ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2011.0021

IMPROVED COMPRESSION OF XML FILES FOR FAST IMAGE TRANSMISSION

S. Manimurugan[1] and K. Porkumaran[2]
[1]Department of Computer Science and Engineering, *Karunya University, Tamil Nadu, India*
E-mail: smanimurgan@yahoo.co.in
[2]*Department of Electrical & Electronics Engineering, Dr.NGP Institution, Anna University, Tamil Nadu, India*
E-mail: porkumaran@gmail.com

Abstract
The eXtensible Markup Language (XML) is a format that is widely used as a tool for data exchange and storage. It is being increasingly used in secure transmission of image data over wireless network and World Wide Web. Verbose in nature, XML files can be tens of megabytes long. Thus, to reduce their size and to allow faster transmission, compression becomes vital. Several general purpose compression tools have been proposed without satisfactory results. This paper proposes a novel technique using modified BWT for compressing XML files in a lossless fashion. The experimental results show that the performance of the proposed technique outperforms both general purpose and XML-specific compressors.

Keywords:
XMl File, Encoding, DeCoding, GST, EC, RLE

1. INTRODUCTION

The health care industry is one of the world's largest and fastest-growing industries, aims at providing quality healthcare to all kinds of people. The development of modern equipments and communication devices has made it possible to take specialty healthcare to the rural and remote population of a country. According to [30], the future of healthcare industry is shaped by teleradiology and technologies such as telemedicine. Patient information, which may include patient details, treatment history, images of previous tests, etc., plays an important role in these technologies. In particular, image data acquired from various medical equipments like X-Ray, CT, MRI, etc. is increasingly transmitted through World Wide Web, telephone, mobile, WANs, etc. Methods of transmitting medical images have been studied by many researchers including [22] and [23]. All these systems face the following three common problems, security, speed and image quality. According to [1], illegal data access has become prevalent in wireless and general communication networks and therefore there is a high demand for techniques that protect data being transmitted. The second problem is the time taken to transmit an image from source to destination. This depends largely on the network bandwidth and traffic but the file size also has a great impact on transmission speed. Most importantly, the main challenge faced by the transmission systems is the demand of doctors and healthcare professionals for a lossless image after transmission. The reason behind this demand is to make correct diagnosis, which is very difficult with a distorted image. According to [13], the current need of medical transmission systems is a new storage technology that can be used by both hospital CIOs and transmission system and which somehow reduces the storage space required while maintaining the quality of data. Transmission time is directly related to the compression and decompression time along with a good compression ratio. Thus, the transmission of medical images is a multidisciplinary process, where transmission speed, bandwidth utilized and security plays a crucial role. SMIT (Secure Medical Image Transmission) is a system proposed to satisfy these three requirements, that is, small file size, maximum speed and lossless image quality.

