	0.371			
acli_sandpit_rocks	comptest_objects2-sandpit2_LWAC01_T00_P00	1262463	262023	0.266	128.0506	0.7523	0.841	25984	0.256	196.2789	0.4908	0.3137
	comptest_objects2-sandpit2_LWAC02_T00_P00	1252404	262107	0.264	128.519	0.7495	0.7769	25974	0.254	167.9019	0.5737	0.3236
	comptest_objects2-sandpit2_LWAC03_T00_P00	1216406	262085	0.263	130.7989	0.7365	0.6969	26217	0.261	147.6395	0.6525	0.3422
	comptest_objects2-sandpit2_RWAC01_T00_P00	1161606	262083	0.255	246.8838	0.3902	0.8764	25917	0.262	282.3728	0.3411	0.281
	comptest_objects2-sandpit2_RWAC02_T00_P00	1230235	261957	0.264	262.0269	0.3676	0.8522	26047	0.267	314.5289	0.3063	0.3482
	comptest_objects2-sandpit2_RWAC03_T00_P00	1201786	261996	0.257	342.3312	0.2814	0.7105	25981	0.268	358.7709	0.2685	0.3371
	L0123-Composite	2601832	393170	0.406				38978	0.395			
	R0123-Composite	2654095	393129	0.385				39033	0.376			
	Average	1500113.167	294771.271	0.281916667	361.310617	0.4616125	0.743660417	29370.5	0.28440625	382.663846	0.4072125	0.32369375

Table 2 - Data collected from compression of images in OpenJPEG (JPEG 2000)

It can be observed from the table that the files in the 'ACE_sandpit_2xpct' folder were removed as mentioned in section 5.1 as well as the coloured composite images.

With the exception of real time, values with decimals - in most cases - have been given to four decimal places. In some instances, there is less than four however, this is due to the spreadsheet software removing trailing zeros. Secondly, real times are only to three decimal places (with the same practice of the removal of trailing zeros) as this is the maximum displayed by both OpenJPEG and the 'time' keyword in a bash shell.

Filenames have extensions struck since these are changeable given the file naming convention adopted for the project. Moreover, file sizes are posted in bytes, because if a larger unit was used, by rule of consistency, they would need to also be to four decimal places. However, if this was the case the table would not provide accurate values for the sizes of the files introducing a substantial amount of bad data into the results. This would be extremely detrimental to the validity of any subsequent analysis that may be performed, moreover, this situation would damage the repeatability of the research.

Finally, the average values in the final row of the table have been left as they are since these are generated through embedded functions in the spreadsheet and changing them is unnecessary. Moreover, it is yet another step in ensuring the accuracy of findings in subsequent calculations.

In the following section analysis will be conducted on this data and answers to the initial question proposed in section 2.2 extracted. Generated graphical representations of the data found in Table 1 and Table 2 will also be presented in section seven. This will be followed by a critical evaluation of the projects overall process in section eight.

7 Analysis

The process of analysis will be as follows; firstly, a general look into the simple statistics generated from the raw data and how they compare to each other to see if there are any preliminary conclusions that can be drawn and either proven or disproven later on in subsequent findings. This will be followed by a discussion as to the specifics of each of the two algorithms, this will include the magnitude of their compression when compared against time, statistical significance between metrics that appeared similar in the first stage of analysis, and briefly how certain metrics affect each other, for instance the effect a high Mean Squared Error value has on the Peak Signal to Noise Ratio. Finally, there will be a discussion as to the distribution of the data and what the presence of anomalies suggests about the different algorithms and their processes. Again, the aim of this research is to try and answer the overall question of:

Given the quality of compressed images, what reasons are there for the use of the CCSDS 122.0-B-2 compression standard on PanCam, and why are other image compression algorithms not used in the context of interplanetary communication?

7.1 Initial Observations

Post data collection, basic averages were calculated as a first step in order to try and establish any early patterns or avenues that could be explored in subsequent stages of analysis. These findings will be explored with respect to compression ratio.

At an x8 compression level, files computed in Bit Plane Encoder had a smaller average size than J2K at 131,072 bytes, as opposed to the average size of 294,771 bytes of files compressed to J2K. More interestingly however, computed files in Bit Plane Encoder were all the same size (131,072 bytes.) This is likely due to the emphasis given in space missions to the predictability of both software and hardware. Particularly given that this is a rover mission, it means that any issues or complications that may be encountered must have corrections and fixes made remotely. As a result, having all possible system operations be as reliably and predictably as possible, greatly reduces this risk meaning that there is a hugely increased probability that missions and research may be conducted with as few issues possible. Additionally to this, having a consistent size of image, that is required to be communicated back to Earth, means bandwidth and hardware resources can effectively be allocated and planned for to meet the demands of the volume of data being transmitted. This is supported by the fact that the time in which compression took place was an average of 0.0789 and 0.0402 seconds in Bit Plane Encoder against an average of 0.2819 and 0.2844 seconds in OpenJPEG at x8 and x80 compression rates, respectively. It may be argued that this is likely only due to the more thorough optimisation of wavelet transform operations over the plethora of distortions involved in J2K compression. However, it supports the idea that efficiency and operating using only the required resources are key considerations when working with equipment bound for space, especially if such equipment is autonomous and/or operated remotely from as far away as Earth is from Mars.

With regard to the Image Quality Assessment metrics for this level of compression. There were some interesting conclusions to be derived from the average results;

Table 3 - Comparative table of average scores for IQA metrics at 8x compression rate

(X8)	*BPE*	*J2K*
MSE	*16.4125*	*361.3106*
PSNR	*19.1702*	*0.4616*
SSIM	*0.7661*	*0.7437*

Table 3 illustrates these. Notice the difference in magnitude of the values for Mean Squared Error and the Structural Similarity Index. This would suggest that perceptively, according to the Human Visual System (as best it *can* be represented by the Structural Similarity Index) the two compression processes produce results that

are comparable. SSIM would suggest the images are very similar in quality and are relatively high in quality too. The Mean Square Error, however, suggests the opposite. In regard to the absolute value of Mean Squared Error, files computed through Bit Plane Encoder are of much higher quality -mathematically – than those computed through OpenJPEG. The implication of this is that J2K has been designed for consumption by a human observer, whereas the CCSDS standard, produces images that are capable of generating accurate results when run through algorithms to measure aspects of the environment on Mars for instance factors such as the reflective nature of surfaces, algorithms that rely on mathematical principles rather than human perception. Because of this factor alone, for the purpose of taking measurements and learning about conditions on other planets, it can be concluded that the CCSDS standard is superior to J2K, the perceived quality may be equitable, but for the purpose of scientific suitability, the CCSDS standard appears to be more capable.

