Wireless Compression for Head Mounted Displays

Omraan M. Khan, *Student, UIUC*

This paper aims to study video compression within the context of wireless head mounted devices (HMD's). The goal was to improve video compression for high resolution binocular video, displayed close to the user's eyes. We attempted to implement a foveation algorithm using MATLAB and analyzed the file sizes before and after to measure our effectiveness. We found that foveation is a very effective method to compress video, especially for HMDs by leveraging the limitation of human vision. We were able to see a 15x improvement over traditional MPEG - 4 compression. These compression levels may allow us to even send video over the WIFI 5 standard.

***Index Terms*— Foveation, Head Mounted Display, Image signal Processing**

I. Context

IMMERSIVE technologies, such as Extended Reality (XR) make extensive usage of Head Mounted Devices or HMDs. HMDs allow content creators to create virtual and augmented reality spaces for users to completely immerse themselves in. Modern HMDs typically have two high resolution, high refresh rate screens mounted near the eyes, with lenses creating the illusion of a truly 3d space.

Some HMDs such as the new Apple Vision Pro have all the compute on board mounted on the user's head. In contrast, older HMDs such as the HTC VIVE use a wire connection to an external computer which performs much of the computation. There are pros and cons to each approach. Standalone HMDs require no base station to operate and function independently; but they often require larger batteries, are typically heavier, and have lower computational power compared to a desktop device. Whereas base station HMDs have more compute available to them, require smaller batteries, and are lighter; but can have higher latency and may require wires to offer a stable connection.

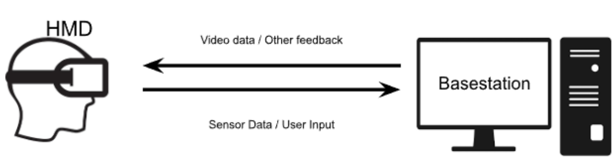

Fig. 1. An example of a Bases Station HMD setup

One way to improve base station HMDs would be to eliminate the need for wires while still providing stable, low-latency high-quality video. This paper aims to explore video compression techniques that can be used for this exact purpose.

II. Constraints and Solutions

The main obstacle to wireless base station HMDs is the need for: very stable, very high resolution, high frame-rate video, and minimal latency. All while limited to WIFI bandwidth and reliability.

Constraints:

- 4k or greater resolution to avoid screen door artifacts
- 120 + frames per second to provide a smooth experience
- Round trip latency <40ms to avoid motion sickness, includes processing time, as well as transmission time
    - Realistically we want to aim for <10 ms latency for transmission
- Video stream must be stable and avoid stuttering and other disorienting side effects even under degraded network conditions
- WIFI bandwidth is lower than a wired connection, and bandwidth fluctuates heavily based on environmental factors.

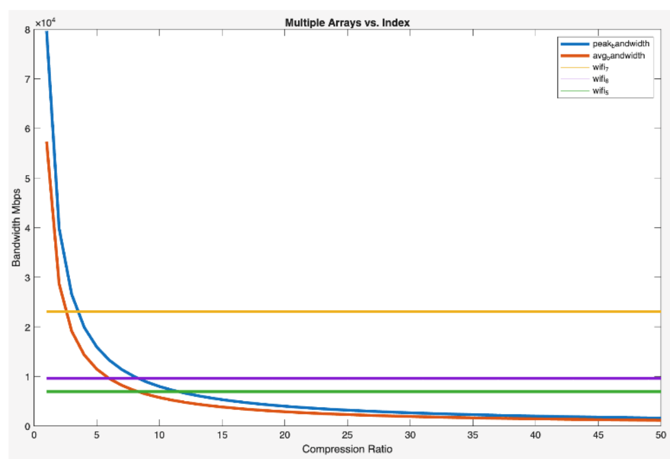

Fig. 2. Illustration of Required Bandwidth for a 4k Video at 144 FPS

As we can see we are sending quite a bit of raw data in a short amount of time, while still limited to the available bandwidth of modern WIFI standards. These available bandwidths are also idealized numbers and do not reflect real world conditions. As such we want to target a required bandwidth that is around 1/10th the available bandwidth. This means for the various we want a compression ratio around 5-10 for WIFI 7 and 6, and a compression ratio of 10-15 for WIFI 5.

