© 2015 IJSRSET | Volume 1 | Issue 4 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Science

# A Novel Video Compression Prototype for Large Scale Videos using Content Mining Techniques

**K. N. Abdul Kader Nihal*[1], Dr. A. R. Mohamed Shanavas[2]**
Department of Computer Science, Jamal Mohamed College, Tamil Nadu, India

## ABSTRACT

The video compression became one of the mandatory elements in the modern digital technology so that the large-scale video should be compressed in order to serve high density digital videos that are transmitted over even on low bandwidth network carriers. The existing video compression techniques are not adequate in order to support high speed transmission needs of video transfer, particularly in medical and defense sector. In this paper, the authors analyzed the existing video compression techniques along with pros and cons. Then, the authors proposed a novel video compression prototype for compressing large scale videos using data mining techniques, which falls under loosely compression. However, the Proof-of-Concept (PoC) and result interpretation shows that the proposed video compression is at least five times better than the available video compression techniques for large scale video compression.

**Keywords:** Video Compression, Loosely Compression, Frame, JPEG, Content Mining, Data Mining

## I. INTRODUCTION

Over the past decades, video compression technologies have become an integral part of the way we create, communicate and consume visual information. Digital video communication can be found today in many applications such as broadcast services over satellite and terrestrial channels, digital video storage, wires and wireless conversational services and etc. The data quantity is very large for the digital video and the memory of the storage devices and the bandwidth of the transmission channel are not infinite, so reducing the amount of data needed to reproduce video saves storage space, increases access speed and is the only way to achieve motion video on digital computers. The video contains much spatial and temporal redundancy. In a single frame, nearby pixels are often correlated with each other. This is called spatial redundancy, or the intra-frame correlation. Another one is temporal redundancy, which means adjacent frames are highly correlated, or called the inter-frame correlation. Therefore, our goal is to efficiently reduce spatial and temporal redundancy to achieve video compression [1-3].

Video clips are made up of sequences of individual images, or "frames". Therefore, video compression algorithms share many concepts and techniques with still image compression algorithms, such as JPEG. In fact, one way to compress video is to ignore the similarities between consecutive video frames, and simply compress each frame independently of other frames. For example, some products employ this approach to compress video streams using the JPEG still-image compression standard [4-6]. This approach, known as "motion JPEG" or MJPEG is sometimes used in video production applications. Although modern video compression algorithms go beyond still-image compression schemes and take advantage of the correlation between consecutive video frames using motion estimation and motion compensation, these more advanced algorithms also employ techniques used in still-image compression algorithms. Therefore, we begin our exploration of video compression by discussing the inner workings of transform-based still-image compression algorithms such as JPEG [7, 8].

The paper is organised as, Section II describes the need of video compression and basics of video compression, the Section III distinguishes the difference between lossless and lossy video compression formats, the Section IV illustrates the proposed video compression and Section V concludes the paper.

## II. VIDEO COMPRESSION: THE NEED

Compression Technology is employed to efficiently use storage space, to save on transmission capacity and transmission time, respectively. Basically, it is all about saving resources and money. Despite of the overwhelming advances in the areas of storage media and transmission networks it is actually quite a surprise that still compression technology is required. One important reason is that also the resolution and amount of digital data has increased (e.g. HD-TV resolution, ever-increasing sensor sizes in consumer cameras), and that there are still application areas where resources are limited, e.g. wireless networks. Apart from the aim of simply reducing the amount of data, standards like MPEG-4, MPEG-7, and MPEG-21 offer additional functionalities [9].

During the last years three important trends have contributed to the fact that nowadays compression technology is as important as it has never been before – this development has already changed the way the user work with multimedia data like text, speech, audio, images, and video which will lead to new products and applications [10]:

a) The availability of highly effective methods for compressing various types of data.
b) The availability of fast and cheap hardware components to conduct compression on single-chip systems, microprocessors, DSPs and VLSI systems.
c) Convergence of computer, communication, consumer electronics, publishing, and entertainment industries.

The high bit rates that result from the various types of digital video make their transmission through their intended channels very difficult. Even entertainment video with modest frame rates and dimensions would require bandwidth and storage space far in excess of that available from CD-ROM. Thus delivering consumer quality video on compact disc would be impossible. This is analogous to an envelope being too large to fit into a letterbox. Similarly the data transfer rate required by a video telephony system is far greater than the bandwidth available over the plain old telephone system (POTS). Even if high bandwidth technology (e.g. fibre-optic cable) was in place, the per-byte-cost of transmission would have to be very low before it would be feasible to use it for the staggering amounts of data required by HDTV. Finally, even if the storage and transportation problems of digital video were overcome, the processing power needed to manage such volumes of data would make the receiver hardware very expensive [11-13].