For x80 compression, the results are similar to that of the lesser compression ratio. Bit Plane Encoder achieved an x80 compression in an average of 0.0402 seconds, whereas J2K did it in 0.2844 seconds on average.

Table 4 demonstrates the results.

Table 4 - Comparative table of average scores for IQA metrics at 80x compression rate

(x80)	*BPE*	*J2K*
MSE	*183.6473*	*382.6638*
PSNR	*1.6406*	*0.4072*
SSIM	*0.2484*	*0.3237*

Table 4 shows that perceptually J2K achieved better results given the Structural Similarity Index value. However, consistent with prior analysis, the CCSDS standard had a better mathematical performance proven by the lesser Mean Squared Error, despite having a greater ratio of distorting noise as per the Peak Signal to Noise Ratio. At this stage the difference in compression ratios become clear. At x8 and x80 compression, BPE reduced the file size to 8.74% and 0.87% of its original size, respectively. In comparison, J2K reduced the file size to 19.65% and 1.96% of its original size, respectively.

This means that despite a greater inventory of distortion methods J2K uses. The CCSDS standard still removed more information from the files, and as mentioned previously, achieves a better Mean Squared Error, similar Structural Similarity Indexes, and does this in both scenarios, with a greater ratio of distorting noise. The CCSDS standard can remove more information from a file, whilst maintaining a higher degree of relevant quality. This would suggest that the type and/or nature of the information being removed by each algorithm differs and does so to a degree that makes it noticeable.

Ultimately these findings would suggest that the algorithms have clearly been designed for different tasks. The CCSDS standard achieves maximum information

removal whilst simultaneously maximising mathematical similarity. Whereas the J2K algorithm achieves a standard that is more in keeping with a human's perception of high quality, compressing images to a lesser extent and slower, but retaining structural information that the Structural Similarity Index relies upon to output high values.

7.2 Algorithm Specifics

This set of figures contains graphs that represent the change in file size in Megabytes, against the real compression time in seconds.

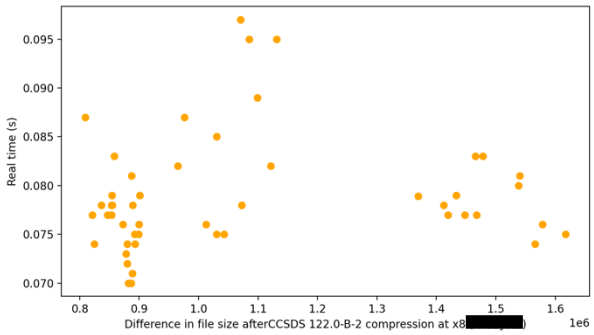


Figure 1 - Graph to show difference in file size (MB) against real compression time (s) for BPE at x8 compression

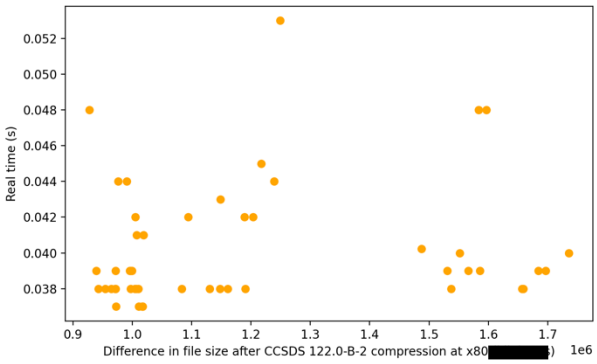


Figure 2 - Graph to show difference in file size (MB) against real compression time (s) for BPE at x80 compression

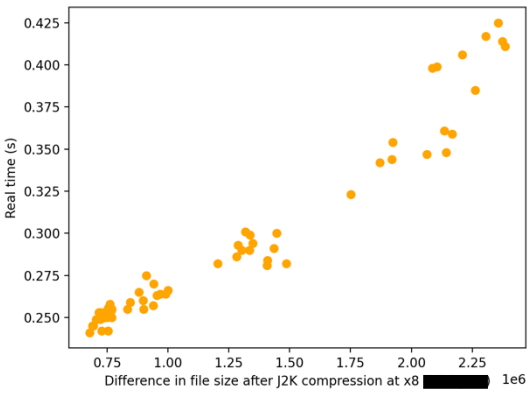


Figure 3 - Graph to show difference in file size (MB) against real compression time (s) for J2K at x8 compression

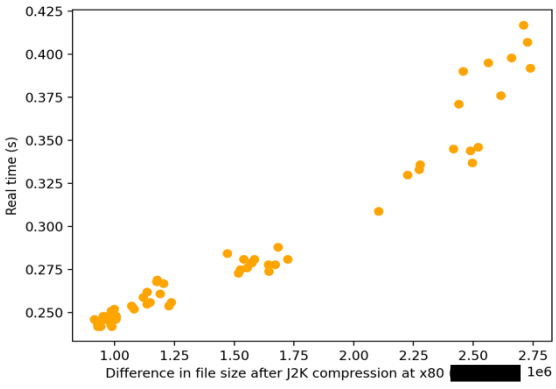


Figure 4 - Graph to show difference in file size (MB) against real compression time (s) for J2K at x80 compression

Table 5 - Table of PCC values for each of Figures 1-4

Pearson's Correlation Coefficient	BPE	J2K
x8 compression	0.1108	0.9445
x80 compression	0.1273	0.9530

It would be expected that for a greater compression, the time requirement to execute the compression would also increase. The figures above demonstrate this relationship for both algorithms and at both compression levels.

From the Figures above it can be observed that for both sets of graphs (Figures 1-2 and 3-4) there is distinct clustering occurring, specifically for Figure 1 between 0.8 – 1.2 MB and 1.35 – 1.6 MB, Figure 2, 0.9 – 1.3 MB and 1.45 – 1.8 MB, Figure 3, 0.75 – 1.6 MB and 1.75 – 2.5 MB, and Figure 4, 0.75 – 1.75 MB and 2 – 2.75 MB. Secondly, there is also an obvious difference in correlation between compressions in BPE and those in J2K.

