e-Health – For Continuity of Care
C. Lovis et al. (Eds.)

doi:10.3233/978-1-61499-432-9-1138

Performance Comparison of Compression Algorithms for Archiving Segmented Volumetric Binary Medical Data

Felix FISCHER[a], M. Alper SELVER[b,1], Oguz DICLE[c] and Walter HILLEN[d]
[a] *Nautavis GmbH, Linnich, Aachen, GERMANY*
[b] *Dokuz Eylul University, Dept. of Electrical and Electronics Eng., Izmir, TURKEY*
[c] *Dokuz Eylul University, School of Medicine, Dept. of Radiology, Izmir, TURKEY*
[d] *FH-Aachen, Juelich Division, Medical Informatics Laboratory, Aachen, GERMANY*

Abstract. Archiving result of a segmentation task allows the representation of the segmented volume at a later time. The segmented volume can be stored in a binary format, which can be restored by a simple combination of the original data with this binary information. Since, the sizes of the segmented binary data have high memory requirements; a lossless compression method should be employed for efficient archiving. Thus, this study examines different approaches for compression and their suitability for restoring binary segmentation results. To evaluate the compressive properties, multiple test cases with diverse spatial structures and acquired with different modalities from clinical practice have been used. The results show that best performance is achieved with JBIG2 method both in terms of compression ratio and processing time.

Keywords. Medical Image, Segmentation, Compression, DICOM, Visualization.

Introduction

Tomographic medical imaging systems (i.e. CT, MR etc.) produce two-dimensional cross-sectional images (i.e. slices) of the human body in DICOM format. The number of these slices may vary from hundreds to thousands based on the acquisition parameters. Mostly, diagnoses are preformed based on these individual slices, which require the physician to interpret the spatial context between slices. This task is very difficult, especially when the amount of data (i.e. number of slices) increases. Since, three-dimensional (3-D) analysis can overcome this difficulty; they are becoming a more frequent tool in parallel to the developments in 3-D technology.

Segmentation is a fundamental step of 3-D analysis that delineates the anatomical structure of interest (i.e. organ, tissue, tumor, vascular tree etc.). A binarized form of segmented volume can be defined as a set of voxels, whose values are 1 if they belong to the object and 0 otherwise. Once the binarized form is obtained; the segmentation result can be restored by a simple combination of the original data with the binary data. This makes the binary data an ideal tool for saving segmentation information, efficiently. Since, the sizes of the segmented binary data have high memory

[1] Corresponding Author.

requirements; they must be compressed with a lossless technique. Thus, this study examines different approaches for compression and their suitability.

To accomplish this task, several compression methods, which exploit the special structure of the (binary) segmentation results, are compared. It must be noted that there must be a lossless compression so that the original data can be fully restored. The following compression methods have been studied:

- Run-Length Encoding [3]
- Entropy Encoding: CCITT T.4/T.6 [4], JBIG2 [5], JPEG 2000 [7]
- Phrase Encoding: ZIP, LZW [4]
- Octree Encoding [8]

To evaluate the compressive properties multiple test cases from clinical practice have been collected. During the selection of datasets, special care was given to select anatomical objects with diverse spatial structure and acquisitions with different modalities. This ensures that the presented compression results are representative of a wide range of applications. Based on the application results, the most suitable method is determined for the compression of binary medical volumetric data sets.

1. Methods

Considering that a 3-D anatomical structure located in a volume of interest (VOI), we can define a segmented object as the set of voxels, whose values are equal to 1 inside the VOI, while all other (background) voxels are equal to 0. The VOI can be used to reduce the complete data set to a partial volume by specifying a rectangular area in all dimensions. Thus, in this study a VOI is defined for each object of interest prior to its segmentation and compression.

The tested compression methods - with the exception of the octree - work on two-dimensional images. Therefore, the data is first stored slice by slice in a series of single images in uncompressed bitmap (BMP) format. As a second step, the BMP image was converted in a Portable Bitmap image (PBM). In the PBM format, the pixels are stored bitwise (i.e. without compression); thus, it does not reduce the size of the original data. Since the volume data exists as a series of two-dimensional images in PBM format, they can be opened in graphics programs without problems and stored in other formats such as PNG or TIFF with additional compression options.

As mentioned above, the octree coding is a special case since, unlike the other methods, the segmented data is considered as a volume (i.e. not as series of slice images). The storage was carried out in two separate files: The values of the voxels are consecutively written to one file. In a second file, the dimensions of the VOI can be stored. With the help of these two files, it is possible to restore the binarized segmentation result as a volume.