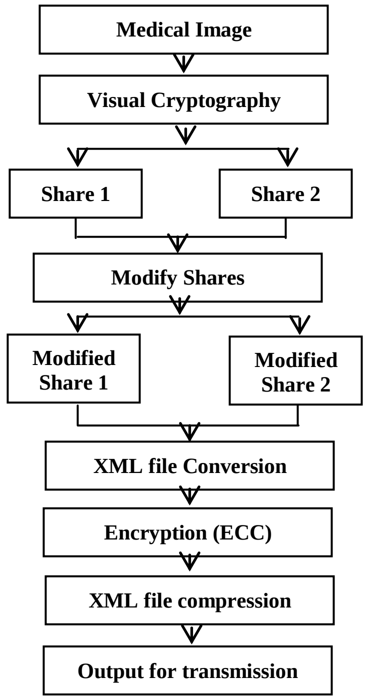


Fig.1. SMIT Architecture

SMIT is proposed as a solution to satisfy these three demands of a medical image transmission system. SMIT can be used to transfer medical images from hospitals to remote areas. SMIT is designed in a way to help healthcare professionals to access images immediately on emergency cases. The architecture used to build SMIT is shown in Fig.1. SMIT starts with an input medical image, which is divided into two share files using visual cryptography. The two share files are modified using several bit wise row and column transformations. The transformations are performed solely to interchange the rows and columns of an image file so as to confuse content stealers. The modified files are then converted into a single XML file. The XML file is encrypted using ECC algorithm and then compressed. This compressed encrypted file is used during transmission. A reverse process is used at the destination to obtain the original image file. SMIT is comprised of five steps: (i) visual cryptography (ii) pixel modification (iii) XML conversion (iv) encryption and (v) XML compression. This paper, concentrates on developing a scheme for Step 3. The method uses BWT for compressing XML file, which uses a star-entropy a novel method using BWT as basis is also proposed.

The paper is organized as follows: Section 2 provides an overview to XML technology. Section 3 explains the XML format used, while the technique used to compress this file is explained in Section 4. The results obtained while testing the proposed compressor is presented and discussed in Section 5. Section 6 concludes the work with future research directions.

2. OVERVIEW OF XML

XML is a markup language for documents containing structured information, which contains both contents (words, picture, etc.) and role of content (heading, footnote, etc.). A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way to add markup to documents. XML was first proposed in 1998 by World Wide Web Consortium (W3C) as an alternative for its two predecessors, HTML (Hypertext Markup Language) and SGML (Standard Generalized Markup Language). From its advent, XML has gained wide acceptance and is being used in many applications as a standardized data exchange format. XML is often referred as "self-describing data" and are designed to use self-defined tags to carry data, but not to display them. It is designed in such a way that the schema is repeated for each record in the document. XML has become the de-facto feature in information exchange on World Wide Web and it is clear that an enormous amount of data in the Internet will be encoded in XML in the near future because of its extensibility and characteristic of cross-platform. Many expectations are surrounding XML, as it simplifies data exchange among heterogeneous computers.

The information explosion has intensified the demand for computers, and, in turn, induced a number of healthcare professionals to connect with one another via the ever-growing World Wide Web. The current era is witnessing several standardized web services, which enable users to receive and transmit patient data and medical images all over the world. It has been noticed in World Wide Web transmissions, users are increasingly using XML (eXtensible Markup Language) formatted files for transmitting data and images [38]. XML is rapidly becoming a standard format for electronic data structuring, storage and exchange [7], [41] and this has motivated the paper to use XML as a standard during transmission.The usage of XML for transferring data has several advantages like its semi-structured nature of representing the data, the semantics that can be added by tag-names, the simple data exchange process among heterogeneous computers [3], etc. These advantages pin point the fact that the usage of XML will grow substantially in the near future. The self describing feature grants the XML great flexibility that it has been enthusiastically adopted in many areas of computer industry; for example, it is being increasingly used in a variety of database and eBusiness applications.

2.1 XML FILE

An example of XML file is shown in Fig.2. An XML file consists of declaration (Line 1), attributes (line 3), comments (lines 4, 8), element (lines 5-7, 9-11), contents (lines 6, 10). Declaration line specifies properties such as XML version, text encoding method, etc. An attribute is similar to a variable and an element consists of elements, comments, attributes and contents. An XML file is case sensitive and a well-formed file contains a single root element (For Example: Article).