As can be noted from Table 5, J2K compressions have a Pearson's Correlation Coefficient of much nearer one, given this near perfect linear trend, it may suggest that J2K is not as scalable as the CCSDS standard and that it is better suited to larger images such as those that can be taken by the HRC. Of course, this is down to the derivative of the CCSDS 122.0-B-2 algorithm that handles multichannel images, and so results may differ greatly. However, this may be another reason as to why the CCSDS standard is being used on Rosalind Franklin, in order to facilitate this scalability. Because images computed through Bit Plane Encoder have a near zero Pearson's Correlation Coefficient, this suggests that larger images may not take an increased amount of time to encode, this may follow the pattern of better reliability and predictability, knowing that for a given compression rate, the time required will most of the time fall between a given range.

Another possible explanation for this, however, is that the results produced by Bit Plane Encoder are more *erratic* and *unpredictable*. A zero Pearson's Correlation Coefficient means no general trend, it does not mean the values are guaranteed to fall within a given range. If this same method was run with a greater sample size of images, the result may be the production of a scatter graph that is truly 'scattered' - meaning no trends and accurate line of best fit. If this is the case, this is the opposite of the reliable, predictable nature of an algorithm bound for space, that has so far been discussed. This is highly unlikely though given the resources and people that have been invested into the creation of such an algorithm, however, in the nature of fullness and totality, it must at least be acknowledged as a possibility and as something that could potentially (although very unlikely to) compromise the workability of ExoMars.

To address the notion of clustered results, upon further investigation, this seems to simply be down to the initial distribution of the original file sizes.

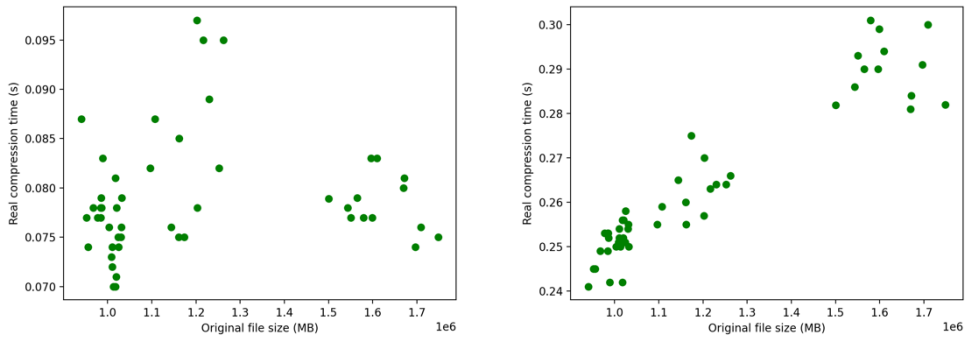


Figure 5 and 6 (Left and Right) - Graph of original image size against real compression time in BPE and OpenJPEG, respectively (x8 compression)

Figure 5 shows a state of clustering. When compared to Figure 1, it is clear that the scatter patterns are incredibly close in likeness, despite a small gain in lateral spread in some instances, most probably produced by the compression algorithm.

Figure 6 further supports this with again, clustering pre compression, however the scatter pattern in this image seems to contain a greater spread than when compared to Figure 3. This is because J2K compression contains a quantization step, which brings the spread of individual pixel values closer together and so results in the reduced spread seen in Figure 3.

Another reflection that can be had from the figures above is that the range of real compression time for files compressed to J2K overall, is the same as the range of real compression time for files compressed through Bit Plane Encoder, that is 0.06 seconds. However, when this range is considered proportionally to the maximum real compression time for each algorithm, a picture is developed with respect to the spread of data within this range. Although the ranges are the same, the data for real compression time in seconds has a greater spread of values for images computed through Bit Plane Encoder, than images computed in OpenJPEG. This is shown in the figures above, where in Figures 3 and 4, the compress times for both x8 and x80 compression rates appear to be very similar (i.e., in both instances between 0.250 – 0.425 on the vertical axis) whereas in Figures 1 and 2, for Bit Plane Encoder, the average time taken to encode images by x80, was significantly quicker at 0.0402 seconds, than x8 which was 0.07892 seconds on average. A difference of 0.03872 seconds. Although the range is equitable, for Bit Plane Encoder, this range is found at a lower magnitude of value.

A two-sampled t-test, is a method in which statistical significance can be determined for the difference between two sets of data. In section 7.1, it was noted that the Structural Similarity Index for both algorithms were very similar. Because of this similarity, performing such a test on the SSIM data will prove valuable in getting an understanding of if the two algorithms produce results that are perceptually - significantly different. Moreover, getting a T-value for the file size post compression, will provide another overall measure as to if there is a significant difference between the two algorithms and confirm that the process of identifying reasons for this significance, is a worthwhile way of proceeding.

SSIM T-test:

For this investigation the hypothesis and null hypothesis will be as follows:

h_0 = There **is no** significant difference between the SSIM of encoded images in BPE and J2K.
h_1 = There **is** a significant difference between the SSIM of encoded images in BPE and J2K.

A standard 95% significance level will be used for this investigation.

Using SciPy to calculate the p value for the Structural Similarity Index of both CCSDS compressed images and J2K compressed images yields a result of:

x8 p-value = 0.2999
x80 p-value = 0.0009

Since x8 p-value > 0.05, h_0 is accepted. Moreover, since x80 p-value < 0.05, h_0 is rejected. This implies that statistically, there is no significant difference between the perceptual quality of encoded images at a lower level of compression, however at higher levels, there is. Again, this suggest that when Bit Plane Encoder encodes files, the rate of decrease in perceptual quality, is greater, than that of an image compressed in J2K. Supporting the earlier implication that numerically, the CCSDS standard is superior, but J2K preserves more structural information and is more appropriate for human observers.

File Size T-test:

This test will follow the same process, with hypotheses' as follows:

h_0 = There **is no** significant difference between the compressed file sizes of the CCSDS compression standard and J2K compression
h_1 = There **is** a significant difference between the compressed file sizes of the CCSDS compression standard and J2K compression

x8 p-value = 0.0009
x80 p-value = 6.2385

x8 p-value < 0.05, and x80 p-value > 0.05.

