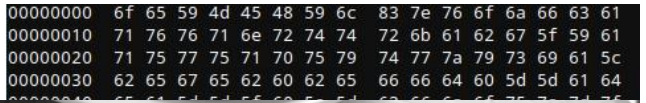
Lossless Image Compression

- using Adaptive Huffman Coding

Marcus Windmark, Oct 23 -14
Data Compression, NTNU

Outline of the Project

- Lossless image compression



- 8 bit grayscale images

Choice of Algorithm(1)

“Huffman code is one of the fundamental ideas that people in computer science and data communications are using all the time”

- Donald E. Knuth, author and professor

Choice of Algorithm(2)

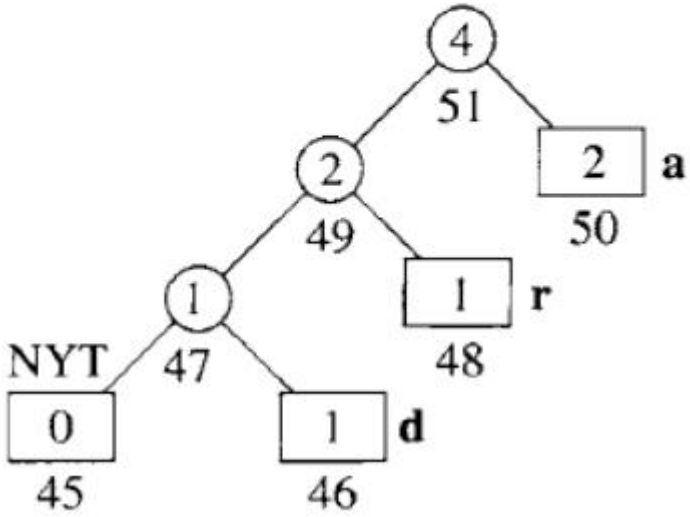
- Huffman Coding is fundamental
- Adaptive Huffman Coding

Algorithm(1)

[a a r d v a r k]

