A Solution for Transmitting and Displaying UHD 3D Raw Videos Using Lossless Compression

Ruan D. Gomes  
Federal Institute of Paraíba  
Guarabira, Paraíba, Brasil  
ruan.gomes@ifpb.edu.br

Yuri G. da Costa  
LAViD/UFPB  
João Pessoa, Paraíba, Brasil  
yuriggc@lavid.ufpb.br

Lucenildo L. A. Júnior  
LAViD/UFPB  
João Pessoa, Paraíba, Brasil  
lucenildo@lavid.ufpb.br

Manoel G. da Silva Neto  
LAViD/UFPB  
João Pessoa, Paraíba, Brasil  
manoelgs@lavid.ufpb.br

Alexandre N. Duarte  
LAViD/UFPB  
João Pessoa, Paraíba, Brasil  
alexandre@ci.ufpb.br

Guido L. de Souza Filho  
LAViD/UFPB  
João Pessoa, Paraíba, Brasil  
guido@lavid.ufpb.br

ABSTRACT
This paper describes a software-based approach for transmitting and displaying ultra high definition (UHD) videos (ex: 4K) in raw format (2D or 3D). The viability of using lossless compression algorithms for transmission and exhibition of UHD raw videos was investigated. This approach allows the transmission of raw videos for supporting the development of distributed edition tools for UHD videos and high quality visualization using a lower bandwidth. We conducted performance studies of eight lossless compression algorithms, aiming at analyze the compression rate for 4K videos, and the decoding latency. Finally, a multithread version of the LZ4HC was integrated with the Fogo Player, which is a software-based 4K player developed in a previous work. We obtained a reduction of 38% in bandwidth requirements, and a decoding latency that allowed the exhibition in real time of 4K videos.

Keywords
UHD videos, 4K, 4K-3D, Lossless video compression

1. INTRODUCTION
With the advent of the 4K technology the development of new applications for advanced visualization became possible, offering an image quality comparable to that offered by 35mm film and presenting the inherent advantages of the digitization, as the easier storage and higher immunity to deterioration. These new applications may have a direct impact in different scenarios such as the Digital Cinema and the transmission and display of artistic performances and sports events in Ultra High Definition (UHD). Besides applications focused on media consumers, new content production applications may emerge, such as tools and technologies to support collaborative editing and visualization of UHD content.

Nevertheless, the development of solutions to deal with UHD content presents many challenging aspects, mainly due to the large amount of data that must be processed. These new applications must deal with limitations related to storage, transmission and computational complexity of the algorithms for encoding and decoding the UHD content.

For example, a 4K video usually have a resolution of 4096 x 2160 pixels. Considering raw videos with 24 bits per pixel and 24 frames per second the resulting stream will have a bitrate of approximately 5 Gbit/s. For 3D videos it is necessary to transfer two streams, one for each eye. Thus, it would require a link of about 10 Gbit/s for real time transmission. All this without considering the overhead introduced by the transmission protocols. Using more quantization levels and higher frame rates would make this situation even worse.

This bandwidth problem could be mitigated with the use of lossy codecs, but there are some applications that require the transmission of the raw data, such as distributed collaborative editing tools for 4K contents. Besides, some applications may require extreme image quality, such as telemedicine applications. For these applications, lossless compression could be used to help diminish the presented bandwidth requirements, while all the video information is preserved.

In this work we investigate the performance of a set of lossless compression algorithms, aiming at analyzing the compression ratio obtained for 4K videos and the speed of the encoding and decoding processes. Finally, we performed the integration of a multi-threaded version of the LZ4HC algorithm with the Fogo Player, a software-based 4K player described in our previous work [1]. We obtained a compression ratio of about 38% and decoding times that are compatible with real-time displaying of 4K videos by software.

2. RELATED WORK
There are several initiatives aiming to develop solutions for transmitting and displaying UHD videos [2, 3, 4, 5, 6]. Most of these works are based on the JPEG-2000 standard, which is the standard recommended by the Digital Cinema Initiative (DCI). However, due to the computational complexity of the algorithms defined by the JPEG-2000 standard, the solutions based on this standard relied on dedicated hardware to perform the encoding and decoding processes. These approaches, when compared to software-based approaches, have as common drawbacks the relatively higher cost and limited flexibility.

In a previous paper [1] we developed a software-based solution, called Fogo Player, for distributing, transmitting and
displaying UHD videos. In this solution, all processing is performed using an architecture of parallel and distributed software components. This software-based solution is flexible in terms of resolution and coding algorithms. In the experiments presented in [1] we used the H.264 standard, however it is possible to integrate different coding standards to the solution.

In 2009, researchers at NTT (Network Innovation Laboratories) conducted the first experimental transmission of 4K raw videos using a 10 Gbit/s link. This experiment, conducted by Shiraia et al. [5], required a bitrate of about 6 Gbit/s since it used no method for data compression. The described solution is simple and presents a very small delay since it is not necessary to decode the video before displaying it. However, the solution requires a very high bandwidth and make it impossible to transmit a 4K-3D raw video using a 10 Gbit/s link.

