

STRANGE BEHAVIORS AND ROOT CAUSE IN THE COMPRESSION OF PREVIOUSLY COMPRESSED VIDEOS

Chiman Kwan and Jude Larkin

Applied Research, LLC, Rockville, Maryland, USA

ABSTRACT

In video compression class projects, students may observe some strange behaviors when using video codecs. Some performance metrics from a mediocre codec such as motion JPEG-2000 (or simply JPEG2000) may have exceptionally high values at certain compression ratios as compared to other high performing codecs. This strange behaviors may be overlooked by instructors and students may never understand why this is happening. In this paper, we will first highlight the strange behaviors. We will then use experiments to systematically determine the root cause. Our experiments show that, if one uses a previously compressed and decompressed video in some compression experiments, then it is highly likely that some strange behaviors will show up. Some advice will be provided to instructors, tutors, and students on how one can prevent such behaviors from occurring.

KEYWORDS

Video codecs, motion JPEG-2000, X264, X265, Daala

1. INTRODUCTION

Videos have been widely used in surveillance [1]-[11], traffic monitoring [13]-[15], and planetary exploration [18]. Video compression [19] helps data storage [20][21], reduces bandwidth usage [15], achieves real-time video communications [16][17], and is a popular course in many universities. When instructors teach video compression techniques in undergraduate or graduate courses, students are usually given some class projects to apply and compare different video codecs in the literature using some videos. Sometimes, students simply search the Internet and find some interesting videos without knowing the quality of the videos. As a result, some strange compression results may be encountered. For instance, a low quality video shown in Figure 1 may be used to compare different codecs. The peak signal-to-noise ratio (PSNR) results of different video codecs at different compression ratios are shown in Figure 2. One can immediately notice that the motion JPEG-2000 [22] results look exceptionally well between 0.01 and 0.02 compression ratios. However, this does not make sense because motion JPEG-2000, a codec of the 2000's, should not work that well as compared to other recent codecs such as X264 [23], X265 [24], and Daala [25]. Daala is a recent open source software developed based on lapped transform [26][27]. What caused such a strange and puzzling behavior? Students usually asked the instructors or tutors for help. They were probably told that their experiments might have some flaws. For example, they might have wrongly calculated the compression ratios, incorrectly set some quality control parameters in the codecs, etc. Then students spent a lot of time trying to fix their codes. However, after many hours of debugging, they were still frustrated because they could not find any errors in their codes.



Figure1. Two frames from a low quality video. There are color distortions in the frames.

From the above simple example, there are two natural and important questions that need to be answered. First, does the strange behavior in Fig. 2 only occur in JPEG2000? In other words, can this behavior occur in other well-known and more recent codecs such as X264, X265, and Daala? Second, what is the root cause of such strange behaviors? Determining it will have important educational values to students, tutors, and instructors.

In an earlier paper [28], we reported some strange behavior in image codecs. In particular, if one compresses a previously compressed image, it is very likely to see abnormally high metrics, no matter what image codecs one uses. At the end of [28], we speculated that we might also see strange behaviors in video codecs as well. In this paper, our main goal is to summarize our study on video codecs. We first present some experiments that illustrate some strange and normal behaviors in video compression. We then move on to identify the root cause of the above behavior shown in Fig. 2. After some extensive experiments, we determined the root cause was caused by the quality of those input videos, which were previously compressed and then decompressed by some codecs. The decompressed videos have some specific artifacts associated with each codec. When the decompressed videos are compressed again at certain compression ratios, the same artifacts show up again. Since the artifacts are similar to those artifacts in the original low quality videos, the PSNR values are high. We observed that this behavior does not only appear in JPEG2000, but it also happens in every recent and well-known codecs.

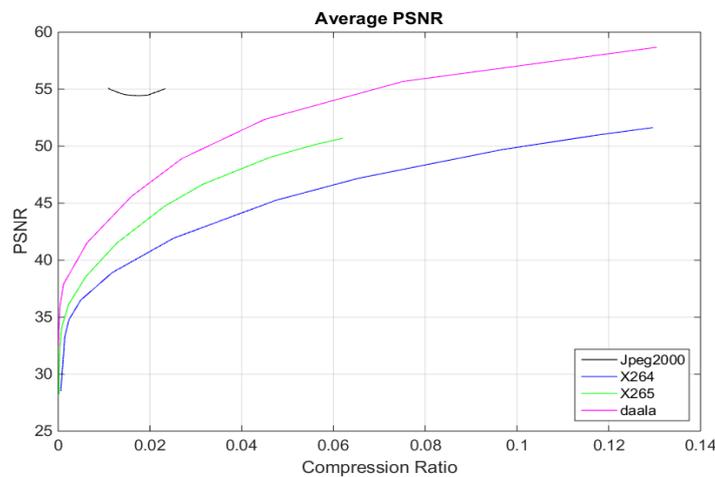


Figure2. Averged PSNR plots of 30 images using four well-known codecs: JPEG2000, X264, X265, and Daala. JPEG2000 has a very strange behavior.

Our contributions are as follows. First, we report some strange behaviors in video compression, which may have occurred in many university classes already. It will be important to let students, tutors, and instructors be aware of such behaviors. Second, we determined the root cause of the strange behavior shown in Fig. 2. If the input videos are compressed and decompressed using some codecs such as JPEG2000, X264, X265, and Daala, then strange and anomalous behavior can occur. For instance, if the images have been compressed and decompressed using JPEG2000 and the decompressed images are compressed again by JPEG2000, then we will see strange behaviors similar to that in Fig. 2. Our paper is organized as follows. In Section 2, we will show the strange and normal compression behaviors. In Section 3, we will show the root cause of those strange behaviors, which can actually happen in any codecs. This may be a surprise finding because people may think recent codecs are more robust to the quality of input images. Finally, we conclude our papers with some remarks and future directions.

2. STRANGE AND NORMAL BEHAVIORS IN VIDEO COMPRESSION

In this section, we focus on showing some strange and normal behaviors in video codecs.

2.1. A Close Look at Strange Behaviors

Let us first take a close look at those video frames shown In Figure 1 . We can notice that the image quality is rather poor as shown in Figure 3. There are a lot of artifacts if we zoom in to certain regions, indicating that the low quality video is probably a reconstructed or decompressed video. In fact, this low quality video was compressed 100 times and decompressed using JPEG2000. However, the PSNR metrics shown in Figure 2 have exceptionally high values, which do not make sense. We believe this puzzling phenomenon has already occurred many times in video compression classes and we have not seen any publications explaining this strange behavior.

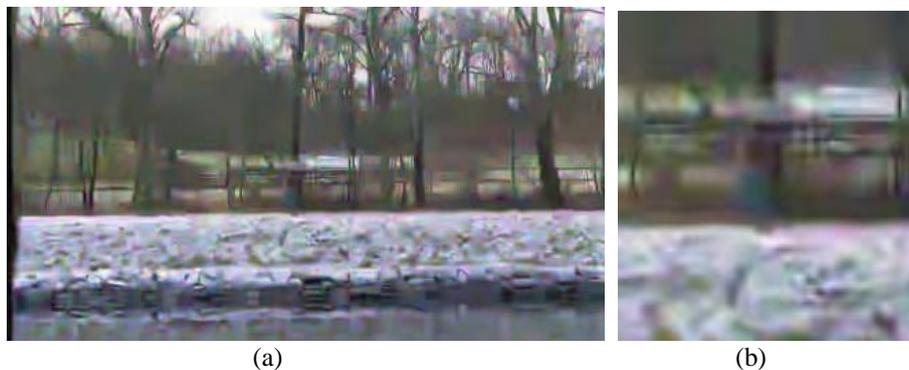


Figure 3. (a) Low quality video frame; (b) Zoomed-in region. This video is a decompressed video which was obtained by compressing a high quality video 100 times using JPEG2000.

