Impact of Compression Algorithms on Data Transmission

Vandana Jindal, A. K. Verma & Seema Bawa

Thapar University, Patiala

***Abstract* – Reduction in file size leads to the reduction in the number of bits required to store it. When data is compressed, it must be decompressed into its original form bit for bit. It means that the restored data is identical to the original data. Such type of data compression is referred to as lossless data compression. Data compression works by the identification of patterns in a stream of data. The most important term in compression theory is the Shannon's Limit. Compression ratio or ratio of the size of a compressed file to the original uncompressed file is the main parameter in which the effectiveness of any compression software may be measured.**

Keywords – Shannon's Limit, Compression ratio, CS&Q, JPEG and MPEG.

I. INTRODUCTION

The word *"compression"* means reduction in size but retaining its meaning. Sometimes "encoding" may be referred to as data compression and "decoding" as decompression. In computers various methods may be employed for reducing the size of a file. Reduction in file size leads to reduction in the number of bits required to store it. Coming to the latest technological developments like in the field of Wireless Sensor Networks (WSNs) etc., data compression holds a vital place. WSNs have a number of sensor nodes which are extremely small in size, consuming low power and costing less. WSNs overweigh traditional networks in the areas of deployment, scalability, ease of use and mobility [1]. Due to lack of structure and resources, WSNs have a few limitations as compared to the traditional network. To name a few – in terms of energy, processing, power consumption, memory, complicated structure topology etc. Storing the data and its transmission involves money. More the information, higher the cost. Moreover, this data, rather the digital data is not stored in the most optimum manner. They may be stored as: ASCII text or binary code. These methods require files, almost double the size of the data actually needed to represent the information. The above problem may be solved most optimally by employing data compression, where this data is changed into its compact form. *"Compression ratio"* [2,3,4] or ratio of the size of a compressed file to the original uncompressed file is the main parameter in which the effectiveness of any compression software may be measured. It is out of question to coin a particular data compression technique as better than the other, as a Lossy technique may prove to be better than the lossless technique in one case and vise-versa.

Data + Compression = information – redundancy

The term ``compression rate" comes from the transmission camp, while ``compression ratio" comes from the storage camp.

$$\text{Compression ratio} = \frac{\text{Compressed size}}{\text{Uncompressed size}}$$

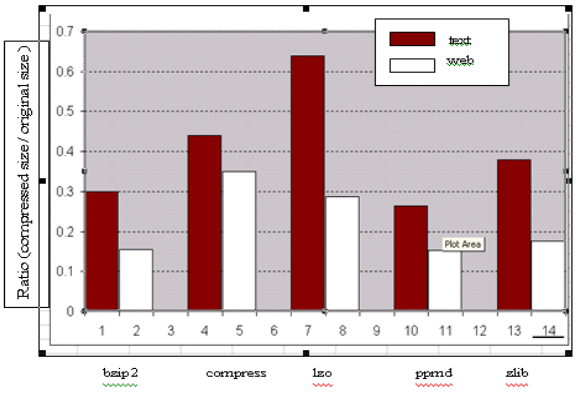

Fig.1 : Compression ratio [11]

II. WHY COMPRESSION?

Very obvious reasons for compressing the data are:

Cost of disk: No doubt that the disks are cheap but disks used for high end systems are expensive. Moreover, if replication or mirroring etc. are used all these will ultimately enhance the hardware cost.

Cost of data management: It takes very long for backup and recovery when the database is very big. Compressed the database, smaller will be the backup.

Memory: Compressed data will occupy less space thus more data can be placed in the same area (memory). For e.g., If the data is compressed by 50%, we can place almost twice the size of the data.

Bandwidth and transfer speed: Compressed data uses lesser bits as compared to uncompressed data resulting in less usage of bandwidth when downloaded. Hence resulting in quicker transfer speed.

If the file has n-characters, then the

Savings = (8n-5n)/8n => 37.5%.

A. *How does compression work?*

Data compression works by the identification of patterns in a stream of data. It chooses a more efficient method to represent the same information. Essentially, an algorithm is applied to the data in order to remove as much redundancy as possible. The most important term in compression theory is – "Shannon's Limit". This limit tells us how far we can compress a given source of data. Beyond that particular point, it is impossible to reliably recover the compressed data. Modern compression algorithms coupled with today's fast processors allow users to approach Shannon's Limit. However, they can never cross it.

