COMP590 Assignment 2 Write-Up

Garrett Bolter: 730091835

1. For my arithmetic encoding compression technique, I first downloaded and watched the .mp4 file to conceptualize any strategic manipulations I could make on the data from visual inference. Noticing a relatively even coloration with minor 0 or 255 pixel values (black and white), I decided to take a unique approach with some of the outlined considerations into account. Pixel value variations were relatively consistent from the beginning to end (I perceived, at least), so from frame to frame I did not anticipate much variation either. So my initial instinct that I followed thru with until the end of my encoding procedure was to take the initial pixel values of the first frame and relate them to the rest of the 299 frames under speculation. From this comparison, the difference between encoded values was retained to a relatively small range that spanned from 0 to 511, since two values have the possibility to reach either end of a 256 – 256 or 256 + 256 spectrum. Utilizing the console and print checks, I produced a nice bell-curve in the relative frequency mapping, so I knew my method was tangible.
2. From conception of my method, I considered accurate pixel prediction in subsequent frames after first viewing the .mp4 video. Staying in one focus for the majority of the 10 second video, the remaining time was minimal in my mentality relative to the compression achievable. Although this concept aligns closest with temporal coherence, it still considers spatial coherence as the pixel values should remain closely related from minor shifts as the frames pass.
3. From the github files, I tested the .dat file against each of the AdaptiveACEncode, ContextAdaptiveACEncode and the StaticACEncode encoders. From their respective compressed file sizes, the ContextAdaptiveACEncode encoder produced the best results by a significant margin when compared to the AdaptiveACEncode and StaticACEncode encoder’s results. Between the former of these results, the AdaptiveACEncode encoder was a slightly more efficient method by a compression difference of about 1,000 bytes compared to the StaticACEncoder. As for my PixelEncode encoder, the relative compression was considerably worse than the three aforementioned algorithms. Unfortunately, the compression achieved would only be desirable if one had millions of files and thousands of hours to allow the compression of all the files to complete, and even then the results would be a small fraction of data. I do believe, however, that there is a niche that my method could exploit such as very long still landscape videos where an occasional bird may fly by with minimal amounts of wind in the air to move the pixels of trees in the far far distance. I realized after compiling my results that my method may have been at a disadvantage in this scenario in that the video is very short. In utilizing a large overhead relative to the length and respective size of the file under compression, it quite possibly could achieve desirable results at some increase in file length.
4. If I went back into my code in aim to optimize and generate a better compression rate, I would most definitely incorporate a frame’s frequency value to its prior frame’s pixel value. Also, I would try to determine a way in which the pixel speculation would incorporate a surrounding pixel buffer and not just the one itself.