

Analysis of LAPAN-IPB image lossless compression using differential pulse code modulation and huffman coding

P R Hakim* and R Permala

Satellite Technology Centre, Indonesian Aeronautics and Space Agency (LAPAN),
Rancabungur, Bogor 16310, Indonesia

E-mail: patriarachmanhakim@yahoo.com

Abstract. LAPAN-A3/IPB satellite is the latest Indonesian experimental microsatellite with remote sensing and earth surveillance missions. The satellite has three optical payloads, which are multispectral push-broom imager, digital matrix camera and video camera. To increase data transmission efficiency, the multispectral imager data can be compressed using either lossy or lossless compression method. This paper aims to analyze Differential Pulse Code Modulation (DPCM) method and Huffman coding that are used in LAPAN-IPB satellite image lossless compression. Based on several simulation and analysis that have been done, current LAPAN-IPB lossless compression algorithm has moderate performance. There are several aspects that can be improved from current configuration, which are the type of DPCM code used, the type of Huffman entropy-coding scheme, and the use of sub-image compression method. The key result of this research shows that at least two neighboring pixels should be used for DPCM calculation to increase compression performance. Meanwhile, varying Huffman tables with sub-image approach could also increase the performance if on-board computer can support for more complicated algorithm. These results can be used as references in designing Payload Data Handling System (PDHS) for an upcoming LAPAN-A4 satellite.

1. Introduction

LAPAN-A3/IPB satellite brings one multispectral push-broom imager with four color channel (RGB-NIR), one digital matrix camera with Bayer color filter array (CFA) filter, and also one analog video camera [1]. With current satellite altitude of 505 km, the multispectral imager will have approximately 15 meter ground sampling distance (GSD) with swath-width of 100 km, while the matrix camera will have around 3 meter resolution with 6 km square coverage area. Special attention must be addressed on the multispectral push-broom imager data transmission, since each channel has 8002 detector, where each detector is represented by 16-bit digital number. Therefore, LAPAN-IPB has 105Mbps data-rate transmission to accommodate real-time transmission of the multispectral imager data. Even though of having quite high transmission rate, LAPAN-IPB multispectral image data still need to be compressed to further optimize bandwidth saving without sacrificing decompressed image quality. In LAPAN-IPB satellite on-board data processing, the multispectral image can be compressed in lossy or lossless method. In general, lossless compression will reduce the image size without affecting its quality, while lossy compression will reduce the size more efficient, but it reduces the quality of the decompressed image [2].

There are several methods that can be used for image compression. Consultative Committee of Satellite Data System (CCSDS) recommends the use of Discrete Wavelet Transform (DWT) for both



satellite image lossy and lossless compression [3]. However, DWT algorithm implementation in satellite on-board computer is quite challenging, in term of algorithm itself and in hardware implementation. Therefore, LAPAN-IPB uses Fast-Fourier Transform (FFT) for lossy compression and Differential Pulse Code Modulation (DPCM) for lossless compression, where both methods use Huffman entropy-coding. The analysis of LAPAN-IPB image lossy compression has been done [4], and this paper deals with the analysis of lossless compression one. In general, LAPAN-IPB lossless compression scheme uses DPCM to de-correlate image pixel values while Huffman table is used to encode the resulted DPCM values, using basic concept of representing a more frequent value with fewer number of bit [5].

There are several potential issues that can be discussed in the lossless compression algorithm that has been implemented in LAPAN-IPB. First thing that will be analyzed in this paper is the use of one single neighboring pixel in DPCM algorithm of LAPAN-IPB. In DPCM algorithm, there are several options that can be used to calculate the value of de-correlated pixels before entropy-coded by Huffman table [6]. It consists of the simplest case of taking differences between current and one previous pixel in the same line, which is the method used in LAPAN-IPB, and the more complex form that taking average of two or more neighboring pixels. In general, the trade-off between compression performance and hardware complexity should be treated carefully to decide which type of DPCM should be used in compression algorithm.