2.2. Normal Behaviors

From the above, we suspect that the image quality may play an important role in the strange behaviors in Figure 2. We then carried out another study. Here, we used a high quality video that has not been compressed before. One video frame is shown in Figure 4, which has much better quality than that of Figure 3. We then repeated the same study as before and generated some performance metrics using four codecs. From Figure 5, we can see that, at high compression cases (less than 0.05 compression ratio), JPEG2000 is actually the lowest performing codec. The results are in agreement with past studies in the literature.



(a) Whole image

(b) Zoomed in image

Figure 4. (a) High resolution video; (b) Zoomed-in region.

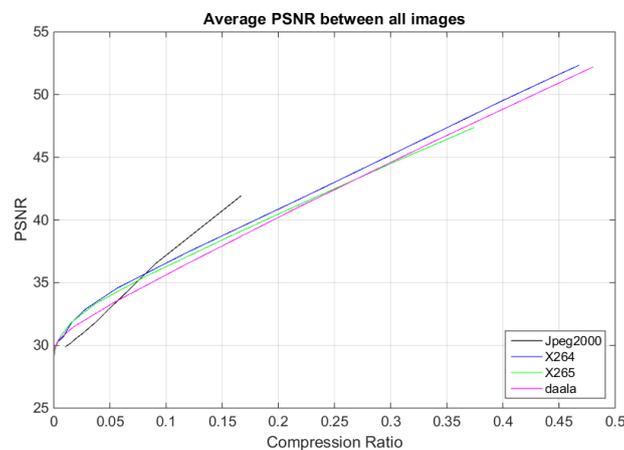


Figure 5. Normal compression behavior when original high quality video was used.

3. ROOT CAUSE OF STRANGE BEHAVIOR IN VIDEO CODECS AND ITS VERIFICATIONS

Based on observations in Section 2, it is quite clear what the root cause may be. We speculate that it is caused by the video quality. Here, we perform a systematic study to confirm the above speculation. We will also answer the second question: Does the strange behavior only happen in JPEG2000? The answer is negative, as we observe the same strange behavior in every recent and well-known codec in the literature.

The following two videos shown in Figure 6 were used in our experiments. The following procedures were used in our study:

- Step 1: Compress each video using a video codec (JPEG2000, X264, X265, or Daala) at several compression ratios, ranging from 100 to 1 to 20 to 1.
- Step 2: Decompress the video using the same codec in Step 1.
- Step 3: Compress the decompressed video in Step 2 using all codecs at different compression ratios and then generate the performance metrics using PSNR.
- Step 4: Analyze the metrics and see if there is any strange behavior in a particular codec.



(a) Running Video. Height: 720; Width: 1280; Frames: 50.



(b) Johnny Video. Height: 720; Width:1280; Frames: 601

Figure 6. High quality videos from Xiph [29].

3.1. Running Video

3.1.1. Study for JPEG2000 Codec

Here, we focus on JPEG2000. First, we compressed the video by 100, 50, and 20 times using JPEG2000 and then decompressed them. The decompressed videos are shown in Figure 7. For ease of comparison, the original video is also included. If one looks closely at the zoomed in area of each decompressed video, one can observe some artifacts due to compression. We then applied four codecs to compress and decompress those previously compressed and decompressed videos in Figure 7

Figure and the PSNR metrics are shown in Figure 8. It can be seen that JPEG2000 has abnormally high values in all metrics.

This study clearly shows that if the input video comes from a decompressed image using JPEG2000, it is highly likely that a strange behavior can also occur in performance metrics corresponding to JPEG2000.



(a) High quality video (frame 1)



(b) Zoomed in area



(c) Decompressed frame 1 (100 times compression);



(d) Zoomed in area

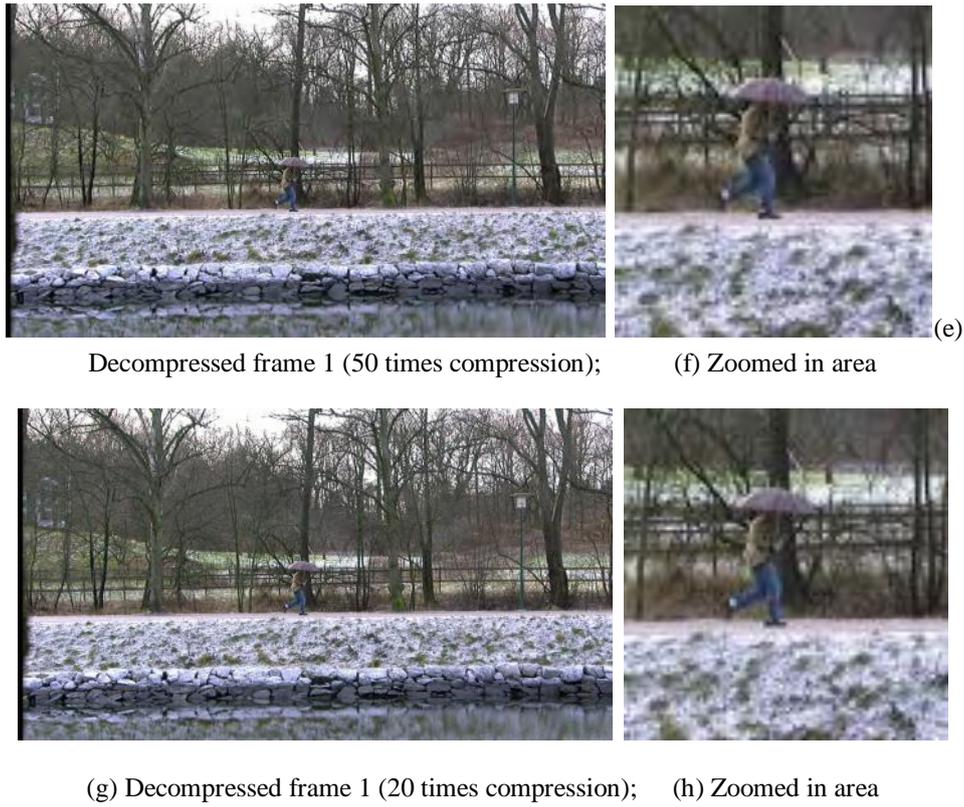


Figure 7. Original and decompressed videos using JPEG2000.

