Lossy Compression of Scanned Map Images

Alexey Podlasov, Alexander Kolesnikov and Pasi Fränti
Speech & Image Processing UnitDepartment of Computer Science
University of Joensuu, Joensuu, Finland
{apodla, koles, franti}@cs.joensuu.fi

Abstract

An algorithm for lossy compression of scanned map images is proposed. The algorithm is based on color quantization, efficient statistical context tree modeling and arithmetic coding. The rate-distortion performance is evaluated on a set of scanned maps and compared to JPEG2000 lossy compression algorithm, and to ECW, which is commercially available solution for compression of satellite, aerial and scanned imagery. The proposed algorithm outperforms these competitors in rate-distortion sense for the most part of the operational rate-distortion curve.

Keywords: *Digital map images, lossy image compression, context modeling, color quantization.*

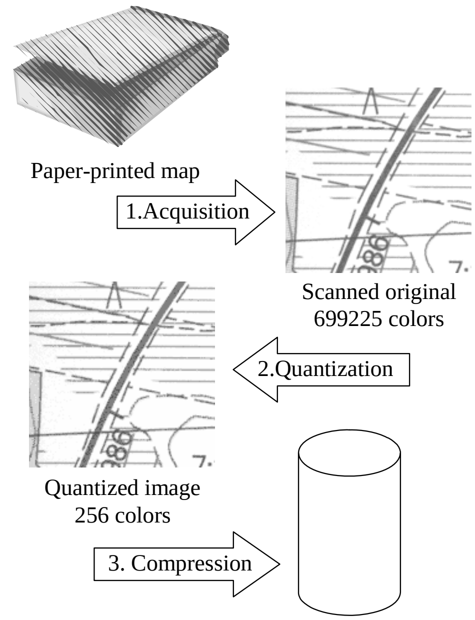

Figure 1: Overall scheme of the proposed compression

1. INTRODUCTION

Nowadays, digital *Geographical Information Systems* (GIS) became more and more popular among all kind of users. Though at the beginning the price of mobile positioning (e.g. GPS) and processing devices restricted the use of electronic navigation to military or corporate applications, today we are facing the extensive growth of this industry in personal user sector. Recent progress in low-cost mobile hardware and, especially, in low-cost memory made computer-aided navigation possible in your personal car on a road trip, as well as in your hand while trekking.

However, raster map image converted from the vector database is not always the case. It is still common that, when needed, geographical information could only be found on the paper printed map. Similar case is the digitization and storage of rare maps, which are too fragile and valuable to be used as such. Though this kind of paper-printed material could be easily digitized and integrated into computerized navigation or archive system, there are still some specific problems. The main problem of raster maps is their storage size. Paper printed material of approximately A4 size scanned with 300dpi in true-color results in about 2500×3500 pixel image requiring 24 bits per pixel, which is 25 megabytes per image. The number of unique colors can vary from hundreds of thousands to several millions depending on the type of the map. For example in our experiments we experienced up to 700000 unique colors in topographic map images. Standard lossless compression techniques such as PNG, GIF or TIFF are able to provide about 1.2:1 compression ratio, which is not enough for effective transmission of the image to the user's device and processing it there. Lossy compression is therefore needed.

There is a wide variety of standard multi-purpose lossy compressions techniques, as well as techniques developed specifically for compression of scanned material. Among the standard algorithms JPEG and JPEG2000 [14] are the most popular. Wavelet-based *Multiresolution Seamless Image Database* (MrSID) [1] by LizardTech is a patented commercial solution for storing large amounts of satellite and aerial images. It is applied for compression of scanned map imagery as well. Wavelet-based *Enhanced Compression Wavelet* (ECW) [3] format by ER Mapper is also a commercially available solution for GIS-based image compression. Well-known DjVu format [2] by LizardTech and AT&T is specially developed for storage of scanned imagery, especially books.

