Image Encoding

Scott Carda, Benjamin Kaiser, Christopher Smith

Overview

- What’s the point?
- Huffman Encoding
- Predictive Encoding
- Run Length Encoding

What’s the point?

- What is it?
  - Process of reducing the amount of data required to represent information
- Uses
  - Saves storage space
  - Network transmission

Huffman Encoding

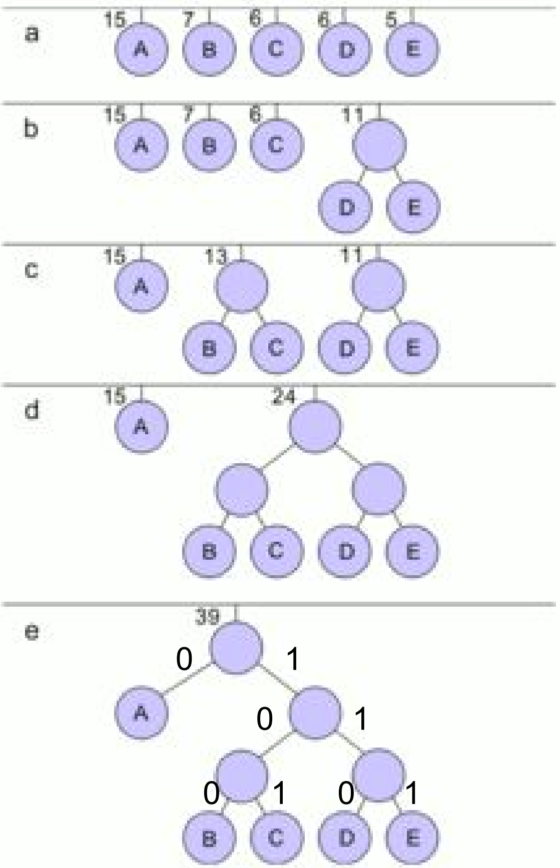
- Data is Chunked into ‘Symbols’
- Lossless, Variable-Length Encoding
- Uses Statistics to Improve Size of Compression
- More Frequently Used Symbols are Given Smaller Codes
- Uses a Binary Tree Data Structure to Determine Encoding

Huffman Encoding Cont.

- First Pass: Get Image Histogram
- Create Tree with the Histogram
- Encode Huffman Tree to Compressed File
- Second Pass: Encode all symbols by their Path in the Huffman Tree

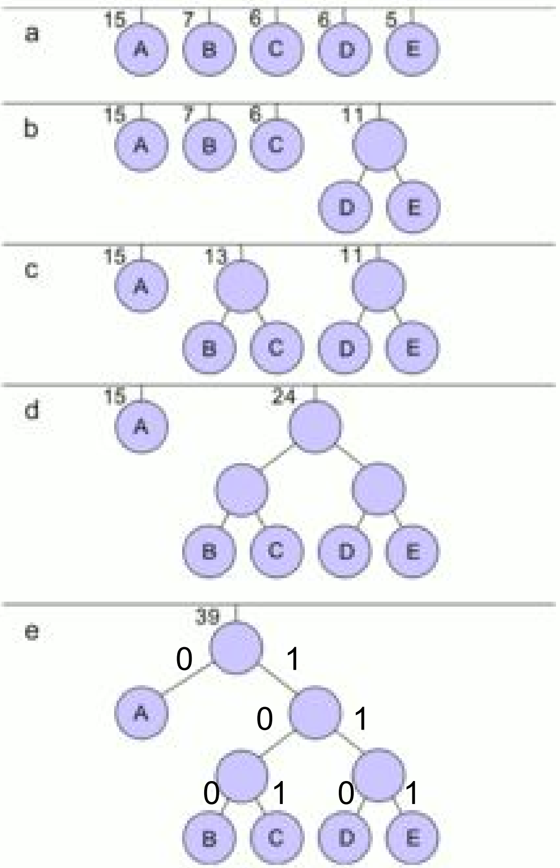
Creating the Huffman Tree

1. Create List of Nodes out of Frequency-Symbol Pairs, Sorted by Frequency
2. Pop Least Frequent Two Nodes
3. Create New Node with Two Popped Nodes as Children
4. New Node's Frequency is Sum of Children's Frequencies