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <Article><Title>XML Compressor</Title>
3 <Info author="Krishnan"></Info>
4          <!-- Introduction to the topic -->
5          <Chapter><Title>History</Title>
6                   <Para>... W3C in 1998...</Para>
7          </Chapter>
8          <!—next chapter -->
9          <Chapter><Title>Summary</Title>
10                  <Para>... XML compression is … </Para>
11         </Chapter>
12 </Article>
```

Fig.2. Sample XML File

2.2 XML COMPRESSION

Even though the XML files have become indispensable in many areas, it has a significant disadvantage of huge size. XML files are text-based and highly verbose in nature, which means that the amount of data that has to be transmitted, processed and stored is often larger than the other data formats. This huge size introduces bandwidth communication overhead, end to end communication delay, increased processing time and increased storage space [21], [34]. These problems can be addressed by using data compression to reduce the amount of storage required to store XML files. This has attracted several researchers and academicians to concentrate on the development of XML-specific compressors and binary formats.The problem of XML file compression has been studied by various researchers and academicians [20], [16] and several XML compression tools have been proposed in the literature of the recent years [6], [26]. Some aim for maximum compression ratio, that is, small size [25], [2], while others focus on striking a balance between maximum bandwidth and compression/decompression times [17], [36]. Research works are also conducted to answer queries directly from compressed XML file [37], [12].

Most of the existing methods focus either on achieving high compression ratio, maximum compression speed or minimum storage space and are lossy. Skibinski *et al*. [35] is of the opinion that producing an exact replica of the compressed file after decompression is an important feature of XML compressors, which are often neglected by the existing compressors. Moreover, they also stressed on the fact that as archived XML files are more often decompressed than recompressed, quick decompression is considered as a key requirement in a XML compressor.

Motivated by these observations, this paper aims to develop an efficient fast unqueriable compressor for lossless compression of XML documents with high compression ratio. The main objective is to produce a compression model that minimizes the storage requirement, time taken to compress and decompress, while maintaining a lossless nature.

2.3 TAXONOMY OF XML COMPRESSION TECHNIQUES

Generally, the XML conscious techniques can be categorized into queriable compression and unqueriable compression techniques. Queriable techniques come with the important feature that they can be queried directly on the compressed data, but unfortunately do not perform well in terms of compression ratio and execution time. Unqueriable techniques need to fully decompress their data before it can be queried again, but they can achieve a much higher compression ratio [39]. Examples of queriable compressors include XGrind [37], XMLZip [42], XPress [27] and XML Skeleton Compression [8] and unqueriable compressors include XMill [25], XMLPPM [11], SCA [24] and Millau [36]. Several generic compression techniques like gzip [14], bzip2 [10], winzip [40] and pkzip [31] have also been used but not with satisfactory results. The performance of various queriable and unqueriable compressors was compared by [29] and [3]. Similar work was also done by [32] and [33] along with some guidelines and recommendations which are useful for helping developers and users for making an effective decision towards selecting the most suitable XML compression tool for their needs. The taxonomy of XML compressing techniques is shown in Fig.3.

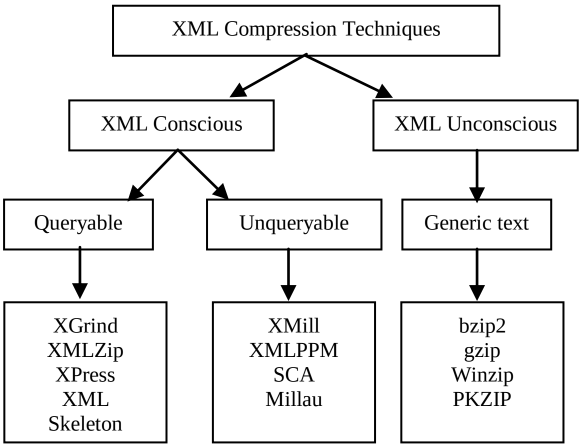


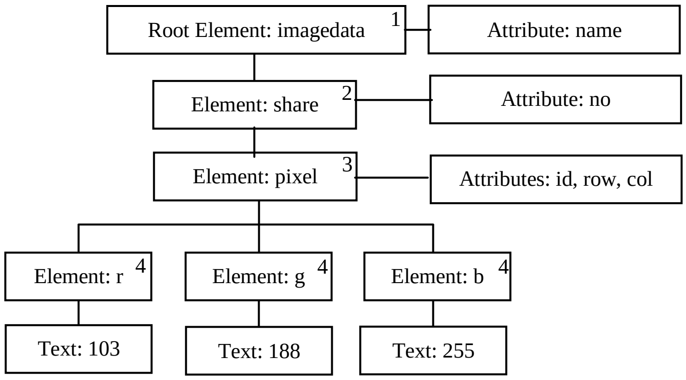
Fig.3. Taxonomy of XML Compression Techniques

3. XML FORMAT USED IN SMIT

A sample XML file produced by Step 3 is shown in Fig.4. The first line is the declaration line, which defines the XML version and encoding technique used. The next line contains the root element (imagedata) of the document. The root a child element, share with an attribute “no” indicating the share number. Each share element has several pixel sub-child elements with three attributes, id, row and column, which correspond to a unique pixel identification number, row and position of the pixel in the share. The red, green and blue colour value of the pixel is stored using the r, g and b elements respectively. After successful creation of the XML file, the file is embedded with an EOF character. The tree structure of the sample XML file is shown in Fig.5.