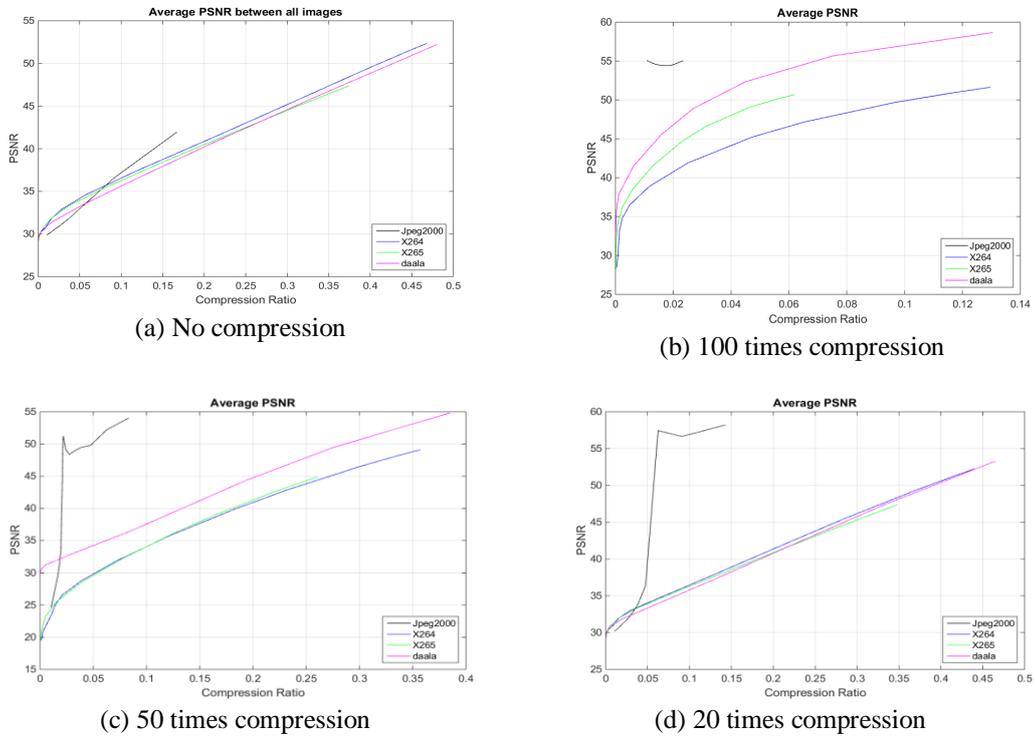


Figure 8. Performance metrics of various codecs. The input videos are decompressed videos using JPEG2000.

3.1.2. Study for Daala Codec

Here, we repeat the study for Daala. We first compressed the Running video using Daala at 100 to 1, 50 to 1, and 20 to 1 compression ratios and then decompressed it using Daala. The decompressed videos are shown in Figure 9Figure . For completeness, we also include the original video for easy comparison. We then used the decompressed videos as inputs to four codecs and generated the PSNR metrics shown in Figure 10Figure . Although we do not see a spike, we do see some exceptionally high values in the Daala plots, as compared to other codecs. From Figure 10Figure (a) for the normal case, we observe that Daala is only slightly better than JPEG2000 when we use high quality images. Here, we see that Daala gains a significant boost in metrics when we used the decompressed videos as input. For example, at 0.05 compression ratio, Daala has 32.5 dBs of PSNR in Figure 10Figure (a), but now it attains 47 dBs of PSNR in Figure 10Figure (b), 45 dBs in Figure 10Figure (c), and 38 dBs in Figure 10Figure (d).

This experiment demonstrates that if the input video to a video codec is a decompressed video using Daala, then it is highly likely that some strange behaviors may show up in the performance metrics corresponding to Daala.



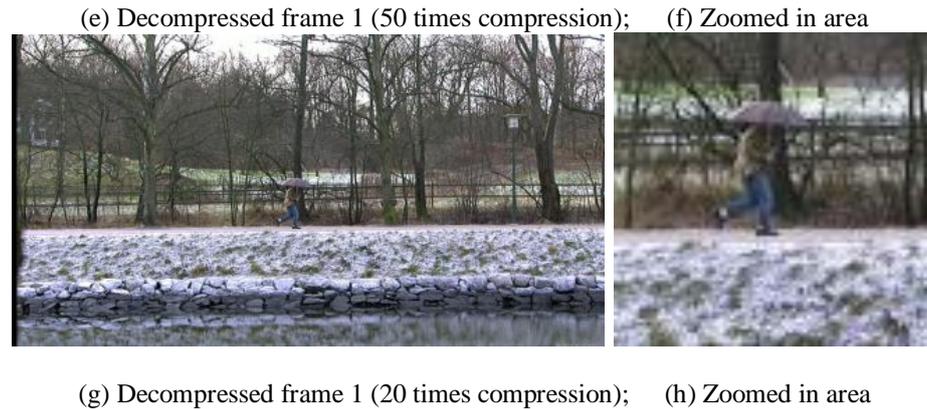


Figure 9. Original and decompressed videos using Daala.

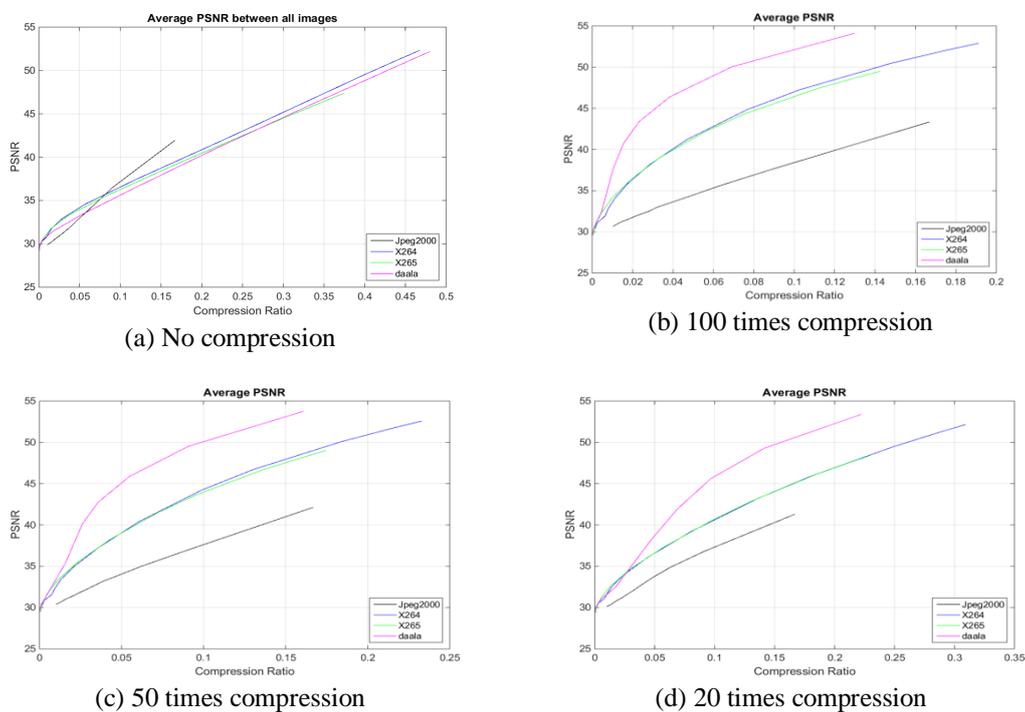


Figure 10. Performance metrics of various codecs. The input videos are decompressed videos using Daala.

3.1.3. Study for X264 Codec

Here, we investigate whether or not we can observe the same strange behavior if we use X264. The same steps were followed as the first two studies. That is, we applied X264 to compress one high quality video at 100 to 1, 50 to 1, and 20 to 1 compression ratios and decompressed them using X264. The decompressed videos, including the original video, are shown in Figure 11Figure . We then used these decompressed videos in Figure 11Figure as input to four codecs. The performance metrics of the different cases are shown in Figure 12Figure .

We can observe the same strange behavior as the earlier cases. As compared to Figure 12Figure (a) for the normal case, all the metrics associated with X264 in Figure 12Figure (b), (c), and (d) have gained an increase in metrics in terms of PSNR. X265 and Daala also gain

an increase in PSNR values likely due to the fact that they are all discrete cosine transform (DCT) based codecs.

This study shows that X264 can also have some strange behavior if the input image is not an original from the camera. In particular, if the input is a compressed and decompressed image using X264, then it is highly likely that we will see some strange behavior in the performance metrics corresponding to X264.