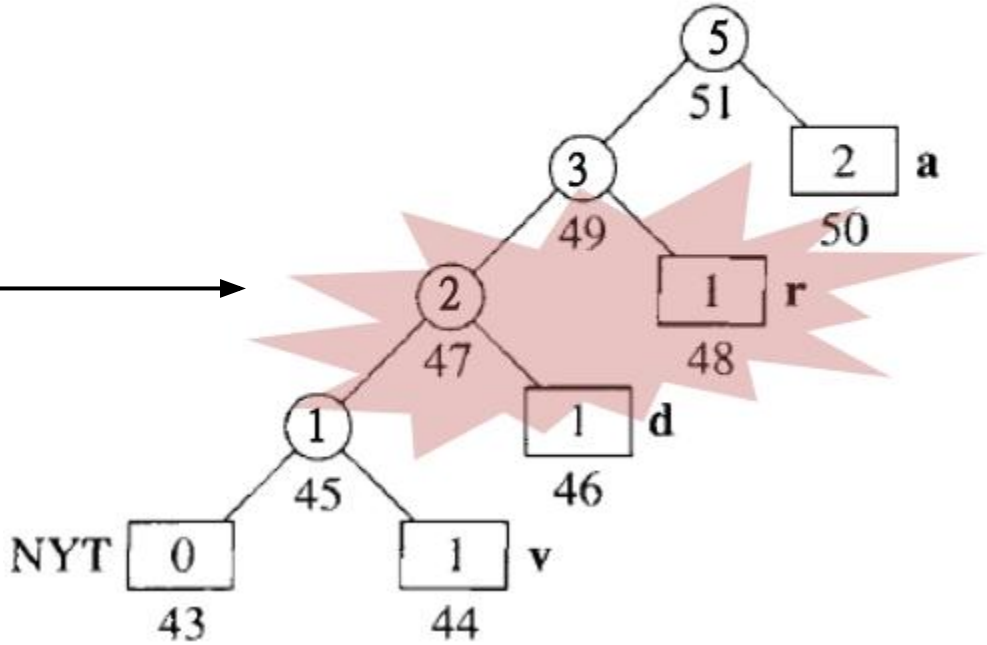
NYT 00

d 00011



NYT 000

v 1011



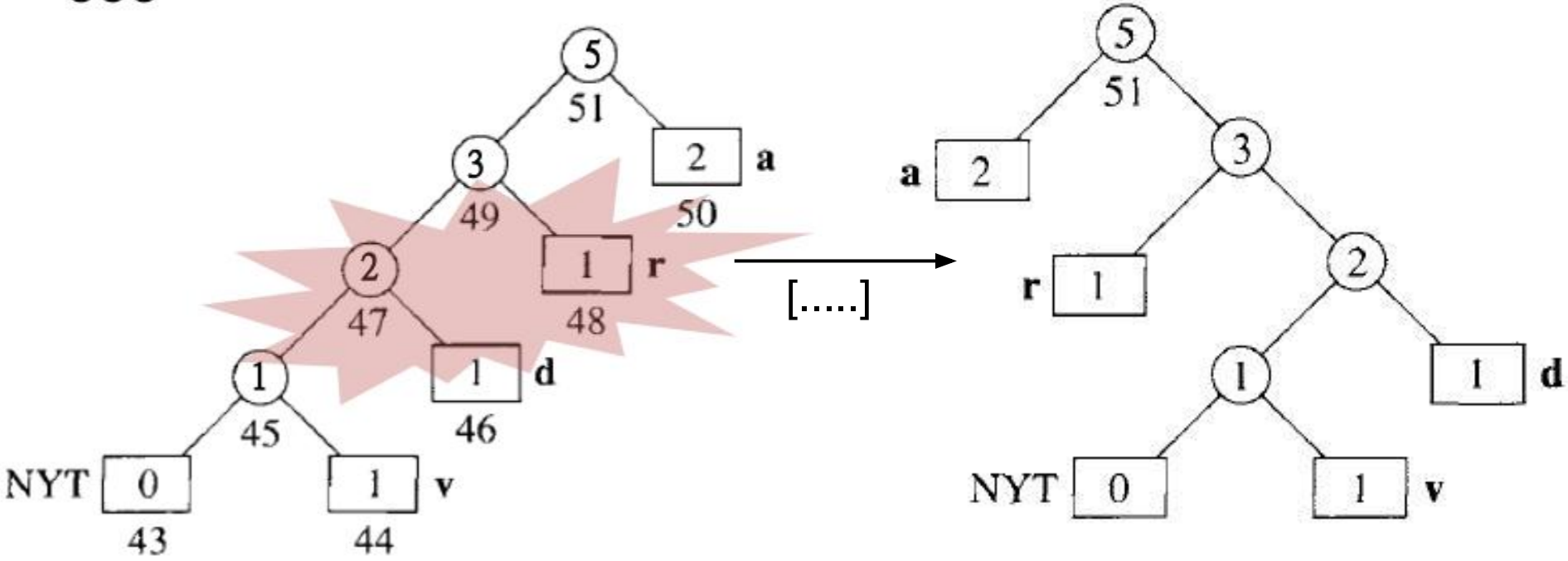
Example from lecture slides

Algorithm(2)

[a a r d v a r k]

NYT 000

v 1011



Algorithm(3)

Encoder

```
initializeModel();
while ((bit = read()) != EOS)
{
    encode(bit, output);
    updateModel(bit);
}
```

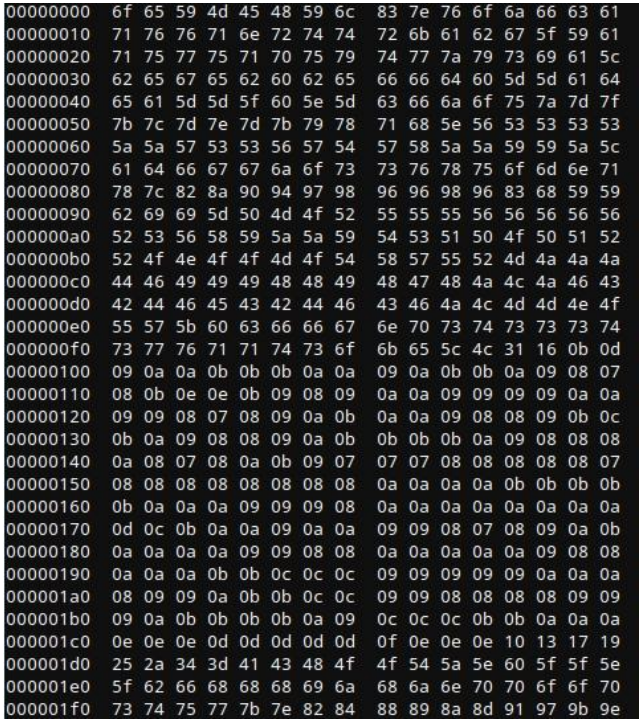
Decoder

```
initializeModel();
while ((bit = read()) != EOF)
{
    decode(bit, output);
    updateModel(bit)
}
```

