Comp 590 Assignment 2 Write-Up

Scott Hofbauer

02/25/19

1. **What scheme or schemes did you try? If you came up your own idea, describe it here.**

The scheme I used was the differential coding scheme. The way I approached this scheme was to create an array of size 512 representing the possible differences of -255 – 255. The array had to range from -255 to 255 in order to handle the worst case if there was a white pixel next to a black pixel and vice versa. Next I created a second array that contained the counts for the differences of the pixels. This array would contain the amount of times a difference of 0 or -7 etc. would occur throughout the video. I observed that the majority of the differences were close to zero with a maximum difference of -161 (worst case scenario never happened). After creating the symbol array for the differences and getting the counts for each difference, I passed those arrays to the FreqCountIntegerSymbolModel to create the model. The work done before that was part of the “preprocessing step” described in the assignment write-up. Finally, I created an encoder with the default range bit width of 40 and wrote into the output file, the 512 difference frequency counts, the first byte of the input file, and the number of symbols. This would represent the header of my compressed file. All this information is necessary for the decoding step. Then I began encoding the file using the encoder and model previously described. After encoding, my compressed file was 915,200 bytes versus the original file of 1,228,880 bytes resulting in the savings of 313,600 bytes.

1. **Why do you think your scheme would do a good job predicting pixel values? How does your scheme exploit temporal and/or spatial coherence?**

This scheme took advantage of Spatial Coherence pattern, where pixel intensity is likely to be very similar to the intensity of the nearby pixels. This was exploited by finding the differences between the values and seeing that for the most part there is relatively small difference in pixel intensity. We can see that we exploit spatial coherence when the counts for differences, the worst-case scenario -255 or 255 never happens, thus by giving the counts of 0 to more symbols, this allows the encoder to emit less bits because it doesn’t allocate any space for the differences that never occur such as -255 and 255.

1. **When applying the English text-based models (static, adaptive, and context-adaptive) to the video data, which scheme performed best? Does the scheme you developed compress better or worse than the English text-based models when applied to video data? If you weren't able to finish and test your own scheme, how do you think your scheme would fare in comparison to the English text-based models?**

When applying the English text-based models, I saw that the context-adaptive was the best by achieving a compressed file of 909KB. The static and adaptive were not very good at compressing the video file. My scheme came very close the context-adaptive model of 915KB so only 6,000 bytes larger. I would consider this relatively good compression for a video file. My scheme proved to be far better than the static or adaptive models.

1. **What is one change you could make to your scheme that might improve its results?**

One change I could make to my scheme is to keep the same differential coding preprocessing step but instead of having just one model that represented all the frequency counts. I could go along the lines of the context-adaptive models and create a model for each difference. I think this might help me achieve better compression.