Adv MM Assignment #1 – x264 experimentation

Kai Wen Yoke, 2020280598
Khang Hui Chua, 2020280442

December 10, 2020

1 Introduction

In this assignment, we explore the effects on `x264`'s video encoding performance of (1) different presets on `x264` and (2) different video types.

2 Methodology

2.1 Parameters used when running `x264`

There are 9 presets in total: `PRESETS='ultrafast superfast veryfast faster fast medium slow slower veryslow'`, and a faster preset would lead to faster video encoding but lower quality. We test out the tradeoff between speed and quality by testing all 9 presets on each of the 10 videos, so a total of 90 experiments were executed. To ensure fairness when comparing across different videos, we set `x264` with a target average bitrate of 0.8 of the original average video bitrate. We evaluate the visual quality of encoded videos using the SSIM and PSNR metrics output by `x264`.
In summary, we ran the following command: `x264 --bitrate $BITRATE_KBPS_TARG --preset $PRESET --no-progress --ssim --psnr -o $FILENAME_OUT $FILE`

2.2 Details of 10 sample videos chosen

We selected 10 sample videos to represent a variety of video types, each of approximately 4k resolution, and trimmed to roughly 10s. (Source: youtube, vimeo). The trimmed source videos can be found here: https://cloud.tsinghua.edu.cn/d/c4e2cb11d7f349db9315/. The media information is shown in Table 1 and the content variety is shown in Figure 1. We tried to choose videos with high and low motion, natural and synthetic scenes, nature and man-made scenes, and detailed and smoother scenes.