However, popular wavelet techniques have some disadvantages when used for compression of scanned maps. Scanned map combines the characteristics of both image classes: discrete-tone and continuous-tone. The image origin is artificial and, therefore, unlike photography, a map image contains of a small number of unique colors and but lots of small-size detailed structures such as letters and signs, solid uniform areas such as waters, forests, fields, sharp edges and almost no gradient color gradation. Besides this, typical map image contains a lot of repetitive patterns and textures. This comes as from the map itself, e.g. areas like swamps or sands are usually represented by textures, as well as from the paper map production technology *i.e.* when printed, color gradation is usually obtained by dithering the available inks forming uniformly textured areas. This dithering is acquired by the scanner and appears in scanned images as a repetitive pattern of color dots.

Lossy compression based on wavelet transform significantly smoothes the edges of the image and destroys thin well-structured areas, such as textures. When higher level of quality is desired, techniques like JPEG2000 or ECW loose efficiency in compression performance since wavelet transform requires more bits to represent high frequencies of image's sharp edges. On the other hand, the compression algorithms optimized for artificial graphics, such as *Piecewise-constant Image Model* (PWC) [4] or *Embedded Image-Domain Adaptive Compression* (EIDAC) [5], are not effective since these algorithms are designed to deal with computer-generated imagery. However scanned image is affected with noise imposed by the acquisition device – a scanner or a camera. The inconsistency in illumination, sensor's perception and other factors results in blurred edges, and significant increase in the number of colors and intensity gradation. This makes lossless algorithms inefficient in providing necessary compression ratio.

In this work, we propose an alternative lossy compression technique for scanned map images based on color quantization and statistical lossless compression. In contrary to wavelet-based lossy compression, the proposed algorithm does not use lossy image transform. The overall compression system under consideration is outlined in Figure 1. Firstly, the paper-printed map is digitized with *e.g.* flatbed scanner. The resulting image, referred further as the *original image*, is the input of the proposed compression algorithm. The proposed algorithm consists of two stages: color quantization and lossless compression. In quantization stage, the number of colors of the original image is reduced. This stage is a lossy part of the algorithm and the degradation of the image *i.e.* the information loss occurs here. The resulting image with reduced number of colors is referred further as the *quantized image*. In the second stage, the quantized image is compressed by the lossless image compression algorithm.

In general, the proposed scheme does not require any specific quantizer and compressor to be used. Though a big variety of approaches can be considered for this task, we consider the using of simple, fast *Median Cut* (MC) quantizer [11], which is a classical approach widely used in image processing applications and is able to process map images in reasonable time.

Among the variety of lossless compression algorithms which could be considered to be used to perform the compression stage one should mention that all we deal with color map images when the most of efficient lossless compression techniques are aimed at gray-scale imagery. Separating the color planes with following gray-scale compression typically means sacrifice in compression performance since color components are usually highly correlated. Besides that, linear prediction, which is a standard tool for continuous-tone lossless compression algorithms such as JPEG-LS or CALIC [7][8] fails on map images since the value of the current pixel depends on its neighborhood configuration, not on the local intensity gradation. This motivates us to choose for compression stage context-based statistical *Generalized Context Tree* (GCT) compression algorithm which recently presented its high efficiency in compression of computer-generated raster maps [6]. Since the algorithm described in [6] is not applicable directly, in this work we consider its modification.

Summarizing the above mentioned, we propose two-stage algorithm for lossy compression of scanned map images: firstly, the number of colors in the image is reduced by median cut color quantization; then the resulting image is compressed losslessly by GCT lossless compression algorithm. The visual comparison of the proposed algorithm and standard JPEG2000 applying to scanned map image is presented in Figure 2. The upper and lower rows represent lower and higher quality levels respectively. The algorithms are applied to compress the test image with the same

