Video Compression Final Project

Group E R08945033施翰嘉 P08922003 凃仲蔚

**Improving Video Compression Rate through Temporal Axis Transposition of Video**

**Experimental Hypothesis**

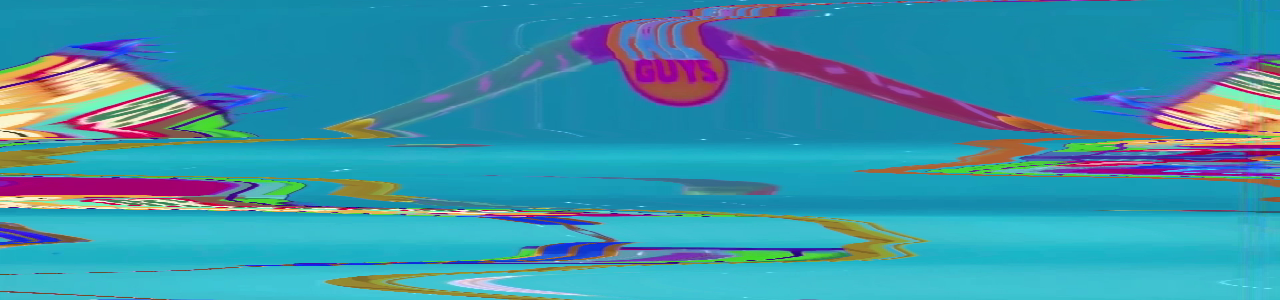
H264 video compression includes inter and intra coding. We hypothesize that if certain scan lines in the video do not change or change minimally throughout the entire video, such as in conference recordings or recordings with a fixed scene, we can transpose the height axis and time axis of the video. For instance, for a 300-frame 1280x720 video, we take the first scan line of each frame to obtain 300 scan lines, which are then concatenated into a 1280x300 image to represent the first frame. Similarly, we take the k-th scan line from each frame to form the k-th frame's 1280x300 image, ultimately resulting in 720 images of 1280x300. These images are then compressed using the H264 encoder, with the expectation of achieving a better compression rate.

**Experimental Process**

The videos selected for this experiment are all 1280x720, 30fps, and 10 seconds long. We used ffmpeg to extract 296 frames saved as bmp images, and then compared the size of the video encoded with H264 from 296 bmp images of 1280x720 with the size of the video encoded with H264 from 720 bmp images transposed to 1280x296. The first video is a gaming live stream (guys.mp4), where only a small portion of the UI area remains unchanged throughout the video. The expected effect is probably not very significant.



After the Transposition of the First Frame



It is observable that there are many changes over time, and the word "GUYS" can be seen in the upper middle part, originating from the "GUYS" sign at the top of the original video, which appears as the character moves forward and passes through the first scan line, indicating that our transposition implementation is correct.



The file sizes of the videos encoded with H264 from 296 non-transposed 1280x720 bmp images and 720 transposed 1280x296 bmp images are:

Non-transposed : **5,171 KB**

Transposed : **8,964 KB**

This is because only the UI part, such as scan line 46, has a fixed and unchanged scene in the image.

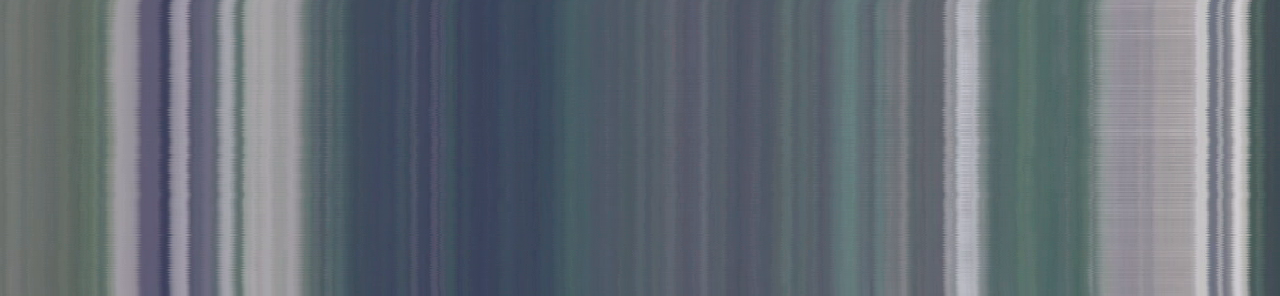


Therefore, the poor compression result is as expected.

The second video is an ecological video shot outdoors with a tripod (bird.mp4).



The original motivation for the topic was that many videos recorded with a tripod have more stable scenes, leading to more scan lines without content changes. However, it was discovered during the experiment that even with a tripod or fixed camera for surveillance recording, the content of the scene still has minor vibrations or changes, making the transposed scan line images appear as if



they mostly remain unchanged, but still with minor changes.

The file sizes of the experiment results are:

Non-transposed : **6,512 KB**

Transposed : **11,410 KB**

ffmpeg output

Non-Transposed

I-Frame : 2 Avg QP : 16.49 size: 145,886

P-Frame : 79 Avg QP : 18.03 size: 55,027

B-Frame : 217 Avg QP : 22.32 size: 9,326

transposed

Frame I : 63 Avg QP : 16.19 size: 30,872 (30,872\*720/296 = 75,094)

Frame P : 238 Avg QP : 20.58 size: 20,991 (20,991\*720/296 = 51,059)