(a) High quality video (frame 1)



(b) Zoomed in area



(c) Decompressed frame 1 (100 times compression);



(d) Zoomed in area



(e) Decompressed frame 1 (50 times compression);



(f) Zoomed in area



(g) Decompressed frame 1 (20 times compression);



(h) Zoomed in area

Figure 11. Original and decompressed videos using X264.

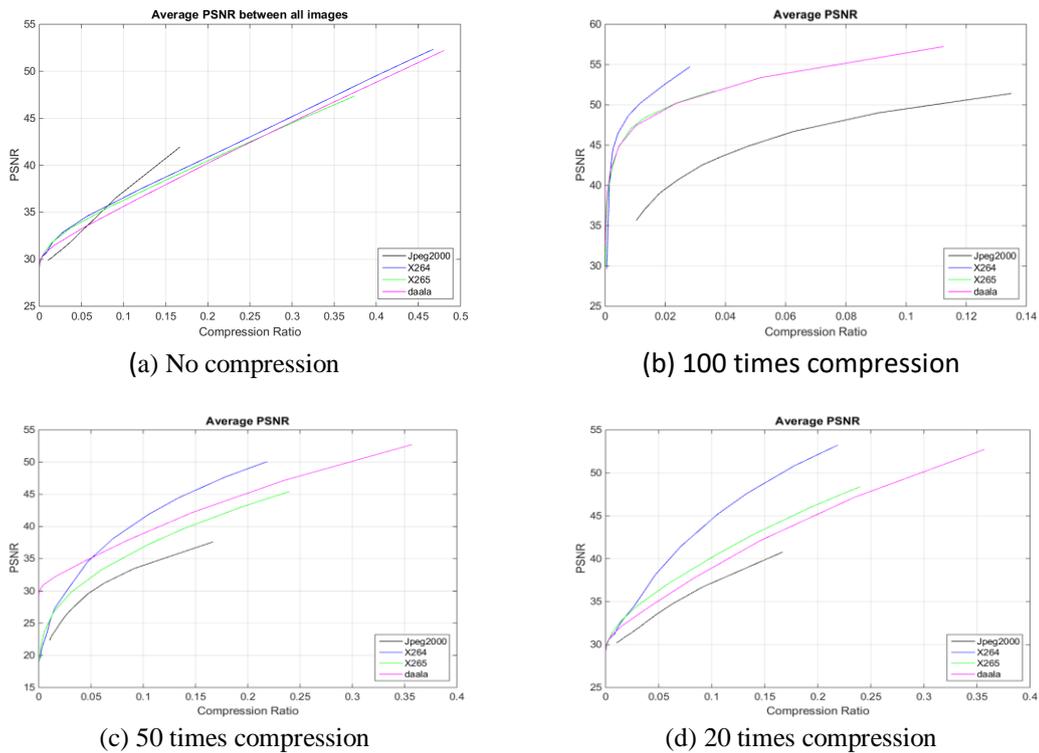


Figure 12. Performance metrics of various codecs. The input videos are decompressed videos using X264.

3.1.4. Study for X265 Codec

Finally, we investigate if the latest codec X265 also has the same behavior as the above three codecs. It turns out that the answer is positive. Here, the original Running video was compressed 100, 50, and 20 times using X265 and decompressed using X265. The decompressed videos as well as the original video are shown in Figure 13. Those decompressed videos were then used as input to four codecs.

As seen in Figure 14 (b), (c), and (d), the boost that X265 receives from this test raises itself, Daala, and X264. However, X265 gains much stronger than X264 in this test. As compared to those plots in Figure 14 (a), X265's PSNR scores in Figure 14 (b), (c), and (d) are much higher than the other methods across most compression rates.

After these case studies, we can conclude that every codec can have strange behaviors if the input video is not coming from the uncompressed raw camera videos.

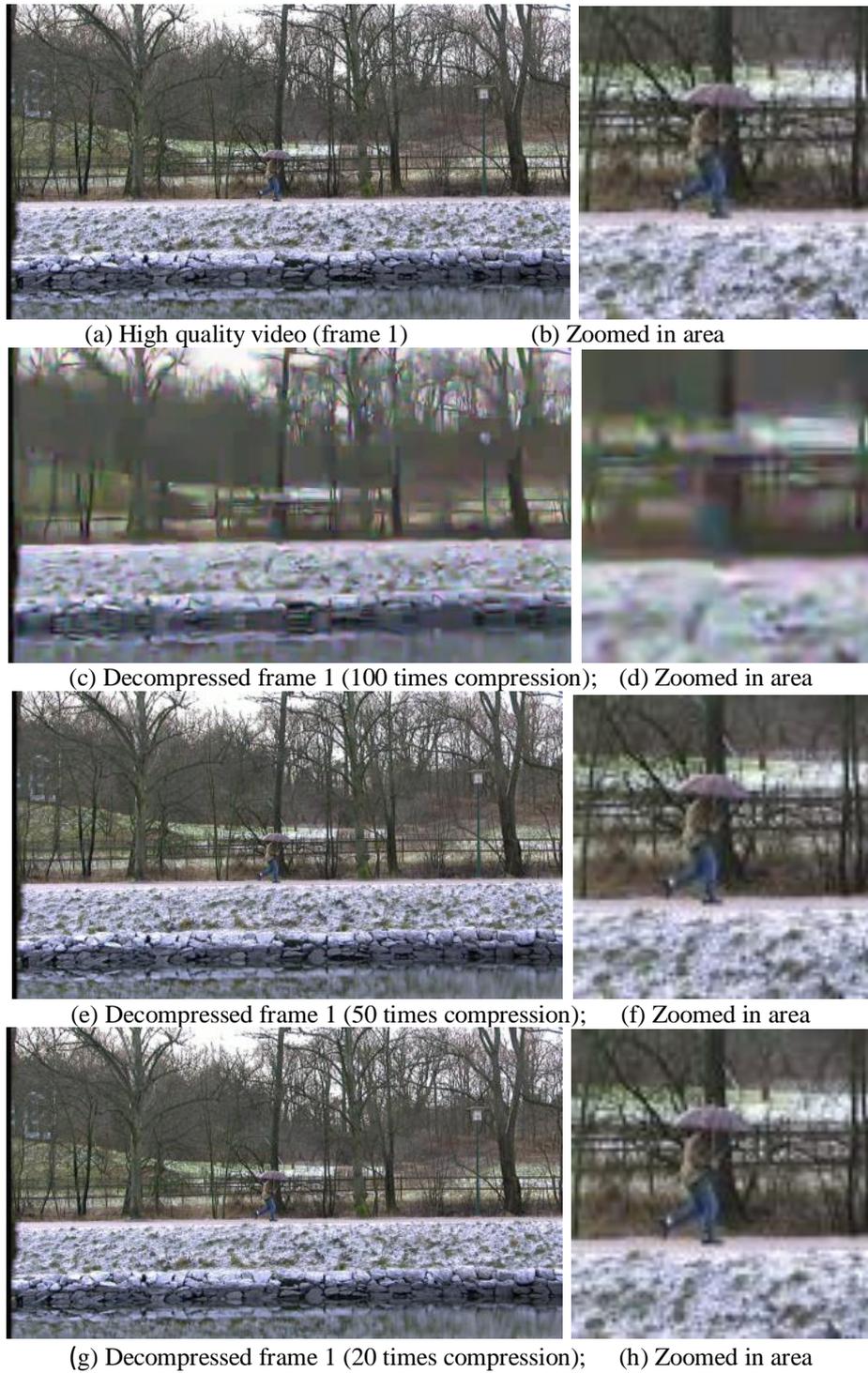


Figure 13. Original and decompressed videos using X265.