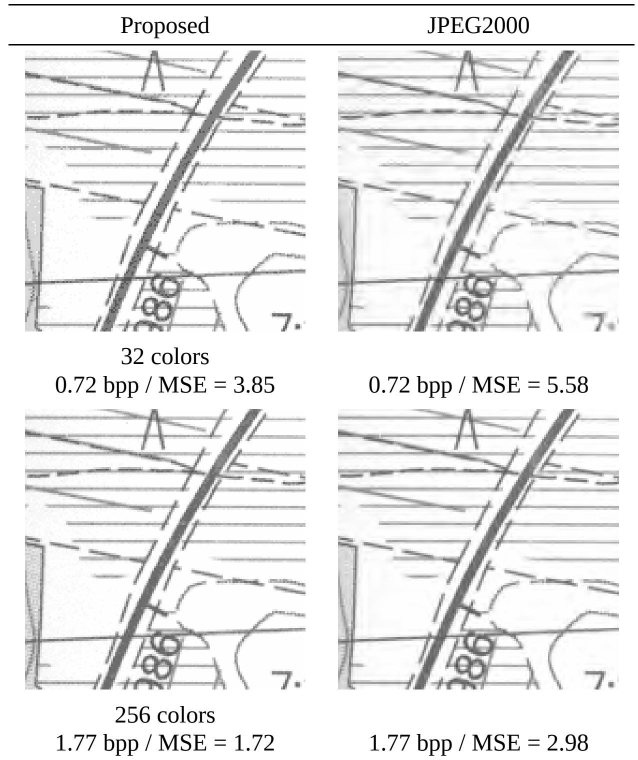

Figure 2: Visual comparison of the proposed and JPEG2000 algorithms**.**

compression ratio – 0.72 bpp for low quality and 1.77 bpp for higher quality. One can see that for equal bitrate the proposed algorithm provides less degradation according to MSE distance. For lower quality level the proposed algorithm preserves edges and does not employ smoothing as JPEG2000. The performance of the proposed algorithm is evaluated on a set of scanned topographic maps and compared to JPEG2000 – standard lossy compressor and ECW – a commercially available compression system. Also in order to prove the efficiency of GCT compressor we consider the comparison with 'trivial approach' where color quantized image is compressed with PWC – an algorithm for compression of computer generated palette images (referred also as *simple images*). We denote this approach as "MC+PWC" *i.e.* median cut plus PWC.

The rest of the paper is organized as follows: describes the proposed compression algorithm is described in Section 2; experiments are presented in Section 3, and Conclusions are drawn in Section 4. Future development of the proposed technique is outlined in Section 5.

2. PROPOSED ALGORITHM

2.1 Median cut quantization

Median cut algorithm is a very popular method for color quantization widely used in image processing practice originally published in [11]. It is relatively simple both conceptually and computationally still providing good results.

The conceptual idea behind the algorithm is to design a color palette in such a way that each color would represent approximately the same number of pixels of the input image. Firstly, the algorithm computes the color histogram of the image. Typically, the image is pre-quantized with uniform quantizer

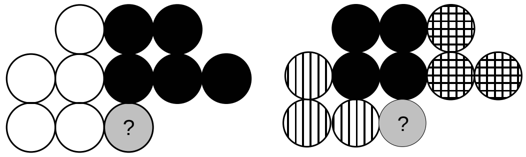

Figure 3: Sample contexts: binary (left) and generalized (right). Pixel which probability is estimated is marked with “?”

since 24-bit color histogram would be difficult to handle. Then, from the color histogram one considers a box enclosing the colors of the image. The idea of median cut is to split the box recursively until the desired number of palette colors is reached. At each step of the algorithm, the box containing largest number of pixels is split along the coordinate that spans the largest range. The split is made at the median point so that approximately equal number of pixels falls into sub-boxes.

2.2 GCT compression

Statistical context-based modeling is a well-known tool in image compression and it is widely used in various compression applications. The general idea is to exploit local correlation among pixels. In typical image, the knowledge about the neighborhood of the unknown pixel significantly improves its probability estimation, e.g. for most of documents, the probability of the current pixel to be white is very high when all its neighbors white. The neighborhood configuration is called a *context* and is defined by the context template. Figure 3, left picture illustrates sample binary contest, where background pixels are drawn as white and foreground as black. The estimated conditional probabilities are usually coded by arithmetic coder [9], as has been done in the very first standard for encoding of bi-level images – JBIG [13].

However, every context-based approach faces two major problems: memory consumption and *context dilution*. The information about estimated probabilities needs to be stored for every context. In case when every possible context is expected to appear in the image this number grows exponentially. For example, for 10-pixel context on a binary alphabet (JBIG) 2^{10} context configurations are possible. In case when K intensity gradations are expected, 10-pixel template results in K^{10} contexts, which is a huge number even for gray-scale images. The problem can be partially solved using the *Context Tree* (CT) modeling originally proposed by Rissanen [10]. This approach organizes the storing of probability estimations in a tree structure. In this way, only the information about the contexts that are really present in the image are stored, which significantly reduces memory consumption.