On the other hand, using the approach presented in this paper, it would be possible to stream 4K-3D raw videos using about 7.5 Gbit/s, considering the flow generated by the video used by NTT and an average compression ratio of about 38%. Furthermore, with the reduction achieved by using lossless compression, it is possible to use four 1 Gbit/s interfaces instead of one 10 Gbit/s interface to transmit 4K raw videos. This represents a huge cost reduction in the network hardware. Fogo Player was designed to allow the use of multiple machines to display different parts of a 4K video synchronously.

2.1 Fogo Player

Fogo Player [1] enables the distribution, transmission and displaying of 4K videos, with or without stereoscopy. It is based on a distributed architecture of software components. Different of other players found in our review, Fogo Player does not use dedicated hardware for decoding and displaying the videos. The Fogo Player’s architecture allows easy integration of different technologies for video coding, which admits the use of the system also for the transmission of raw video using lossless compression algorithms. However, the algorithms used need to be quick enough to enable the transmission and display of UHD videos in real-time.

To handle 4K videos, Fogo Player slices the videos into quadrants in order to process them in parallel. To display the videos, the Fogo Player decodes in parallel each of the quadrants and displays them in a synchronized way on a wall of screens or using a projector. Parallel processing may be performed using multiple processors in a single machine or using multiple machines.

The Fogo Player’s architecture allows the replacement of the modules responsible for decoding without affecting the operation of the rest of the player. In this work we performed the integration of a decoding module based on the LZ4HC algorithm.

3. A COMPARISON BETWEEN LOSSLESS COMPRESSION ALGORITHMS

The compression speed is not a critical requirement for on demand video streaming. However, decompression must be performed in real time as the video is received and displayed. On the other hand, real-time compression is also required for live video transmissions and this usually requires dedicated encoding hardware. In the case of collaborative editing applications, the videos can be compressed in parallel as professionals perform modifications in individual frames since the compression is applied only intra-frame.

We evaluated eight lossless compression algorithms in order to attest the feasibility of using this kind of compression for streaming 4K raw videos. The software libraries used are all open source and the algorithms and their license types are shown in Table 1.

All eight tested algorithms are based on the Lempel-Ziv algorithm, which makes use of a dictionary. These algorithms are optimized to achieve high speed of compression or decompression but have a relatively low compression ratio, especially when compared with the lossy algorithms used specifically to compress images and videos. However, these lossless algorithms have the advantage of allowing the complete reconstruction of the original video on the target, which is critical for some applications.

The LZ4 is a dictionary based algorithm which was developed for high speed compression and decompression of text files [8]. In general, dictionary-based algorithms convert variable-length symbol sets into fixed-size codes. Accordingly, sequences with more symbols can be mapped continuously into smaller code [9].

The LZ4HC algorithm is a variation of LZ4 with a higher compression ratio and a higher decompression speed. The disadvantage of LZ4HC over LZ4 is the compression time, which is substantially higher. However, for applications that are restricted on bandwidth for data transport and that perform more decompressions than compressions, the LZ4HC may be a good alternative.

Oliveira et al. [10] performed a comparison between the LZ4 and LZ4HC and some versions of ZLIB algorithm to compress data from the LHC particle accelerator. In the experiment, the LZ4HC had a compression ratio of about 57%, against 50% of LZ4 and the decompression speed was about 20% higher compared to the LZ4. However, the LZ4 compression speed was 15 times higher compared to the LZ4HC.

Although the work of Oliveira et al. [10] have carried out a comparison between the LZ4 and LZ4HC algorithms, it is important to conduct studies of the performance of these algorithms considering image files in order to verify the feasibility of its application in video transmission and displaying systems. Our paper analyzes LZ4, LZ4HC and other six algorithms.

This study also includes an analysis on the effects on performance and compression ratio of the usage of multiple threads for encoding and decoding video frames. In this case, before being encoded, the frames are divided into smaller blocks and the blocks are encoded in parallel using multiple threads.

Since the vast majority of current processors have multiple cores, you can exploit this feature to get real parallelism in
the encoding and decoding processes. However, subdividing the image into smaller blocks may result in a lower compression ratio.

4. RESULTS
The experiments were conducted on an IBM x3650 with 2 Xeon E5620 processors (eight 2.40 GHz physical cores) and 32 GB of RAM. The workload consisted in a set of 3000 frames with 4096 x 2160 resolution extracted from the 4K movie EstereoEnsaios [12], the first stereoscopic 4K film produced in Brazil.

Figure 1 shows the average compression speed of the 3000 frames using the eight selected algorithms. LZO presented the highest compression speed, processing 1353 MByte/s in the best case. This compression speed would make it possible to encode 4K-3D videos in software in real-time using a single machine. The LZ4 algorithm achieved the second highest compression speed, about 518 MByte/s in the best case. The LZ4HC, as expected, had a very low compression speed in comparison with the other algorithms, being able to process about 44 MByte/s in the best case.

Figure 2 shows the average decoding speed of the eight algorithms. The LZO algorithm achieved again the best results, with a speed of about 1584 MByte/s in the best case. Following the LZO, the best performances were achieved by the LZ4, Snappy, and LZ4HC algorithms, respectively.