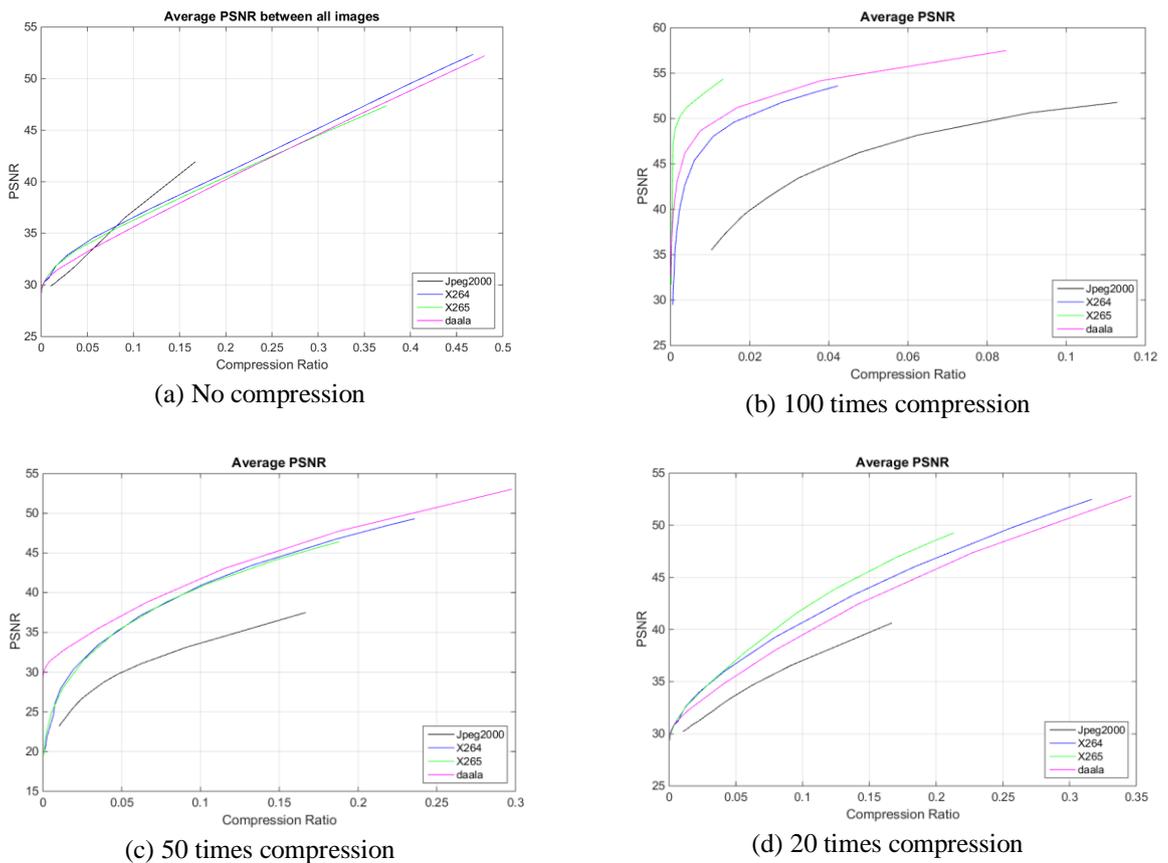


Figure 14. Performance metrics of various codecs. The input videos are decompressed videos using X265.

3.2. Running Video

To further confirm the results in Section 3.1, we include some results using another video known as the Johnny Video.

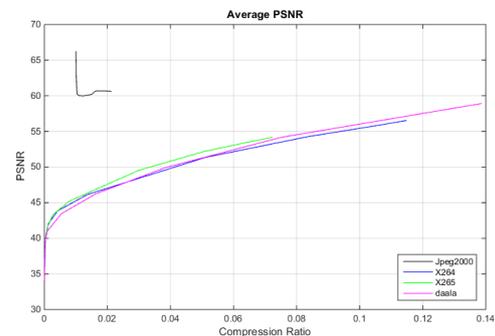
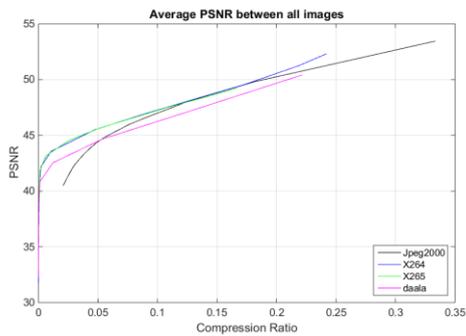
3.2.1. Study for JPEG2000 Codec

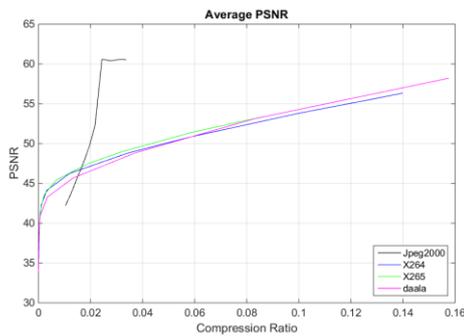
The same protocol in Section 3.1 has been used. Here, we focus on JPEG2000. First, we compressed the Johnny video by 100 and 20 times using JPEG2000 and then decompressed them. We did not include the 50 to 1 compression case because the quality difference of the 100 to 1 and 20 to 1 is almost unnoticeable. The decompressed videos are shown in Figure 15. For ease of comparison, the original video is also included. We then applied four codecs to compress and decompress those previously compressed and decompressed videos in Figure 15 and the various performance metrics are shown in Figure 16. It can be seen that JPEG2000 has abnormally high values in all metrics.

This study clearly shows that if the input video comes from a decompressed image using JPEG2000, it is highly likely that a strange behavior can also occur in performance metrics corresponding to JPEG2000.



Figure 15. Original and decompressed videos using JPEG2000.





(c) 20 times compression

Figure 16. Performance metrics of various codecs. The input videos are decompressed videos using JPEG2000.

3.2.2. Study for Daala Codec

Here, we focus on Daala and see if we still observe those strange behavior in Section 3.1.B. We first compressed the Johnny video using Daala at 100 to 1 and 20 to 1 compression ratios and then decompressed them using Daala. The decompressed videos are shown in Figure 17Figure . For completeness, we also include the original video for easy comparison. We then used the decompressed videos as inputs to four codecs and generated the performance metrics shown in Figure 18Figure . Although we do not see a spike, we do see some exceptionally high values in the Daala plots in Figure 18Figure (b), (c), as compared to other codecs.

In particular, from Figure 18Figure (a) for the normal case, we observe that Daala is only slightly better than JPEG2000 when we use high quality images. Here, we see that Daala gains a boost in metrics when we used the decompressed videos as input. For example, at 0.05 compression ratio, Daala has 44 dBs of PSNR in Figure 18Figure (a), but now it attains 52 dBs of PSNR in Figure 18Figure (b) and 48 dBs in Figure 18Figure (c).

This experiment demonstrates that if the input video to a video codec is a decompressed video using Daala, then it is highly likely that some strange behaviors may show up in the performance metrics corresponding to Daala.



(a) High quality video (frame 1)

(b) Zoomed in area.

