Development and Implementation of Compression and Split Techniques for Multimedia Messaging Service Applications

A. Setyono, A. Md. Jahangir, and C. Eswaran, *Member, IACSIT*

Abstract—**MMS is a mobile messaging system that can be used for sending multimedia messages such as text, image, audio, video and a combination of them. This message can be sent to one or many mobile phones or external servers such as email and fax servers. In this paper, we present compression and splitting techniques for enhancing the capability of MMS technology. With the enhanced technology, one can send large files of multimedia messages without any degradation in the quality of the data which is not possible with the existing MMS technology. The developed techniques are tested using images with JPEG format. The experimental results demonstrate that the proposed system outperforms the conventional MMS in handling large data files. The modified MMS presented in this paper serves as a basis for developing practical internet applications.**

Index Terms—**MMS, multimedia message, compression, split message, JPEG.**

I. INTRODUCTION

Multimedia Messaging Service (MMS) is a rich content message sending system that not only sends text but also sends image, voice, animation, video and combination of them [1], [2]. MMS can be sent to mobile phone number or to external servers such as email servers. Several types of multimedia files such as text, images, audio and video can be sent using MMS. Open Mobile Alliance (OMA) organization has proposed the standards for MMS versions 1.1, 1.2 and 1.3 in the years 2002, 2005 and 2008 respectively [3]-[5]. Currently, it is possible to send multimedia messages up to 600 KB in a single delivery using MMS version 1.3 [5].

In this paper, we focus on image files with JPEG format. We choose the JPEG file format because it is relatively easy to implement the compression and split techniques with this format. JPEG is a standardized image compression mechanism. JPEG stands for Joint Photographic Experts Group, the original name of the committee that wrote the standard. JPEG is "lossy," meaning that the decompressed image is not exactly the same as the original image [6]. There are lossless image compression algorithms, but JPEG achieves much greater compression than is possible with lossless methods. A useful property of JPEG is that the degree of loss (or the degree of degradation) can be varied by adjusting the compression parameters. This means that one

can trade off file size against the output image quality. We use the split technique to facilitate the delivery of multimedia messages if the compressed file is still too large to be sent directly using MMS. Though usage of split technique requires a fairly high cost as it involves payment for each individual split file, this technique is essential if one wants to send large multimedia files without any degradation in the quality of data.

In this paper, we present a systematic procedure for implementing the MMS technology for sending images with JPEG format by using compression and split techniques. The proposed method is suitable for the development of internet applications for transmitting and displaying large size images without affecting the quality of the image.

The rest of the paper is organized as follows: Section II discusses data compression and split message techniques. The proposed framework of the MMS system and its description are presented in Section III. Implementation and evaluation details are presented in Section IV and finally conclusion and future work are given in Section V.

II. COMPRESSION AND SPLIT TECHNIQUES

Data compression is a branch of computer science that comes from information theory. Information theory itself is a branch of mathematics that has been developed in the late decades of the 1940s. The aim of data compression is to reduce the redundancy (useless information) in the message [7].

A. The Baseline JPEG Image Algorithm

JPEG compression technique involves several stages starting from DCT, quantization, zigzag run-length encoding and lossless compression using Huffman coding for building the compressed file. The various steps are explained below [8]-[10]:

1) The image is separated into intensity and color channels. They are subdivided into 8x8-pixel blocks. They are subsequently processed left to right and top to bottom.

2) Each 8x8-pixel block is processed using discrete cosine transform (DCT) and applied to each block.

3) Each block is compressed through quantization. The 64 coefficients for each block are quantized to a lower precision by dividing a fixed table of values that gives the least precision for high-frequency term.

4) The first of the 64 coefficients for each block is the average brightness or "DC" term and remaining 63 coefficients for each block are scanned in zig-zag diagonal order that starts with the lowest frequencies and progressed to the highest.

5) Entropy coding is the process of using entropy algorithm, such as Huffman or Arithmetic for encoding. The DCT and quantization processes will eliminate the pixels with zero values.

