Compression presentation

Lossless/lossy prediction

Lorenzo Predictor
- N-Dimensional predictor
- Add two adjacent vertices, subtract opposite vertex
- 1st row is 0 order scheme
- 2nd row and on use 3 neighbors

A prediction is actually a transform
- In the case of a worst-case scenario we can say that the value only goes from 0-255 meaning if you go below 0, then remap to 256+correction (loop around)
- This allows for better entropy
- You want the highest Huffman leaf to have the highest probability
- You want to change the distribution of frequencies

Lorenzo Predictor Again
- 200 MB, you only need 16KB as working memory for the compression (no need to store entire data set in memory
- You can't obtain a random point without its predecessors
- Solution: Hierarchical schemes

Bi-Lorenzian
- Increase the number of points
- Have a higher degree prediction
- Now you need a 3x3 array of neighbor points
- Make Lorenz predictions on all the neighbor points
- Find the Lorenz predictions of the Lorenz predictions
- $\text{LPR}(C) = B + D - E$
- $\text{BLPR}(C) = \text{LPR}(C) + \text{LPR}(B) + \text{LPR}(D) - \text{LPR}(E)$
- $\text{KB} = B - A - E + F$, $\text{KE} = E - F - H + G$