Therefore, at x8 compression, there is a significant difference between compressed file sizes, but as this compression ratio increases, the difference between the rates of compression becomes less statistically significant. This suggests that given the initial x8 compression p-values, either Bit Plane Encoder's rate of change in distorting power per increase in compression ratio - reduces, or J2Ks rate of change increases,

either of these scenarios would explain the findings illustrated by the two-sampled t-test.

Finally, an illustration of the relationship between Mean Squared Error and Peak Signal to Noise Ratio.

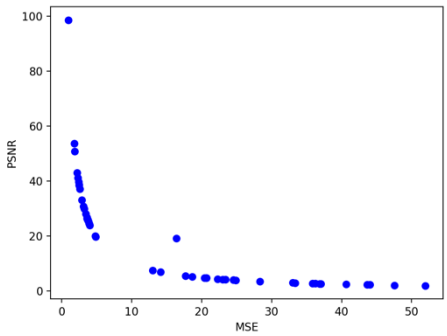


Figure 7 - Graph showing the relationship between MSE and PSNR for images encoded in BPE at x8

Figure 7 demonstrates the asymptotic relationship between the Mean Squared Error and Peak Signal to Noise Ratio, Image Quality Assessment methods. This is necessary as it is major testimony to the fidelity of the script that was developed in parallel to the research that was involved in this project.

Since Peak Signal to Noise Ratios value increases means a better quality, and a lower Mean Squared Error, also means better Quality. It would be expected that as MSE increases, PSNR decreases (but cannot be zero, as there is still noise in the compressed image.) This is the observable trend found in Figure 7.

Note that other graphs were constructed for the two algorithms and compression rates, however the scatter pattern is almost identical so the utility provided by their inclusion in this report would be negligible.

7.3 Distribution of Datum

A general comment on the distribution of data. Although, the distributions appear similar in respect to the Inter Quartile Ranges, these distributions occur at different magnitude ranges, as a result, there is still clear conclusions that can be drawn.

The following figures are a series of box plots of various data distributions for both algorithms, at both levels of compression. These images include plots of; the change in file sizes, distribution of Image Quality metrics, and compression times.

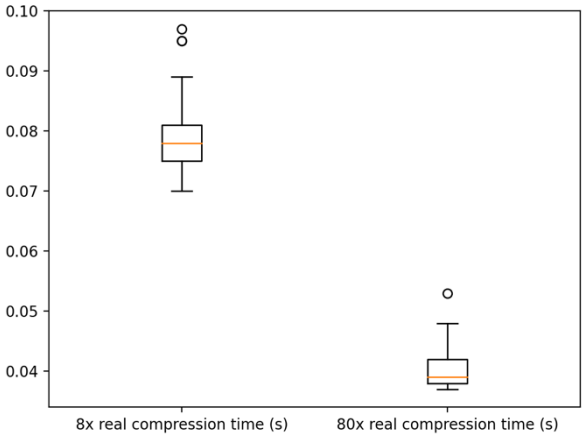


Figure 8 - Boxplot of compression times in BPE

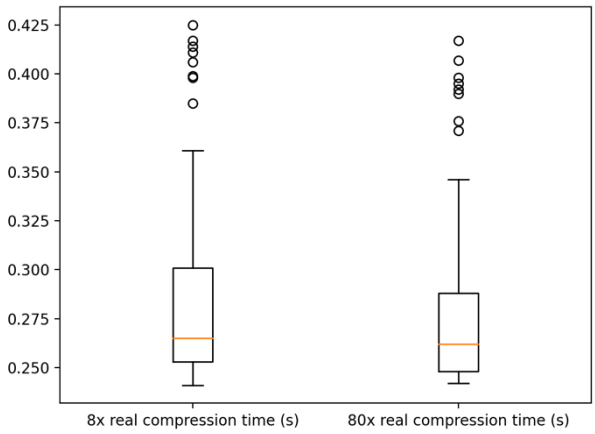
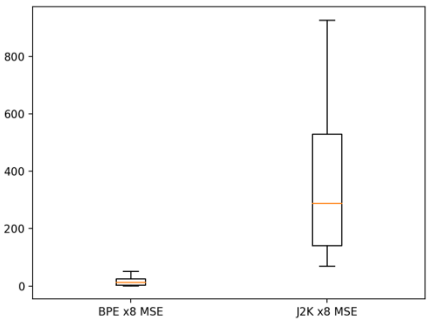
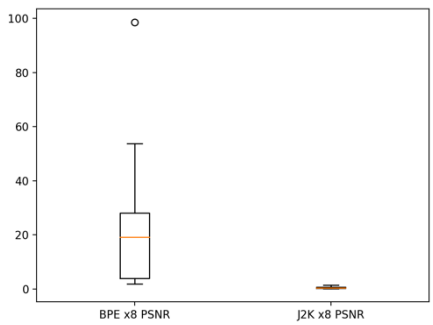
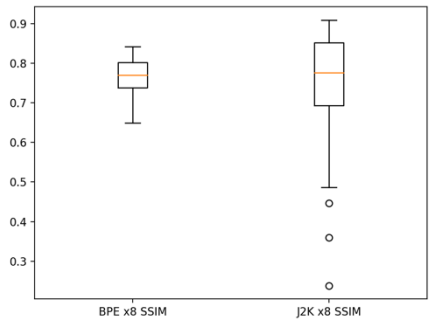


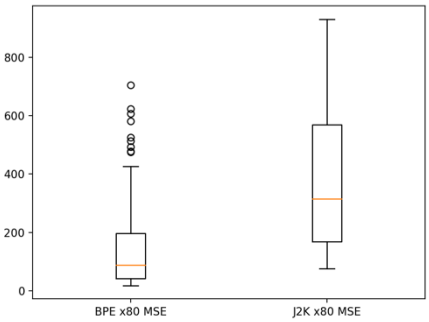
Figure 9 - Boxplot of compression times in J2K