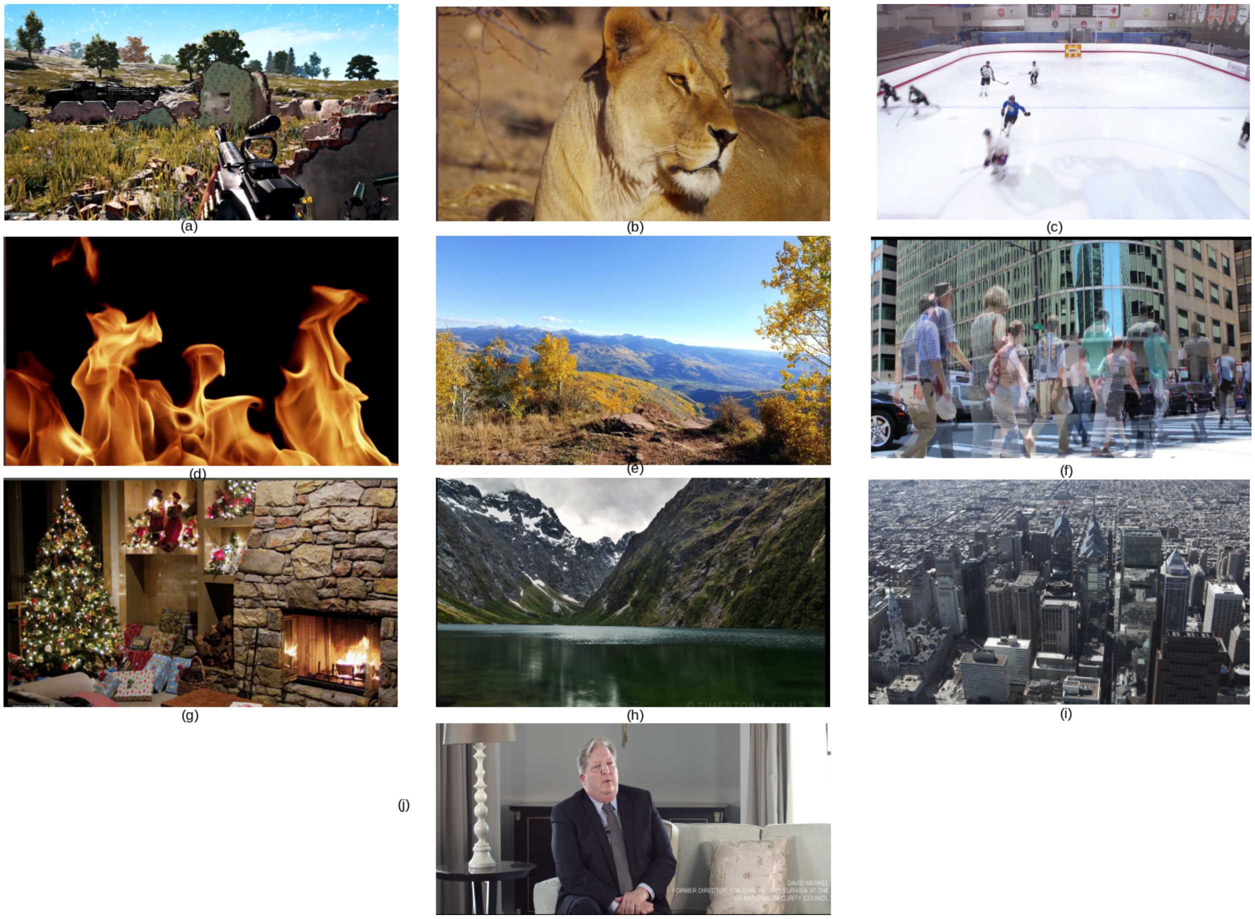

Figure 1: Content range of 10 sample videos: (a) PUBG gameplay; (b) Lion; (c) Hockey; (d) Fire; (e) Drone flying over mountains; (f) City streets timelapse; (g) Christmas at home; (h) Water and clouds in New Zealand; (i) Aerial view of Philadelphia; (j) Interview

VIDEO	***FileSz (Mb)***	***Res***	***BitRate (Mbps)***	***FPS***	***Frames***	***Remarks***
(a) PUBG gameplay	78.3	3840 2160	59.5	29.81	313	Synthetic, high motion
(b) Lion	19.8	4096 2160	14.3	30	320	Quite smooth and homogeneous, low motion
(c) Hockey	17.7	3840 2026	13.6	29.97	307	High motion
(d) Fire	21.6	4096 2160	16.4	39	304	Relatively high motion, rippling fire
(e) Drone flying over mountains	140	3840 2160	106	30	305	Nature, fine details
(f) City streets timelapse	31.3	4096 2048	24.2	30	311	Man-made, still background moving foreground

(g) Christmas at home	64.6	3840 2160	48.2	30	309	Low motion, still except for fireplace
(h) Water and clouds in New Zealand	88.8	3840 2160	70.6	30	301	Nature, fine details, rippling clouds and water
(i) Aerial view of Philadelphia	29.8	4096 2048	23.7	30	302	Man-made, fine details
(j) Interview	6.21	3840 2160	4.72	29.532	299	Low motion, still except the face

Table 1: Table showing details of 10 sample videos

3 Comparing encoding performance

When we called the `x264` command, we set a target output bitrate to be 0.8 of the original for fair comparison across different content types. However, `x264` can only attempt to achieve the target bitrate depending on the difficulty of the video type. Hence, we measured the output bitrate with each preset, averaged the output bitrate over all nine presets per video, and plotted a bar chart of the average output bitrate as a ratio of the original video bitrate in Figure 2.

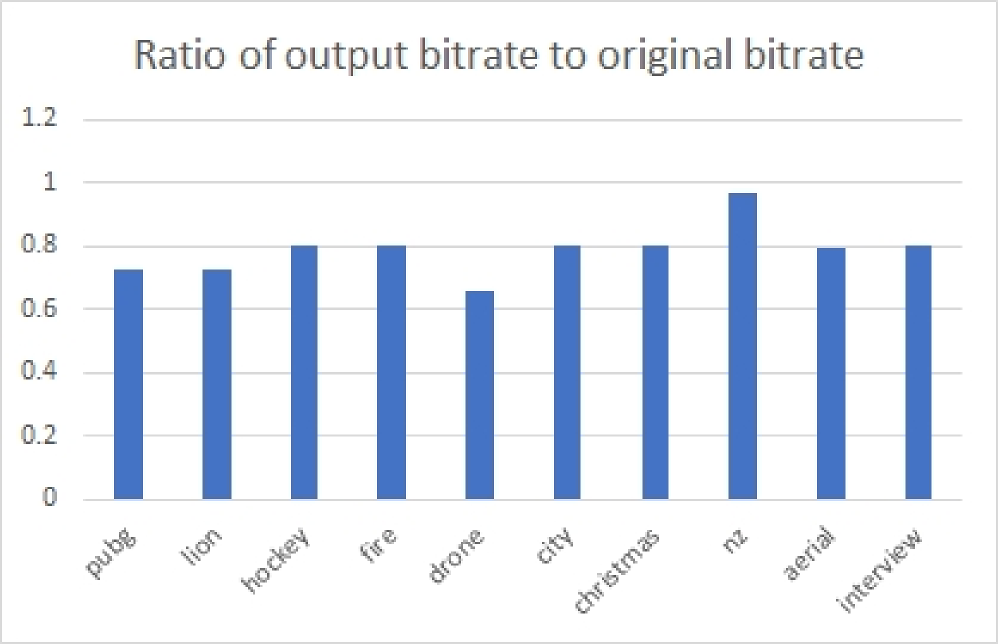

Figure 2: Ratio of output bitrate to original bitrate for each video, averaged over different presets