Implementation

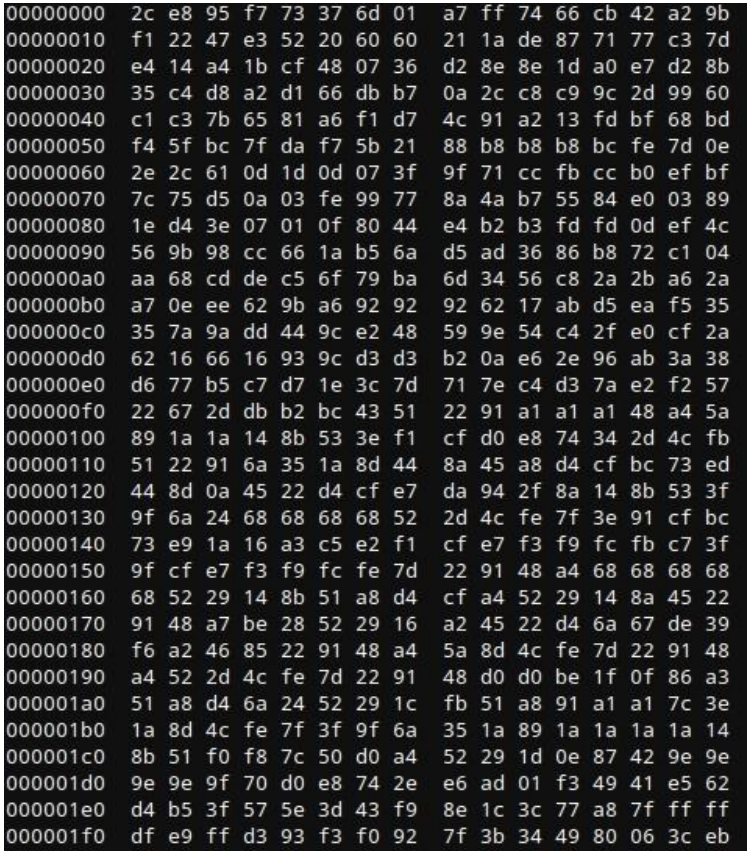
```
private static void encode(InputStream in, BitOutputStream out, PrintWriter HTOutputStream) throws IOException {
    CodeFrequency frequencyTable = new CodeFrequency(IMAGE_BIT_SIZE);
    CodeTree codeTree = frequencyTable.generateCodeTree();
    CodeWriter codeWriter = new CodeWriter(out, codeTree);

    int bitCount = 0;
    while (true) {
        int bit = in.read();
        if (bit == -1) break; // EOS

        codeWriter.write(bit);
        frequencyTable.increment(bit);
        bitCount++;

        if (isUnbalanced(bitCount)) {
            CodeTree updatedCodeTree = frequencyTable.generateCodeTree();
            codeWriter.setCodeTree(updatedCodeTree);
        }
        if (toLimitFreqTable(bitCount)) {
            frequencyTable = new CodeFrequency(IMAGE_BIT_SIZE);
        }
    }
    codeWriter.write(256); // EOF
```

Result



```
00000000  6f 65 59 4d 45 48 59 6c  83 7e 76 6f 6a 66 63 61
00000010  71 76 76 71 6e 72 74 74  72 6b 61 62 67 5f 59 61
00000020  71 75 77 75 71 70 75 79  74 77 7a 79 73 69 61 5c
00000030  62 65 67 65 62 60 62 65  66 66 64 60 5d 5d 61 64
00000040  65 61 5d 5d 5f 60 5e 5d  63 66 6a 6f 75 7a 7d 7f
00000050  7b 7c 7d 7e 7d 7b 79 78  71 68 5e 56 53 53 53 53
00000060  5a 5a 57 53 53 56 57 54  57 58 5a 5a 59 59 5a 5c
00000070  61 64 66 67 67 6a 6f 73  73 76 78 75 6f 6d 6e 71
00000080  78 7c 82 8a 90 94 97 98  96 96 98 96 83 68 59 59
00000090  62 69 69 5d 50 4d 4f 52  55 55 55 56 56 56 56 56
000000a0  52 53 56 58 59 5a 5a 59  54 53 51 50 4f 50 51 52
000000b0  52 4f 4e 4f 4f 4d 4f 54  58 57 55 52 4d 4a 4a 4a
000000c0  44 46 49 49 49 48 48 49  48 47 48 4a 4c 4a 46 43
000000d0  42 44 46 45 43 42 44 46  43 46 4a 4c 4d 4d 4e 4f
000000e0  55 57 5b 60 63 66 66 67  6e 70 73 74 73 73 73 74
000000f0  73 77 76 71 71 74 73 6f  6b 65 5c 4c 31 16 0b 0d
00000100  09 0a 0a 0b 0b 0b 0a 0a  09 0a 0b 0b 0a 09 08 07
00000110  08 0b 0e 0e 0b 09 08 09  0a 0a 09 09 09 09 0a 0a
00000120  09 09 08 07 08 09 0a 0b  0a 0a 09 08 08 09 0b 0c
00000130  0b 0a 09 08 08 09 0a 0b  0b 0b 0b 0a 09 08 08 08
00000140  0a 08 07 08 0a 0b 09 07  07 07 08 08 08 08 08 07
00000150  08 08 08 08 08 08 08 08  0a 0a 0a 0a 0b 0b 0b 0b
00000160  0b 0a 0a 0a 09 09 09 08  0a 0a 0a 0a 0a 0a 0a 0a
00000170  0d 0c 0b 0a 0a 09 0a 0a  09 09 08 07 08 09 0a 0b
00000180  0a 0a 0a 0a 09 09 08 08  0a 0a 0a 0a 0a 09 08 08
00000190  0a 0a 0a 0a 0b 0b 0c 0c  0c 09 09 09 09 0a 0a 0a
000001a0  08 09 09 0a 0b 0b 0c 0c  09 09 08 08 08 08 09 09
000001b0  09 0a 0b 0b 0b 0b 0a 09  0c 0c 0c 0b 0b 0a 0a 0a
000001c0  0e 0e 0e 0d 0d 0d 0d 0d  0f 0e 0e 0e 10 13 17 19
000001d0  25 2a 34 3d 41 43 48 4f  4f 54 5a 5e 60 5f 5f 5e
000001e0  5f 62 66 68 68 68 69 6a  68 6a 6e 70 70 6f 6f 70
000001f0  73 74 75 77 7b 7e 82 84  88 89 8a 8d 91 97 9b 9e
```

