Implementation of New Proposed Video Compression Technique for Surveillance System and Clinical Medicine Videos

K.N. Abdul Kader Nihal, Dr. A.R. Mohamed Shanavas

Abstract— **Data Compression shrinks down a file so that it takes up less space. This is desirable for data storage and data communication. Storage space on disks is expensive so a file which occupies less disk space is "cheaper" than an uncompressed file. Smaller files are also desirable for data communication, because the smaller a file the faster it can be transferred. A compressed file appears to increase the speed of data transfer over an uncompressed file. Video compression is the last step before uploading your file online. As the name implies, compressing video reduces its file size. This is very important, because smaller files upload faster, save bandwidth and storage costs, and load quicker when played back. However, if video is compressed too much, the file can lose its detail, resolution, clarity, and much more. In this paper, the authors proposed new video compression techniques for surveillance videos and clinical medicine videos using 'frame cut-off' technique which is a novel approach in compression video where the movement is minimum. Also the authors \ implemented the proposed algorithm using Java technologies and MATLAB image processing tool and tested with CCTV and medical video processing such as EGC, Intensive Care Unit observations. The interpreted results show that the proposed algorithm compresses the videos five times better than the existing video compression techniques. The pros of the proposed algorithm are that it is well suited for the videos which have minimum movement, for example CCTV footage. The cons of this proposed solution is that the better results cannot be achieved for normal videos which have high movement such as cine movies. However, the content mining techniques are applied efficiently in order to increase the compression ratio and it can be further investigated to increase the high movement videos to compress better.**

Index Terms— **Video compression, Data Mining, /content mining, MPEG, Frame compression**

I. INTRODUCTION

Compression algorithms have been developed for various applications at an international level. These algorithms have been tested and standardized by several organizations. The methods are alternatives to a certain extent, because they were developed with different aims and are tailored to specific applications. Overall, the range of available standardized video compression methods covers practically all applications. Some compression algorithms permit a degree of variation within the standard to meet technology constraints. The most common video compression methods use the statistical similarity of adjacent pixels. The image is converted to a frequency space block by block using the DCT (Discrete Cosine Transformation). In video coding, it is also possible to use the strong time correlation of consecutive frames – the objects shown are usually static or move in a more or less constant manner. If the difference between the images is calculated with motion estimation, it is only necessary to transfer the changes from frame to frame [1].

Transforming a video sequence into a bit-stream is called coding or encoding. This is usually performed by an encoder inside or close to the camera. Transforming a bit-stream back to an image is called decoding. A decoder is needed each time the bit-stream has to be displayed. A device able to perform both functions is called a codec (coder/decoder) [2]. Compression is a reversible conversion (encoding) of data that contains fewer bits. This allows a more efficient storage and transmission of the data. The inverse process is called decompression (decoding). Software and hardware that can encode and decode are called decoders.

The performance of a video codec is always a trade-off among the three variables: (i) Quality, (ii) Bit Rate, and (iii) Computational Cost.

Quality: This notion is inseparable from the fact that all video compression schemes are loss. This means that the cycle consisting in coding a video sequence into a compressed bit-stream, then decoding the bit-stream, yields a recovered uncompressed video sequence which is not strictly the same as the original one. There are information losses. Incidentally, there are lossless compression schemes, but their performance along the bit rate axis is poor, and they cannot be used for the vast majority of applications. An exception might be medical imaging for diagnosis purposes.

Bit Rate: Subjecting a video sequence to a particular encoder yields a compressed bit-stream which is progressively generated. The average production rate of bits is the bit rate. Bit rates for compressed video may range somewhere between 50 kbits/s and 250 Mbits/s. The bit rate is strongly related to the compression performance of the coder, but it also depends on the size of the original images (SDTV, HDTV) and their frame rate. You can use it to compare encoders compressing images of identical size and rate [9].

Manuscript received February 08, 2016.

K.N. Abdul Kader Nihal, Research Scholar, Dept. of Computer Science, Jamal Mohamed College (Autonomous), Tiruchirappalli – 620 020, India

Dr. A.R. Mohamed Shanavas, Associate Professor, Dept. of Computer Science, Jamal Mohamed College (Autonomous), Tiruchirappalli – 620 020, India

Computational Cost: Encoding and decoding images involve a large amount of real-time computation. Generally speaking, increased compression performance involves higher computational cost. What kind of computing devices can we use to perform the computation? In the early days, general-purpose computers were inadequate, even for the relatively simple compression standards available then. With the ever increasing processing power of computing hardware (the famous Moore's law), it became possible to run sophisticated compression or decompression algorithms on low-cost PCs. However, it does not come as a free lunch. If you entrust your computer with such a high-priority task as real-time video coding or decoding, you will experience a performance reduction for all other software applications [2].