B. *Advantages of using Compression*

Fits more data onto your backup device (e.g. Hard disk), Reduces size of database significantly, Lowers total cost of ownership (TCO), No application changes, Performance gains, Easy to enable/disable.

C. *Disadvantages of compression*

Backup file must be uncompressed before performing restoration, Temporary storage- like hard disk is required to perform compression, Backup will take longer to run, because compression can be slow and is performed as an additional step once the backup has finished.

III. APPLICATIONS

1. Generic file compression.
   - Files: GZIP, BZIP, BOA.
   - Archives: PKZIP.
   - File systems: NTFS.
2. Multimedia
   - Images: GIF, JPEG.
   - Sound: MP3.
   - Video: MPEG, DivX™, HDTV.
3. Communication
   - ITU-T T4 Group 3 Fax.
   - V.42bis modem.
4. Databases
   - Google.

IV. CATEGORIES OF DATA COMPRESSION

The categories into which the data compression techniques may be classified are as follows:

1. Lossy V/s Lossless
2. 0D V/s 1D V/s 2D V/s 3D.
3. Context dependent V/s Non-Context dependent.
4. Variable-to-block V/s Block-to-variable.
5. Short (< ~ 200 char), medium (~20K char), large (>~ 2 M char).
6. Zero latency decoding, medium latency coding, Full-latency.

A. *Lossy V/s Lossless*

Lossless Compression: When data is compressed it must be decompressed into its original form bit for bit. It means that the restored data is identical to the original data. The lossless compression algorithm usually use statistical redundancy in representing the data, free of errors. This lossless compression is feasible as there exists statistical redundancy in almost all the real-world data. E.g. of compression techniques: Run-length, Huffman, Delta and LZW.

Compressing a number say 82.999999999 will be written as 82.[9]9 (in its compressed form).

Input message = Output message

Lossy data compression or Perceptual Coding: In this type of compression, the data is compressed but the decompressed data does not change into bit for bit. It results into some degradation of the data. The resultant decomposed data is always an approximation. The lossy techniques are more effective when compressing the data rather than the lossless methods. For e.g. compressing 82.999999999 will be compressed as 83. A few examples of Lossy data compression techniques are CS&Q (coarser Sampling and/or Quantization), JPEG and MPEG etc.

Input message ≈ Output message

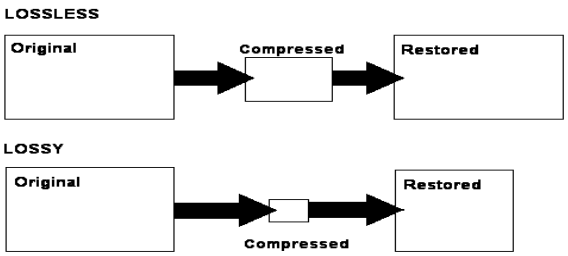

Fig.3 Compressions: Lossless and Lossy

B. 0D V/s 1D V/s 2D V/s 3D.

0D: 0D algorithm operates very efficiently when the data is "small" and is quite inefficient when the data is "large".

1D and 2D: Both 1D and 2D work efficiently when the data is large. They either compress the data fully or not at all. 1D data compression is specially designed to compress 1D stream of symbols. E.g., English text.

Since images are 2D, the contexts are more complicated than 1D string. 2D image compressions are *"wavelets"* and *"fractals".*

Fractal compression: It is a lossy image compression method using fractals. The algorithm compresses textures and natural images. Fractal algorithm is used to convert these parts into mathematical data called fractal codes, which are employed for encoding the same image.

Wavelet image compression: It is a form of compression, used for compressing images. It may also be used for audio and video compression, which may be either lossy or lossless. It is implemented using JPEG 2000 for still images. REDCODE, Dirac, torlien (for video compressions) are good for representing transients but not for harmonic compression.

3D: 3D compression aims at compressing 3D models etc. which find their applications in the areas like compressing graphics, virtual reality, video games, CAD/CAM, medical applications etc. The existing 3D compression algorithms use techniques which are already used by 1D and 2D. The compression tools available for 3D are – Sun's Java 3D compression std., IBM's MPEG/Topological surgery method, Virtue, Direct X. Methods for 3D compression may be grouped into 3 categories:

1. Mesh-based methods - those that traverse a polygon mesh representing an object's surface. E.g. Edge breaker, Java 3D compression, Topological surgery etc.

2. Progressive and hierarchical methods - those that transmit a base mesh and a series of refinements. E.g. Compressed progressive meshes, Sub-division based approaches, compressed normal meshes.