Surprisingly, in our experiments LZ4HC had a slower decompression speed than LZ4, what differs from the results obtained by Oliveira et al. [10] for the files generated by the LHC.

Once decompressed, a 4K video generates a bitstream of about 5 Gbit/s. Any of the four algorithms with the higher decoding speeds would be able to decode a 4K video in real time. With these algorithms is would also be possible to decode 4K-3D videos in real time, since the decoding and displaying of 4K-3D videos may be done in a distributed way using the Fogo Player.

Figure 3 shows the compression ratio of the eight algorithms. The LZ4HC algorithm presented the highest average compression ratio (1.38 in the best case). The second best compression ratio was obtained with the LZMAT algorithm, but the speeds of compression and decompression of this algorithm were very low. The LZO algorithm presented a compression ratio of about 1.19 and the LZ4 presented a compression ratio of 1.23. Thus, although LZO and LZ4 have shown higher compression and decompression speeds, they showed a significantly lower compression ratio when compared to the LZ4HC.

As we see in the Figures, the number of threads have a direct influence in the three evaluated parameters. As the computer used in the experiments had 8 physical cores (16 virtual cores), we obtained a significant increase in speed of compression and decompression by using multiple threads. However, there is a limit to the number of threads used, due to the limitation on the number of cores and the scheduling overhead. For example, the LZ4HC algorithm presented the

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Site</th>
<th>License Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZF</td>
<td><a href="http://oldhome.schmorp.de/marc/lzflzf.html">http://oldhome.schmorp.de/marc/lzflzf.html</a></td>
<td>BSD-type</td>
</tr>
<tr>
<td>Snappy</td>
<td><a href="http://code.google.com/p/snappy/">http://code.google.com/p/snappy/</a></td>
<td>New BSD</td>
</tr>
<tr>
<td>LZ4</td>
<td><a href="http://code.google.com/p/lz4/">http://code.google.com/p/lz4/</a></td>
<td>New BSD</td>
</tr>
<tr>
<td>LZ4HC</td>
<td><a href="http://code.google.com/p/lz4hc/">http://code.google.com/p/lz4hc/</a></td>
<td>GNU Lesser</td>
</tr>
<tr>
<td>FastLZ</td>
<td><a href="http://code.google.com/p/fastlz/">http://code.google.com/p/fastlz/</a></td>
<td>GPL</td>
</tr>
<tr>
<td>LZMAT</td>
<td><a href="http://www.matcode.com/lzmat.htm">http://www.matcode.com/lzmat.htm</a></td>
<td>MIT</td>
</tr>
<tr>
<td>LZO</td>
<td><a href="http://www.oberhumer.com/opensource/lzo/">http://www.oberhumer.com/opensource/lzo/</a></td>
<td>GPL v2+</td>
</tr>
<tr>
<td>QuickLZ</td>
<td><a href="http://www.quicklz.com/">http://www.quicklz.com/</a></td>
<td>GPL 1, 2 e 3</td>
</tr>
</tbody>
</table>
best results using eight cores.

Furthermore, we observed a worsening in the compression ratio in some algorithms when the number of threads was increased. This is due to the fact that smaller picture blocks have a lower level of redundancy compared to the full image. However, the compression ratio for the frames used does not vary significantly even considering 16 threads.

Although having achieved a low compression speed, the LZ4HC algorithm showed the best performance in our evaluation, since it presented a compression ratio significantly higher compared to the other algorithms and a fast enough decompression speed to decode 4K or 4K-3D videos in real time. This algorithm can be used to support distributed 4K editing applications since the coding process do not need to be performed in real time. This algorithm also works satisfactorily for the transmission of 4K raw videos on demand. In the latter scenario the encoding process can be done in a distributed way, as defined in the Fogo Player’s architecture.

5. CONCLUSIONS AND FUTURE WORKS

The general goal of this work is to evaluate the feasibility of using lossless compression to reduce the bandwidth requirements of a software-based solution for transmitting and displaying UHD raw videos, with or without stereoscopy.

We performed a comparative study of 8 lossless compression algorithms in order to observe their compression and decompression speeds and the compression ratio achieved by each of the algorithms. We performed also a study to assess the impact of using multiple threads in these three parameters.

The results showed that LZ4HC algorithm is best suited for the application, since it presents a good compression ratio in comparison to the other algorithms, reaching 38% on average for the frames used in the experiments. In addition, the LZ4HC presented a decompression speed fast enough to allow the decoding of 4K videos (2D or 3D) in real time. We observed that the use of multiple threads can cause a significant increase in the compression and decompression speeds without significantly worsening the compression ratio.

As future work we intend to conduct performance studies of the Fogo Player using the LZ4HC algorithm for transmission and display in real-time of 4K raw videos over long distance networks, in order to verify the overall performance of the player and analyze network metrics such as delay and bandwidth usage. We also intend to perform the world’s first demonstration of a real time transmission of a 4K-3D raw video.

6. ACKNOWLEDGMENTS

The authors would like to thank the National Education and Research Network (RNP), CNPq and CAPES for their financial support.

7. REFERENCES