To do this we have a wide array of possible solutions:

- P frame compression, sending an I frame and then differentials and motion compensation from the I frame as a P frame
- B frame compression, sending an I frame and then B frames which are motion compensated from I frames in the past and P frames from the future
- foveation, leveraging the limitations of the human eye to reduce image quality in areas where it will not be noticed

We would really like to be able to leverage motion compensated P and B frame compression, to bring the required bandwidth down to suitable levels. However, B frame compression introduces large amounts of latency, which we are trying to minimize to prevent motion sickness in users, as well as risking stuttering under poor network conditions if P or I frames are lost. We could use P frame compression with a low GOP, to reduce latency to negligible levels, but we still run the risk of stuttering when I frames are dropped which would significantly hurt user experience. This leads us to turn to foveation which is a rendering method that can reduce the size of each frame that we need to transmit.

III. Approach and Methodology

To get around these limitations we may be able to take advantage of the close distance between a user's eyes and the screens of an HMD. The screen will take up most of the user's field of view. This means that the limitations of human vision will have a significant impact on what parts of an image the user can actually see.

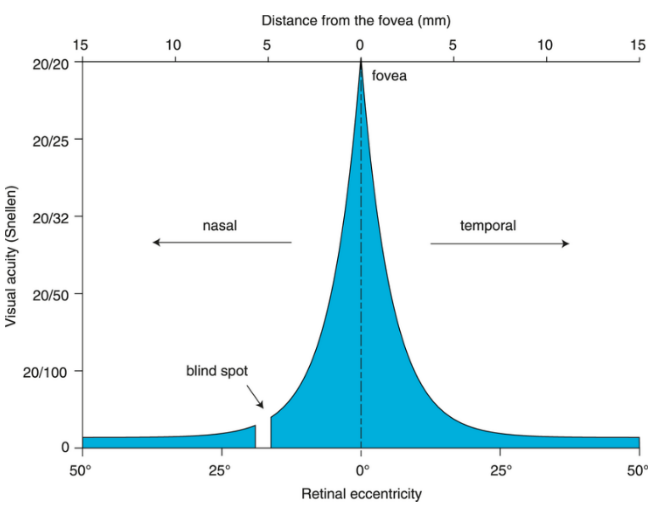

Fig. 3.: Visual acuity as a function of degrees of retinal eccentricity, for the left eye [1]

As we can see, visual accuracy decreases significantly as we move away from the center of our vision. Foveation is a rendering technique that aims to leverage this limitation of the human eye. If we assume that the full 100° vision is taken up by a HD video, we only need an 80 wide pixel strip of video to account for visual acuity > 20/100. Foveation takes advantage of this by significantly reducing resolution on the periphery of an image, where our user is not likely able to notice this change. This ultimately reduces the number of pixels that we need to send between the base station and HMD, reducing the required bandwidth for the video stream.

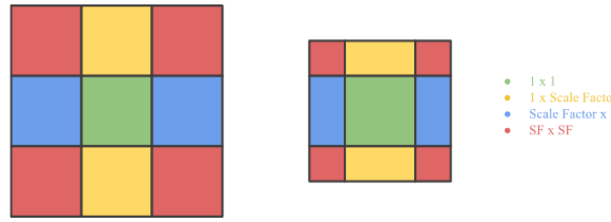

Fig. 4. Representation of the Foveation Algorithm

We aimed to implement a basic foveation algorithm in MATLAB. We would divide each frame into 9 sectors and then vertically, and/or horizontally resize the 8 sectors reducing their size in a form of lossy compression. Effectively reducing the resolution along the periphery of the image. This method leaves the 9th central Sector untouched, preserving the video quality there. Then the receiver, if given the input parameters, can simply undo this transformation thus decompressing the video.

IV. Data and Analysis

Below is the data as we swept center resolutions from 1920x1080 through to 100x100.