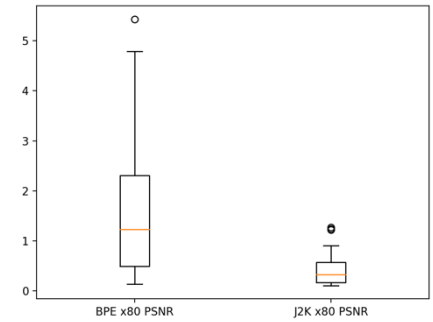


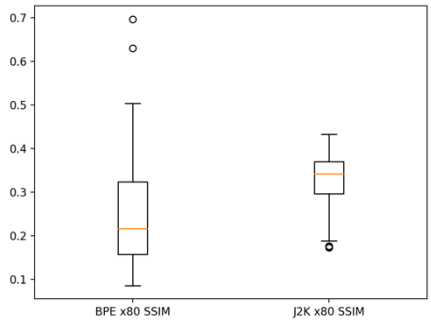




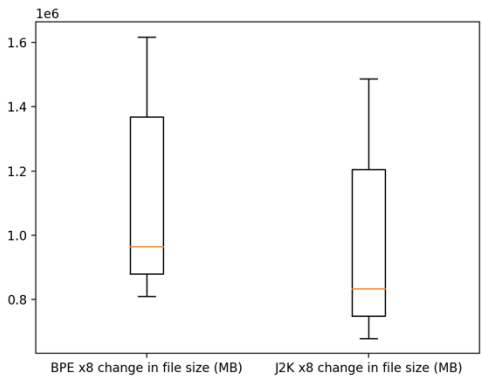
Figures 10, 11, 12 (Left, Centre, Right) - Boxplots of MSE, PSNR, and SSIM at x8 compression







Figures 13, 14, 15 (Left, Centre, Right) - Boxplots of MSE, PSNR, and SSIM at x80 compression



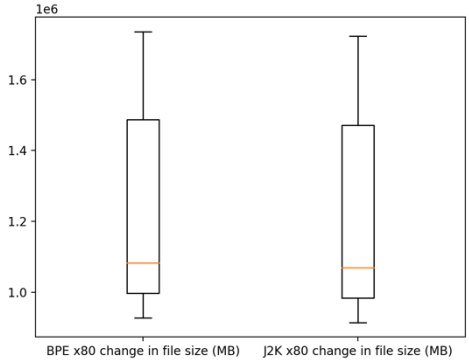


Figure 16, 17 (Left, Right) - Boxplots of change in file size

To begin, some noticeable comparisons that can be drawn from inspection of these figures. Figures 8 and 9, work to concur with the idea proposed in section 7.2, that despite the ten times difference in compression rates, J2K compressed the images all within a similar window of time. Whereas proportionally, Bit Plane Encoder took a lot longer to compress images at x8 compression than images compressed by x80.

One possible explanation for the reduced time at a greater rate of compression, could be the idea that, at x80 compression, there is fewer conditional operations that are required to be executed. In other words, the CCSDS algorithm is less selective with what information is removed since the user has requested a greater rate of compression, and so fewer checks are performed since a concession is made that the goal of maximum compression and maximum numerical quality, cannot both be achieved given such a high rate of compression.

It may also be ascertained from Figures 8 and 9, that there are much fewer anomalies in timings generated from Bit Plane Encoder, than there are generated from OpenJPEG. More reinforcement that the CCSDS standard is focused on process, as opposed to results, like J2K is, (results in the context of perceptual quality as determined by the Structural Similarity Index.)

The plots may help also in determining some of the more consistent processes, as well as the most inconsistent. For instance, Bit Plane Encoder compression time (Figure 8), BPE x8 Peak Signal to Noise Ratio (Figure 11), J2K x8 Structural Similarity Index (Figure 12), and BPE x80 Mean Squared Error (Figure 13), are among some of the more consistent data given the small interquartile ranges of their respective plots. In contrast, both algorithms change in file size at both rates of compression (Figures 16 and 17) both rates of J2K compressions Mean Squared Error (Figures 10 and 13) and BPE x8 Mean Squared Error (Figure 10) have plots demonstrating an inconsistent process.

Moreover, in most cases the data is skewed right (or in this instance vertically.) This means that for a given measure, the distribution is either not normal, or is not equally distributed around two distinct points, the data is biased towards the values of greater magnitude (in this instance.) The only cases where this seems to not be true is BPE x8 compression time (Figure 8) and J2K x80 Structural Similarity Index (Figure 15.)

Additionally, it should be noted how similar the distributions in change in file size are. Particularly at x8 compression (Figures 16 and 17). Again, the implication of this is that the same volume of information is removed, just the nature of the information is different, as mentioned in section 7.1.

The standard deviations for the change in file size are as follows:

Table 6 - Standard Deviations of change in file size (bytes)

Standard Deviation	BPE	J2K
x8	261615	202415
X80	261615	189855

Observing the standard deviation for the data that is represented in Figures 16 and 17 shows the accurate numerical similarity between the two distributions. Interestingly, Bit Plane Encoder, for both compression levels has a standard deviation of 261,615 bytes. Again, supportive of the idea of reliable, predictable algorithm design.

7.4 Interesting Cases

Throughout the data collection process, there appeared to be some instances of compressed images either not fitting the overall trend or standing out as cases that might prove more information if explored further. It is these images that will be discussed in this section, the factors that make them interesting cases described, and whether or not this has or could potentially have any bearing on the data as a whole or the overall scenario of image compression with these particular algorithms, at these particular compression rates.

Absolute Error Maps (AEMs) are a good way to visualise the error between an original image and a compressed version of the image. The developed script is capable of generating such maps by subtracting the pixel values of the original image from those pixels compressed by either algorithm and taking the absolute value. However, the pixel values must first be converted to be within the range of 0 – 255 (making the final image an 8-bit image), this is an obvious downside to this process. It is still unclear if this is an intended behaviour of the PIL library, however, the issue has been worked around and an implementation found within the developed script, to generate images that are still suitable for their intended purpose. That purpose being, human observation within this report as a way to help visualise what has changed during the process of compression, it is not a critical issue and so can be tolerated. Moreover, given that values for Mean Squared Error can range from 0 to ∞, and that in this project, the maximum Mean Squared Error across both algorithms and both compression levels does not exceed 1000. This leads to the generated Absolute Error Maps appearing very dark, this is because with a reduced level of error (indicative of the low MSE scores), there is a smaller numerical difference between the original and compressed pixel values. This means that the absolute values are smaller rather than larger (in most cases below 100.) Given in 8-bit, grayscale images, 255 represents absolute presence of white, this results in the images that are generated being dark. With this knowledge, visually speaking, the best compression rate to generate Absolute Error Maps will be x80, since this compression generates the most distortion and so pixel values skewed closer to 255, making the final image more visible as an absolute error map.

Some such interesting cases are those images that have the lowest quality scores. As mentioned prior, only x80 compressed images will be considered.