Context dilution problem is of different nature and cannot be solved only with optimized memory allocation. The problem is that larger context template does not always provide the increase in compression performance. With increasing of size, particular contexts do not appear frequently enough in the image for probability to be estimated accurately. Incorrect estimation degrades the efficiency of the entropy coder, and therefore, the compression efficiency. In CT modeling, this problem is solved by applying various tree pruning techniques. The idea is that if the parent node (smaller context) provides better compression than its children (larger context), then the children nodes of the tree are pruned and the parent is used instead for the probability estimation. The efficiency of compression is estimated by the entropy of the model.

Generalized Context Tree (GCT) modeling is a generalization of CT modeling. The tree is built for a K-color image; a sample context is illustrated in Figure 3 (right), where different colors of context pixels are illustrated with texture. Pruning is performed by *steepest descent search* algorithm resulting in sub-optimal tree configuration which, however, is very close to the best one obtained by full search. At the moment, GCT compression presents the best performance for lossless compression of computer-generated raster map images [6].

However, we experienced some difficulties when applying GCT on practice. First problem we met was the memory consumption. Though the algorithm was successfully applied in [6] for 67-color images, in our experiments we deal with up to 256-color images which required better implementation of context tree structure. We discovered that in case when storage of pixel counters in tree nodes is implemented as an array, about 90% of array elements are not used since in many-color images the actual variety of contexts is much smaller than it potentially could be. Therefore we modified the implementation of context tree and realized the storage of pixel counters as a linked list. This simple technical improvement dramatically increases the number of colors which GCT compressor is able to process same time making context tree faster to traverse.

Second improvement we considered for GCT is a fast pre-pruning of the tree. In our experiments we discovered that the most part of the tree is not filled with representative statistics since the most of the contexts do not appear in the image frequently enough but just one or twice. Though these contexts are pruned out by steepest descent search algorithm, it is computationally expensive and the vast of total processing time is spent on it. Therefore we considered a simple threshold-based pre-pruning. The idea is that the node (and the represented context) is pruned in case that its occurrence number falls below the predefined threshold. The surviving nodes are then processed by standard pruning algorithm.

The effect of optimization is illustrated in Table 1 for sample 1250×1250 image of 42 colors. Rows of the table represent memory consumption and processing time for original GCT, GCT with optimized memory allocation and for GCT with optimized memory allocation and pre-pruning. For images with more colors the effect is even more significant. In general, the use of these simple and effective optimization techniques made the algorithm applicable for 256-color 3000×3000 pixel images and 20-pixel context on a personal computer with 1G operative memory. Note that no optimization would deal with 256^{20} possible context configurations.

Table 1: The effect of memory optimization and pre-pruning

	Memory, MB	Time, sec
Original GCT	128	334
Optimized memory	30	326
Opt. memory + pre-pruning	30	72

3. EXPERIMENTS

3.1 Compression performance

We compare the performance of the proposed algorithm, referred further as Lossy Generalized Context Tree Modeling (L-GCT), with JPEG2000 [14], which is the recent standard for lossy image compression, and with ECW compressor [3]. For a test set we

Topo1
2461×3345
699925 colors

Topo2
2359×3497
703965 colors

Topo3
2291×3484
669724 colors

Figure 4: Samples of the test set images.

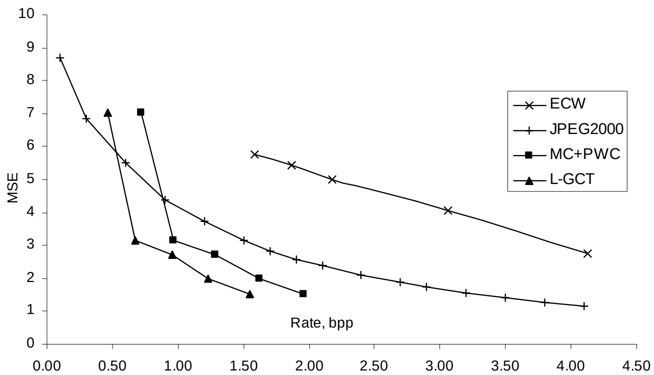

Figure 5: The compression performance of the proposed algorithm (L-GCT) and its competitors.