TABLE I
FOVEATION COMPRESSION DATA

Starting Dimensions	Center Dimensions	Edge Scale Factor	Ending Dimensions	Size Bytes (Foveation)	Size Bytes (MPEG 4)	Foveation Compression Ratio	MPEG Compression Ratio	Overall Compression Ratio
1920 x 1080	1920 x 1080	5	1920 x 1080	6,220,800	2,411,501	1.00	2.58	2.58
1920 x 1080	1000 x 1000	5	1184 x 1016	3,608,832	1,449,821	1.72	2.49	4.29
1920 x 1080	900 x 900	5	1104 x 936	3,100,032	1,194,185	2.01	2.60	5.21
1920 x 1080	800 x 800	5	1024 x 856	2,629,632	935,138	2.37	2.81	6.65
1920 x 1080	700 x 700	5	944 x 776	2,197,632	820,935	2.83	2.68	7.58
1920 x 1080	600 x 600	5	864 x 696	1,804,032	663,071	3.45	2.72	9.38
1920 x 1080	500 x 500	5	784 x 616	1,448,832	548,222	4.29	2.64	11.35
1920 x 1080	400 x 400	5	704 x 536	1,132,032	435,018	5.50	2.60	14.30
1920 x 1080	300 x 300	5	624 x 456	853,632	339,631	7.29	2.51	18.32
1920 x 1080	200 x 200	5	544 x 376	613,632	248,760	10.14	2.47	25.01
1920 x 1080	100 x 100	5	464 x 296	412,032	181,891	15.10	2.27	34.20

As we can see, foveation not only compressed quite a bit on its own but also compressed better under MPEG 4. This means that between foveation and MPEG 4 we were able to compress quite a lot without the added complexity/latency of motion compensation.

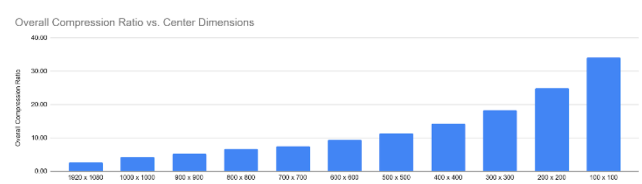

Fig. 5. Foveation Compression Ratio Vs Preserved Center Dimension

Although the application we are aiming to optimize typically demands 4k we were only able to source 1080p video and performed our analysis on that. As we can see as we reduce the center-dimensions, the file size decreases. We can reduce the

1080p image quite a bit even while maintaining large center dimensions. Keep in mind we are still maintaining full quality in the center, so we can still perform other forms of compression on the whole frame before foveating.

In fact, we can compress so much that with a 300 x 300 center preserved (Roughly 1/10th the original area) we would be able to send the video data on WIFI 7, a 200 x 200 enables WIFI 5 and 6 bandwidths to be utilized.

V. Impacts

By reducing the payload size, we will improve our video streaming performance in several ways. Less data to transmit means we can send our data under worse wireless conditions where bandwidth may be limited. Under good conditions we will be able to accommodate higher frame rates as the # of packets per frame will have dropped significantly. Lower payloads lead to less packets which allows us to do two things: we can send fewer packets reducing latency, or we can keep latency the same but can now resend packets for more reliable video. The benefits of a more compressed video stream continue. Ultimately the kind of video stream a wireless HMD requires becomes more practical to transmit, which means that we will be able to create lighter and more comfortable devices for the user.

IV.FURTHER STUDY

Further study must be done into different foveation algorithms are other transforms (especially nonlinear ones) that can better compress peripheral information. Are there other things we can do to make an image easier to transmit? Can foveation be done dynamically to compress less relevant information (such as the background) and preserve points of interest? This study should also be repeated with 4k video data to ensure that the reduction in payload scales with video resolution. Study should be done on what foveation scale factors are imperceptible to humans. We should also consider applying foveation in a gradient compressing more on the edges and reducing the compression as we approach the center.

IIV. Conclusion

We have successfully shown that foveation can significantly improve I frame compression. We end up with almost 35 compression factors, when we reduce our center dimensions down to 100 x 100. This is impressive when we consider that 80 pixels is all we need to account for the most accurate portion of our vision. With further study put into exact parameters, as well as more effective algorithms we should be able to achieve compress video suitable for wireless HMDs even under poor network conditions.

References

[1] Lambertus S, Bax NM, Fakin A, Groenewoud JMM, Klevering BJ, et al. (2017) Highly sensitive measurements of disease progression in rare disorders: Developing and validating a multimodal model of retinal degeneration in Stargardt disease. PLOS ONE 12(3): e0174020. https://doi.org/10.1371/journal.pone.0174020