An Efficient Type Codec for Point Data in Lightweight Applications Scene Representation (LASeR)

YeSun Joung, Jihun Cha, Won-Sik Cheong, Young-kwon Lim, and Kyuheon Kim

ABSTRACT— Recently, MPEG has opened activity to standardize scene representation for lightweight applications such as in mobile phones. The standard is named lightweight applications scene representation (LASeR) and can be applied to improve and make efficient rich media applications and services on mobile devices. In this standard, we proposed an efficient type codec for point data to maximize the bit efficiency of LASeR. In this paper, we describe the new method and the test results of the proposed scheme.

Keywords—LASeR, MPEG-4, point data, type codec.

I. Introduction

While mobile phone and wireless devices are in the spotlight of a promising market, mobile multimedia services are still stuck with a degraded version of the Web with the wireless application protocol (WAP) and I-mode. Lightweight applications scene representation (LASeR) is a standard dedicated to improving and making efficient rich-media applications and services on mobile devices. Using LASeR, advanced multimedia services such as mobile TV services, web-like navigation services, and so on can be provided.

Current LASeR binary representation uses fixed-length coding (FLC) for point data as shown in section 2.6 of the final committee draft [1]. In the final committee draft of LASeR, the actual values of point data are encoded if the number of points (nbPoints) is less than 3. When nbPoints is bigger than or equal to 3, the dynamic range of the point sequence is determined, and point sequence data are then fixed-length encoded using the acquired dynamic range. This method is very simple; however, there is a 10-bit overhead to specify the length fields for each point sequence as well as the numbers of unnecessarily pre-assigned bits in many data fields thereafter.

When dealing with sequences of an alphabet, we can always consider entropy coding for data compression. The given test set falls into this category. Although a few test files suffer from a lack of data redundancy and abundance, many files still provide a chance to perform decent lossless data compression. We focused on this possibility and came up with a proposal.

We present a test set in section II, with a reasoning to select a subset of the test set, and a data analysis on selected files. In section III, we propose an efficient encoding scheme to apply point data, which is called a point type codec in the LASeR specification. The experimental results of the new encoding scheme are presented in section IV. Finally, we follow with our conclusions.

II. Test Set Analysis

1. Data Selection

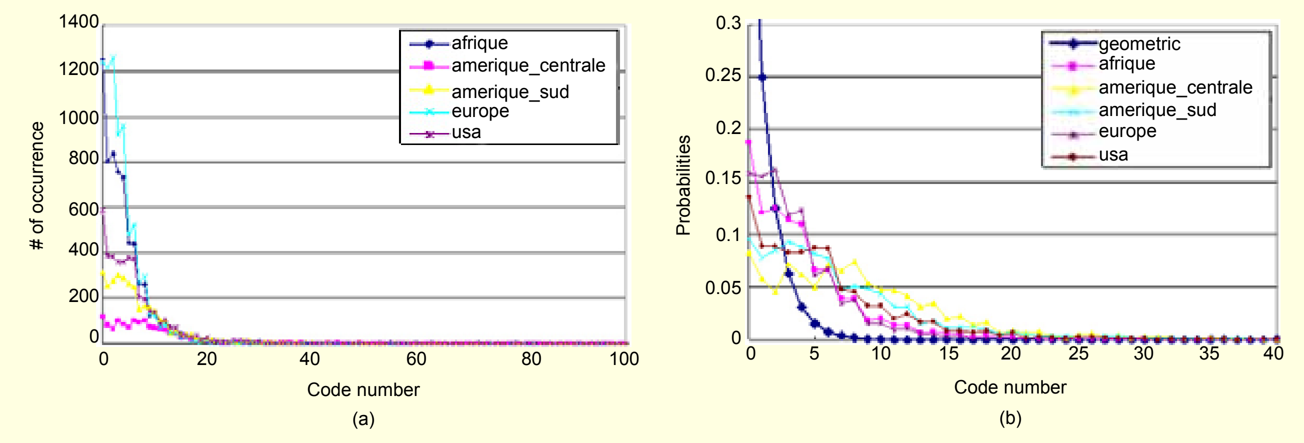
A test set for LASeR is available on the site directed in the core experiment (CE) document [2]. Since we focus on the compression of point data, files containing a meaningful number of points were chosen to be analyzed from the test set. We categorized those files into three groups for the sake of convenience. The three groups are *map*, *cartoon*, and *still image*. In the *map* group, there are five files available in the test set: afrique, amerique_centrale, amerique_sud, europe,

---

Manuscript received June 21, 2005; revised Sept. 07, 2005.