Although significant gains in storage, transmission, and processor technology have been achieved in recent years, it is primarily the reduction of the amount of data that needs to be stored, transmitted, and processed that has made widespread use of digital video a possibility. This reduction of bandwidth has been made possible by advances in compression technology. Advances in compression technology more than anything else have led to the arrival of video to the desktop and hundreds of channels to the home. Compression reduces the bandwidth required to transmit and store digital video [14].

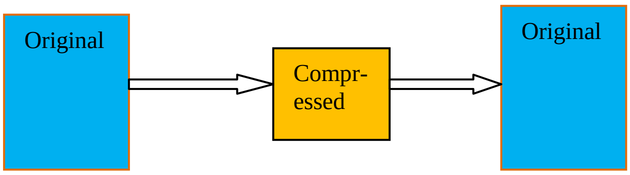
## III. LOSSLESS vs LOSSY VIDEO COMPRESSION FORMATS

Lossless compression techniques, as their name implies, involve no loss of information. If data have been lossless compressed, the original data can be recovered exactly from the compressed data after a compress/expand cycle. Lossless compression is generally used for so-called "discrete" data, such as database records, spread sheets, word-processing files, and even some kinds of image and video information. Lossy compression works very differently. These programs simply eliminate "unnecessary" bits of information, tailoring the file so that it is smaller. This type of compression is used a lot for reducing the file size of bitmap pictures, which tend to be fairly bulky [15].

Few lossless video formats are in common consumer use, as they would result in video files taking up a huge amount of space. Common formats like H.264, MKV, and WMV are all lossy. H.264 can provide smaller files with higher qualities than previous generations of video codecs because it has a “smarter” algorithm that’s better at choosing the data to throw out. Some of these lossless formats also provide compression. For example, a WAV file is a completely uncompressed audio file, and takes up quite a bit of space. FLAC and ALAC are both lossless types of audio files that contain the same data as a WAV file, but they use a form of compression to create smaller files. Formats like FLAC and ALAC don't throw any data away – they keep all the data and compress it intelligently, like ZIP files do. However, they are still significantly larger in size than MP3 files, which throw much data away.

Lossy compression is best used to reduce the size of video data, where defects in the picture/video can be hidden as long as the general structure of the picture/video remains intact. This type of compression is called lossy compression because part of the reason that it compresses so well is that actual data from the video image, is lost and then replaced with some approximation. The typical model for lossless and lossy compression is shown in Figure-1.

LOSSLESS VIDEO COMPRESSION:




LOSSY VIDEO COMPRESSION:

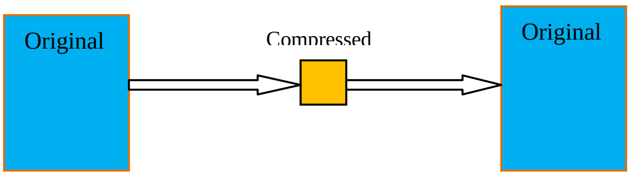



**Figure 1:** Lossless vs Lossy Video Compression

The image compression techniques used in JPEG and in most video compression algorithms are "lossy." That is, the original uncompressed image can't be perfecty reconstructed from the compressed data, so some information from the original image is lost. Lossy compression algorithms attempt to ensure that the differences between the original uncompressed image and the reconstructed image are not perceptible to the human eye [16, 17].

The first step in JPEG and similar image compression algorithms is to divide the image into small blocks and transform each block into a frequency-domain representation. Typically, this step uses a discrete cosine transform (DCT) on blocks that are eight pixels wide by eight pixels high. Thus, the DCT operates on 64 input pixels and yields 64 frequency-domain coefficients. The DCT itself preserves all of the information in the eight-by-eight image block. That is, an inverse DCT (IDCT) can be used to perfectly reconstruct the original 64 pixels from the DCT coefficients. However, the human eye is more sensitive to the information contained in DCT coefficients that represent low frequencies (corresponding to large features in the image) than to the information contained in DCT coefficients that represent high frequencies (corresponding to small features). Therefore, the DCT helps separate the more perceptually significant information from less perceptually significant information. Later steps in the compression algorithm encode the low-frequency DCT coefficients with high precision, but use fewer or no bits to encode the high-frequency coefficients, thus discarding information that is less perceptually significant. In the decoding algorithm, an IDCT transforms the imperfectly coded coefficients back into an 8x8 block of pixels.