```
<?xml version="1.0" encoding="UTF-8"?>
<imagedata>
<share no="1">
<pixel id="0" row="0" column="0">
<r>21</r> <g>171</g> <b>81</b>
</pixel>
<pixel id="1" row="0" column="1">
<r>203</r> <g>75</g> <b>78</b>
</pixel>
...........
</share>
<share no="2">
<pixel id="0" row="0" column="1">
<r>13</r> <g>188</g> <b>255</b>
</pixel>
<pixel id="1" row="0" column="2">
<r>100</r> <g>20</g> <b>91</b>
</pixel>
...........
</share>
</imagedata> EOF
```

Fig.4. XML File



1 – Parent; 2 – Child; 3 – subchild; 4 – Siblings

Fig.5. XML Tree

4. PROPOSED SYSTEM

The present research work a transform encoding method for compressing XML files. Several transformation techniques like Fourier transform, Discrete Cosine Transform (DCT) or wavelet transforms [BGGu98] exists for video and image compression. In the text compression field, a frequently used transformation technique is the Burrows-Wheeler Transform (BWT) [9]. The BWT is one of the best lossless compression methods available. It is an approach that exploits the redundancy of data in the source file and has been applied in various fields as a compression method. It is also called as a "block sorting" compression technique, because it takes a block of text and permutes them. BWT was designed for textual data which works on block redundancy; it is expected to perform well on XML data [3]. The output of BWT is combined with ad-hoc compression techniques (run length encoding and Move-To-Front (MTF) encoding [4] [5] and Huffman coding [15], [18] to provide one of the best compression ratios available on a wide range of data. Several text compressors are designed based on BWT with bzip2 being the most famous one [19]. The process of BWT is shown in Fig.6. The transform encoding used should have the important feature that it should be exactly reversible, so as to achieve lossless compression. Further, it should help to improve the compression efficiency and save time.

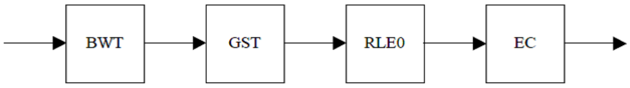


Fig.6. BWT Based Compression

The BWT stage converts the text into a form that is more suitable for compression. However, the BWT has some serious flaws, like, time consuming and extensive usage of memory. To solve these problems, the BWT transformation is modified to use a simple pre-processing step, which uses a internal dictionary based technique. The modified system is termed as Short Tag Dictionary Based BWT (STD-BWT). STD-BWT works by replacing repeated tags with short codes from a given static dictionary. Care was taken while constructing the dictionary, to keep the dictionary size small and easy to maintain. Dictionary based technique was preferred since it does not take into account any higher level correlations relating to the XML structure and can perform a lossless coding. The dictionary used in the present work is created in advance and is shared by both compressor and decompressor. The symbols used to create codewords are given in Table 1.The code replacement procedure is simple. The XML file is traversed from the beginning till it reaches EOF character. As soon as a “<” symbol is encountered, the next “>” symbol is searched and all the characters within and including “<” and “>” is replaced by the short