The paper is organized as, Section II describes the literature review of video compression technologies and Section III distinguishes Different Compression Algorithms, the Section IV illustrates the proposed video compression algorithm and Section V concludes the paper.

II. LITERATURE REVIEW

The main function of video compression technologies is to reduce and remove redundant data (i.e. no motion and no change in scenery) in video so it can be effectively stored or sent over a network. Modern efficient compression techniques can achieve a significant reduction in file size without sacrificing video quality.

Video compression technologies are commonly known as codecs. They use an algorithm called an encoder that compresses data at the source (i.e. before storage or transmission). At the destination, they use another algorithm called a decoder to decode and decompress the data. Codecs are used in pairs and are not cross compatible. A video encoded using MPEG4 technology cannot be decoded using a H264 decoder [10-12].

A. MPEG: Motion JPEG or M-JPEG is a video compression format based on the popular JPEG compression used for still images. It forms a video sequence and gives the impression of motion by combining several still JPEG images. MJPEG is a robust video compression format. Since each frame of video is independent, even if one is dropped, the rest of the video is unaffected. This robustness however also translates to higher bandwidth requirements and higher storage space used.

B. MPEG4: MPEG4 compression technology in video surveillance refers to MPEG4 Visual. Unlike M-JPEG, MPEG4 is a licensed standard. DVRs, IP cameras and other surveillance products must pay a license fee for each channel of video that uses MPEG4 technology. In terms of compression, MPEG4 supports low bandwidth applications that require decent quality images. It uses video compression techniques to locate redundant data in video and reduce its size.

C. H264: H264 also known as MPEG4 Part 10/AVC (Advanced video coding) is the latest MPEG standard for video encoding. As of 2010, it is the preferred video encoding standard since it can compress video significantly without compromising video quality. It provides up to 80% reduction in file size compared to M-JPEG and up to 50% reduction compared to MPEG4. H264 is jointly defined by two standardization groups, the ITU-T's Video Coding Experts Group from the telecommunications sector and the ISO/IEC Moving Picture Experts Group from the IT sector. Because of this backing, it is expected that H264 will surpass the other standards in terms of adoption [4].

Most CCTV compression algorithms used in CCTV are lossy compressions, because such algorithms offer higher compression ratio (the ratio of the resulting video file size compared with the original file size).

III. ASSESSMENT OF VIDEO COMPRESSION ALGORITHMS

Video compression algorithms are divided in two groups: (i) Frame based compression (JPEG, Wavelet, JPEG 2000), and (ii) Stream based compression (MPEG-2, MPEG-4, H.264, MPEG-7). Usage of stream based compression algorithms enables greater savings on storage space and network bandwidth but as a trade-off these algorithms require higher computing performance [13].

Four of the most common type of compressions widely used in CCTV are: (i) Motion JPEG, (ii) MPEG-4, (iii) H.264, and (iv) JPEG2000. IP Video System Design Tool can help to see how choosing of compression methods can affect required network bandwidth and CCTV storage space, so one can clearly see the difference in terms of compression ratio. Motion JPEG is very popular compression format. MJPEG fits very well for video archives because of its frame based nature. MPEG4 can be 3 times more efficient in terms of compression ratio in compare with Motion JPEG. But MPEG4 is a bad choice for systems with frame rate less than 5-6 frames per second. H.264 can be 50-100% more efficient in compare with MPEG-4 MPEG-4 and H.264 are ideal for CCTV systems with limited but stable bandwidth [14, 16].

JPEG2000 is similar to JPEG, but uses wavelet transform instead of discreetest-cosine-transform (DCT) of JPEG. JPEG2000 offers a better image quality on higher compression levels [15]. Another great advantage is a possibility to decompress lower resolution representation of the image. This feature is good for motion detection algorithms. However JPEG2000 compression needs way higher CPU performance, than JPEG [5].