6) The image is reconstructed through decompression, a process that uses the Inverse Discrete Cosine Transform (IDCT).

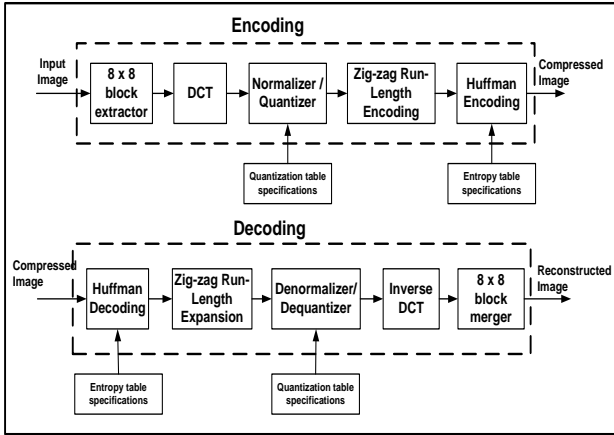The steps in implementing the JPEG image compression are shown in Fig. 1.

Fig. 1. JPEG compression process [8].

B. Discrete Cosine Transform (DCT)

The DCT is a converting process of unit data values into a sum of cosine functions [11]. As described in the previous section, the process of JPEG compression uses DCT process. Fig. 2 illustrates the concept DCT.
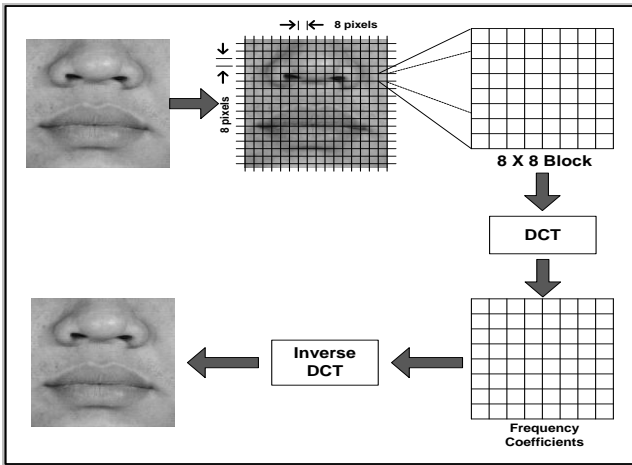

Fig. 2. Concept of DCT for JPEG.

The most common DCT definition of a 1-*D* sequence of length *N* is formulated by [11]

$$F(u) = \sqrt{\frac{2}{N}} C(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \qquad (1)$$

for $u = 0,1,2...N-1$, where

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u = 0 \\ 1 & otherwise \end{cases} \qquad (2)$$

The function $f(x)$ represents the value of the xth sample of the input signal. $F(u)$ represents a DCT coefficient for $u = 0, 1,..., N-1$ [11].

The two-dimensional DCT (2-*D DCT*) of a block of $M \times N$ samples of a two-dimensional signal $F(x,y)$ is formulated as follows[11]:

$$F(u,v) = \frac{2}{\sqrt{MN}} C(u)C(v)$$

$$\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos \left[\frac{\pi(2x+1)u}{2N} \right] \cos \left[\frac{\pi(2y+1)v}{2M} \right] \qquad (3)$$

for $u = 0, 1,2...N-1$ and $v = 0, 1, 2... M-1$, where

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & for\ k = 0 \\ 1 & otherwise. \end{cases} \qquad (4)$$

The three-dimensional variant of the DCT (3*D-DCT*) is a composition of three of 1-*D DCT* along each dimension [12]. The formal definition is as given below.

$$F(u,v,w) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{z=0}^{N-1} f(x,y,z) . Cux . Cvy . Cwz \qquad (5)$$

