Kevin Young
COMP 590
4/28/2019
(late submission)
collaborators: rdarden1, sumit97

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

Ended up using the prior value scheme because I talked a little bit about temporal coherence in a film theory class, granted that was unrelated. Attempted the average-neighbor but quickly scrapped that attempt as average coherence seemed like it involved more calculations in my algorithm; also, in the context of video, temporal coherence seems like it makes more sense, but I realize that's a bold assumption.

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

Frames of a video move in such a way that they give off the perception of motion. The "irrational cut" between said frames is the space between, and their coherence is why information can be carried from one frame to the next and look fluid. In my algorithm I tried to implement an index that would referer back to a prior intensity value and used that as its prediction model, i.e. prior value.

3.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?

Adaptive test: 1063224bytes

Context Adaptive test:909144bytes

Static test: 1064024bytes

Prior Value Context Adaptive:1063224bytes

Ended up being exactly the same as the adaptive model, performing worse than the Context adaptive model. I would say this performed worse and that the algorithm I implemented was not successful, regrettably

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

Mostly concerned with my algorithm. Right now I'm in algorithms and analyses and hopefully I'd be able to improve the run time of said algorithm. A nested for loop means that my current algorithm's run time is O(n^2). My additional "check" for loop is in parallel and shouldn't be affecting the run time. Compared to the Context Adaptive run time, O(n), also had the best compression, so I would prioritize improving the run-time of my algorithm.