The computations performed in the IDCT are nearly identical to those performed in the DCT, so these two functions have very similar processing requirements. A single two-dimensional eight-by-eight DCT or IDCT requires a few hundred instruction cycles on a typical DSP. However, video compression algorithms must often perform a vast number of DCTs and/or IDCTs per second. For example, an MPEG-4 video decoder operating at CIF (352x288) resolution and a frame rate of 30 fps may need to perform as many as 71,280 IDCTs per second, depending on the video content. The IDCT function would require over 40 MHz on a Texas Instruments TMS320C55x DSP processor (without the DCT accelerator) under these conditions. IDCT computation can take up as much as 30% of the cycles spent in a video decoder implementation.

Because the DCT and IDCT operate on small image blocks, the memory requirements of these functions are rather small and are typically negligible compared to the size of frame buffers and other data in image and video compression applications. The high computational demand and small memory footprint of the DCT and IDCT functions make them ideal candidates for implementation using dedicated hardware coprocessors [18-21].

Some of the exiting algorithms for lossless video compression:

- CCITT group 3 & 4 compression
- Flate/deflate compression
- Huffman compression
- LZW compression
- RLE compression

Some of the exiting algorithms for lossy video compression:

- Motion JPEG
- H.264/MPEG-4 AVC
- Ogg Theora
- Dirac
- Sorenson video codec
- VC-1
- H.265/HEVC

## IV. PROPOSED PROTOTYPE FOR VIDEO COMPRESSION

The proposed prototype is an enhanced version of existing proven video compression techniques so that it is best suited for compressing large-scale videos. The high level flow of the proposed prototype for video compression is as shown in Figure-2. The large scale video is taken for compression as input. The video can be stored in terms of files or any other source such as video stream. After receiving the video, it will be split into multiple equal length segments. This is to reduce the complexity while applying the video compression as small segments of video requires only minimum amount of main memory and computing power. Otherwise, high speed processor and huge amount of main memory is needed for loading the video. The video segments should be loaded into main memory before starting the process of compression. However, very small sized segment will not be advised as the compression ratio will be minimized in this case. In the proposed video compression PoC, this value is set to be configured and default recommended size also shown based on the quality and size of the entire video.

The proposed prototype will be applied on each segment of the video to do compression. Here, we can use any promised frame compression technique such as MPEG. Even the image compression techniques can be tuned to compress the frames if high compression ratio is targeted. For each frame, the previous and the next frame is compared, naturally they will be almost same with 10 to 20 precent changes. The delta difference will be recorded and any one of the frame will be deleted from the video. In the reverse process of compression, the delta difference will be added to the current frame to construct the next frame. This prototype will heavily reduce the size of the video and we are applying the standard and provided image compression technique for compression each frame. So the proposed prototype will take advantage of existing video compression techniques and frames will be removed based on delta difference. This kind of image compression will heavily useful for non-frequent frame videos such as medical image processing, for example reading human heart beats. Finally the compressed video will be stored on the physical device or the specified output stream for storage. This approach is more dependent on lossy compression, which is the only disadvantage.

## V. CONCLUSION

Video compression basically means reducing frames data in videos. The existing video compression algorithms target to reduce colour nuances within the image/frame, reduce the colour resolution with respect to the prevailing light intensity and remove small/invisible parts of the picture. Of course, they can achieve good compression ratio. However, these compression techniques are totally inadequate for large-scale videos, particularly used for medical images, military operations and CCTV recordings. In this paper, the author analysed the exiting video compression downsides and proposed a novel video compression prototype for compressing large-scale videos, which uses content mining -one of the data mining techniques. The Proof-of-Concept (PoC) application is developed using Java on Linux platform and tested with medical video processing. The results are interpreted and identified the proposed techniques compress the video at least five times more than the exiting standard video compression techniques.