The other important thing in LAPAN-IPB lossless compression method that will be discussed is the use of static Huffman table to encode every image captured, although the table can be updated for any acquisition period. One particular static table might be good to compress some particular images, but might produce quite bad performance when applied to other different type of images [7]. For example, the camera could capture town, river, mountain, desert, beach or ocean in one single acquisition period, where each of them has different image characteristic that should be encoded using different Huffman table to produce the best performance. However, the implementation of static Huffman table is much easier compare to variable Huffman table that changes continuously between images. Again, the trade-off between performance and computational-cost should be considered when designing the Huffman coding scheme that gives the best result.

In general, the Huffman entropy-coding will work optimally on homogenous data, and in case of image data, the histogram of pixel value should have small standard deviation. In LAPAN-IPB image lossless compression scheme, due to on-board computer limitation, the Huffman algorithm is applied to one single image line. This is in fact, has made DPCM values compressed by Huffman coding be less homogenous. The third issue that will be investigated in this paper is the possibility of using sub-image data, instead of one single line, when applying Huffman coding algorithm. Given the same data buffer to process DPCM values, it is interested to see whether bigger-image data will give significant improvement or not.

This paper aims to analyze LAPAN-IPB lossless compression algorithm, particularly with current LAPAN-IPB on-board computer limitation. Based on the analysis done, this paper will also propose an improved compression scheme, considering several above-mentioned compression aspects, which are the type of DPCM code used to de-correlate pixel values, the type of Huffman entropy-coding used to compress the result of de-correlation process, and also the use of sub-image approach for entropy coding. Apart of being used as references to optimize LAPAN-IPB multispectral imager real-time data acquisition, these analysis results will also used as reference for designing Payload Data Handling System (PDHS) that is currently being developed for the next LAPAN-A4 satellite.

2. Methodology

First step on this research is study literature about image compression in general with lossless method in particular. In general, image compression consists of image de-correlation process and followed by entropy-coding process [8]. De-correlation process is done to exploit the similarity value between neighboring pixels, so that the de-correlated values will have more uniform pattern compare to original image data, therefore can be encoded more efficient by the entropy-coding. There are many

image de-correlation algorithms that can be used for image compression purpose, such as Discrete-Cosine Transform (DCT), Fast-Fourier Transform (FFT) and Discrete Wavelet Transform (DWT) [9]. Meanwhile, entropy-coding process is done to compress the resulted de-correlated values, where more frequent values will be encoded with fewer number of bit compare to less frequent values. Some common examples of entropy-coding algorithm are Huffman and arithmetic coding [10]. Figure 1 shows general block diagram of image compression process. De-correlation process is the heart of image compression algorithm since it distinguishes image data compression from any other ordinary data compression.

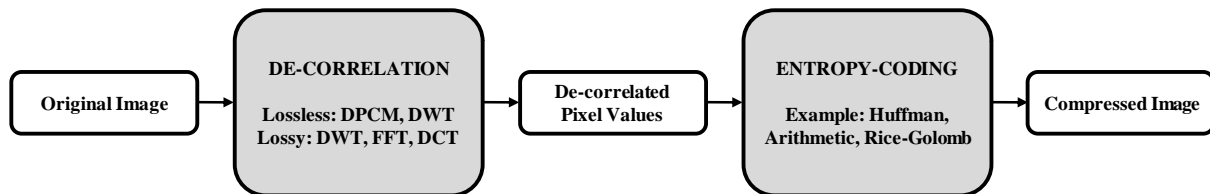


Figure 1. General block diagram of image compression process

Next, LAPAN-IPB lossless compression algorithm is modeled and implemented using MATLAB software to determine the quality of compression algorithm. In LAPAN-IPB lossless compression, DPCM is used to de-correlate pixel values and static Huffman table is used to encode (compress) the de-correlated pixel values. To provide robustness to accommodate various types of image, special-character is added to Huffman table to limit number of bits generated by the entropy-coding process. Figure 2 shows flowchart of image lossless compression that is used in LAPAN-IPB satellite.