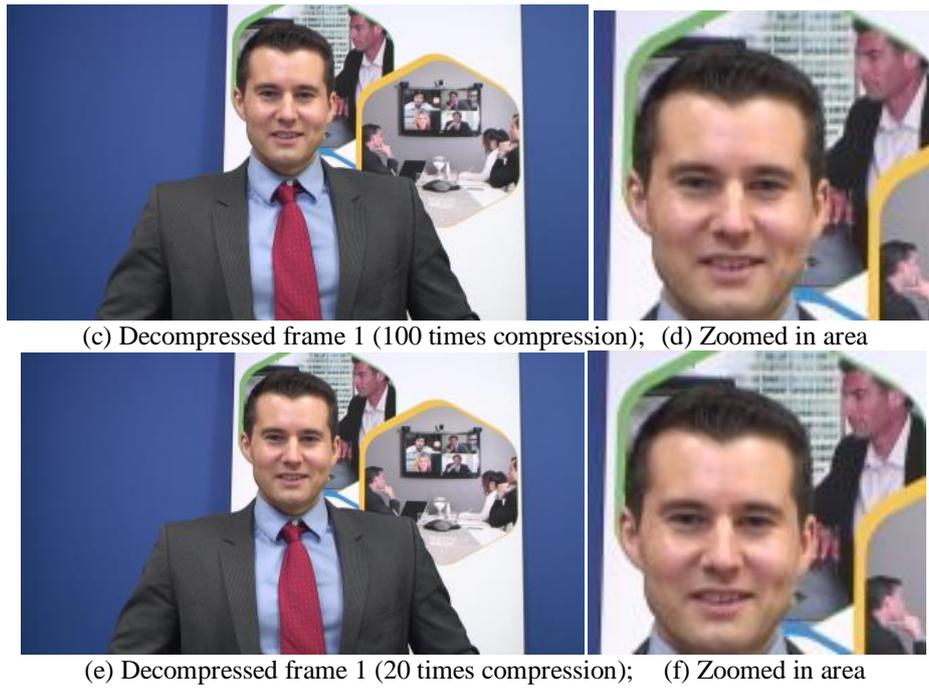


Figure 17. Original and decompressed videos using Daala.

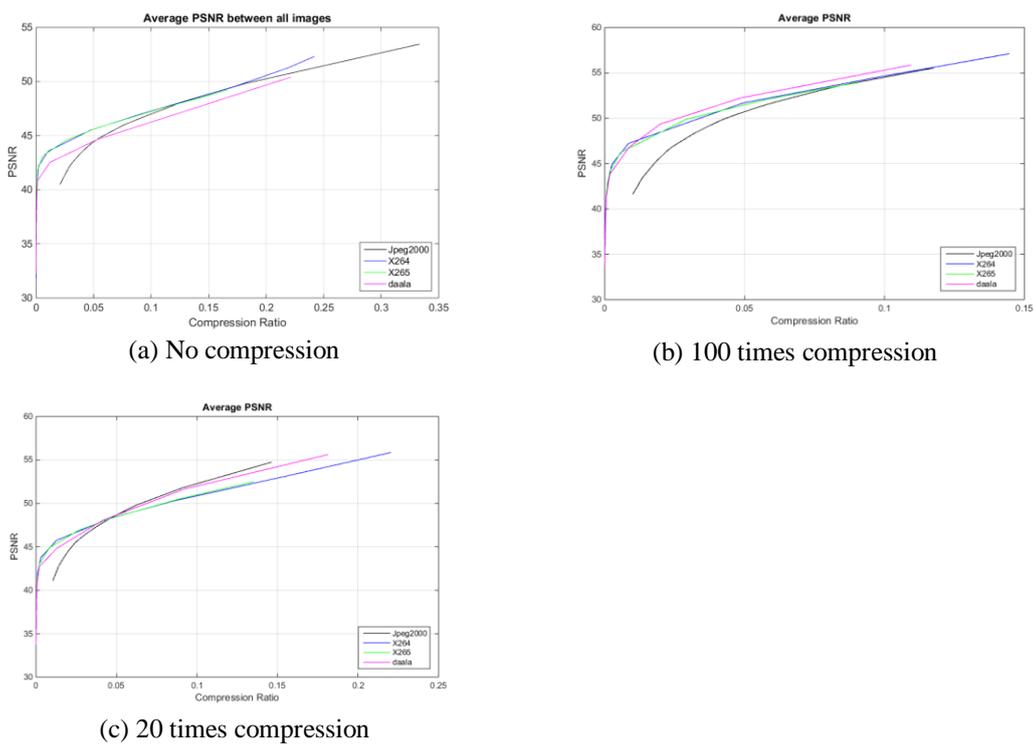


Figure 18. Performance metrics of various codecs. The input image is a decompressed image using Daala.

3.2.3. Study for X264 Codec

Here, we investigate whether or not we can observe the same strange behavior if we use X264. The same steps were followed as the first two studies. That is, we applied X264 to

compress one high quality video at 100 to 1 and 20 to 1 compression ratios and decompressed them using X264. The decompressed videos, including the original video, are shown in Figure 19. We then used these decompressed videos in Figure 19 as input to four codecs. The performance metrics of the different cases are shown in Figure 20.

We can observe the same strange behavior as the earlier cases. As compared to Figure 20 (a) for the normal case, all the metrics associated with X264 in Figure 20 (b), (c), and (d) have gained an increase in metrics in terms of PSNR. X265 and Daala also gain an increase in PSNR values likely due to the fact that they are all discrete cosine transform (DCT) based codecs.

This study shows that X264 can also have some strange behavior if the input image is not an original from the camera. In particular, if the input is a compressed and decompressed image using X264, then it is highly likely that we will see some strange behavior in the performance metrics corresponding to X264.

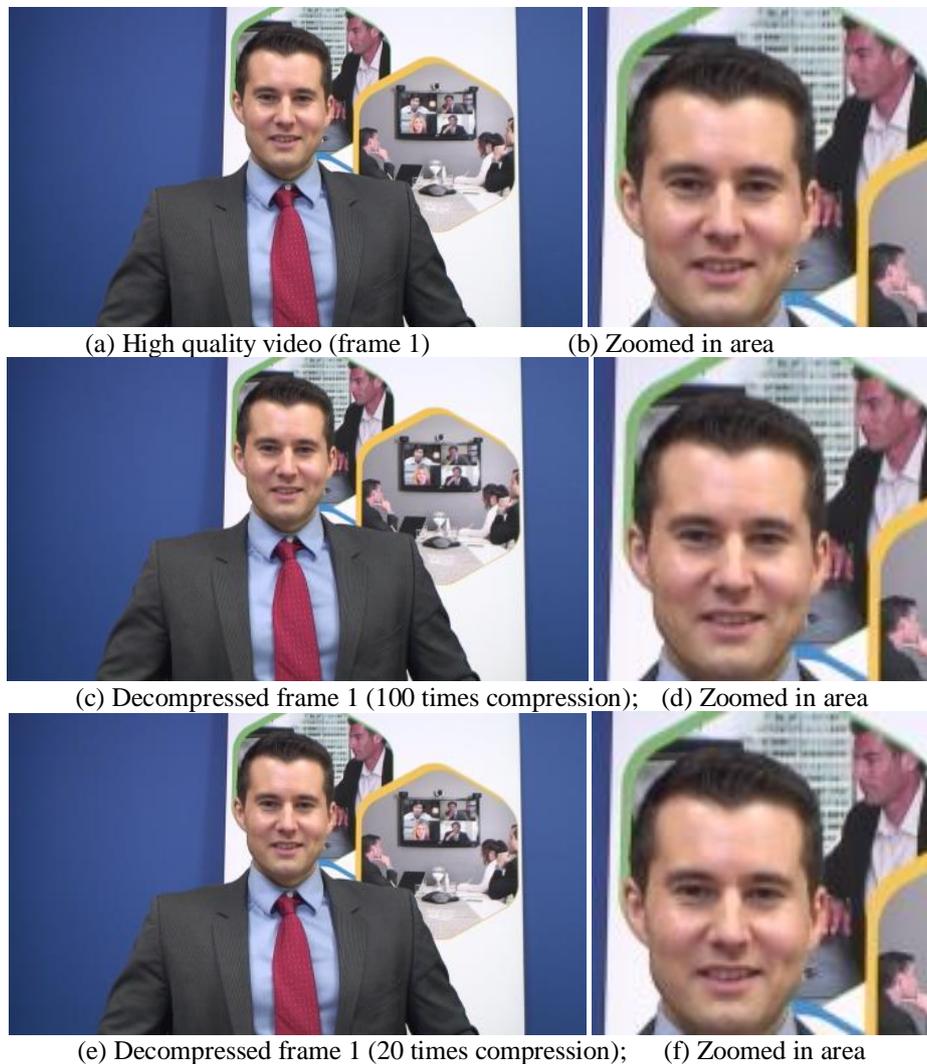


Figure 19. Original and decompressed videos using X264.