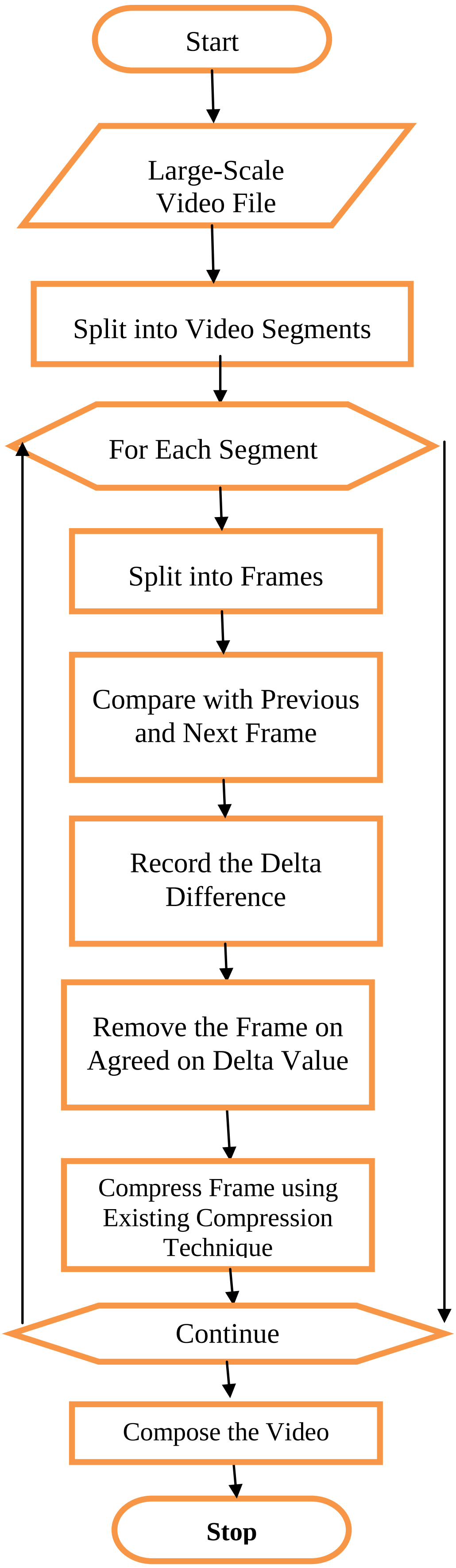



**Figure 2:** Flow chart of the proposed prototype

## VI. REFERENCES


[1] Dr. Andreas Uhl. 2013. Compression Technologies and Multimedia Data Formats, Lecture Notes, University of Salzburg, Osterreich

[2] Ohm, J-R., and Gary J. Sullivan. "High efficiency video coding: the next frontier in video compression [Standards in a Nutshell]." Signal Processing Magazine, IEEE 30.1 (2013): 152-158.

[3] Koo, Heon-mo, Atthar H. Mohammed, and Thuan H. Pham. "Techniques for rate-distortion optimization in video compression." U.S. Patent No. 20,150,172,650. 18 Jun. 2015.

[4] Li, Zhicheng, Shiyin Qin, and Laurent Itti. "Visual attention guided bit allocation in video compression." Image and Vision Computing 29.1 (2011): 1-14.

[5] Banitalebi-Dehkordi, Amin, et al. "Compression of high dynamic range video using the HEVC and H. 264/AVC standards." Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), 2014 10th International Conference on. IEEE, 2014.

[6] Taubman, David, and Andrew Secker. "Method of signalling motion information for efficient scalable video compression." U.S. Patent Application 13/421,788.

[7] Lee, Jong-Seok, and Touradj Ebrahimi. "Perceptual video compression: A survey." Selected Topics in Signal Processing, IEEE Journal of 6.6 (2012): 684-697.

[8] McCarthy, Sean T., et al. "Method of bit allocation for image & video compression using perceptual guidance." U.S. Patent Application 13/841,865.

[9] Daribo, Ismael, Gene Cheung, and Dinei Florencio. "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression." Image Processing (ICIP), 2012 19th IEEE International Conference on. IEEE, 2012.

[10] Colin Manning. http://newmediarepublic.com/dvideo/ Referred: 2015.

[11] Milani, Simone, et al. "Multiple compression detection for video sequences." Multimedia Signal Processing (MMSP), 2012 IEEE 14th International Workshop on. IEEE, 2012.

[12] Hadizadeh, Hadi, and Ivan V. Bajic. "Saliency-aware video compression." Image Processing, IEEE Transactions on 23.1 (2014): 19-33.

[13] Gao, Yu, et al. Encoder-Driven Inpainting Strategy in Multiview Video Compression. No. EPFL-ARTICLE-200490. Institute of Electrical and Electronics Engineers, 2014.

[14] http://www.newmediarepublic.com/dvideo/compression/adv03.html Referred: 2015

[15] Thyagarajan, Kadayam S. Still Image and video compression with MATLAB. John Wiley & Sons, 2011.

[16] Esche, Marko, et al. "Adaptive temporal trajectory filtering for video compression." Circuits and Systems for Video Technology, IEEE Transactions on 22.5 (2012): 659-670.

[17] http://www.eetimes.com/document.asp?doc_id=1273618 Referred: 2015

[18] Van Wallendael, Glenn, et al. "3D video compression based on high efficiency video coding." Consumer Electronics, IEEE Transactions on 58.1 (2012): 137-145.

[19] Richardson, Iain E. The H. 264 advanced video compression standard. John Wiley & Sons, 2011.

[20] Hanhart, Philippe, et al. "Subjective quality evaluation of the upcoming HEVC video compression standard." SPIE Optical Engineering+ Applications. International Society for Optics and Photonics, 2012.

[21] Revaud, Jérôme, et al. "Event retrieval in large video collections with circulant temporal encoding." Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.