Most of the videos are compressed to roughly 80% which is what we aimed for, however the ’(b) Water and clouds in New Zealand’ video only managed an average compression of 97%, while ’(e) Drone flying over mountains’ was compressed much more to 66%. Perhaps video (b) was more difficult to compress due to varying fine details in clouds and water texture, but we are not sure why (e) was compressed much more than instructed in the settings. Despite having two significant outliers in terms of extent of compression, we decided to go ahead with this setup and compare encoding performance across different videos later on, but keeping in mind that (b) and (e) had quite different bitrates from the other eight videos.

3.1 Comparing effects of different presets

We plotted across different presets the execution time in Figure 3, PSNR in Figure 4 and SSIM in Figure 5. The results seem to suggest that there is no need to go beyond the `medium` preset, and more often than not, `faster` is a pretty good balance between speed and quality.

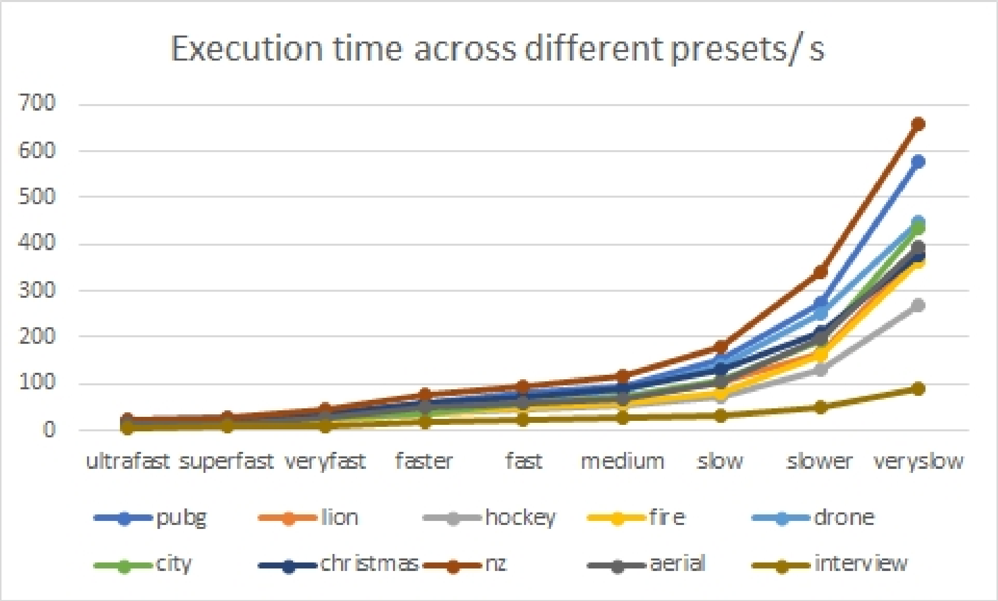

Figure 3: Execution time across different presets

Execution time seems to increase significantly beyond the `medium` preset.

