- Snarf the huff project
  - I am assuming that most of you have already done this

- Start reading/re-reading the Huff assignment
  - I am assuming that most of you have already done this

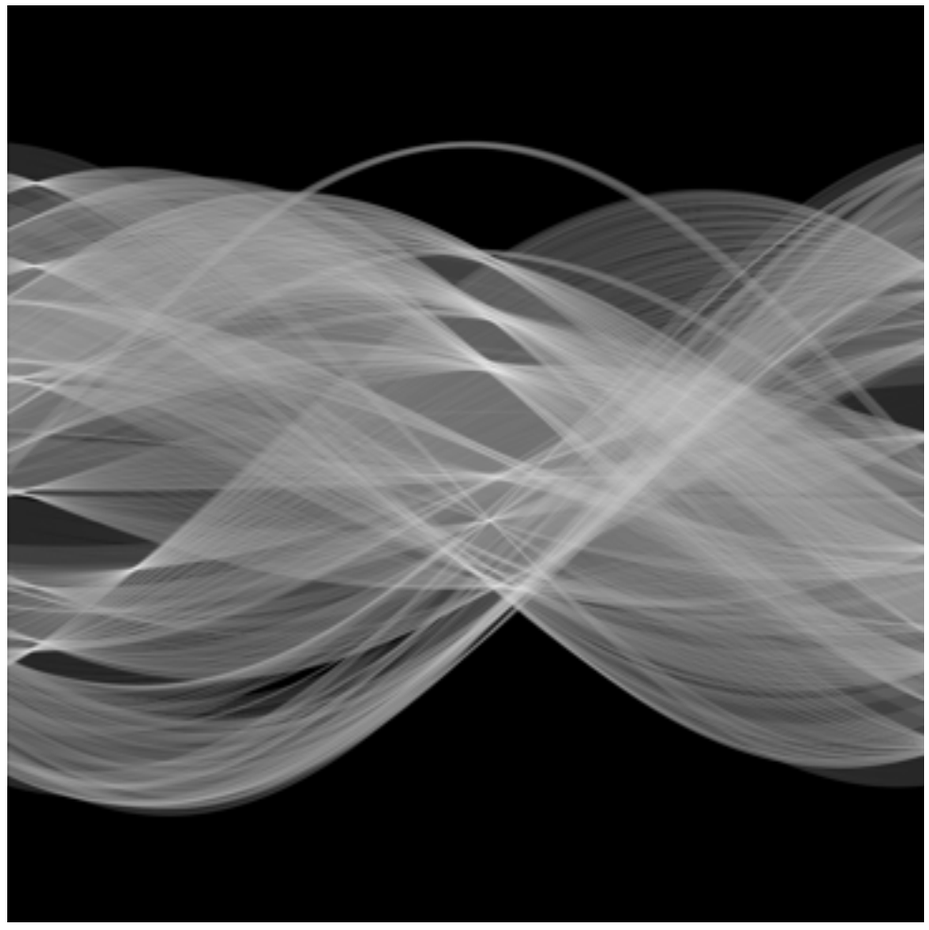
Today

- Practice building a Huffman tree
- Develop skills for the Huff assignment

- By the end of class you should:
  - Know how to make a Huffman tree
  - Have a simple example that can help test your assignment
  - Have code for reading in data from

Side Note

- Huffman (Huff) ≠ Hough
- Hough Transform - edge detection in images



Huffman Assignment

- Step 1: Compress a file
- Step 2: Uncompress a file
- Step 3: Profit

4

Huffman Assignment

- Compress

1. Read a file and count occurrences for each character
2. Build Huffman tree from counts
3. Use tree to construct a map from character -> Huffman code
4. Output the compressed file using codes from step 3

Huffman Assignment

- A compressed file
  - magic number - info on how to decode header
  - header - info on how to decode data
  - data

Huffman Assignment

- Uncompress

1. Check file is well formed (magic number)
2. Read header (counts of all characters including PSEUDO_EOF)
3. Build Huffman tree from header
4. Use tree to construct a map from character -> Huffman code
5. Output the uncompressed file using codes from step 4

Huffman Assignment

- Compress

1. Read a file and count occurrences for each character
2. Build Huffman tree from counts
3. Use tree to construct a map from character -> Huffman code
4. Output the compressed file using codes from step 3

Huffman Assignment

- Compress

1. Read a file and count occurrences for each character
2. Build Huffman tree from counts
3. Use tree to construct a map from character -> Huffman code
4. Output the compressed file using codes from step 3

Build Tree

- Go to the recitation webpage
- Complete the questionnaire

- We will start this together

Huffman Assignment

- Compress

1. Read a file and count occurrences for each character

2. Build Huffman tree from counts

3. Use tree to construct a map from character -> Huffman code

4. Output the compressed file using codes from step 3

Huffman Assignment

- Compress

1. Read a file and count occurrences for each character

  - “go go gophers”

- Let’s write some code!

Today

- Practice building a Huffman tree
- Develop skills for the Huff assignment

- By the end of class you should:
  - Know how to make a Huffman tree
  - Have a simple example that can help test your assignment
  - Have code for reading in data from