Creating the Huffman Tree

5. Insert New Node back into List of Nodes in Sorted Order
6. Repeat From Step 2, until only One Node Remains, this is Root Node
7. Assign ‘0’ to left edges, ‘1’ to right edges

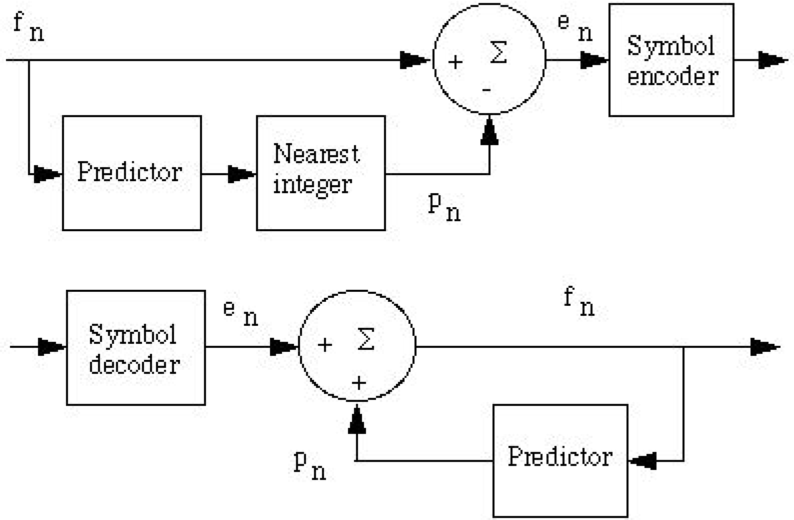


Predictive Encoding

- Can be lossless and lossy
- Good compression
- Eliminates redundancies of closely spaced pixels
- Difference between the actual and predicted value of the pixel

Lossless Predictive Coding

- Encoder
  - Predictor generates anticipated value
  - Predictor output is rounded
- Decoder
  - Uses same predictor as encoder



Previous Pixel

4	6	8
5	0	2

→

4	2	8
5	0	2

...