The following three Figures represent the images involved in the creation of, and the resultant, Absolute Error Map for one image in the sample. This particular image had the lowest Structural Similarity Index generated by Bit Plane Encoder, at a value of 0.1039.

Figures 19 and 20 representing the original and compressed versions respectively. Figure 18 is the absolute error map of these two images. The first glaring difference is the artifacts that have emerged on the lower right-hand side of the compressed version of the image.

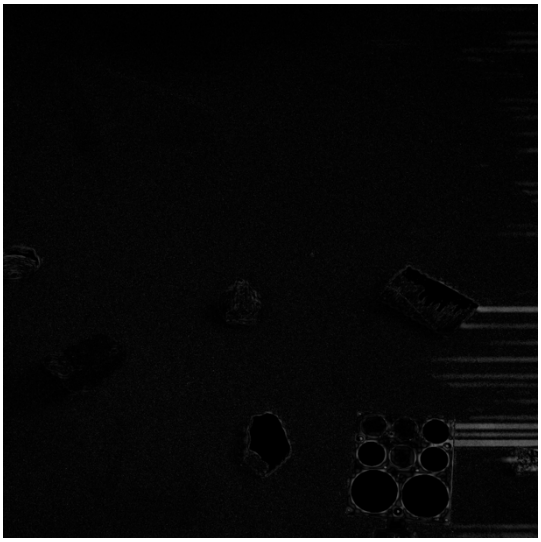


Figure 18 - Absolute Error Map

Figure 19 - Original image of AEM in Figure 18

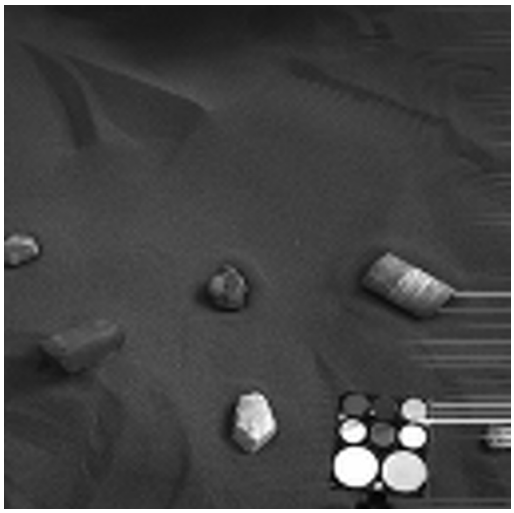


Figure 20 - Original image from Figure 19 compressed through BPE by x80

These created streaks show up clearly on the Absolute Error Map. The artifacts do extend along the whole right-hand edge of the image, however, the ones in the lower corner are more clear to see. Moreover, a second feature of the Absolute

Error Map (Figure 18) is the edges on some of the rocks. The outlines of said rocks can be made out in the map. This is one reason for the low Structural Similarity Index of this compressed image (Figure 20), because the gradient of change in pixel values between those within the bounds of the rock, and those outside, is lesser, than in the original. This creates a blurred edge between the object in question and the background. It is this edge that can be seen in the difference map. This same effect occurs on the calibration unit pictured in the lower right corner also. Along with the overall rectangular shape of the unit, the individual colour pallets can be made out. Finally, significantly less visible but still relevant, are imperfections in the sand background. This can be seen all over the space in the Error Map (Figure 18) appearing like static. Minute specs of lighter shaded Gray (near white) are visible. An error map can also be created from the compressed J2K version of this image – seen in Figure 21.

Figure 21 - AEM of images in Figure 19 and x80 J2K compressed version

In this version of the Absolute Error Map, the same issue with the sand background remains. However, the edges seen in this error map (Figure 21) are created more so from the difference in pixel value within the bounds of the given object. These differences effect the detailed patterning on objects, the texture. Given that this difference is consistent across the entirety of the image, it means that objects that have a greater difference in average pixel value - for instance the rock that appears white in the lower centre of the image, and the whiter colour pallets on the calibration unit - when compared to the neighbouring pixels of neighbouring objects, will stand out more in the Absolute Error Map (Figure 21.) Because the aforementioned objects are sitting on darker (lower pixel valued) objects (sand background and black background of calibration unit respectively) it means these objects are the ones that stand out the most on the Error Map (Figure 21) given the consistent nature of distortion that J2K applies over the whole image.

Another image that produces interesting results is that with the highest Mean Squared Error. This happens to be the first image processed. The collection of images for this case are as follows:

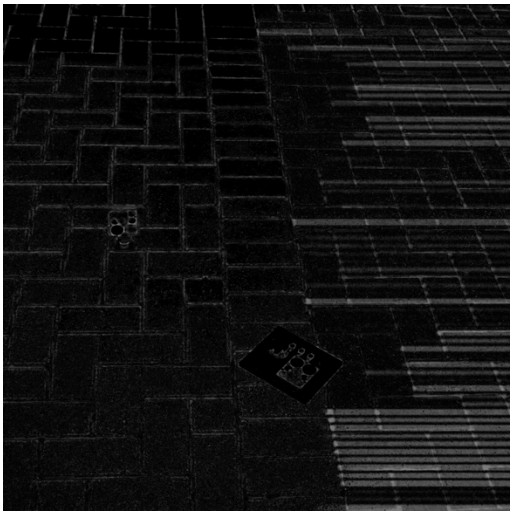


Figure 22 - AEM of x80 compressed image in sample image

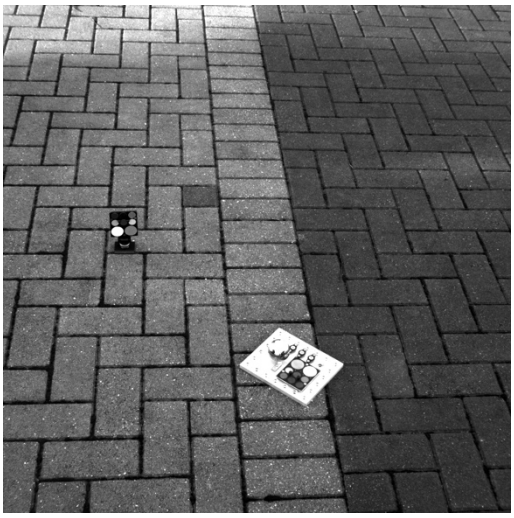


Figure 23 - Original image that would go to produce AEM in Figure

Figure 24 - Compressed version of Figure 23, x80 in BPE

As can be seen from Figures 22 – 24. This Absolute Error Map has the same characteristics as those explored in Figure 18, those being – the obvious presence of artifacts encroaching from the right-hand side of the image, the edges of objects being clear in the Error Map (Figure 22), and to a limited degree the same affect seen on the sand in Figure 18, but this time on the texture of the brickwork, making up the background.