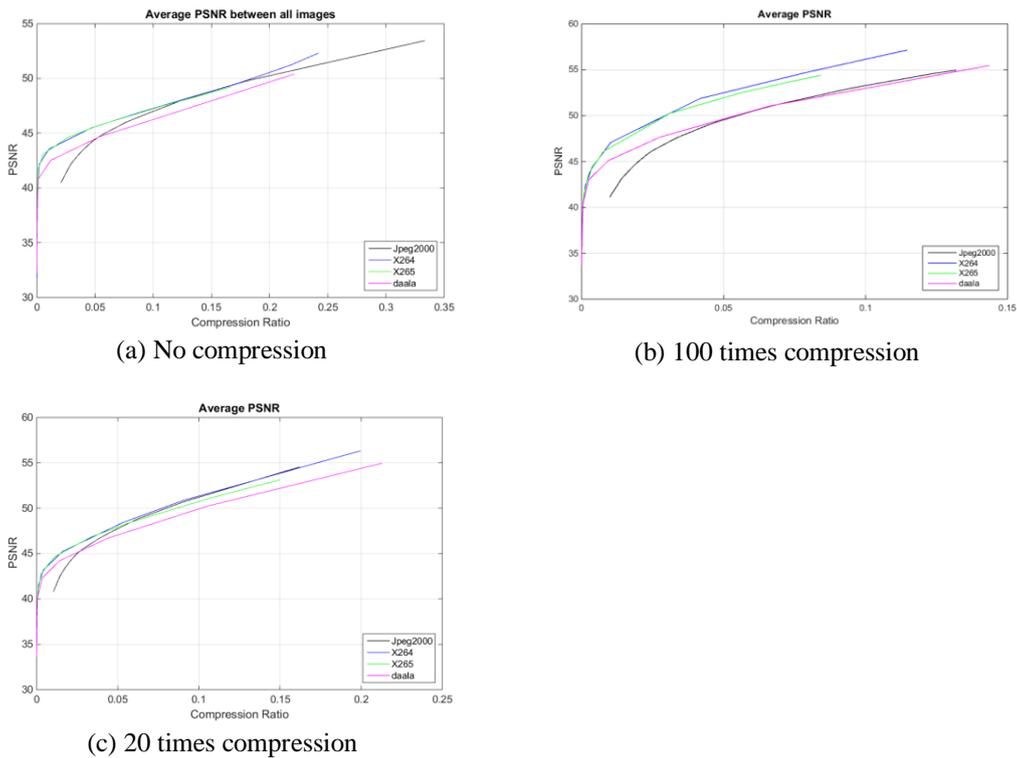


Figure 20. Performance metrics of various codecs. The input image is a decompressed image using X264.

3.2.4. Study for X265 Codec

Here, we investigate if the latest codec X265 also has the same behavior as the above three codecs. The answer is positive. Here, the original Johnny video was compressed 100 and 20 times using X265 and decompressed using X265. The decompressed videos as well as the original video are shown in Figure 21Figure . Those decompressed videos were then used as input to four codecs.

As seen in Figure 22Figure (b), (c), and (d), the boost that X265 receives from this test raises itself, Daala, and X264. However, X265 gains much stronger than X264 in this test. As compared to those plots in Figure 22Figure (a), X265’s PSNR scores in Figure 22Figure (b), (c), and (d) are much higher than the other methods across most compression rates.

After these case studies, we can conclude that every codec can have strange behaviors if the input video is not coming from the uncompressed raw camera videos.



(a) High quality video (frame 1)

(b) Zoomed in area

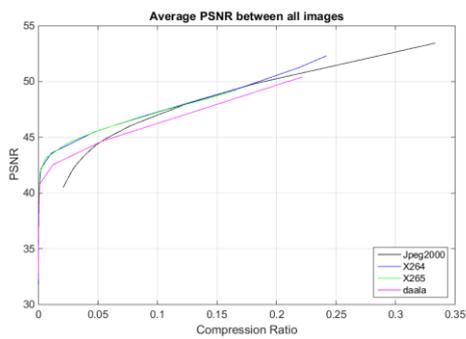


(c) Decompressed frame 1 (100 times compression); (d) Zoomed in area

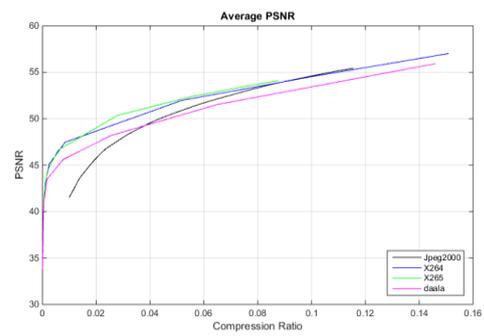


(e) Decompressed frame 1 (20 times compression); (f) Zoomed in area

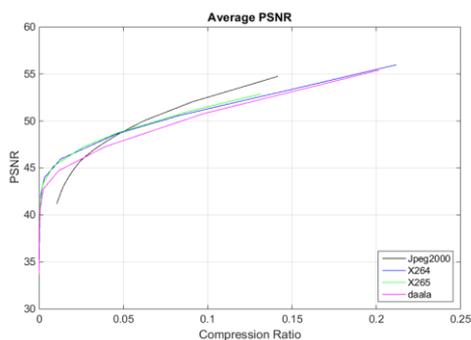
Figure 21. Original and decompressed videos using X265.



(a) No compression



(b) 100 times compression



(c) 20 times compression

Figure 22. Performance metrics of various codecs. The input image is a decompressed image using X265.

3.3. Advice to Students, Instructors and Tutors

From the above studies, we can observe that the strange behaviors in the performance metrics are caused by certain compression artifacts in the input video. Different codecs may have some unique artifacts. When the decompressed videos are compressed again at certain compression ratios, the same artifacts show up again. Since the artifacts are similar to those artifacts in the original images, the performance metrics are high. For instance, if a decompressed image from a particular codec, say Daala, is used again as an input image to the same Daala codec, then it is possible to see some strange behaviors in the performance metrics. Hence, a piece of advice should be given. If an instructor assigns some class projects to students, he needs to tell the students that the input videos to be fed into various video codecs need to be from uncompressed videos (raw camera videos). If instructors forgot to mention the above, tutors or students need to be careful about the input video quality. The highest quality videos should be used in the experiments. One way to check the quality is to zoom in to the areas with edges and see if there are any blocky or ringing artifacts. Ideally, uncompressed videos from digital camcorders should be used.

3.4. Discussions

Besides teaching, the strange behaviors can be used to detect whether a video is genuine or not. For instance, suppose a video has been compressed and decompressed by X-264 before. Now, we want to determine whether this video is genuine or not. One way to verify this is to compress it again by X-264 and other codecs. If we see some strange peaks in the performance metrics of X264 and not in others, then we can say that the given video is not genuine. Therefore, our paper can be useful for video forensic, which is to determine whether a video is genuine or not.