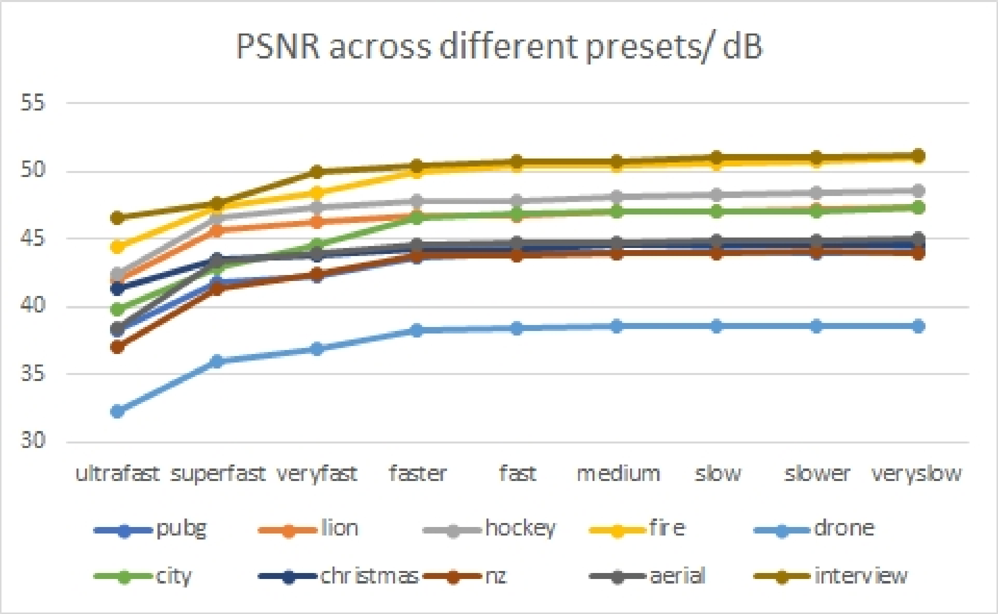

Figure 4: PSNR across different presets

PSNR seems to stop increasing significantly beyond the `faster` preset.

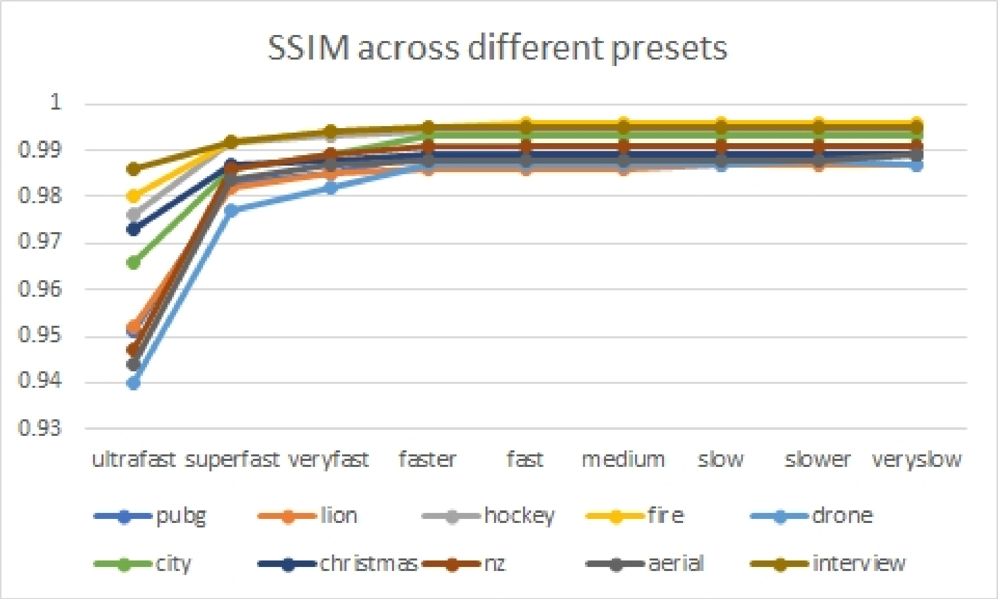

Figure 5: SSIM across different presets

Similar to PSNR, SSIM seems to stop increasing significantly beyond the `faster` preset.

3.2 Comparing effects of different image content

The PSNR and SSIM metric do not perfectly capture the visual quality of the videos (does not take into account motion, and also do not correspond to human visual perception). However, for the ten videos, they seem to correlate quite well as seen in Figure 6 so we will use them for evaluating compression quality. Since eight of the videos were compressed by the same relative amount according to bitrate, if the output has a higher PSNR/SSIM score, then we can conclude that this kind of video is easier to compress by x264. This inference does not apply to '(h) Water and clouds in New Zealand' and '(e) Drone flying over mountains'.