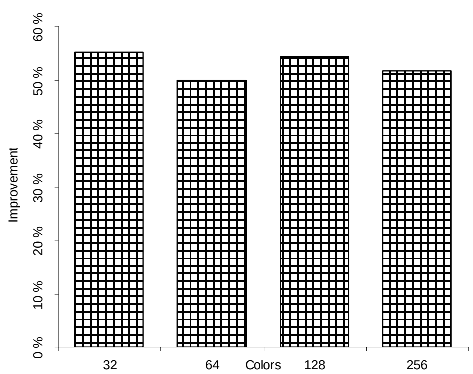

Figure 6: The relative compression improvement provided by L-GCT comparing to JPEG2000.

consider three scanned topographic maps of Finland: topo1, topo2 and topo3. Raster images are acquired by a flatbed scanner at 300 dpi. Samples and image dimensions are illustrated in Figure 4. The experiments are performed on P4-3GHz 1G RAM computer.

We measure the distortion caused by the lossy compression algorithm as MSE distance in $L^*a^*b^*$ color space [15]. The distance is measured from the degraded image to the scanned original. The operational rate-distortion function for JPEG2000 is estimated by considering 16 quality levels varying bit rate approximately from 0.1 to 4 bpp, and respectively, MSE distortion from 8.69 to 1.16. For the proposed compressor we consider 5 quality levels by defining the number of colors in the image as 256, 128, 64, 32 and 16. Images of 256-color are the practical limit of the proposed algorithm. In our experiments for L-GCT, we use 20-pixel context modeling with pre-pruning threshold level set to 32. The compression results – bits per pixel and MSE distance are measured as the average over the test set.

The compression performance of L-GCT and its competitors is illustrated in Figure 5. The proposed algorithm outperforms its competitors starting from 32-color images. Better performance is presented for the rest of quality levels up to 256-color images. The relative improvement over JPEG2000 with respect to the similar objective quality level is illustrated in Figure 6. The improvement of the proposed algorithm varies around 50% for images of 32 to 256 colors. The comparison with 'trivial approach' MC+PWC proved that GCT provides better lossless compression. ECW in our experiments performs worse than JPEG2000.

The processing time required by the proposed algorithm depending on the quality of the image is represented in Table 2. One can see that the most of the time is spent on the construction of the context tree. Encoding and decoding times are almost equal and are much smaller than the tree construction time.

As a disadvantage of the proposed algorithm one can still consider its compression time and memory consumption. For example for highest quality levels the compression of single image takes about one and a half hour. This restricts the use of the proposed approach in real-time applications, though the offline archiving is practical since decompression does not require significant time or memory.

Table 2: L-GCT processing time (sec) depending on the amount of color in the image.

	16	32	64	128	256
Tree constr.	204	333	591	1816	5021
Encoding	7	12	22	40	62
Decoding	11	16	28	49	71

4. CONCLUSION

We proposed a lossy compression method for scanned map images. The algorithm is based on color quantization, which is the lossy part, and context tree modeling, which is a lossless compression technique. The quantization is performed by median cut algorithm, and the compression is by Generalized Context Tree lossless compression algorithm. The compression efficiency is achieved mostly by the use of efficient GCT compressor, which utilizes specific properties of the scanned map imagery, such as low amount of unique colors, solid uniform or highly textured areas and sharp edges. The rate-distortion performance of the proposed algorithm is evaluated on a set of scanned topographic maps and compared to JPEG2000 and ECW wavelet-based lossy compressors. JPEG2000 is a recent standard for common lossy image compression and ECW is a commercial proprietary format for aerial and satellite image storage used also for the compression of scanned imagery. Also, in order to prove the efficiency of GCT we compared the proposed algorithm to the 'trivial approach' where the compression is performed by standard PWC compressor. The proposed algorithm outperforms JPEG2000 for about 50% in average by the provided rate for similar MSE distortion level. However, one can consider processing time and memory consumption as the drawbacks of the proposed technique.