Again, a second version of the Absolute Error map can be created using the J2K compressed version, that can be seen below in Figure 25:

Figure 25 - AEM of original image in Figure 23, with J2K x80 compression

In this version of the Absolute Error Map, the artifacts that in the previous version (Figure 22) protrude from the right-hand edge of the image are not present or cannot be seen with the same clarity. This is because J2K doesn't perform such an intrusive and extensive removal of data, proven by prior analysis. Because of this the chance of artifacts being introduced such as the ones observed in Figure 22, is greatly reduced. Only when J2K compressed images reach incredibly high levels of compression, does this become a concern. For reference, OpenJPEG allows compression ratios of up to x9000, at this point the quantization becomes overbearing and the image is reduced almost entirely to the same colour, most probably a value that is located somewhere centrally between the minimum (0) and maximum pixel values for that particular bit-depth of image i.e., the median value (although this hasn't been proven.) Secondly, it can again be observed that objects of high contrast, in the original image (and therefore also the compressed image, just possibly to a lesser extent) have this contrast preserved in the J2K version of the Absolute Error Map (Figures 21 and 25.) Moreover, another repeated observation is the 'static' effect from the background surface. Both algorithms seem to be limited in their ability to accurately reflect both natural and manmade (in this instance sand and brick respectively) surfaces. Given this is inevitable when Rosalind Franklin begins operating on Mars, it is fair to say this is a limitation of both algorithms. It may, however, be argued that J2K could be considered 'better' for this type of image, given its aversion to introducing artifacts, that could – in the right position of the right image – be confused for a feature of the subject, of said image.

Finally, something that hasn't yet been observed properly in these Absolute Error maps is the effect of shadows on the result. In Figure 25, the region in question can clearly be observed in the upper left corner. When comparing Figures 18 and 22, to Figures 21 and 25, there seems to be a particular lack of representation of this colour shift – particularly for Figures 22 and 25. Given that PanCam, among other things, will be seeking to gain an understanding of the terrain, the idea that the CCSDS standard is accurately compressing this type of colour change is one that is reassuring. Because the shift is so clear on the J2K compressed version, this means this information is not accurately recorded in the compression (since the difference

in the true and compressed pixel values is greater.) Another reason for the CCSDS standards superiority for missions in space.

7.5 Further Avenues of Research

Throughout this report reasons have been given to answer the question of:

Given the quality of compressed images, what reasons are there for the use of the CCSDS 122.0-B-2 compression standard on PanCam, and why are other image compression algorithms not used in the context of interplanetary communication?

This report explored two different compression ratios, for two different algorithms. The next most obvious question is how much compression? How much compression can be applied to an image before said image becomes unusable by the software and algorithms that require it? Answering such a question would involve testing the algorithm that has been selected to be involved in operations with PanCam (CCSDS compression standard) against multiple images -similar to those used in this project – at multiple levels of compression to try and ascertain a level of compression for that image that is suitable for the task it is required for.

Because the images taken by PanCam include a default level of compression, and a compression level for 'quick look' images (images destined to be observed by humans) answering such a question may require a study to be performed. A study that involves human observers to give a perceived quality score. This may then be translated into a mean opinion score and used alongside other Image Quality Assessment metrics – possibly including but not limited to the ones used in this project – to aid in the determination of suitable levels of compression. It is important to note however, that any subsequent research should involve the coloured composite images. It will require access to the second CCSDS compression algorithm that is capable of processing such images but means the research would be more representative of scenarios PanCam and Rosalind Franklin will find themselves in, i.e., having to send both colour and Gray scale images back to Earth.

Secondly, another question that's answers could prove useful is that of - Which collection of Image Quality Assessment algorithms or processes are best suited for the assessment of images taken in the context of space missions?

As has been seen in this project, not one Image Quality Assessment algorithm can be used in isolation, particularly the non-Human Visual System based algorithms. They all provide different information and different 'reasonings' for the quality scores they provide. In this report it has been noted that compression algorithms have interesting effects on various scenarios seen in the sample images, that ultimately may be, and probably will be, present in actual images taken by PanCam, for instance, the effects looked at caused by intricate/small details and shadows in images and how the presence of artifacts may affect the quality score of images. In all, one particular collection of Image Quality Assessment algorithms will provide a more accurate overall conclusion on the quality of an image – given its intended use

– than another, answering this question will involve testing these compositions to find one that is most optimal.

It goes without saying, that there will be more questions that arise from this report, however, the two questions listed in this section are the most substantial in furthering understanding of Image Quality Assessment in a more general setting, and Image Quality Assessment in the context of space missions.

8 Critical Evaluation

To begin, a note on the state of the project. Selecting a research question for this project was complicated. The focus of the project changed from strictly a comparison of Image Quality Assessment algorithms, to one of using them to assess compression algorithms. Because of this change, the research that had gone into the project in the initial weeks became less useful and relevant, and the limited preliminary reading that was done before the project began became a crutch. In the first version of the project, the research that was done was substantial, and thorough. Post project change however, this research proved to be of limited use. When establishing the direction that the project would take in the opening weeks. It would have proved beneficial to go in a direction that relied less on technical and substantial input from others in order to execute the original plan for the project.

This project was a research project and as such the ultimate goal was to answer the research question. In doing this, success was found, however, it could have been achieved to a greater extent. Statistical evaluation of the collected data was performed as noted in sections 7.2 and 7.3. However, this is arguably not comprehensive enough to leave the question answered to an extent that means it is not up for dispute. Moreover, in providing reasons to answer the research question. A few well supported and well substantiated reasons were identified. In review it may have been better to establish a greater assortment of reasons, this would have proven the extent of the research and analysis, and would have required said analysis to have been more substantial and thorough; a good remedy to this issue.