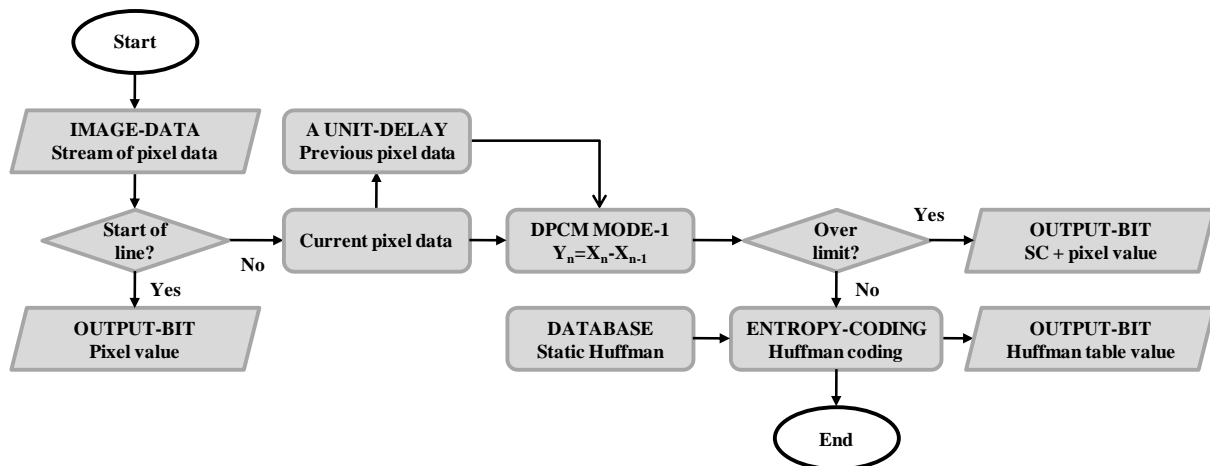


Figure 2. Flowchart of LAPAN-IPB image lossless compression

Finally, several tests and simulations are done to determine the quality of LAPAN-IPB lossless compression, such as the averaged nominal compression ratio produced and how well single static Huffman table adapt to various type of image. Since lossless compression does not reduce the quality of decompressed image, the only important parameter that will be considered is image compression ratio, which is ratio of number of bits of original image compare to the compressed image [11]. Based on these simulation results, several additional simulations are done to see if it is possible to improve the compression quality by changing the type of DPCM and Huffman table used, as well as the use of sub-image as the input-image for entropy-coding process. Figure 3 shows flowchart of experiment and simulation done in this research.