code from the dictionary. A simple lookup procedure is used to find the code corresponding to the extract tag in the dictionary. A space is always encoded as “*” and all other symbols are written as it is but within “{“ and “}”.The XML file apart from tags, spaces and data values also has CR/LF symbols (Carriage Return / Life Feed). The CR/LF are used to separate each line in the XML file, which is the most frequently used symbol. The combined CR/LF symbols are replaced by ##. After replacement, the file will have frequent “*” and “#” symbols which will be best utilized for high compression ratio. After transformation, the next step performs a Global Structure Transform (GST) which transforms the local context of the symbols to global context. A typical representation of GST is a Move-To-Front transformation (MTF), which is a List Update Algorithm (LUA), which replaces the input symbols with corresponding ranking values. The third stage shrinks the number of symbols by applying a simple Run Length Encoding (RLD) and finally in the last stage the result is entropy coded (EC). In the present work, arithmetic coding is used in this stage. The paradigm used in this scheme is given in Fig.7.

Table.1. Internal Dictionary

Tag	Assigned Code
<?xml version="1.0" encoding="UTF-8"?>	A
<imagedata>	B
</imagedata>	C
<share no="1">	D
<share no="2">	E
</share>	F
<pixel id="	G
" row="	H
" column="	I
</pixel>	J
<r>	K
<g>	L
<b>	M
</r>	N
</g>	O
</b	P

5. EXPERIMENTAL RESULTS

The proposed compressor was vigourously tested with several XML files. All the experiments were conducted using a Pentium IV Dual Processor with 2GB RAM. Six XML files with varying size were used to evaluate the proposed system. The proposed system was tested with various existing compression techniques, namely, Winzip, gzip, pkzip, XMill and BZlib algorithms. The existing compression techniques for convenience sake are called as base algorithms in this paper. The algorithm was tested using three performance metrics, namely, compression ratio, compression time, decompression time and efficiency gained. The compression ratio can be measured as the ratio of the number of bits required to represent the image before compression to the number of bits required to represent the same image after compression and is given by the formula,

CR = size of input image / size of output image (1)

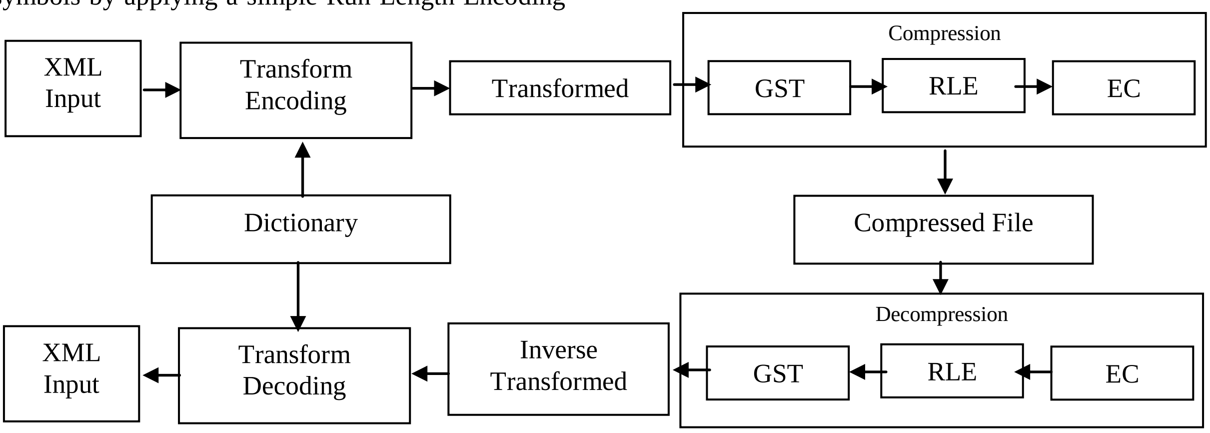


Fig.7. Proposed Scheme

Table.2. Compression Ratio

File Name	Original Size	Win zip	G zip	Pk zip	X Mill	B Zlib	STD-BWT
File32.XML	3Kb	82	86	88	66	86	88
File64.XML	12Kb	84	93	88	72	94	96
File128.XML	44Kb	86	94	87	77	90	92
File256.XML	678Kb	88	92	88	79	92	93
File512.XML	3116Kb	84	90	86	89	93	94
File1024.XML	6114Kb	83	91	87	90	95	96
Average		84.50	91.00	87.33	78.83	91.67	93.17

From the above equation, it is obvious that as the compression ratio increases the compression technique employed is more effective. Compression and decompression time are the basic measurements used to evaluate an image compression system. Compression and decompression time denotes the time taken for the algorithm to perform the encoding and decoding algorithm respectively. The results obtained are presented in the following subsections. The compression ratio obtained using the proposed and base algorithms for the six test corpora are given in Table.2.