3. Image based approaches - those that encode not an object but a set of pictures. E.g. QuickTime VR and IPIX.

C. Variable-to-block V/s Block-to-variable.

Block-to-variable codes: These make use of variable number of output bits for each input symbol. E.g., Huffman coding, symbol ranking, Arithmetic coding etc. It aims at assigning shorter codes for the most occurring symbols and longer codes for rarely occurring symbols; thus providing a reduction in the average code length and thus compression.

Variable-to-block codes: These use fixed length output code to represent a variable length part of the input. These codes are also referred to as parse methods as there is no fixed format to divide the input message into coded form. e.g. Substitution compression.

D. Short (< ~ 200 char), medium (~20K char), large (> ~ 2 M char).

"Small" data compression algorithms, designed to be used on slow machines with very little RAM and ROM.

E. Zero latency decoding, medium latency coding, Full-latency.

Zero Latency decoding – It starts emitting decompressed data immediately after receiving the first code word, emitting one output character for every compressed code word. E.g. LZR W.

Medium Latency decoding – It starts emitting decompressed data after only a few more characters. E.g. Huffman.

Full latency decoding - It must have the entire file available before the first characters can be decoded. e.g. bzip.

V. COMPRESSION TYPES

The compression methods which are used on PCs are listed below:

Utility-Based File Compression: It is the most common method used widely on the PCs. In this compression, a utility program is employed to compress one or more files on our PC and also to decompress the same. E.g. PKZIP, WinZip.

Operating System File Compression: some operating systems perform the compression of files on an

individual basis within the operating system itself. E.g. Windows NT, Windows 2000.

Volume Compression: In this type of compression a lot of disk space is saved by compressing whole of the disk volumes. E.g Utility programs, third-party packages.

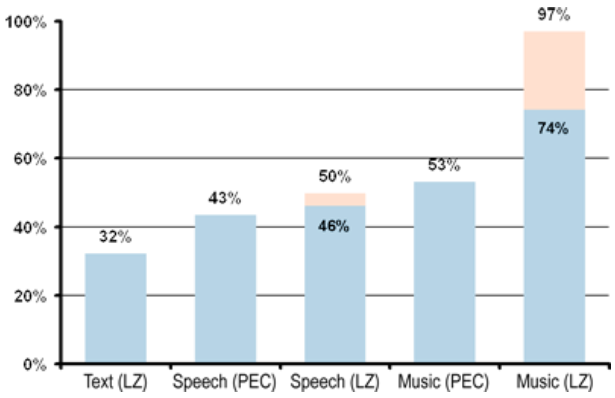

Fig.4 : Text and audio compression [11] (size as percentage of original)

VI. DATA COMPRESSION TECHNIQUES

Various data compression techniques [5,6,7,8,9] are available and currently new methods are under development too. By no means do we claim that the techniques mentioned below are the only ones available but have made an attempt to cover at least one in each category. To mention a few already available and the latest developments are:

A. Lossless compression methods

Run-Length Encoding- In this the data contains repeated strings, which are then replaced by special marker.

8644444444358882782222222} 86#40835#803278#206

Statistical compression- Short codes are used for frequent symbols and long codes are used for infrequent symbols. The 3 common principles are: Morse code, Huffman encoding, Lempel-Ziv-Welsh encoding

Relative Compression- This type of compression is extremely useful for sending video, commercial TVs and 30 frames per second.

i) Run-Length Encoding

It is sometimes called recurrence coding. Data files frequently contain the same character repeated many times in a row. The data contains repeated strings, which are then replaced by special marker. ``AAAAAABBBBCCC'' could be more efficiently represented as ``*A6*B4*C3'', thus saving 6 bytes. The code consists of a flag character, a count byte, and the repeated character.