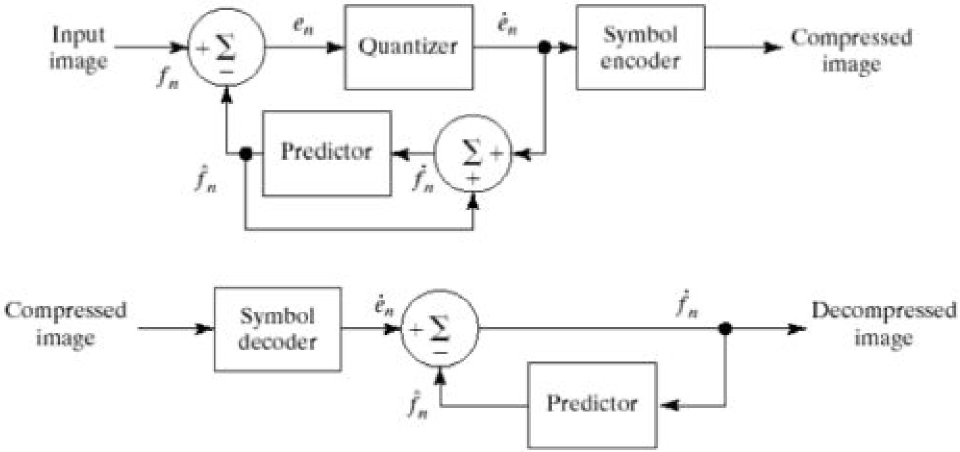
4	2	6
5	-5	2

→

4	2	6
5	-5	-3

Lossy Predictive Coding

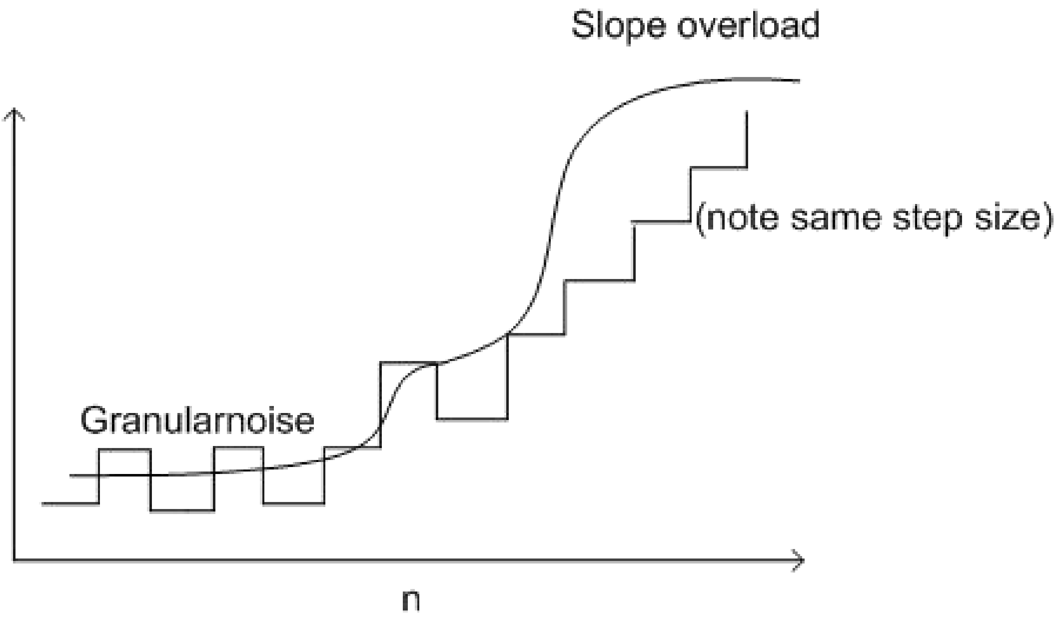
- Encoding
  - Quantization step added
    - Maps prediction error into limited range
  - Predictor
    - Uses past predictions
- Decoding
  - Unchanged



Delta Modulation

- Predictor
  - Same as Previous Pixel
- Quantizer
  - If e(n) >= 0
    - Positive delta is used
  - If e(n) < 0
    - Negative delta is used

Code Rate: 1 bit/pixel



Delta Modulation Results

Original

Delta of 4.3

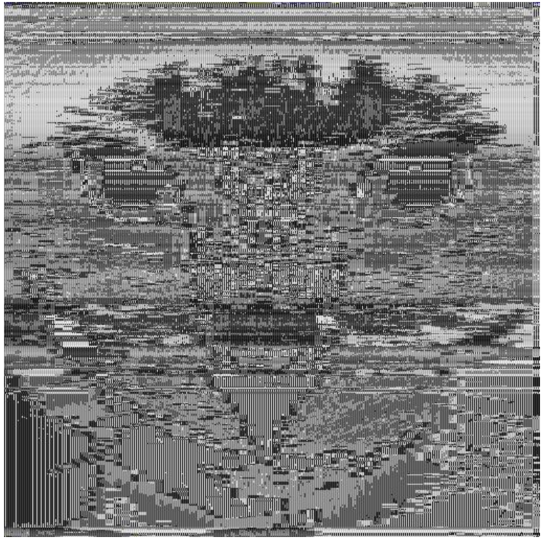
Delta of 1

Delta Modulation Results

Original

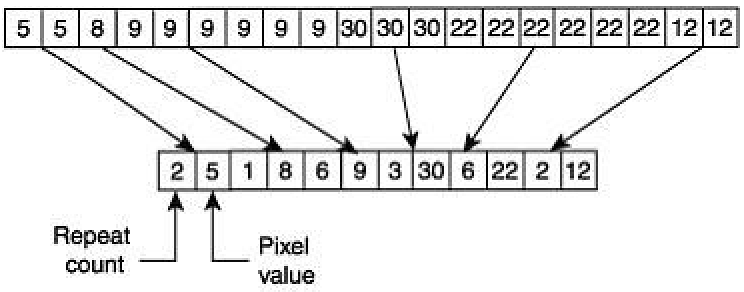
Delta of 4.3

Delta of 50



Run-Length Encoding

- Creates “runs” of horizontal pixel intensities
- Can be lossy or lossless



https://www.mathworks.com/matlabcentral/fileexchange/31123-rle-run-length-encoding