For *u, v, w* = 0, 1, 2 ..., *N*-1, where *Cux, Cvy, Cwz* are as follow:

$$Cux.Cvy.Cwz =$$

$$\cos \left[\frac{\pi}{N} \left(x + \frac{1}{2} \right) u \right] . \cos \left[\frac{\pi}{N} \left(y + \frac{1}{2} \right) v \right] . \cos \left[\frac{\pi}{N} \left(z + \frac{1}{2} \right) w \right] \qquad (6)$$

C. Huffman Coding

In general, a data compression technique consists of two major activities, namely modeling and coding. The fundamental process of a data compression technique is to determine a set of parts of the data (a stream of symbols) to turn it into code (a stream of codes). If the compression process is effective, then the stream of codes will be smaller than the size in terms of streams of symbols. The decision to make certain symbols with certain codes is the core of the modeling process. Generally means that a model is a collection of data and rules that determine the symbol elements as input and output of the code for the compression process. In Huffman coding, symbols that often appear will be transformed into code with a small number of bits, while the less frequent will be the code with a large number of bits [13]. The block diagram of the static Huffman model is shown in Fig. 3.
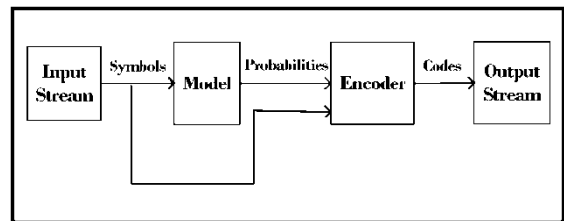

Fig. 3. Static compression of Huffman coding.

Procedure to build this tree is simple and easy to understand. Each symbol is sorted according to frequency, this frequency is considered as the weight of each symbol, and then followed by steps as follows:

1) Two free nodes with the lowest weights are paired.
2) Parent node for the second node created in the previous step. Summing up the the frequencies and use them as weights.
3) Now parent node serves as a node free.
4) Give code node 0 to 1 for the left and right nodes.
5) Repeat the steps above until only one node left [14].

D. *Split Message*

This technique is used to split a large file into a number of small files. These files are sent one by one with a single key press. The files are merged back into a single file on the mobile phone recipient. This technique is similar to the process of splitting an SMS message with large number of characters into multiple SMS each with a maximum number of 160 characters.

The MMS packages the compressed message in MIME format and encodes into a binary message. We need to split the compressed message file into several smaller files in order to send a large message. The splitting method is employed if the original file size (OFS) is larger than the maximum capacity (MC) permitted. If $M = $ OFS / MC, the number of split files N is given by

$$N = \begin{cases} M & if\ M\ is\ an\ integer \\ Integer\{M + 1\} & if\ M\ is\ a\ fraction \end{cases}$$

Let the original image be of size $(w \times h)$ where w and h denote the width and height of the image respectively. Then the size of each of the N split images will be equal to $\{w \times (h / N)\}$.

III. PROPOSED FRAMEWORK

The proposed MMS application is developed using Java 2 Micro Edition. The application has been simulated using the Net Beans emulator. The simulated results show that the developed application can be installed on a mobile phone. In this study, we have developed the MMS application to send image files for JPEG format using compression and the split message techniques. This aims to enhance the existing ability of the MMS within the message delivery. After developing the MMS application, the next step is to install it on a mobile phone to send multimedia messages to other mobile phones or external servers, such as email servers. The proposed framework of the MMS is shown in Fig. 4.
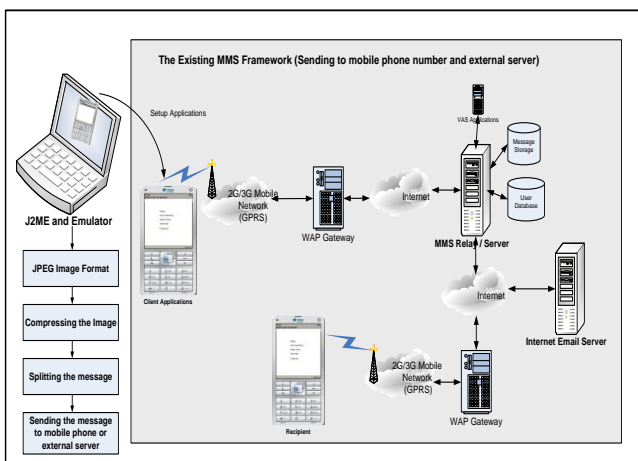

Fig. 4. Proposed Framework of MMS technology.