From the results projected, it can be seen that the proposed algorithm produces better compression than the base algorithms. In terms of compression ratio, a percentage gain efficiency of 11.45%, 3.79%, 7.84%, 10.75% and 1.32% was obtained over Winzip, gzip, pkzip, XMill and BZlib base algorithms. The XMill algorithm performance is better with large sized files, but its performance degraded with small sized files. The STD-BWT compressor is consistent in its results and produce good compression ratio, which means small size during transmission. The compression and decompression time (in seconds) obtained is given in Table.3 and Table.4 respectively.

Table.3. Compression time

File Name	Winzip	Gzip	Pkzip	XMill	Bzlib	STD-BWT
File32.XML	0.83	0.89	0.87	0.64	0.51	0.49
File64.XML	0.85	0.91	0.93	0.66	0.62	0.48
File128.XML	0.88	0.95	0.99	0.72	0.62	0.63
File256.XML	0.88	0.98	0.103	0.79	0.77	0.53
File512.XML	0.91	0.113	0.111	0.86	0.83	0.78
File1024.XML	0.94	0.121	0.101	0.93	0.91	0.88

Table.4. Decompression Time

File Name	Winzip	Gzip	Pkzip	XMill	BZlib	STD-BWT
File32.XML	0.77	0.82	0.66	0.64	0.54	0.32
File64.XML	0.71	0.80	0.72	0.67	0.64	0.41
File128.XML	0.65	0.77	0.76	0.73	0.60	0.53
File256.XML	0.65	0.76	0.73	0.79	0.74	0.47
File512.XML	0.62	0.64	0.86	0.88	0.80	0.66
File1024.XML	0.61	0.59	0.89	0.94	0.89	0.80

Again from the table it can be seen that STD-BWT compressor is the fastest in compressing as well as decompressing process. Another desirable feature noticed is that the decompression time is less than the compression time which is a desirable feature as the healthcare professionals normally do not have patience or time to weight for decompression. Further, the STD-BWT compressor outperforms all the base compressors both in terms of compression ratio achieved and time. According to [28], compression ratio and speed are the two most important performance factors of any compression algorithm. From the results obtained it can safely concluded that the proposed algorithm makes it an attractive option for several advanced communication applications where XML files are used for transferring image data over World Wide Web and wireless networks.

6. CONCLUSION

Transmission of images over wireless networks and WWW is tremendous and is expected to increase in future. A well-known method of transmitting images is using XML file format where the image is converted to XML file. The main drawback XML files are its huge file size. This paper attempts to produce a novel method to solve this problem by using a modified BWT based lossless compression method. The modified method uses a dictionary which assigns small codes to the tags and other elements of the XML file. The results indicate that the proposed model meets the requirement of the current industry standards and can be used in by all transmission applications which needs a fast compressor with high compression ratio. In future, the STD-DWT model will be incorporated in SMIT and the effect of compressor on quality will be studied.

REFERENCES