The project outline that was created at the start of the project, made clear not only the direction the research would go to begin with, but also listed project deliverables. The direction the project has taken has already been discussed. The deliverables listed in that document included:

- This report
- A log or diary for the project
- Bank of references

In all, these deliverables have been achieved and met, however, to comment on the bank of references. The extent of referencing in this report is not representative of the extent of material that was read and recorded in the GitHub for this project. The reason for this is the project change that occurred. It goes without saying however that this would not be the case had image compression methods been the subject of

research in the initial stages of the project, beyond that of preliminary reading, before the start of the project. Additionally, the proposed tasks all proved necessary tasks and were all achieved during the lifecycle of the project.

To discuss the developed script that proved crucial in data processing but also very valuable to data analysis. It has become more of a small library, incorporating many others. It has been tailor made for the specific set of tasks involved in the production of this report and the execution of this project. It could have had, and probably would benefit from, some refinement. Possibly creating a Graphical User Interface, or even just a simple text-based UI. The former, was outside the scope of this project, particularly given that this is a research project, and that the technical submission was never going to play as an important role as it may have in other more engineering-based projects. The latter was never of a high enough priority to warrant doing. However, if this research was to be continued and taken further, this would absolutely be an issue that would be resolved. The script does however enable the computation of the three Image Quality Assessment algorithms used in the project, the collection and manipulation of pixel data from different types of images, the generation of Absolute Error Maps between two images, and general statistics used in this project. Ultimately making it very valuable in terms of analysing images in the way that has been done in this project. Optimisation is still required but the script serves its purpose well in this context.

Regarding the overall process of the project. The evolution of specific ordering of tasks was very natural. The first few weeks involved doing the majority of the background reading and information collection. This time was spent reading papers on how Image Quality Assessment methods work, are categorised and how comparisons between them may be made. When sample images became available data started to be collected before running the statistical tests and recording them in this report. This was beneficial for maximising working time since the process didn't have to be planned to as fuller an extent as a project of an engineering nature. However, it did mean that there was an increased reliance on the project diary for recording the process. Since there was no documented process plan, recording what *had* been done was the next best thing. This works nearly as well, however is limited when it comes to proving originality of thought and process. Unfortunately, for this project, this mainly took the form of rough handmade notes, not available on this projects GitHub repository. This is something that would definitely be changed if the project was to be revisited or further research conducted.

If this process was to be conducted again and essentially restarted. It would be prudent for experimentation with various tools, libraries, and algorithms to have begun sooner in the project's lifecycle. This would have allowed more time for analysis, optimisation of any developed technical work, and thew resolution of any issues that may have presented themselves at later stages.

Overall, the project was challenging and yet interesting. There were some substantial difficulties throughout the process, as well as many more trivial issues, however these were dealt with as best as possible and ultimately what has been performed, is

some original research leading to some informed conclusion and the generation of a small technical submission along with this report to demonstrate those findings.

9 References

1. Clark, A.M. (1998), The qualitative-quantitative debate: moving from positivism and confrontation to post-positivism and reconciliation. Journal of Advanced Nursing, 27: 1242-1249. https://doi.org/10.1046/j.1365-2648.1998.00651.x

2. Business Research Methodologies https://research-methodology.net/research-methods/ - Accessed 20/04/2022

3. Bit plane encoder http://hyperspectral.unl.edu/ - Accessed 22/04/2022

4. Bit-Plane Slicing https://theailearner.com/2019/01/25/bit-plane-slicing/ - Accessed 22/04/2022

5. F. Auli-Llinas and M. W. Marcellin, "Scanning Order Strategies for Bit plane Image Coding," in IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 1920-1933, April 2012,doi:10.1109/TIP.2011.2176953.

6. Description of the process of wavelet transforms - https://www.youtube.com/watch?v=kuuUaqAjeoA – Accessed 22/04/2022

7. A Tutorial of the Wavelet Transform – Chun-Lin, Liu; 2010 https://www.researchgate.net/profile/Vladimir-Kulchitsky/post/Is-there-any-difference-in-the-tiling-of-the-time-frequency-plane-by-the-orthogonal-and-biorthogonal-wavelet-basis-functions/attachment/59d629c179197b807798844a/AS%3A337039875690496%401457367976408/download/WaveletTutorial.pdf - accessed 22/04/2022

8. JPEG2000 compression https://jpeg.org/jpeg2000/ - Accessed 22/04/2022

9. OpenJPEG GitHub repository https://github.com/uclouvain/openjpeg/tree/v2.4.0 - Accessed 22/04/2022

10. Umme Sara, Morium Akter, Mohammed Shorif Uddin, “Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study” (2019) **DOI:** 10.4236/jcc.2019.73002

11. Zhou Wang, Alan C. Bovik, “Modern Image Quality Assessment Synthesis Lectures on Image, Video, and Multimedia Processing” - (2006), Morgan & Claypool https://doi.org/10.2200/S00010ED1V01Y200508IVM003

12. Yusra A. Y. Al-Najjar, Dr. Der Chen Soong “Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI” International Journal of Scientific & Engineering Research, Volume 3, Issue 8, August-2012 1 ISSN 2229-5518 https://www.ijser.org/researchpaper/Comparison-of-Image-Quality-Assessment-PSNR-HVS-SSIM-UIQI.pdf

13. Wang, Z.; Simoncelli, E.P.; Bovik, A.C. (2003-11-01). Multiscale structural similarity for image quality assessment. Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2004. Vol. 2. pp. 1398–1402 Vol.2. doi:10.1109/ACSSC.2003.1292216. ISBN 978-0-7803-8104-9.

14. ImageMagick - image modification tool https://imagemagick.org/script/index.php - Accessed 23/04/2022

22. PILLOW documentation on image modes - https://pillow.readthedocs.io/en/stable/handbook/concepts.html#concept-modes – Accessed 23/04/2022

23. NumPy.fromfile() function - https://numpy.org/doc/stable/reference/generated/numpy.fromfile.html - Accessed 23/04/2022

10 Appendix A – Third-Party Code and Libraries

15. Python Image Library (PILLOW) https://python-pillow.org/ - Accessed 23/04/2022

16. SciKit-Learn https://datagy.io/python-scikit-learn-introduction/ - Accessed 23/04/2022

17. Scikit-image https://scikit-image.org/ - Accessed 23/04/2022

18. Numpy https://numpy.org/ - Accessed 23/04/2022

19. SciPy https://scipy.org/ - Accessed 23/04/2022

20. MatPlotLib https://matplotlib.org/ - Accessed 23/04/2022

21. Pandas https://pandas.pydata.org/ - Accessed 23/04/202