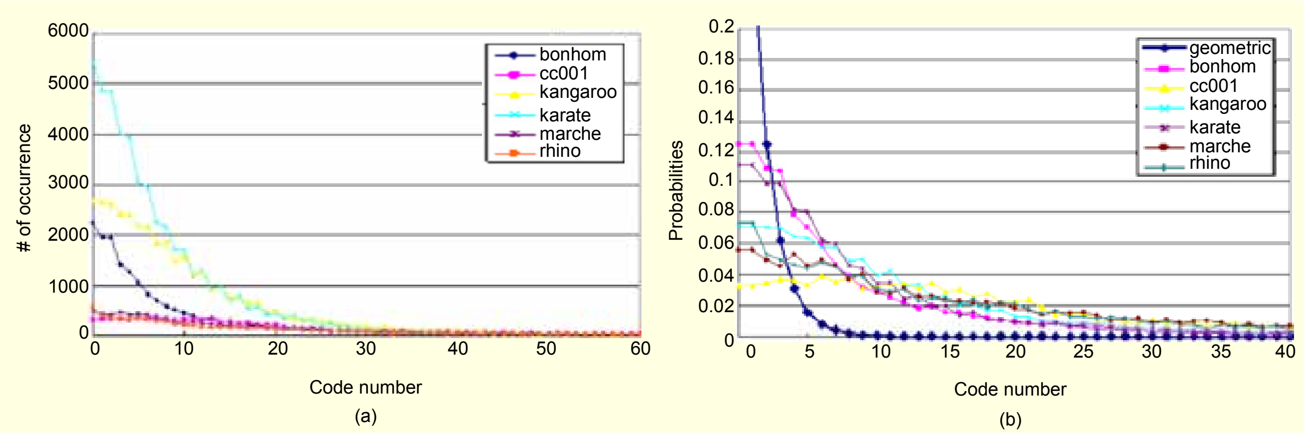
YeSun Joung (phone: +82 42 860 1557, email: ysjoung@etri.re.kr), Jihun Cha (email: jihun@etri.re.kr), Won-Sik Cheong (email: wscheong@etri.re.kr), and Kyuheon Kim (email: kkim@etri.re.kr) are with Digital Broadcasting Research Division, ETRI, Daejeon, Korea.

Young-kwon Lim (email: young@netntv.co.kr) is with the Department of Electronic and Computer Engineering, Hanyang University, Seoul, Korea.



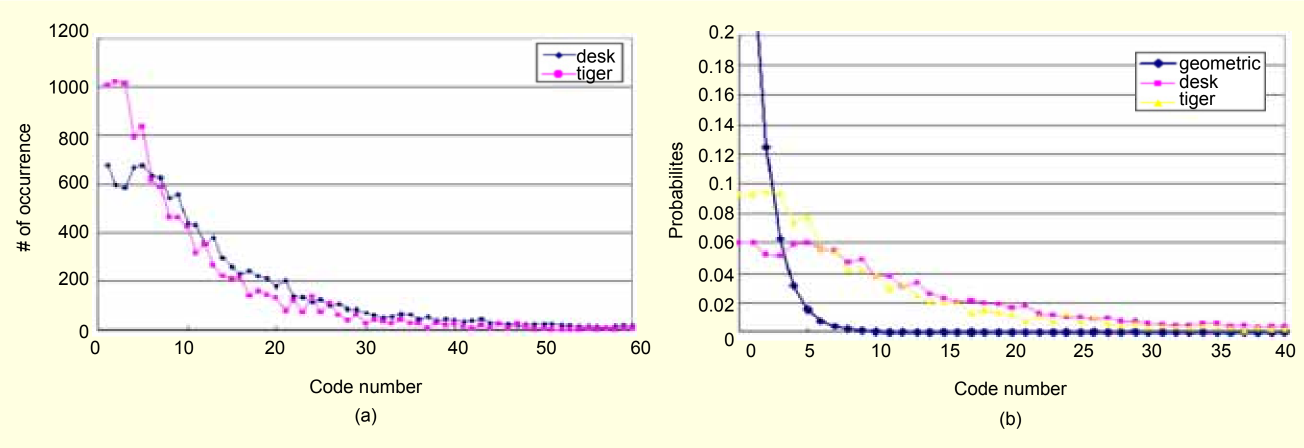
(a) (b)

Fig. 1. Graphs for the *map* group.