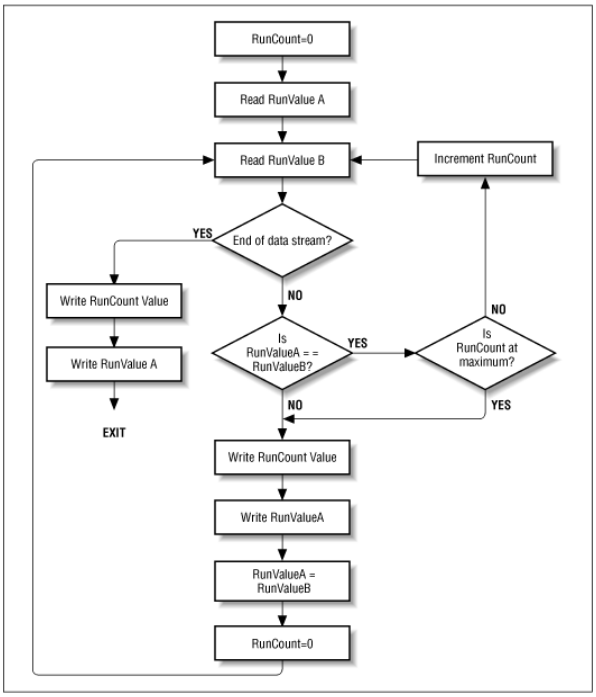

Fig.5 : Basic Run- length Encoding Flow

Is run length encoding practical for images?

Chances of three or more identical consecutive pixels are low for most real images. Especially images with large color depth. Some images do have lots of consecutive pixels. Especially images with low color depth. RLE is used for fax machines, by BMP, TIFF and PCX files. Advantages and Disadvantages

- Easily implemented.
- Does not require much CPU horsepower.
- Efficient with files that contain lots of repetitive data. These can be text files if they contain lots of spaces for indenting but images containing white or black areas are more suitable. Computer generated color images can also be highly compressed.

Areas of RLE compression usage

RLE compression can be used in the following file formats: TIFF files and PDF files.

ii) Huffman Encoding

This method is named after D.A. Huffman, who developed the procedure in the 1950s. More than 96% of this file consists of only 31 characters out of 127 available- like the lower case letters, the space, the comma, the period, and the carriage return. These 31 common characters are assigned binary codes like-

00000 =”a”, 00001=”b” etc. thus enabling 96% of the files to be reduced by 5/8.

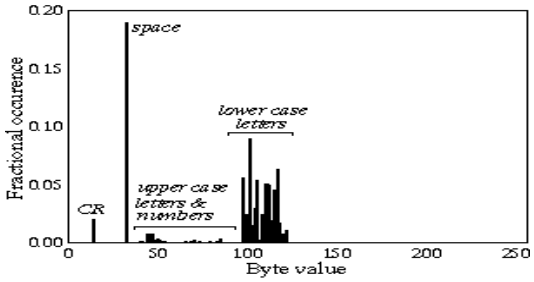

Fig. 6 : Huffman Encoding

To Compute Huffman code:

- Count frequency p_s for each symbol s in message.
- Start with one node corresponding to each symbol s (with weight p_s).
- Repeat until single tree formed:
  - select two trees with min weight p1 and p2
  - merge into single tree with weight p1 + p2

Advantages and Disadvantages:

- This compression algorithm is mainly efficient in compressing text or program files.
- Images like they are often used in prepress are better handled by other compression algorithms.

Areas of Huffman compression usage

- In pkZIP, lha, gz, zoo and arj.
- JPEG and MPEG compression.

iii) LZW Compression

LZW compression [10] is named after its developers, A. Lempel and J. Ziv, with later modifications by Terry A. Welch. It is the foremost technique for general purpose data compression due to its simplicity and versatility

LZW compression flowchart- The variable, CHAR, is a single byte. The variable, STRING, is a variable length sequence of bytes. Data are read from the input file (box 1 & 2) as single bytes, and written to the compressed file (box 4) as 12 bit codes.

Advantages

- simple, fast and good compression
- can do compression in one pass
- dynamic codeword table built for each file
- decompression recreates the codeword table so it does not need to be passed

Disadvantages

- not the optimum compression ratio
- actual compression hard to predict

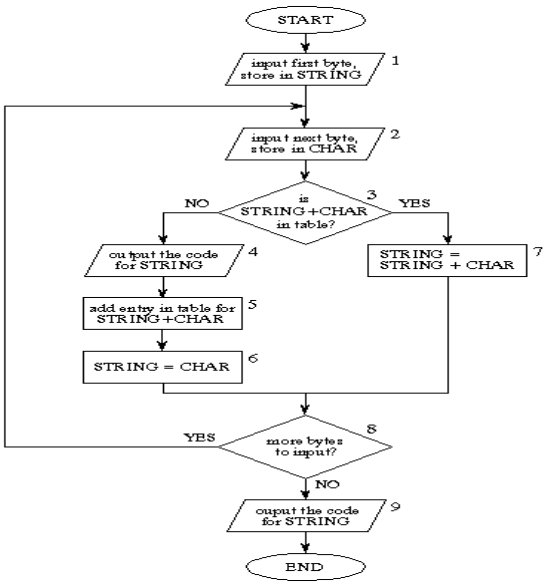

Fig.7 : LZW Compression flowchart

Areas of LZW compression usage

LZW compression can be used in the following file formats: TIFF files and GIF files.