The following paragraphs introduce further details of the compression methods used in this study and corresponding software programs to implement them:

Run-length encoding (RLE): The run-length encoding has been performed with the freely available Java software Birle [3], where the PBM image was used as the input file. However, this program is not designed specifically for the encoding of binary data. Although, run-length coding can be applied to PWM files, the data is processed byte by byte. This result with eight voxel values that are combined into one byte and this byte is encoded as a symbol. The result of the compression is stored in a raw file.

CCITT T.4/T.6: With the freely available graphics program IrfanView 3.98 [4] the images of different formats can be opened, viewed and converted into other graphics formats. By storing the PBM image in TIFF format, the test cases with the CCITT T.4 procedure and CCITT T.6 were compressed directly by the saving process, when the compression CCITT Fax 3 and CCITT Fax 4 were selected.

JBIG2: The encoding of the PBM image with the JBIG2 method was performed using the C + + program jbig2enc [5]. The compression obtained this way was written to the standard output and redirected to a file in JBIG2 format. With the C program jbig2dec -0.9 [6], the JBIG2 file was decoded again.

JPEG 2000: The free command-line program GeoJasper [7] allows the conversion of common graphic formats to JPEG 2000 (JP2) format and therefore, was used for the JPEG 2000 encoding. However, it only accepts image formats in which a pixel has at least one byte. For this reason, BMP image was used as the input file.

ZIP: The ZIP code was also carried out with IrfanView [4]. Through a lossless conversion of the PBM file in the graphic format PNG data, images were automatically compressed using the ZIP method. Also the run-length encoding of eight bits are grouped into a byte and then compressed in this case.

LZW: CCITT T.4 and CCITT T.6 encoding was chosen for the application of LZW under IrfanView under TIFF format, but with the compression option LZW. Again, the data processed byte by byte similar to the run-length encoding and the ZIP method.

Octree: Because there was no program available for the octree encoding, this was implemented independently according to an algorithm from [8], and then optimized for speed using the user manual of [9]. Input of the octree coding requires the two files containing the voxel values and the dimensions of the VOI.

2. Data sets

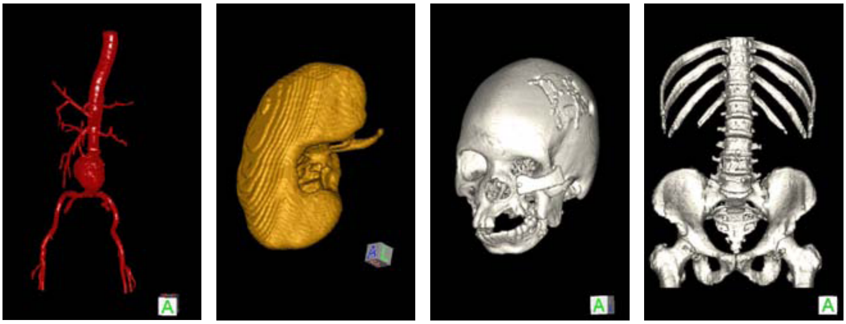
The data sets for testing compression methods are selected from acquisitions of different modalities and diverse anatomical objects with diverse spatial structure so that actual performance of the techniques can be evaluated for a wide range.

The first data set is called aorta, which is acquired with contrast medium (288 slices, slice thickness (ST): 1.5 mm), by a CT and segmented with the Connected Threshold method used in [2]. With a VOI of 139x322x288 voxels, "aorta" has the greatest dimensions and file size among all test objects (Figure 1.a).

The kidney was segmented from two different data sets (i.e. CT, MR), which provided different segmentation results for the interior of the kidney. The first data set is a coronal MRI image series with 72 slices with a slice thickness of 1.4 mm. The segmentation of the (right) kidney was performed using the Fast Marching method [2]. The selected VOI has the dimensions 121x52x205 voxels. The second kidney data set is a CT series with 238 slices with a slice thickness of 1 mm. Again, the Fast Marching method was used to segment kidney and the size of the VOI is 114x101x112 voxels.

The skull CT data set consists of 361 slices with a thickness of 0.7 mm. For the segmentation of the skull, the Connected Threshold method was chosen. For the skull, a VOI size 175x214x302 voxels was set (Figure 1.c).

Skeleton data set includes the ribs and the hip of a CT dataset (288 slices, slice thickness: 1.5 mm) and it was segmented with the Connected Threshold method (Figure 1.d). The corresponding VOI has a size of 239x146x288 voxels.



(a) (b) (c) (d)

Figure 1. Volume rendering based illustrations of selected medical data sets used for testing the compression algorithms, (s) aorta, (b) MR kidney, (c) skull, (d) skeleton.