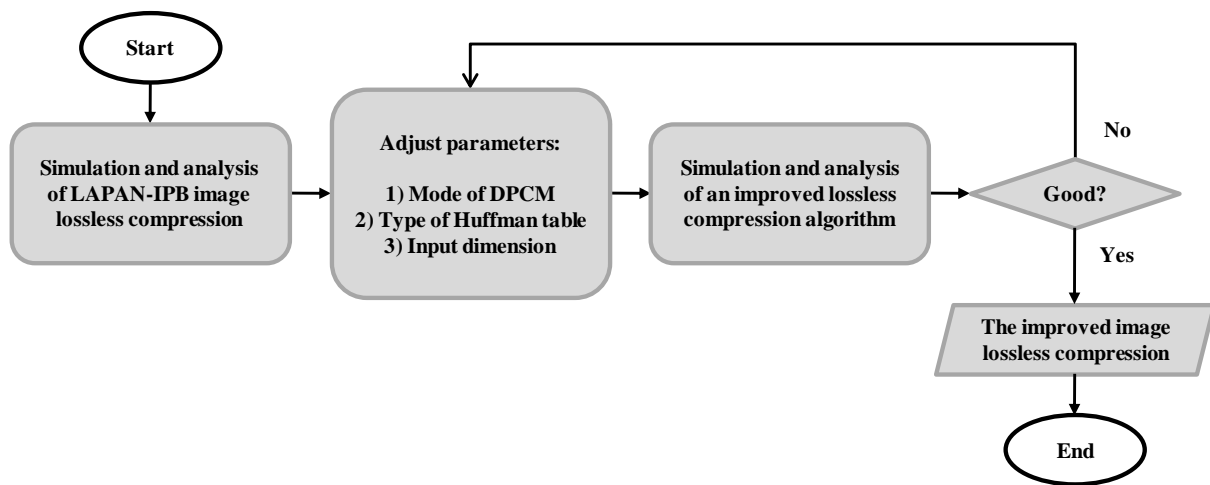


Figure 3. Flowchart of experiment and simulation in this research

3. Simulation Results

Based on figure 3, the experiment and simulation done in this research can be divided into two parts. In the first part, LAPAN-IPB image lossless compression performance will be simulated and analyzed using MATLAB and JAVA software, both for encoding and decoding process. Meanwhile in the second part, several algorithm improvements will be analyzed using MATLAB software, including the different type of DPCM and Huffman-table used in compression algorithm, and multi-dimensional image-input for entropy coding.

3.1. LAPAN-IPB lossless compression

As mentioned earlier in figure 2, LAPAN-IPB image lossless compression uses the simplest method of DPCM algorithm, where it takes a simple difference between current pixel and previous pixel value to be feed for Huffman-coding. Static Huffman-table is then used to encode resulted differenced value. The compression algorithm is simulated by using six images, which consists of three artificial images and three images taken during LAPAN-IPB payload flight test campaign. The flight test images used in this simulation are shown in figure 4. Table 1 shows simulation results, where the first three are for artificial images and the last three are for flight test images. Ideal compression ratio can be achieved when each image is encoded using ideal Huffman-table corresponds to the image, which gives a better compression ratio compare to when static Huffman-table is used to encode all images. In worst-case scenarios, where the static-table has significant characteristic difference to the image, the compression ratio can be reduced significantly.



Figure 4. LAPAN-IPB payload flight test images used in simulation: urban area (image-4) in the left, farmland (image-5) in center, and river area (image-6) in the right

Table 1. Compression ratio of LAPAN-IPB image lossless compression

Image Compressed	Ideal Ratio	Static Huffman (mean)		Worst-Case Scenario	
		Ratio	Reduction (%)	Ratio	Reduction (%)
Image-1	1.432	1.259	13.74	1.109	29.13
Image-2	1.904	1.747	8.99	1.658	14.84
Image-3	1.979	1.861	6.34	1.763	12.25
Image-4	1.762	1.703	3.46	1.640	7.44
Image-5	1.774	1.703	4.17	1.641	8.10
Image-6	2.346	2.163	8.46	1.886	24.39
Performance	1.826	1.695	7.73	1.570	16.31

LAPAN-IPB image lossless compression algorithm uses special-character to minimize the above problem. If the resulted differenced DPCM value is beyond a pre-determined range value, then it produces special-character bits appended to its pixel value. The Huffman table is only applied if the resulted differenced DPCM value is inside the range. The special-character bits are chosen carefully so that it can be differentiated from all other Huffman-table output. The use of this special-character can improve compression ratio slightly, as can be seen in Table 2.

Table 2. Influence of special-character to improve compression ratio

Image Compressed	Ideal Ratio	Static Huffman (mean)		Worst-Case Scenario	
		Ratio	Reduction (%)	Ratio	Reduction (%)
Image-1	1.432	1.290	11.01	1.156	23.88
Image-2	1.904	1.820	4.62	1.723	10.50
Image-3	1.979	1.898	4.27	1.765	12.12
Image-4	1.762	1.704	3.40	1.641	7.37
Image-5	1.774	1.703	4.17	1.641	8.10
Image-6	2.346	2.163	8.46	1.886	24.39
Performance	1.826	1.720	6.16	1.595	14.48