[1] Achanta, H.K. and Korrai, D.R., 2009, "A secure scheme for medical image transmission", Joint IEEE North-East Workshop, Circuts and Systems and TAISA Conference (NEWCAS-TAISA'09), pp.1-4.

[2] Adiego, J., Navarro, G. and de la Fuente, P., 2007, "Using structured contexts to compress semistructured text collections", Information Processing and Management, Vol.43, No.3, pp.769-790.

[3] Augeri, C.J., Bulutoglu, D.A., Mullins, B.E., Baldwin, R.O. and Baird, L.C., 2007, "An Analysis of XML Compression Efficiency", Proceedings of Experimental computer science, No. 7, ACM Publications, pp. 1-12.

[4] Bentley, J.L., Sleator, D.D., Tarjan, R.E. and Wei, V.K., 1986, "A Locally Adaptive Data Compression Scheme", Proc. 22nd Allerton Conf. On Communication, Control, and Computing, Vol.29, No.4, pp.320-330.

[5] Bonifati, A., Lorusso, M. and Sileo. D., 2009, "XML Lossy Text Compression: A Preliminary Study", Proceedings of the Sixth International XML Database Symposium (XSym 2009), LNCS 5679, pp.106-113.

[6] Bottcher, S. and Steinmetz, R., 2006, "DTD-Driven Structure Preserving XML Compression", Flexible and Efficient Information Handling, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol.4042, pp. 266-269.

[7] Buneman, P., Grohe, M. and Koch. C., 2003, "Path Queries on Compressed XML", Proceedings of the 29th International Conference on Very Large Data Bases, pp.141-152.

[8] Burrows, M. and Wheeler, D.J., 1994, “A Block-sorting Lossless Data Compression Algorithm”, Technical Report 124, Digital Equipment Corporation.

[9] Bzip2. http://sources.redhat.com/bzip2

[10] Cheney, J., 2000, “Compressing XML with Multiplexed Hierarchical PPM Models”, Proceedings of the IEEE Data Compression Conference, pp. 163-172.

[11] Ferragina, P., Luccio, F., Manzini, G.and Muthukrishnan.S., 2006, “Compressing and searching XML data via two zips”, Proc. Int’l Conf. on the World Wide Web. ACM, pp. 751–760.

[12] Fluckinger, D., 2010, “New medical image storage technology may ease burden on hospital CIOs”, Health IT and Electronic Health News, TechTarget Corporate, http://www.techtarget.com.

[13] Gailly, J. and Adler, M., 2010, “gzip 1.2.4”, http://www.gzip.org.

[14] Gallager, R.G, 1978, “Variations on a theme by Huffman”, IEEE Trans. Information Theory, Vol.24, No.6, pp. 668-674.

[15] P. Skibinski, J. Swacha and Sz. Grabowski, 2008, “A highly efficient XML compression scheme for the web”, Information Technology and Libraries, Vol.28, No.3, pp.143-153.

[16] Girardot, M. and Sundaresan, N., 2000, “Millau: an encoding format for efficient representation and exchange of XML over the Web”, Computer Networks, Vol. 33, pp.747–765.

[17] Huffman, D.A., 1952, “A Method for the Construction of Minimum Redundancy Codes”, Proc. IRE, Vol. 40, No.9, pp.1098-1101.

[18] Lansky, J., Chernik, K. and Vlckova, Z., 2007, “Syllable-Based Burrows-Wheeler Transform”, Proceedings of the Dateso, Annual International Workshop on Databases, Texts, Specifications and Objects, Vol.235, pp. 1–10.

[19] League, C. and Eng, K., 2007, “Schema-Based Compression of XML Data with Relax NG”, Journal of Computers, Vol. 2, No. 10, pp. 9-17.

[20] League, C. and Eng. K., 2007, “Type-Based Compression of XML Data”, Data Compression Conference (DCC '07), pp. 273-282