MMS application already installed on the mobile phone can also be used as input media for building various kinds of internet applications. This is because the multimedia message that has been sent to the mobile phone number and email servers can be retrieved and stored into a database server. From that database, we can develop Internet applications

such as telemedicine, monitoring and e-commerce system. Fig. 5 shows the flowchart of sending MMS using the proposed method.
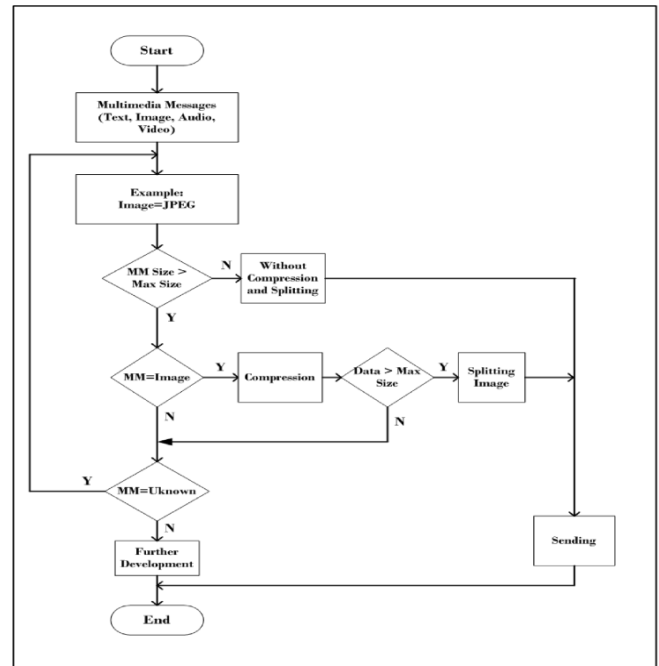

Fig. 5. Flowchart of the algorithm for sending MMS.

The multimedia message in the form of text, image, audio or video that will be sent using MMS technology is compressed first, if the size of the file exceeds the capacity allowed. Then if the size of the compressed file is still larger than the maximum capacity allowed then the message will be split into a number of small size files so that they can be sent using the MMS technology. Multimedia messages that have been split are sent one by one until all successfully delivered.

IV. IMPLEMENTATION AND EVALUATION

In this research work, we have implemented the compression and split message techniques before sending the JPEG file. We have also performed various evaluations and analysis of the implementation process. The following section discusses the details of implementation and evaluation processes.

A. *Implementation*

To implement the compression and split techniques for MMS application on a mobile phone, we use mobile phones, Sony Ericsson K630i and J105i. These mobile phones have fairly comprehensive features including MMS and Java. The MMS application is installed on the mobile phones. We have simulatedan experiment to determine the outcome of the implementation of the compression and split techniques. For this simulation, we have used test JPEG images with file sizes exceeding the maximum capacity permitted.

In this simulation, we have used Java 2 Micro Edition (J2ME). Java library allows only up to 30 KB of data for each delivery. If a file of size exceeding 30 KB is transmitted, the data will be compressed to get a smaller size file. If the compressed file size is still more than 30 KB, it will be split into smaller files. Fig. 6 shows the simulation results.

Fig. 6. Simulation of the proposed framework.

The explanation of the simulation results shown in Fig. 6 are: a) user interface for sending MMS, b) file (image) that will be sent, c) selecting a file that will be sent, d) the information in the form of notification that the file is more than 30 KB, e) compressing files with a selected value of the quality factor, f) determining the destination number, g) splitting file to be smaller files, h) delivery process, where in one time multiple messages are sent sequentially and i) MMS inbox (sending the multimedia message successfully).