3.2. Differential Pulse Code Modulation (DPCM)

In DPCM algorithm, there are seven modes that can be used to calculate de-correlated value that will be feed to Huffman-coding [6]. Usually, the neighboring pixels that are used in DPCM calculation can be labeled as W (west), N (north), and NW (north-west), corresponding to its location based on current pixel under consideration. In standard image processing, pixel data is processed by line from north to south and by column from west to east. For example, LAPAN-IPB lossless compression uses DPCM mode-1 that differentiates current pixel with previous pixel in the west, therefore can be labeled as $Y=P-W$, where Y is de-correlated value, P is current pixel value and W is west-side pixel value. With the same naming convention, DPCM mode-2 uses $Y=P-N$ and mode-3 uses $Y=P-NW$. Meanwhile, the rest DPCM modes uses more complex form, where DPCM mode-4 uses $Y=P-(W+N-NW)$, mode-5 uses $Y=P-(W+(N-NW)/2)$, mode-6 uses $Y=P-(N+(W-NW)/2)$ and mode-7 uses $Y=P-(W+N)/2$. Table 3 shows compression ratio produced by each of these DPCM modes, using the same six images that are used in previous LAPAN-IPB lossless compression analysis. It can be seen that the last four modes which are use more than one neighboring pixels, give a much better compression ratio compare to the first three modes which are only use one single neighboring pixel. This shows that LAPAN-IPB lossless compression performance can be improved by approximately 10 percent by using one of these multi-neighboring pixel modes, which are mode-4, mode-5, mode-6 and mode-7. DPCM mode-7 in particular has slight advantage to the other multi-neighboring pixel modes since this mode has the simplest equation and therefore lowest computation complexity.

Table 3. Performance of different type of DPCM calculation

DPCM	Image Compression Ratio						Average
	Artificial Dummy-Image			Digital Camera Image			
	Image-1	Image-2	Image-3	Image-4	Image-5	Image-6	
Mode-1	1.432	1.905	1.979	1.762	1.774	2.346	1.826
Mode-2	1.428	1.833	1.928	1.789	1.803	2.370	1.818
Mode-3	1.337	1.717	1.796	1.642	1.638	2.128	1.678
Mode-4	1.513	1.903	2.096	1.948	1.973	2.567	1.952
Mode-5	1.536	1.991	2.146	1.955	1.985	2.652	1.992
Mode-6	1.533	1.961	2.116	1.970	2.002	2.666	1.989
Mode-7	1.527	1.999	2.130	1.947	1.961	2.662	1.984

3.3. Different Huffman table implementation

Based on Table 1, ideal compression ratio can be produced if the image is encoded using its own Huffman-table that represents its histogram characteristic. This can be done by calculating histogram distribution of de-correlated values for each image before entropy-coding. This histogram distribution is then used to form Huffman-table corresponds to the de-correlated image, therefore compression ratio produced can be considered ideal (best). However, this method needs massive memory to store entire pixel image data, as well as complex on-board circuit to calculate Huffman table value. Note that this method is different from adaptive-Huffman coding, where the value of Huffman-table is progressively changing respect to new pixel data calculated by the algorithm. Adaptive-Huffman coding is beyond the scope of this research, in this research different Huffman-coding implementation in entropy-coding are investigated.

The idea is to use the best possible Huffman-table to encode each image. Since the ideal Huffman-table is expensive to calculate, then this research proposes the use of parallel encoding using a set of different static Huffman-table for each image. Each image is encoded several times, separately but simultaneously, using several pre-determined table, and the best compressed image is then produced as the output. The index of which the table is used can be inserted in the header of the image, to provide information for decoder to decompress the compressed image. Table 4 shows the compression ratio produced by using this parallel encoding compare to original LAPAN-IPB lossless compression algorithm. It can be seen that the compression ratio can be improved by approximately 5 percent. Note that the result can be different if using another set of static Huffman table, but this simulation shows the idea of how parallel Huffman-coding could improve the compression performance.