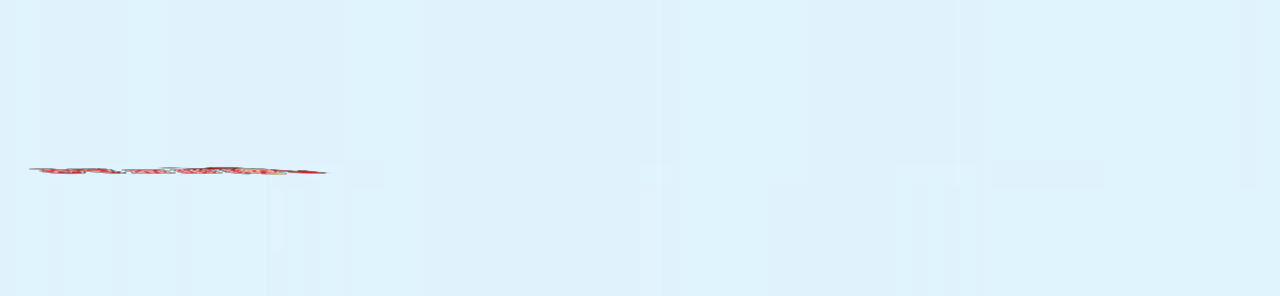
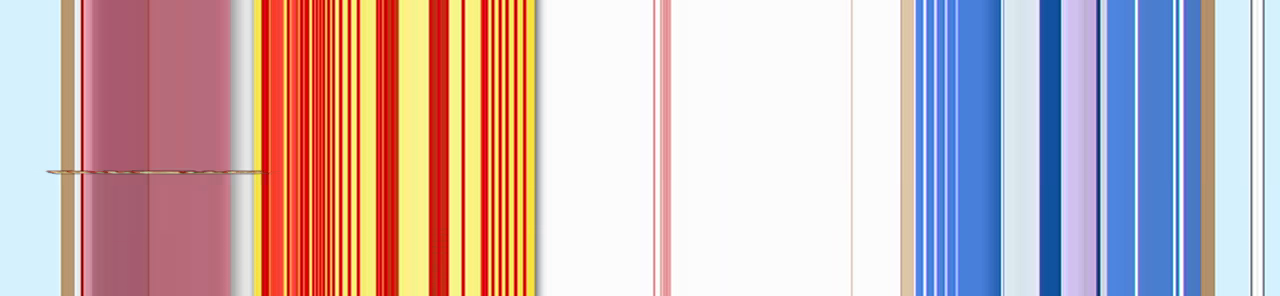
Frame B : 420 Avg QP : 21.45 size: 11,270 (11,270\*720/296 = 27,413)

After transposition, the number of I Frames required during the encoding process increased significantly. Although the recalculated size of each I Frame to a height of 720px (Size \*720/296) requires less space (75,094) compared to the storage space for non-transposed bmp (145,886), the substantial increase in the number of I Frames leads to a significant increase in the overall size of the video.

The third video is from a stock market analysis show. Since the image of the host changes while most other areas remain unchanged, a better compression effect is anticipated.



Most scan lines produce a very stable linear pattern after transposition.



After H264 compression, the video file sizes are:

Non-transposed : 562KB

Transposed : 1,084KB

Non-transposed

I-Frame : 2 Avg QP : 13.24 size: 166,110

P-Frame : 77 Avg QP : 14.91 size: 2,209

B-Frame : 219 Avg QP : 13.52 size: 309

Transposed

I-Frame : 54 Avg QP : 15.20 size: 2,147

P-Frame : 558 Avg QP : 19.33 size: 1,728 (1,728\*720/296 = 4,203)

B-Frame : 109 Avg QP : 15.52 size: 203 (203\*720/296 = 493)

The increase in I Frames is significant, believed to be due to the lack of correlation in height between most pairs of scan lines in the video, leading to an increase in I Frames after transposition. The third video experiment clearly shows a significant decrease in the size used by I Frames, but since the sizes used by P and B Frames are not low and the proportion of P Frames is much higher than in the non-transposed video, the overall video size is still larger.

Since direct compression with H264 did not yield good results, we attempted to compress the transposed bmp images with jpeg and perform Huffman Table Optimization to see if the total size of the 720 transposed 1280x300 .jpg images would perform better than H264 compression.

The results show that at quality=95, the size of a single image is 14KB, totaling 10MB for 720 images, far exceeding the file size of the H264 video. At quality=50, the file size is 10KB per image, with a total file size of 7.2MB, still far exceeding the 562KB produced by H264 compression of the non-transposed bmp.

In the end, we found that restoring the transposed bmp compressed video. Significantly reduces image quality. The left image is a screenshot from the video produced by the transposed bmp, and the right image is from the non-transposed bmp video.



The non-transposed bmp compressed with jpeg to 352KB and the transposed bmp compressed to 14KB significantly reduce the total file size, but neither is as small as the video directly compressed with H264.

Conclusion

The experiments conducted with three videos confirmed that H264's compression effectiveness for inter frames is already very good. Transposing the frame and height axes before compression does not yield better results. Instead, transposition results in significant differences in content between consecutive frames, leading to a substantial increase in the number of I Frames. Even though the space required for each I Frame decreases after transposition, it does not compensate for this difference. In the third video experiment, the proportion of P Frames also significantly increased, resulting in a higher overall file size.

The final experiment with bmp compression to jpeg also failed to reduce the size of individual jpeg compressed files enough to outperform H264's compression efficiency.

**程式碼**

Transport.py對於輸入影像 (Line.51) 產生轉置的bmp和轉置bmp的壓縮影片(transpose\_bmp.mp4)，以及壓縮影片再次轉置還原回原始影像(transpose\_mp4\_extract\_transpose\_back.mp4)

其產生的original\_bmp.mp4是使用未轉置bmp壓縮的影像，檔案大小總是少於transpose\_bmp.mp4

test\_jpeg.py則是以jpeg壓縮轉置/不轉置bmp的功能

實驗產生的輸出檔案都在 src/data/exp\_{video\_name} 中

實驗影片為src/data/guys.mp4 bird.mp4 stock.mp4