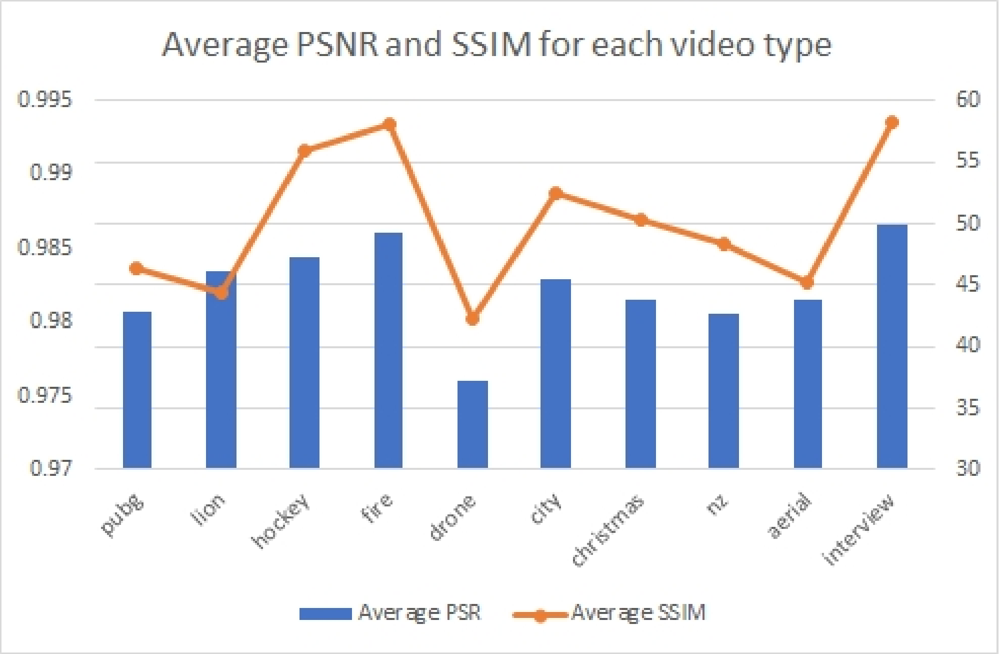

Figure 6: Average PSNR and SSIM across different video types

3.2.1 Easy videos - low motion, homogeneous background

The easiest to compress video is '(j) Interview' which is not surprising since there is no movement except for the interviewee talking, hence x264 is able to exploit the temporal redundancy between frames. The background of the interviewee is also relatively smooth and not too finely detailed, and thus thee is a lot of spatial redundancy within each single frame as well, leading to the best visual quality.

For a similar reason, '(d) Fire', and '(b) Lion' are also quite easy to compress. '(d) Fire' has a homogeneous black background, but the flames move quite significantly from frame to frame and I thought that would present some difficulty, but perhaps the rippling of the flames was smooth and predictable so x264 could still perform well on '(d) Fire'.

For '(b) Lion', the lion is the main focus of the frame and the background is blurred, so the texture is rather homogeneous, and also the lion moves very slowly throughout the video, so that is why it was easy to encode. However, I was expecting '(b) Lion' to be one of the easiest, but it was only the fourth easiest.

3.2.2 Unexpectedly easy videos - high motion

'(c) Hockey' was a video of people playing hockey in an ice arena, with the players moving rather fast, on a quite homogeneous white arena. It turns out that '(c) Hockey' was the third easiest to encode surprisingly despite the high motion of the players skating. However, it is true that the players were quite small compared to the arena which was a homogeneous white space, so perhaps the spatial redundancies significantly outweighed the high motion factor leading to good compression performance.
Similarly, for '(f) City Streets timelapse', the city buildings background is constant, with people moving quickly in the foreground in a time lapse fashion. I think the constant background possibly contributed significantly to the ease of compression despite the high motion.

3.3 Unexpectedly difficult video - low motion but fine details

'(g) Christmas at home' is a video of a living room with a christmas tree and fireplace, and it is motionless except for fire crackling at the fireplace. There is a lot of detail in each frame, but due to the low motion, I was expecting '(g) Christmas at home' to be one of the easier ones to compress, however, it lies roughly in the middle, so it seems to confirm the trend that the level of detail is more significant then level of motion in terms of influence on compression quality.

3.3.1 Difficult videos - fine details

Even with very little bitrate compression to 97% of original bitrate (see Figure 2), '(h) Water and clouds in New Zealand' still performed among the worst. The worst performing is '(e) Drone flying over mountains' but note that it also underwent the highest bitrate compression to 66% (see Figure 2). Even with these considerations, it is clear that both videos are difficult to compress, owing to the fine details captured of grass, water, skies, clouds with shadows. Particularly for '(h) Water and clouds in New Zealand', there is also very fine and rapid changes to the water surface and clouds between every frame which increases encoding difficulty.

3.3.2 Other sample videos

'(a) PUBG gameplay' is a synthetic screen capture of an online game, and it does not seem any easier or more difficult than the natural videos because of its artificiality. In fact, I think because the PUBG video contained quite a lot of details and motion, it was one of the more difficult ones to encode.
'(i) Aerial view of Philadelphia' is an aerial view of skyscrapers and buildings, with the drone's camera moving slowly. There is quite a lot of detail in the video, which is probably why it was among the worst peforming as expected.

4 Summary of findings

Overall, I think videos with high level of detail in each still frame are the most difficult to encode, and higher motion also increases the level of difficulty but not as significantly as finer details. Synthetic or natural images do not matter.