Table 4. Performance of parallel Huffman-coding

Huffman-coding	Image Compression Ratio						Average
	Artificial Dummy-Image			Digital Camera Image			
	Image-1	Image-2	Image-3	Image-4	Image-5	Image-6	
Ideal	1.432	1.904	1.979	1.762	1.774	2.346	1.826
Static Huffman	1.259	1.747	1.861	1.703	1.703	2.163	1.695
Parallel encoding	1.359	1.840	1.933	1.741	1.745	2.292	1.775
Improvement	7.91	5.31	3.89	2.22	2.48	5.98	4.74

Another aspect that was investigated was about dimension of image data that will be encoded by Huffman-coding. In LAPAN-IPB lossless compression, and also in the proposed parallel Huffman-coding, single Huffman-table is applied to entire image. Actually this is not the best condition to exploit the similarity between pixels. In complete images, anything can be captured by the imager, ranging from homogenous object like ocean or desert, to heterogeneous object such as city with traffic and buildings. This diverge object characteristic usually cannot be encoded optimally to produce the best compression ratio. The idea is to divide the complete image into several sub-images, hoping that

each sub-image might have homogenous characteristic, thus can be encoded optimally using parallel Huffman-coding proposed above. Therefore, each sub-image might be encoded by using several pre-determined Huffman-table, each one is best table corresponds to its sub-image, and table-indexing is still same as before, inserted in the beginning of each sub-image data. However, based on several simulations that have been done, this approach gives insignificant performance improvement, only under 5 percent compression ratio increase.

4. Analysis

All of simulations that have been done in this research show that there are some aspects of LAPAN-IPB lossless compression algorithm that can be improved to increase compression ratio. The use of DPCM mode 4-7 could increase the compression ratio about 10 percent, while parallel Huffman table and sub-image approach could further increase the ratio by 5 percent each. If all of these proposed improved methods can be realized, then the lossless compression algorithm could produce about 20 percent better compression ratio compare to LAPAN-IPB algorithm. However, some of the proposed method needs additional on-board processing resources and power, and depending on Payload data Handling System (PDHS) that the satellite has, these methods might not be able to be implemented. This section discusses the trade-off analysis between the compression performance improvement achieved by the proposed method and additional on-board processing complexity needed, in order to see which proposed improvement technically viable to be implemented.

Mode-1 of DPCM algorithm that is used in LAPAN-IPB lossless compression is very simple. It consists of single operation by subtracting current pixel value with previous pixel value, thus only needs one byte-register to store previous pixel value and one addition-subtraction circuit. In the other hand, Mode 4-7 of DPCM algorithm which produce significant compression ratio increase, need more complicated on-board processing. Since it needs more than one neighboring pixels in calculation, at least one of them coming from previous line, it means that more byte-registers are needed to store all pixel values of previous line. In case of LAPAN-IPB multispectral imager where there are 4-bands of 8002 pixels of 2 bytes data each line, then additional 64-kbyte registers are needed. This additional complexity is quite significant, especially if the on-board processor has limited resources. Meanwhile since in mode 4-7 there are one division by two and two or three addition-subtraction operation, then one bit-shift register and some additional addition-subtraction circuit are needed, but these additional complexity are not significant compare to previous additional memory needed to store all pixel values of previous line.

In parallel Huffman-coding scheme, it practically encodes source data simultaneously using several Huffman-tables. However, these parallel Huffman-coding use the same de-correlated values produced by DPCM algorithm, so the additional hardware complexity only comes from the needs of several additional memory-buffer to temporarily store produced bits by each Huffman-table. How significant are these additional complexity depends on the size of image that being compressed. In LAPAN-IPB lossless compression, all bits produced by Huffman-coding are temporarily stored in small data buffer, and then processed to produce byte-data before sending the compressed byte-data to RAM memory. In parallel Huffman-coding, this is not the case. Since the algorithm needs to compare compression ratio of all Huffman-table result, then all bits produced by each Huffman-table must be stored in different data buffer. In standard LAPAN-IPB multispectral image, each image has roughly 50 thousand of lines, that was 50k-lines x 4-bands x 8002-pixels x 2-bytes, or approximately 3-Giga bytes data. Thus, using five Huffman-tables in parallel-coding needs additional 15-GB memory as data buffer, which is quite astonishing.