(a) (b)

Fig. 2. Graphs for the *cartoon* group.



(a) (b)

Fig. 3. Graphs for the *still image* group.

and usa. In the *cartoon* group, there are six files available in the test set: bonhom, cc001, kangaroo, marche, karate, and rhino. In the *still image* group, there are two files available in the test set: desk and tiger.

An entropy coding is a coding scheme that involves assigning codes to symbols so as to match code lengths with the probabilities of the symbols. The assigned code is usually called a code number or code word. For analysis, in this section, two graphs are presented for each data group: occurrence distribution function and probability density function.

As shown in Figs. 1(a) through 3(a), the shape of the occurrence distributions of each group files generally follow a geometric distribution. In other words, lower code numbers have a high occurrence, and the number of occurrences drops dramatically as the code number increases. Although the shapes of a few files like amerique_centrale and cc001 do not seem to follow the shape of the geometric distribution, it is due to the low number of occurrences on those files compared to other files.

Figures 1(b) through 3(b) show the probability density functions (pdfs) of the geometric distribution. These graphs clearly show the similarity of the general shape and the difference of their slope severity.

2. Analysis

As shown in II.1, the distributions of many test files generally follow a geometric distribution. Therefore, we expect that applying an entropy coding would produce a decent data compression. Although there are many entropy coding methods, a scheme that satisfies LASeR requirements, namely an efficient binary representation and small footprint, should be considered. It is known that exponential Golomb (Exp-Golomb) coding produces a very high compression ratio when the pdf of given data follows a geometric distribution.

The ideal pdf for Exp-Golomb coding, which is shown in Figs. 1(b) through 3(b) with a dark bold line, has a very high probability at code number 0; the probability decreases dramatically as the code number increases. Although the probability at code number 0 is not high enough and their variance is much larger than the one for the ideal pdf for Exp-Golomb, many pdfs of test files generally follow the shape of the ideal pdf. Exp-Golomb has a parameter that can be fitted into various geometric distributions with different variances.

Although there are many entropy coding methods, we can classify them into two groups: one with a coding table and another without a coding table. The Huffman coding method represents a group that utilizes a coding table. This method can produce a near-optimal compression efficiency; however, to decode the point data, this method requires transmitting a coding table and putting overhead on the decoder (terminal device) side to access the memory location every time. Since LASeR pursues a light-weighted footprint and complexity, these methods are excluded from consideration.

Another group of coding schemes includes arithmetic coding and exponential Golomb coding. Although arithmetic coding is quite an efficient coding scheme, its lack of error resilience prohibits it to be applied for LASeR. Exponential Golomb coding has several characteristics that suit a LASeR environment. With a little overhead on the encoder side, it can be easily converted to a reversible variable-length coding for added error resilience [4]. Another merit is low overhead on the decoder side. Since the decoding process can be done using only addition and a bit shift, it can be implemented without a large overhead on small devices such as mobile phones.

III. Proposed Method

In this section, we describe a point data compression scheme. The aiming point of our scheme is to improve the compression efficiency of LASeR binary representation by applying Exp-Golomb coding to point sequence data, while the complexity on the decoder side is comparable to the one used by the current method. The details of Exp-Golomb coding used in our scheme are founded in [3]-[5].

As mentioned in section I and shown in Fig. 4, current LASeR binary representation uses FLC for point data. The FLC method is very simple and less complex in the decoder side than the proposed method; however, there are 10-bit overheads to specify the length fields for each point data sequence. In this proposed scheme, the choice between Exp-Golomb coding and fixed length coding is made first. When fixed length coding is chosen, point sequence data are encoded as described in the final committee draft of LASeR [1]. On the other hand, when Exp-Golomb coding is chosen, the value of parameter k is determined, and a point sequence is then encoded by the Exp-Golomb coding method.

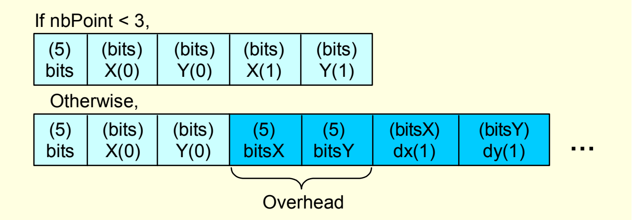


Fig. 4. Binary sequence for FLC.