In this research, we only focus on developing an MMS application to send JPEG images. The sequence diagram of the MMS application that has been installed on a mobile phone is shown in Fig. 7.
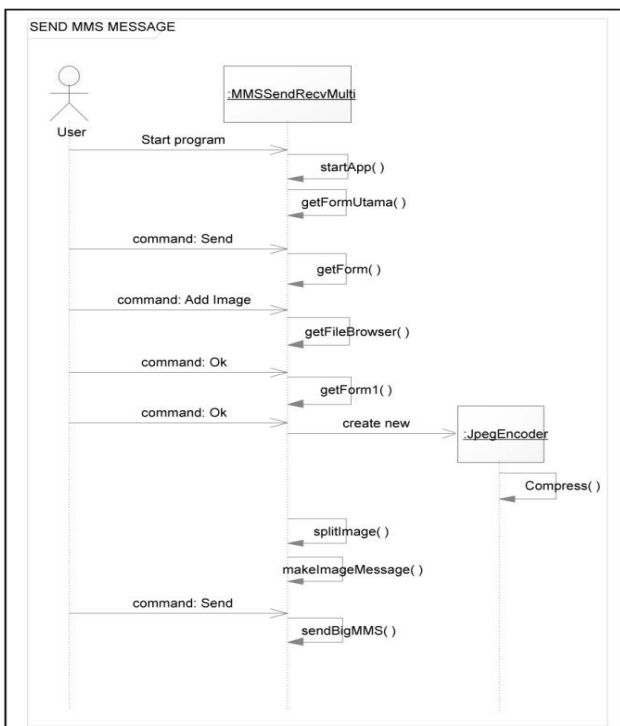

Fig. 7. Sequence diagram for sending MMS.

B. Evaluation

The compression and split techniques provide benefits, such as i) the application can send files larger than the one that is possible with the existing application, ii) the application can maintain the quality of the original file by merging and decompressing the split files on the mobile phone receiver and iii) the method is suitable for client-server applications which requires a good quality data, such as telemedicine system. A comparison between the existing and the developed systems are shown in Table I.

TABLE I: COMPARISON THE METHODS

The Existing	The Developed
1. Max Size 30 KB.	1. Max size > 30 KB.
2. Resize Image.	2. Compress and Split Image
3. Single message.	3. Multiple message single sending.
4. Size and quality reduced	4. Size and quality are not reduced. It can be merged and decompressed
5. The file cannot be reconstructed to become similar to the original one.	5. The files can be merged and decompressed to become similar to the original file.

In comparing the applications, we assume that both the existing and developed applications have been developed using Java technology. From this comparison, we can see that the developed system has many benefits: it can send large size image file, it can send several files at a single sending and it can also merge and decompress several files to become a single file. Moreover, it keeps the quality of the image file intact, same as the original one.

To evaluate the performance of applications that are developed, we have conducted experiments using JPEG images of sizes larger than the maximum capacity of 30KB. The results of the compression and split file techniques are shown in Table II.

TABLE II: MEASUREMENT PERFORMANCE BASED ON COMPRESSION RESULT (QUANTIZATION 50%)

Data Size	Reduced File Size After Compression	Split files	Split File Sizes
153 KB	15 KB	1 files	15
115 KB	33 KB	2 files	16, 17
112 KB	52 KB	2 files	25, 27
101 KB	12 KB	1 file	12
100 KB	69 KB	3 files	20, 26, 23

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented a novel technique to improve the ability of the existing MMS for sending large size JPEG image files. This is achieved by implementing compression and split techniques on the mobile phone. The compression technique is used to reduce the file size. The splitting technique is used to break the original file into smaller files according to the limitation of the MMS. The simulation results show that the proposed method can be successfully implemented on a mobile phone to send large amount of data using MMS application. The proposed framework is useful for developing mobile applications using MMS technology such as mobile telemedicine systems. The extension of this method for the audio and video files is under investigation.