[21] Lee, S., Lee, T., Jin, G. and Hong, J., 2008, “An Implementation of Wireless Medical Image Transmission System on Mobile Devices”, Journal of Medical Systems, Vol. 32, No.6, pp. 471-480.

[22] Lee, Y., Kwak, J., Cho, S., Kim, J. and Kang, H., 2007, “A Study on the Medical Image Transmission Service Based on IEEE 802.15.4a”, Advances in Hybrid Information Technology, Vol. 4413, pp. 159-167.

[23] Levene, M. and Wood, P.T., 2002, “XML Structure Compression”, Proceedings of the Second International Workshop on Web Dynamics.

[24] Liefke, H. and Suciu, D., 2000, “XMill: An Efficient Compressor for XML Data”, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 153-164.

[25] Maneth, S., Mihaylov, N. and Sakr, S., 2008, “XML Tree Structure Compression”, Proceedings of the 3rd International Workshop on XML Data Management Tools and Techniques, pp. 243-247.

[26] Min, J.K., Park, M.J. and Chung, C. W., 2003, “XPRESS: A Queriable Compression for XML Data”, Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 122-133.

[27] Müldner, T., Leighton, G. and Diamond, J., 2005, “Using XML compression for WWW communication”, Proceedings of the IADIS WWW/Internet 2005 Conference.

[28] Ng, W., Lam, W. and Cheng, J., 2006, “Comparative analysis of XML compression technologies”, World Wide Web, Kluwer Academic Publishers, Vol.9, No.1, pp.5–33.

[29] Norcen, R., Podesser, M., Pommer, A., Schmidt, H.P. and Uhl, A., 2003, “Confidential storage and transmission of medical image data”, Computers in Biology and Medicine, Vol. 33, pp.277–292.

[30] pkzip. http://www.pkware.com

[31] S. Sakr, 2009, “An Empirical Evaluation of XML Compression Tools”, Proceedings of the 1st International Workshop on Benchmarking of XML and Semantic Web Applications, Vol. 5667, pp. 49-63.

[32] S. Sakr, 2009, “XML compression techniques: A survey and comparison”, Journal of Computer and System Sciences, Vol. 75, No.5, pp. 303-322.

[33] Sayeed, S., Liscano, R. and Ahmad, A., 2007, “Performance Measurement of XML Compression Algorithms for DSRC Messages”, 32nd IEEE Conference on Local Computer Networks (LCN 2007), pp. 677 – 684

[34] Skibinski, P., Grabowski, S. and Swacha, J., 2007, “Effective asymmetric XML compression”, Software: Practice and Experience, Vol.38, No.10, pp.1027 – 1047.

[35] Sundaresan, N. and Moussa, R., 2001, “Algorithms and Programming Models for Efficient Representation of XML for Internet Applications”, Proceedings of the 10th International WWW Conference, pp. 366-375.

[36] Tolani, P.M. and Haritsa, J.R., 2002, “XGRIND: A Query-friendly XML Compressor”, Proceedings of the 18th ICDE, pp.225-234.

[37] Toman, V., 2004, “Syntactical Compression of XML Data”, Proceedings of the 16th International Conference on Advanced Information Systems Engineering.

[38] Veldman, I., Keijzer, A. and Keulen, M., 2009, “Compression of Probabilistic XML Documents”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Vol. 5785, pp. 255-267.

[39] Winzip. http://www.winzip.com

[40] World Wide Web Consorcium, 2010, Extensive Markup Language (XML) 1.0. URL: http://www.w3.org/TR/2000/REC-xml-20001006, Last Access Date: 03-02-2010.

[41] XMLZip - XML Solutions. http://www.xmls.com.