4. CONCLUSIONS

In this paper, we first report some puzzling observations that students in video compression courses may encounter in their class projects. That is, if students use some low quality videos that may have been compressed and decompressed before in their compression experiments, then they may observe some unexpected high performance metrics in some codecs. We then use experiments to demonstrate that such anomalous behavior can happen in any codec. The recommendation to instructors, tutors, and students is that one should use high quality videos that directly come out of the digital camcorders when one tries to compare the performance of different codecs. The observations in this paper can also provide some guidance to video forensics.

ACKNOWLEDGEMENTS

This research was supported in part by NASA Jet Propulsion Laboratory under contract 80NSSC17C0035. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of NASA or the U.S. Government.

REFERENCES

- [1] C. Kwan, B. Chou, J. Yang, A. Rangamani, T. Tran, J. Zhang, and R. Etienne-Cummings, "Target tracking and classification directly using compressive sensing camera for SWIR videos," *Journal of Signal, Image, and Video Processing*, June 7, 2019.

- [2] C. Kwan, B. Chou, J. Yang, A. Rangamani, T. Tran, J. Zhang, and R. Etienne-Cummings, "Target Tracking and Classification Using Compressive Measurements of MWIR and LWIR Coded Aperture Cameras," *Journal Signal and Information Processing*, vol. 10, no. 3, August 8, 2019.
- [3] Kwan, C., Chou, B. and Kwan, L. M., (2018) "A Comparative Study of Conventional and Deep Learning Target Tracking Algorithms for Low Quality Videos," *15th International Symposium on Neural Networks*.
- [4] Kwan, C., Chou, B., Yang, J. and Tran, T., (2019) "Compressive object tracking and classification using deep learning for infrared videos," Proc. SPIE 10995, *Pattern Recognition and Tracking XXX (Conference SH20)*.
- [5] C. Kwan, B. Chou, J. Yang, and T. Tran, "Target tracking and classification directly in compressive measurement for low quality videos," SPIE 10995, *Pattern Recognition and Tracking XXX*, 1099505, 13 May 2019.
- [6] Kwan, C., Chou, B., Echavarren, A., Budavari, B., Li, J. and Tran, T., (2018) "Compressive vehicle tracking using deep learning," *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*.
- [7] Kwan, C., Gribben, D. and Tran, T. (2019) "Multiple Human Objects Tracking and Classification Directly in Compressive Measurement Domain for Long Range Infrared Videos," *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York City.
- [8] Kwan, C., Gribben, D. and Tran, T. (2019) "Tracking and Classification of Multiple Human Objects Directly in Compressive Measurement Domain for Low Quality Optical Videos," *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York City.
- [9] C. Kwan, B. Chou, J. Yang, and T. Tran, "Deep Learning based Target Tracking and Classification Directly in Compressive Measurement for Low Quality Videos," *Signal & Image Processing: An International Journal (SIPIJ)*, 11/16/2019.
- [10] C. Kwan, B. Chou, J. Yang, and T. Tran, "Deep Learning based Target Tracking and Classification for Infrared Videos Using Compressive Measurements," *Journal Signal and Information Processing*, 11/1/2019.
- [11] C. Kwan, B. Chou, J. Yang, A. Rangamani, T. Tran, J. Zhang, and R. Etienne-Cummings, "Deep Learning based Target Tracking and Classification for Low Quality Videos Using Coded Aperture Camera," *Sensors*, 19(17), 3702, August 26, 2019.
- [12] J. Zhou and C. Kwan, "Anomaly Detection in Low Quality Traffic Monitoring Videos Using Optical Flow," Proc. SPIE 10649, *Pattern Recognition and Tracking XXIX*, 106490F, 30 April 2018.
- [13] C. Kwan, J. Zhou, Z. Wang, and B. Li, "Efficient Anomaly Detection Algorithms for Summarizing Low Quality Videos," Proc. SPIE 10649, *Pattern Recognition and Tracking XXIX*, 1064906, 27 April 2018.
- [14] C. Kwan, J. Yin, and J. Zhou, "The Development of a Video Browsing and Video Summary Review Tool," Proc. SPIE 10649, *Pattern Recognition and Tracking XXIX*, 1064907, 27 April 2018.
- [15] C. Kwan, J. Larkin, B. Budavari, E. Shang, and T. Tran, "Perceptually Lossless Compression with Error Concealment for Periscope and Sonar Videos," *Signal & Image Processing: An International Journal (SIPIJ)*, April 30, 2019.
- [16] C. Kwan, B. Li, R. Xu, T. Tran, and T. Nguyen, "Very low-bit-rate video compression using wavelets," *Wavelet Applications VIII, Proc. SPIE (vol. 4391)*, pp.176-180, 2001.
- [17] C. Kwan, E. Shi, and Y. Um, "High performance video codec with error concealment," *Data Compression Conference, Snowbird, Utah, March 27-30, 2018*.
- [18] C. Kwan and J. Larkin, "Perceptually Lossless Compression for Mastcam Images," *IEEE Ubiquitous Computing, Electronics & Mobile Communication Conference*, New York City, November 2018.
- [19] G. Strang and T. Nguyen, *Wavelets and filter banks*, Wellesley-Cambridge Press, 1997.
- [20] J. Zhou and C. Kwan, "A Hybrid Approach for Wind Tunnel Data Compression," *Data Compression Conference, Snowbird, Utah, March 27-30, 2018*.
- [21] C. Kwan and Y. Luk, "Hybrid sensor network data compression with error resiliency," *Data Compression Conference, Snowbird, Utah, March 27-30, 2018*.
- [22] JPEG-2000, <https://jpeg.org/jpeg2000/>, accessed April 7, 2020.
- [23] X264, <http://www.videolan.org/developers/x264.html>, accessed April 7, 2020.
- [24] X265, <https://www.videolan.org/developers/x265.html>, accessed April 7, 2020.

- [25] Daala, <http://xiph.org/daala/>, accessed April 7, 2020.
- [26] C. Kwan, B. Li, R. Xu, X. Li, T. Tran, and T. Q. Nguyen, "A complete image compression codec based on overlapped block transform," *Eurosip Journal of Applied Signal Processing*, pp. 1-15, January 2006.
- [27] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre-and post-filtering," *IEEE Transactions on Signal Processing*, 2003.
- [28] C. Kwan, J. Larkin, B. Budavari, and T. D. Tran, "A short review of well-known image codecs and observations and root cause of some strange behaviors in image compression codecs," *Journal of Signal, Image, and Video Processing*, October 28, 2019.
- [29] Test videos, <https://media.xiph.org/video/derf/>

AUTHORS

Chiman Kwan received his BS with honors in Electronics from the Chinese University of Hong Kong in 1988, and MS and Ph.D. in Electrical Engineering from the University of Texas at Arlington in 1989 and 1993, respectively. Currently, he is the President of Applied Research LLC, leading research and development efforts in image and video processing, and remote sensing. He has 15 patents, 65 invention disclosures, 360 papers in journals and conferences, and over 550 proprietary technical reports. Over the past two decades, he has secured more than 120 diverse research projects totaling more than 36 million dollars from various government agencies and private companies. He also received numerous awards and recognitions from IEEE, NASA, and other agencies

Jude Larkin graduated in Computer Science from the Franciscan University of Steubenville, Ohio, USA. In the past three years, he has worked on image demosaicing, image super-resolution, pansharpening, image fusion, change detection using satellite images, and contingency planning for UAVs.