The JPEG standard was developed by the Joint Photographic Expert Group (part of ISO) for efficient storage of individual frames. Motion JPEG or M-JPEG is a series of separate JPEG images that form a video sequence. When 16 JPEG image frames or more are joined together per second, the result is an illusion of motion video. Video reproduction at 30 frames per second (FPS) for NTSC signals or 25 FPS for PAL signals is called full motion video or continuous-motion video. Although Motion JPEG is an unlicensed standard it is widely compatible with many applications that require low frame rates or technologies such as Video Analytics where frame by frame analysis is crucial [16-18]. Advantages include,

- Ability to support multi-mega pixel resolution
- Ideal for courtroom single frame evidence
- Clearer images at lower frame rates than MPEG-4
- Frame by frame playback offers more frames to view
- Technology is simpler; this can reduce the cost of a camera or video codec
- At low bandwidth priority is given to Image Resolution [19]

Disadvantages include,

- No M-JPEG standard often means incompatibility issues
- High bit rate for scenes with little or no activity increases bandwidth and storage
- Video quality deteriorates at higher compression ratios
- Converting M-JPEG into another format reduces video quality
- Dated technology superseded by more bandwidth-efficient encoding techniques [6, 20].

IV. PROPOSED VIDEO COMPRESSION ALGORITHM

In the proposed video compression, every second of video is composed of a series of still frames (typically 24-60 frames per second). Depending on the subject, only part of the image changes from frame to frame. Instead of storing two nearly-identical frames within a video file, only the parts of the image that have changed are recorded. So if one has a friend waving to the camera, and your friend's arm is the only thing moving in the shot, the image information of your friend's arm is the only thing that is recorded. The method a computer uses to determine the amount of change between frames is called the codec.

Every couple of frames, the codec will pick a key-frame that will serve as a reference for all the frames after it. Each key-frame consists of the entire image, and the following frames base their changes off of it. It's important to select a key-frame interval that isn't too frequent, or too infrequent. If you set the interval too frequent, you'll end up with a large video file, set it too infrequent and you'll have strange ghosting artifacts. Matching your key-frame interval to your frame rate (essentially making a new key-frame once every second) is usually a good place to start, but depending on what you're shooting this may vary.

Bit rate (also known as data rate) is the amount of data that is used for each second of video. In the world of video, this is generally measured in kilobits per second (kbps). Bit rates come in two flavors: constant and variable. A constant bit rate (CBR) uses the same amount of data every second, while a variable bit rate (VBR) adjusts the amount of data used depending on the complexity of the changes between frames. Generally, a variable bit rate will produce a smaller file, using less data for less complex frames. However, if there is a lot of action going on in your video, a variable bit rate will use more data to account for more movement. The proposed video compression algorithm is given in Fig. 1, and the frame comparison sub-routine is given in Fig. 2.

Algorithm: Frame_Cut_Off_Video_Compression

```
Declare
Input:
    iVideo: Input Video Filename
Process:
        Segments[]: Array[0..Segment]
        Segment: Sub-division of segmented video
        Delta values: Retrieved from XML Code
        Frames: Subdivision of segment
        Frame: Subdivision of frames
        Index[ ]: Array[0,.n]
        SplitSize: division od video
        DefinedLength: Minimum Video Size to Encode
        VideoLength: Actual Video Size before encode
        PrevFrame: Defaulted to Null
        CurrFrame: Defaulted to Null
Output:
        Output: Compressed Video Filename
Begin:
  Step I:  Start
  Step II: Input iVideo
  Step III: Scan for VideoLength
          If Videolength< <DefinedLength> then
          Return "Reject Compression due to lower value
          Else
          Goto Step IV
  Step IV: Split the video into Segments
          Segments[iVideo]← split[iVideo,SplitSize]
  Step V: Each Segment in Segments[iVideo]
          Do
          Split Segments[Index] into Frames[Segments];
          where Index starts from 0;
          Perform Step VI;
          done;
  Step Vi: PrevFrame=nuil;
          CurrFrame=null;
          Each frame in Frames[Segments];
          do
          If PrevFrame==null;
          PrevFrame=Frames[index];
          Continue;
          elseif PreFrame!=null and CurrFrame!= null then
          Call Compare(PreFrame,CurrFrame);
          If CurrFrame<Configured<DeltaValue>
          Continue;
          Else
          Store CurrFrame;
          PrevFrame=CurrFrame;
          Else CurrFrame==Null;
          Close the iVideo;
          Release All Resource Handled
          Release Memory Allocation
          End if
            While (CurrFrame != Null)
Step VII :Stop
```