3. Applications and Results

Tables 1 and 2 show the results of the compression in terms of the compression capacity. Since the size of PBM data correspond to uncompressed size of the original data ([width x height x depth of the VOI] x 1 bit + header of the PBM Data), it has the smallest possible compression ratio. Thus, these values are used as the benchmark for comparing the compression results (see first row of Table 1).

Table 1 shows the sizes of the original/PBM files and compressed files in kilobytes (kB). According to the VOI dimensions of the individual test cases, the size of the original files are relatively small for kidney data (i.e. 166 kB and 167 kB), while the files of the aorta, cranial and skeleton are significantly larger (i.e. over 1000 kB).

Table 2 is based on the data of Table 1 and shows the compression ratio (i.e. the ratio between the compressed file and the original file in percentage) for each of the algorithms. The procedures are arranged in descending order according to their average efficiency over all test cases.

In Table 3, the compression time of each method is given. The test cases are grouped into two categories: 1) test cases, whose PBM file is smaller than 1000 kB (i.e. kidney CT and kidney MRI) and 2) test cases, whose PBM file is larger than 1000 kB (i.e. aorta, skull and skeleton).

The compression of the test cases was carried out with the help of various software programs (described in section 2) using a computer with Intel Pentium 4 Processor (3.40 GHz), 2 GB of RAM and the Windows XP operating system.

4. Discussion

With a reduction of the VOI of the original data below 5% for all test cases even for very large volume data, the application of JBIG2 method performs the best among all other compression techniques used in this study. Moreover, the compression ratio of the JBIG2 method reduces the segmented binary volume data to acceptable file sizes. Another important advantage is that the compression time of JBIG2 is negligible with less than one second. Therefore, the JBIG2 method can be used effectively for compressing the binary segmentation result of volumetric data.

Table 1. File sizes of compressed test cases of all tested methods (in kilobytes (kB))

	Aorta	Kidney CT	Kidney MR	skull	skeleton
Original/PBM	1631	166	167	1389	1232
RLE	73	51	42	477	208
CCITT T.4	467	67	65	496	208
CCITT T.6	36	10	14	90	70
JBIG2	10	5	8	59	44
JPEG 2000	234	148	186	1442	882
ZIP	24	12	15	121	92
LZW	59	20	21	212	131
Octree	160	99	150	1393	825

Table 2. Compression ratios for all test cases (from top to bottom: high). The compression ratio is the ratio between the compressed file and the original file (PBM) in percent.

	Compression ratio in %				
	Aorta	**Kidney CT**	**Kidney MR**	**skull**	**skeleton**
JBIG2	0,61	3,01	4,79	4,25	3,57
CCITT T.6	2,21	6,02	8,38	6,48	5,68
ZIP	1,47	7,23	8,98	8,71	7,47
LZW	3,62	12,05	12,57	15,26	10,63
RLE	4,48	30,72	25,15	34,34	16,88
CCITT T.4	28,63	40,36	38,92	35,71	26,46
Octree	9,81	59,64	89,82	100,29	66,96
JPEG 2000	14,59	89,16	111,38	103,82	71,59

Table 3. Compression duration of each process in seconds.

	RLE	CCIT T T.4	CCITT T.6	JBIG2	JPEG 2000	ZIP	LZW	Octree
File<1000 kB	2-3	<1	<1	<1	<1	<1	<1	30
File> 1000kB	5-6	<1	<1	<1	6-7	<1	<1	35-700

References

[1] Salomon D, Data Compression: The Complete Reference, Springer-Verlag, New York, Inc., 1998

[2] Fischer F, Selver MA, Hillen W, Güzeliş C, Integrating Segmentation Methods from Different Tools into a Visualization Program Using an Object Based Plug-In Interface, *IEEE Transactions on Information Technology in Biomedicine*, **14(4)** (2010), 923-934.

[3] Tyler T, BIRLE – Bijective Run Length Encoding (RLE) compressor, http://mandala.co.uk/birle/, Last checked: 30.10.2008

[4] Skiljan I, IrfanView 4.20g, http://www.irfanview.de, Last checked: 30.10.2008

[5] Langley A, JBIG2 Encoder 0.27, http://www. imperialviolet.org/jbig2.html, Last checked: 30.10.2008

[6] Giles R, Levien R, jbig2dec 0.9, http://sourceforge.net/projects/jbig2dec, Last checked: 30.10.2008

[7] Levit D, Fedorov V, GeoJasper, http://www.dimin.net/software/geojasper/, Last checked: 09.09.2008

[8] Gargantini I, Linear Octrees for Fast Processing of Three-Dimensional Objects, *In Proceedings of Computer Graphics and Image Processing (CGIP)* **20** (1982), 365–374.

[9] Samet H, Applications of Spatial Data Structures, Addison Wesley, 1990.