Lossless - Bit Plane

```
For each bit plane
    Set starting state as bit set or not set
    Write starting state
    For each pixel in image
        If bit state != starting state
            Starting state becomes bit state
            Write count
            Reset count to 0
        Increment count
```

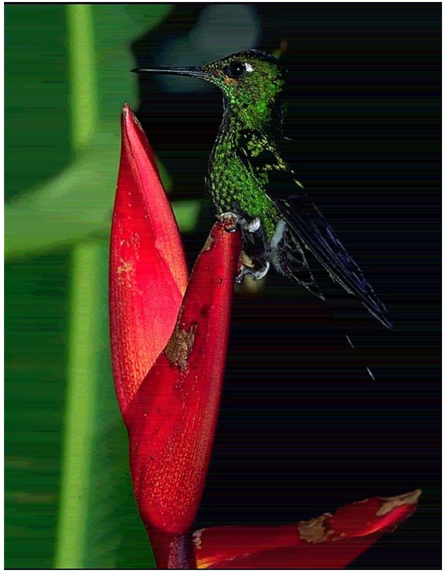
Lossy - Range of Intensities

```
Set base intensity to first intensity in image
    For current pixel in image
        If current pixel intensity outside of +/- range of base intensity
            Write count and base intensity
            Set base intensity to current pixel intensity
            Reset count to 0
        Increment count
    Write count and base intensity
```

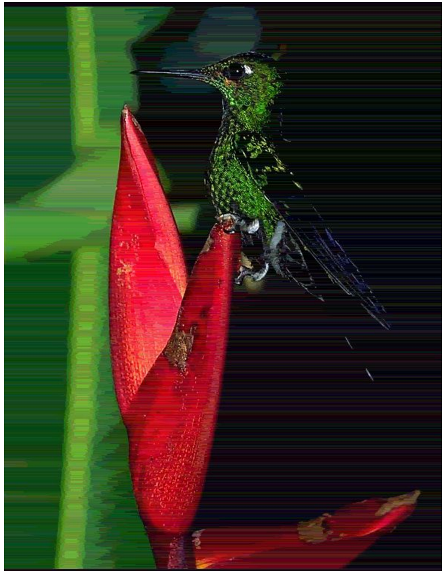
Lossy - Range of Intensities

Lossy - Range of Intensities

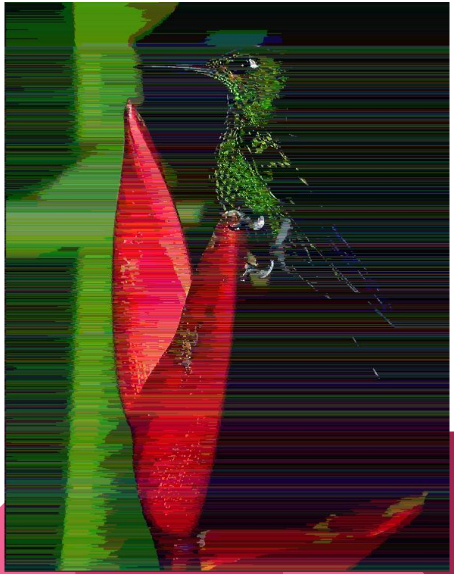
16 level intensity



32 level intensity



64 level intensity



Statistics and Analysis

- Compression
  - Lossy Compresses on average between 40% and 60% while retaining quality
    - Binary images can get as much as 93% compression
    - Best case (all white image): 99%
  - Lossless on average actually increases image size
    - Binary images and those with little to no variation work the best
    - Best case 93%
- Percent Error
  - Lossy: Less than 10% for +/- 4 range
  - Lossless: Not Applicable

Applications

- Fax machines
  - Primarily binary images
- Truevision TGA (TARGA)
  - Simple icons etc
  - Commonly used in early 2000’s video games

Questions?

Resources

http://www.fileformat.info/mirror/egff/ch09_03.htm

https://en.wikipedia.org/wiki/Quadtree

https://upload.wikimedia.org/wikipedia/commons/thumb/d/d8/HuffmanCodeAlg.png/220px-HuffmanCodeAlg.png

https://en.wikipedia.org/wiki/Truevision_TGA

https://www.mathworks.com/matlabcentral/fileexchange/31123-rle-run-length-encoding