Images transmitted over the internet pose to be the best example of *why data compression is required*. Downloading an uncompressed image (a TIFF file) will need approximately 600 Kbytes of data in comparison to 300 Kbytes for using a lossless technique for compressing the image (GIF format). JPEG is the best choice for digitized photographs, while GIF is used with drawn images, such as company logos that have large areas of a single color.

iv) Delta Compression

Delta encoding- It is a way in which the data is stored in the form of differences i.e. deltas between sequential data rather than data themselves. It is sometimes called *delta compression* because some instances of the encoding can make encoded data shorter than non-encoded data. In the delta compression problem, we have two files, f_{old} and f_{new}, Our aim is to compute a file f_δ of minimum size, such that we are able to reconstruct f_{new} from f_{old} and f_δ. The way adopted for solving the problem were based on finding the largest common subsequence of the two strings using dynamic programming and adding all remaining characters to f_{new} explicitly.

Applications

The applications of data compression are aimed at reducing networking and storage costs. The various areas of delta compression are:

Software Revision Control Systems: delta compression techniques are employed in the context of systems used for maintaining the revision history of software projects and other documents [13,14, 15].

Delta Compression at the File System Level: It provides efficient implementation of revision control systems, as well as some other applications.

Software Distribution: Techniques are employed for generating software patches that can be transmitted over a network in order to update installed software packages.

Exploring File Differences.

Improving HTTP performance: It is employed for improving the latency for web accesses, by exploiting the similarity between current and outdated versions of a web page, and between different pages on the same web site.

Efficient Web Page Storage.

B. Lossy compression methods

Lossy data compression techniques are used for pictures, videos and sounds. The techniques employed are: JPEG, MPEG, CS & Q.

i) JPEG (Transform Compression)

JPEG is named after its origin the *Joint Photographers Experts Group*. This involves reducing the number of bits per sample or entirely discards some of the samples. JPEG is used for still images and is the standard used on the web for photographic images (the GIF format is often used for textual images). It employs transform compression- a Lossy compression scheme for color and gray-scale images. The JPEG compression scheme is divided into the following stages: Initially transform the image into an optimal color space, then Down sample chrominance components by averaging groups of pixels together, Apply a Discrete Cosine Transform (DCT) to blocks of pixels, thus removing redundant image data. Then Quantize each block of DCT coefficients using weighting functions optimized for the human eye and finally Encode the resulting coefficients (image data) using a Huffman variable word-length algorithm to remove redundancies in the coefficients.

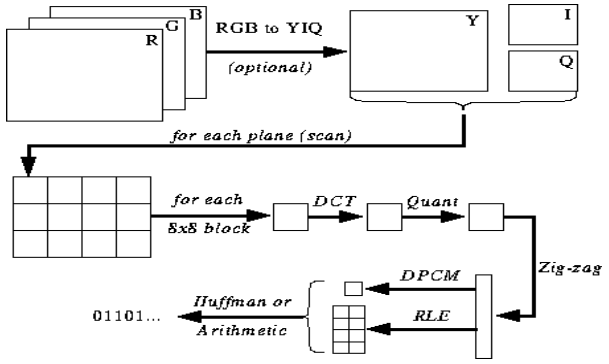

Fig. 8 : Steps in JPEG [12] Compression

Advantages

- Remarkable compression ratios.
- JPEG decompression is supported in PostScript level 2 and 3 RIPs. It means that smaller files can be sent across the network to the RIP which frees the sending station faster, minimizes overhead on the print server and speeds up the RIP.

Disadvantages

- JPEG is not used for images with sharp changes in tone.

Areas of JPEG usage

JPEG compression can be used in the following file formats: EPS-files, EPS DCS-files, JFIF-files and PDF-files.

ii) Multimedia Compression

Multimedia compression is a general term referring to the compression of any type of multimedia, most notably graphics, audio, and video.

MPEG (Moving Pictures Experts Group) The future of this technology is to encode the compression and decompression algorithms directly into integrated circuits. The job of MPEG is to take analog or digital video signals and convert them to packets of digital data that are more efficiently transported over a network.The approach used by MPEG can be divided into two types of compression: within-the-frame and between-frame. MPEG is used for video. The MPEG transform coding algorithm includes these steps: Discrete Cosine Transform (DCT), Quantization and Run-Length Encoding.