Fig. 1: Proposed Video Compression Algorithm

```
Algorithm: Frame_Comparison

Declare
Input:
        PrevFrame:
       CurrFrame:
Process:
      PrevFram1: Assigned to PrevFrame
      CurrFrame1: Assigned to CurrFrame

Begin:
   Step I:   Start
   Step II: Input  PrevFrame, CurrFrame:

   Step III:  Read PrevFrame, CurFrame
   Step IV: PrevFrame1=PrevFrame
             CurrFram1=CurrFram
   Step V: do
               if(PrevFrame1.Read()!=-1) &&
               ((CurrFrame1.read() )<!=-1)
               PrevFrame=CurrFrame
               Return 0
               Else
               Return 1
                While(PrevFrame1=CurrFrame2)
   Step VI: end
```

Fig. 2: Frame Comparison Subroutine

V. CONCLUSION

For security and surveillance, image compression is primarily used for storage and real-time transmission. As technology progresses, the market is demanding higher frame-rates as well as higher resolution video, primarily intended for transmission over LANs and relatively high-speed WANs. JPEG and M-JPEG is suitable for the lower frame-rates, and wavelet is fine for high-bandwidth yet high-compression situations where the video needs to appear smooth but in fact much of the detail has been lost. MPEG-2 offers the best video money can buy but the MPEG-4 compression algorithm still offers the best video quality for the most common networks available today. After the interpretation of the implementation and analysis of the result of the proposed solution for compression surveillance and clinical medicine videos, it is found that the proposed algorithm result in better compression ration compared to the existing video compression techniques as discussed in Literature Review Section. The compression ratio achieved is five times better than the current available compression techniques especially for CCTV and medical video compression.

REFERENCES

[1] Bosch, “White Paper: Video Compression for CCTV,” Referenced: July 2015.

[2] Yves Joskin, “White Paper: The right Video Compression for Digital CCTV,” Euresys, December 2008

[3] https://vimeo.com/blog/post/video-compression-basics: Referred:May 2015

[4] http://philsecurity.com/guides/dvr/compression/

[5] https://www.jvsg.com/cctv-compression-guide/

[6] http://www.controlware.co.uk/en/start/knowledge-base/security-articles/cctv-video-compression/motion-jpeg.html

[7] Le Gall, Didier. "MPEG: A video compression standard for multimedia applications." Communications of the ACM 34.4 (1991): 46-58.

[8] Sayood, Khalid. Introduction to data compression. Newnes, 2012.

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

[10] Sullivan, Gary J., and Thomas Wiegand. "Rate-distortion optimization for video compression." Signal Processing Magazine, IEEE 15.6 (1998): 74-90.

[11] Marpe, Detlev, Heiko Schwarz, and Thomas Wiegand. "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard." Circuits and Systems for Video Technology, IEEE Transactions on 13.7 (2003): 620-636.

[12] Sullivan, Gary J., and Thomas Wiegand. "Video compression-from concepts to the H. 264/AVC standard." Proceedings of the IEEE 93.1 (2005): 18-31.

[13] Sullivan, Gary J., and Thomas Wiegand. "Video compression-from concepts to the H. 264/AVC standard." Proceedings of the IEEE 93.1 (2005): 18-31.

[14] Ghanbari, Mohammed, and Institution of Electrical Engineers. "Video coding: an introduction to standard codecs." Stevenage: Institution of Electrical Engineers, 1999.

[15] Pourazad, Mahsa T., et al. "HEVC: The new gold standard for video compression: How does HEVC compare with H. 264/AVC?." Consumer Electronics Magazine, IEEE 1.3 (2012): 36-46.

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

[17] Nguyen, Viet-Anh, Dongbo Min, and Minh N. Do. "Efficient techniques for depth video compression using weighted mode filtering." Circuits and Systems for Video Technology, IEEE Transactions on 23.2 (2013): 189-202.

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

[19] Chen, Doris, and Deshanand Singh. "Fractal video compression in OpenCL: An evaluation of CPUs, GPUs, and FPGAs as acceleration platforms."Design Automation Conference (ASP-DAC), 2013 18th Asia and South Pacific. IEEE, 2013.

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