REFERENCES

[1] G. L. Bodic, *Mobile Messaging Technologies and Services SMS, EMS, MMS*, 2nd ed. John Wiley & Sons, Chichester, England, 2005, pp. 197-220.

[2] G. L. Bodic, *Multimedia Messaging Service: An Engineering Approach to MMS*, John Wiley & Sons, Chichester, England, 2003, pp. 68-91.

[3] Open Mobile Alliance, *Multimedia Messaging Service Architecture Overview*, OMA-WAP-MMS-ARCH-v1_1-20021101-C, Approved Version 1.1, 2002.

[4] Open Mobile Alliance, *Multimedia Messaging Service Encapsulation Protocol*, OMA-MMS-ENC-V1_2-20050301-A, Approved Version 1.2, 2005.

[5] Open Mobile Alliance, *Multimedia Messaging Service Requirement Candidate Version*, OMA-MMS-ENC-V1_3-20080128-C, Approved Version 1.3, 2008.

[6] JPEG image compression FAQ. (2010). [Online]. Available: http://www.faqs.org/faqs/jpeg-faq/part1/

[7] M. Nelson and J. L. Gailly, *The Data Compression Book*, 2nd ed. M&T Books, 1996.

[8] T. Acharya and P. S. Tsai, *JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures*, John Wiley & Sons, Canada, 2005.

[9] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 2nd ed., Prentice Hall, USA, 2001.

[10] J. C. Russ, *Image Processing Handbook Fifth Edition*, CRC Taylor and Francis, USA, 2007.

[11] S. A. Khayam, "The Discrete Cosine Transform (DCT): Theory and application," Department of Electrical and Computer Engineering, Michigan State University, 2003.

[12] J. Hermany, "Compression Methods Using 3D DCT," Doctoral Degree Programme.

[13] Forum Nokia, Nokia MMS FAQ Version 1.1. (2004). [Online]. Available: http://www.forum.nokia.com

[14] D. Salomon, *Data Compression: the Complete Reference*, 3rd ed., Springer, USA, 2004.

Andik Setyono was born in Indonesia on February 12, 1980. He received his bachelor and master degrees in computer science at Dian Nuswantoro University, Semarang, Indonesia in 2003 and 2005, respectively. He graduated the Ph.D. degree in the Faculty of Computing and Informatics at Multimedia University, Cyberjaya, Malaysia. Since 2006 he has been with Nuswantoro University as a lecturer in the Faculty of Computer Science at Dian Nuswantoro University, Indonesia. He is also as a member of IACSIT.

Md. Jahangir Alam received the B.Sc. (Honours) and M.Sc. degrees in in physics from Jahangirnagar university, Dhaka, Bangladesh in 1989 and 1991, respectively. He received Ph.D. in Information Technology from Osaka University, Osaka, Japan in 2004. He worked as an assistant professor in the Department of Electronics and Computer Science at Jahangirnagar University. Currect he is working as a lecturer at Multimedia University, Cyberjaya, Malaysia.

C. Eswaran received his B.Tech, M.Tech and Ph.D degrees from the Indian Institute of Technology Madras, India, where he worked as a professor in the Department of Electrical Engineering until January 2002. Currently he is working as a professor in the Faculty of Computing and informatics, Multimedia University, Malaysia. He is also serving as the chairman of the research center, "Communication Infostructure" at this university. Dr.C. Eswaran served as a visiting faculty and research fellow in many international universities such as Ruhr University Bochum, Germany, Concordia University, Canada, University of Victoria, Canada, and Nanyang Technological University, Singapore. He has received two prestigious international fellowships: Humboldt-Fellowship (Germany) and Advanced Systems Institute Fellowship(British Columbia, Canada). He has supervised successfully more than 25 Ph.D/M.S students in the areas of Information Systems,Communications, Digital signal Processing, Neural Networks, and Telemedicine. Prof.C. Eswaran is a senior member of IEEE.