The needs of extra-ordinary additional data buffer by parallel Huffman-coding scheme can be reduced by using sub-image compression scheme proposed above. By using sub-image approach, the algorithm does not need to compare several compressed full-image, instead the algorithm can compare the result of each Huffman-table compression for pre-determined number of lines. For example, the LAPAN-IPB multispectral image captured can be divided into several sub-images which has 100 lines each, which corresponds to around 250 ms acquisition period. This way, parallel-coding only needs

64-MB data buffer for each table. This requirement is still enormous, but it is significantly easier to implement in low-cost on-board processor.

Table 5 shows several methods that have been proposed to improve the quality of image lossless compression of LAPAN-IPB multispectral imager, with their performance in terms of compression ratio and their feasibility in terms of on-board processor implementation. The method proposed initially is designed for multispectral image but can be easily adopted for matrix image.

Table 5. Trade-off between performance improvement and additional on-board complexity

Method Proposed	Compression-Ratio Improvement	Additional-Hardware Complexity	Remark
DPCM Mode 4-7	10%	Small memory registers and small arithmetic circuit	Very feasible to implement
Parallel Huffman-coding	5%	Very huge RAM memory and small comparative circuit	Not feasible for full-image, must employ sub-image approach
Sub-Image Approach	Under 5%	Small RAM memory and small comparative circuit	Minor compression improvement

5. Conclusions

LAPAN-A3/IPB image lossless compression method produces good compression ratio of around 1.5 to 2 when using ideal Huffman table, however the use of static Huffman table in LAPAN-IPB could reduce the compression ratio for about 10%. To mitigate this performance degradation, the algorithm uses special-character to limit number of bits produced by Huffman table, which could improve the compression ratio up to 5%. Based on simulations done, algorithm performance can be improved by modifying DPCM mode and Huffman-table type that are used. The use of two-or-more neighboring pixels in DPCM calculation could increase compression ratio for about 10%, while the use of parallel Huffman-table with sub-image approach could also increase the ratio for another 10%. All of these improvement shows that it is possible for LAPAN-A4 satellite to produce lossless image compression ratio up to 2.5, or 20% better compare to current LAPAN-IPB, provided that Payload Data Handling System (PDHS) of LAPAN-A4 has enough resources to handle the algorithm proposed.

Acknowledgments

The authors would like to thank Mr. Abdul Rahman as director of LAPAN Satellite Technology Center and Mr. Wahyudi Hasbi as chief engineer of LAPAN-IPB satellite, for their support and assistance so that this work can be well completed.

References

- [1] Hasbi W and Suhermanto 2013 Development of LAPAN-A3/IPB satellite an experimental remote sensing microsatellite *ACRS Bali*
- [2] Buchman A, Buchman A and Sabou S 2010 Image processing in transform domain: a survey *Carpathian Journal of Electronic and Computer Engineering* **3** pp. 1-4
- [3] The Consultative Committee for Space Data Systems 2005 *Image data compression* Blue Book CCSDS **122.0-B-1**
- [4] Hakim P R and Permala R 2015 Compression algorithm performance analysis of LAPAN-A3 satellite image using Fast-Fourier Transform *ISAST Bali*
- [5] Sharma M 2010 Compression using Huffman coding *International Journal of Computer Science and Network Security* **10** pp. 133-141
- [6] Pennebaker W B and Mitchell J L 1993 *JPEG Still Image Data Compression Standard* (New York: Van Nostrand Reinhold)
- [7] Salomon D 2008 *A concise introduction to data compression* (London Springer: Verlag London Limited)
- [8] Kaur N 2013 A review of image compression using pixel correlation and image decomposition

- with results *International Journal of Application of Innovation in Engineering and Management* Vol. **2**
- [9] Sharma N, Garg I, Sharma P K and Sharma D 2012 Analysis of transform techniques for 2D image compression *International Journal of Engineering and Technology* Vol. **1** pp. 78-82
- [10] Patel D, Bhogan V and Janson A 2013 Simulation and comparison of various lossless data compression techniques based on compression ratio and processing delay *International Journal of Computer Application* Vol. **81**
- [11] Anitha S 2015 Lossless image compression and decompression using Huffman coding, *International Research Journal of Engineering and Technology* Vol. **2**