5. FUTURE WORK

We believe that the potential of the algorithm needs to be investigated in more details. Such application areas could be considered as lossy compression of simple graphics – architectural schemes, engineering drawings; different types of scanned map images – historical maps, city plans, navigational and atlas-type maps. The effect of different type of sensor could also be studied; for example, simple graphics obtained with a digital camera. The optimal choice of the quantization scheme also an open question as well as the question of faster processing time of the algorithm.

6. REFERENCES

[1] LizardTech web site, http://www.lizardtech.com, accessed 18.3.2007.

[2] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, Y. Le Cun, "High Quality Document Image Compression with DjVu". Journal of Electronic Imaging, vol. 7 (3), pp 410-425, SPIE, 1998.

[3] ER Mapper web site, http://www.ermapper.com/ecw/, accessed 18.3.2007.

[4] P. Ausbeck, "The piecewise-constant image model", Proceedings of the IEEE, vol. 88 (11), pp. 1779-1789, 2000.

[5] Y. Yoo, Y. Kwon, A. Ortega, "Embedded image-domain adaptive compression of simple images", Conference record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1256–1260, 1998.

[6] A. Akimov, A. Kolesnikov and P. Fränti, "Lossless compression of color map images by context tree modeling", IEEE Trans. on Image Processing, vol. 16 (1), 2007.

[7] M. Weinberger, G. Seroussi, G. Shapiro, "The LOCO-I lossless image compression algorithm: principles and standartization into JPEG-LS", IEEE Trans. on Image Processing, vol. 9 (8), pp. 1309–1324, August 2000.

[8] X. Wu, N. Memon, "Context-based, adaptive, lossless image coding", IEEE Trans. on Communications, vol. 45 (4), pp. 437–444, 1997.

[9] J. Rissanen, G. Langdon, "Arithmetic coding", IBM Journal of Research, Development, vol. 23, pp. 146–168, 1979.

[10] J. Rissanen, "A universal data compression system", IEEE Trans. on Information Theory, vol. 29 (5), pp. 656–664, 1983.

[11] P. Heckbert, "Color image quantization for frame buffer display", Comput. Graph. 16, pp. 297-307, 1982.

[12] A. Podlasov, P. Fränti, "Merge–based color quantization and context tree modeling for compression of color quantized images", Proc. of IEEE Int. Conf. on Image Processing (ICIP'2006), Atlanta, 8-11 Oct. 2006, pp. 2277-2280.

[13] ITU-T recommendation T.82, "Information technology – coded representation of picture and audio information – progressive bi-level image compression", 1993.

[14] D. Taubman, M. Marcellin, JPEG2000: Image Compression Fundamentals, Practice and Standards, Kluwer Academic Publishers, 2001.

[15] CIE, Colorimetry, CIE Pub. No. 15.2, Centr. Bureau CIE, Vienna, Austria, 1986.

About the authors

Alexey Podlasov received his MSc degree in applied mathematics from Saint-Petersburg state University, Russia, in 2002, and the MSc degree in computer science from the University of Joensuu, Finland, in 2004. Currently, he is a doctoral student in computer science in the University of Joensuu. His research topics include processing and compression of map images.

Alexander Kolesnikov received the M.Sc. degree in physics in 1976 from the Novosibirsk State University, U.S.S.R., and the Ph.D. degree in computer science in 2003 from the University of Joensuu, Joensuu, Finland. From 1976 to 2003, he was a Senior Research Fellow with the Institute of Automation and Electrometry, Russian Academy of Sciences, Novosibirsk, Russia. In 2003, he joined the Department of Computer Science, University of Joensuu. His main research areas are in signal and image processing, vector map processing, and compression.

Pasi Fränti received his MSc and PhD degrees in computer science in 1991 and 1994, respectively, from the University of Turku, Finland. From 1996 to 1999 he was a postdoctoral researcher of the Academy of Finland. Since 2000, he has been a professor in the University of Joensuu, Finland. His primary research interests are in image compression, clustering and speech technology.