The proposed scheme can be summarized as follows:

For each point sequence data

Step 1. Choose coding method between Exp-Golomb coding or fixed length coding. When fixed length coding is chosen, go to step 4.

Step 2. Decide the value of parameter k.

Step 3. Encode point sequence data to Exp-Golomb code using k decided in step 2. Go to step 5.

Step 4. Encode point sequence data to fixed length code.

Step 5. End.

The binary sequence after applying the proposed scheme is shown in Fig. 5.

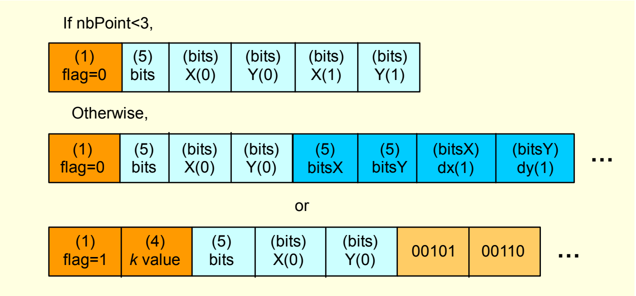


Fig. 5. Binary sequence of the proposed scheme.

IV. Experimental Results

LASeR Reference software was modified to accommodate our proposal. In this experiment, the thirteen test files selected in section II are tested. The experimental results are shown in Table 1. In this table, the original file size is the binary file size for each test file, the new file size is the binary file size acquired by the proposed coding scheme, and the last column shows the percentages of bit reduction when the proposed scheme is applied to the test files against the original scheme. As shown in Table 1, a 4 to 29% bit reduction is acquired for the selected test files. From the above results, we have verified that the proposed scheme can improve the efficiency of binary representation. All of the results were measured by file size, which is provided using a Windows system.

Table 1. The compression efficiency of the proposed scheme on the selected test set.

Category	Test files	Original file size	New file size (bits)	Bit reduction ratio (%)
Map	afrique	41,512	29,616	28.66
	amerique_centrale	11,104	9,592	13.62
	amerique_sud	21,328	17,400	18.42
	europe	45,536	34,624	23.96
	usa	33,744	24,936	26.1
Cartoon	bonhom	171,464	159,656	6.89
	cc001	100,808	95,416	5.35
	kangaroo	587,200	558,488	4.89
	karate	701,192	675,152	3.74
	marche	100,472	96,640	3.81
	rhino	75,952	69,424	8.59
Still image	desk	97,056	87,008	10.35
	tiger	55,392	55,192	16.94

Another important factor in LASeR is the complexity of the decoder. Since the proposed scheme uses variable length coding, operations to convert a codeword to a symbol are required. These operations may increase the complexity of thedecoder. However, the proposed scheme can be implemented with a few bit shifts and additions. Thus, adoption of the proposed scheme slightly increases the complexity of the decoder. More specifically, the proposed scheme needs M+2k shifts, two additions to decode each point sequence.

V. Conclusion

In this document, we described a coding scheme that can improve the efficiency of LASeR binary representation. To verify the proposed scheme, we implemented the scheme in reference software with small modification and performed a series of experiments on a test set.

The experimental results showed that the proposed scheme can achieve around a 4 to 29% bit reduction with a small number of additional operations. Therefore, the proposed type codec for point data has been adopted as the LASeR binary representation.

For further work, we need to improve the encoding scheme in a more intelligent way and experiment on more test sequences. Currently, when the proposed scheme decides on parameter k in the encoder, the encoder tests all parameters k and chooses the optimal one. This mechanism produces some overhead to the encoder. By using scene information or other parameters, we may find a more smart way to encode.

References

[1] ISO/IEC JTC1/SC29/WG11, *LASeR Text of ISO/IEC 14496-20/FCD*, Oct. 2004, pp. 49-50.

[2] ISO/IEC JTC1/SC29/WG11, *LASeR Core Experiment on LASeR Binary Representation*, July 2004, pp. 1-7.

[3] S.W. Golomb, "Run-Length Encoding," *IEEE Trans. on Inf. Theory*, vol. 12, issue 3, 1966, pp. 399-401.

[4] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8), *Reversible VLC for Added Error Resilience,* N1164, Mar. 3, 1999.

[5] Iain E.G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons Press, Dec. 2003, pp. 198-200.