Advantages and Disadvantages

- Complex.
- Requires a great deal of processor time to encode the file.

- Requires a great deal of processor power to view the video.

Areas of MPEG usage:

Video Kiosk, Video on Demand, Video Dial Tone, Training, Corporate Presentations, Video Library, CATV(Cable Television), DBS (Direct Broadcast Satellite) and HDTV (High Definition Television).

iii) CS&Q (coarser sampling and/or quantization)

It employs a fixed-input and a fixed-output scheme. It means that, a fixed number of bits are read from the input file and a smaller fixed number of bits are written to the output file. Simply speaking, it reduces the number of bits/sample or entirely discards some of the samples thus resulting into poor picture (image) quality.

VII. CONCLUSION

Fathom 3.0 is developed by Intel in cooperation with Microsoft and scientific Atlanta. It works with media files for mobiles, portable, web and high definition.

Geolocation data compression is used for compressing data between sensors and *Robust digital compression* requires 1mb of additional software on web client. New techniques are coming up each day, to cater to the needs of various upcoming technologies. Various experiments are being conducted for examining the best technique in a particular area. As WSNs is an upcoming field, based on the above observations, it is possible that we might employ any one of the data (text) compression techniques for processing a query in Wireless sensor nodes.

Table 1: Summarizing the properties of various data compression algorithms

Algorithm/ Method	Type	Group Size I/P	Group Size O/P	Comp. Ratio	Comp. Rate	Best Suited	Area of Application	Used In Files
RLE	Lossless	Var.	Fixed	2:1	17.5%	Images with large areas of solid white/black	Text used in C 64 pgm.	TIFF,PCX, BMP,PDF
Huffman's Code	Lossless	Fixed	Var.	1.5: 1	94%(Image) 66% (Text) 65%(Speech)	All images	backend to some other compression method	GZIP, PkZIP, BZIP2,Iha, PNG, gz, zoo,Arj, JPEG & MPEG
DELTA	Lossless	-	-			Online backup services, displaying differences, merging changes, distributing updates, transmitting video sequences	LPC, mainly been used for Windows Updates, RFC 3299,	
LZW	Lossless	Var.	Fixed	5:1	88%(Image) 44%(Text) 64%(speech)	Text files, All images	Line-art images	TIFF, GIF
CS&Q	Lossy	Fixed	Fixed	3:1			reduces the number of bits/sample	
JPEG	Lossy	-	-	20:1 to 25:1(Good quality)	15-25% 66%(low quality)	Grayscale/ color images	Color images	EPS,JFIF,PDF, TIFF,SPIFF,ERS DCS-files
MPEG	Lossy	-	-	6:1 to 14:1		Sound track associated with video.	Direct Broadcast Satellite, Cable TV, Media Vaults, DVD Videos, HDV, MOD & TOD, XDCAM, DVB, ATSC, ISDB-T.	Memory Stick Video Format, MP4,AVI, AVCHD, DivX, WMV

VIII. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci, "Wireless sensor networks: A survey," Computer Networks, vol. 38, pp. 393–422, 2002.

[2] Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP, John Miano, August 1999

[3] Introduction to Data Compression, Khalid Sayood, Ed Fox (Editor), March 2000.

[4] Managing Gigabytes: Compressing and Indexing Documents and Images, Ian H H. Witten, Alistair Moffat, Timothy C. Bell, May 1999.

[5] www.data-compression/index.shtml

[6] www.data-compression/lossless.shtml

[7] http://searchciomidmarket.techtarget.com/ Definition

[8] http://localtechwire.com/business/local-tech-wire/

[9] http://www.futureofgadgets.com/ futureblogger/show/1730

[10] http://www.cs.mcgill.ca/~cs251/OldCourses/ 1997/topic23/#JavaApplet

[11] http://dvd-hq.info/data_compression_1.php

[12] http://www.cs.cf.ac.uk/Dave/Multimedia/ node234.html

[13] B. Berliner. CVS II: Parallelizing software development. In Proc. of theWinter 1990 USENIX Conference, pages 341–352, January 1990.

[14] M. Rochkind. The source code control system. IEEE Transactions on Software Engineering, 1:364–370, December 1975.

[15] W. Tichy. RCS: A system for version control. Software - Practice and Experience, 15, July 1985.

◈◈◈