Raw



```
00000000  2c e8 95 f7 73 37 6d 01  a7 ff 74 66 cb 42 a2 9b
00000010  f1 22 47 e3 52 20 60 60  21 1a de 87 71 77 c3 7d
00000020  e4 14 a4 1b cf 48 07 36  d2 8e 8e 1d a0 e7 d2 8b
00000030  35 c4 d8 a2 d1 66 db b7  0a 2c c8 c9 9c 2d 99 60
00000040  c1 c3 7b 65 81 a6 f1 d7  4c 91 a2 13 fd bf 68 bd
00000050  f4 5f bc 7f da f7 5b 21  88 b8 b8 b8 bc fe 7d 0e
00000060  2e 2c 61 0d 1d 0d 07 3f  9f 71 cc fb cc b0 ef bf
00000070  7c 75 d5 0a 03 fe 99 77  8a 4a b7 55 84 e0 03 89
00000080  1e d4 3e 07 01 0f 80 44  e4 b2 b3 fd fd 0d ef 4c
00000090  56 9b 98 cc 66 1a b5 6a  d5 ad 36 86 b8 72 c1 04
000000a0  aa 68 cd de c5 6f 79 ba  6d 34 56 c8 2a 2b a6 2a
000000b0  a7 0e ee 62 9b a6 92 92  92 62 17 ab d5 ea f5 35
000000c0  35 7a 9a dd 44 9c e2 48  59 9e 54 c4 2f e0 cf 2a
000000d0  62 16 66 16 93 9c d3 d3  b2 0a e6 2e 96 ab 3a 38
000000e0  d6 77 b5 c7 d7 1e 3c 7d  71 7e c4 d3 7a e2 f2 57
000000f0  22 67 2d db b2 bc 43 51  22 91 a1 a1 a1 48 a4 5a
00000100  89 1a 1a 14 8b 53 3e f1  cf d0 e8 74 34 2d 4c fb
00000110  51 22 91 6a 35 1a 8d 44  8a 45 a8 d4 cf bc 73 ed
00000120  44 8d 0a 45 22 d4 cf e7  da 94 2f 8a 14 8b 53 3f
00000130  9f 6a 24 68 68 68 68 52  2d 4c fe 7f 3e 91 cf bc
00000140  73 e9 1a 16 a3 c5 e2 f1  cf e7 f3 f9 fc fb c7 3f
00000150  9f cf e7 f3 f9 fc fe 7d  22 91 48 a4 68 68 68 68
00000160  68 52 29 14 8b 51 a8 d4  cf a4 52 29 14 8a 45 22
00000170  91 48 a7 be 28 52 29 16  a2 45 22 d4 6a 67 de 39
00000180  f6 a2 46 85 22 91 48 a4  5a 8d 4c fe 7d 22 91 48
00000190  a4 52 2d 4c fe 7d 22 91  48 d0 d0 be 1f 0f 86 a3
000001a0  51 a8 d4 6a 24 52 29 1c  fb 51 a8 91 a1 a1 7c 3e
000001b0  1a 8d 4c fe 7f 3f 9f 6a  35 1a 89 1a 1a 1a 1a 14
000001c0  8b 51 f0 f8 7c 50 d0 a4  52 29 1d 0e 87 42 9e 9e
000001d0  9e 9e 9f 70 d0 e8 74 2e  e6 ad 01 f3 49 41 e5 62
000001e0  d4 b5 3f 57 5e 3d 43 f9  8e 1c 3c 77 a8 7f ff ff
000001f0  df e9 ff d3 93 f3 f0 92  7f 3b 34 49 80 06 3c eb
```

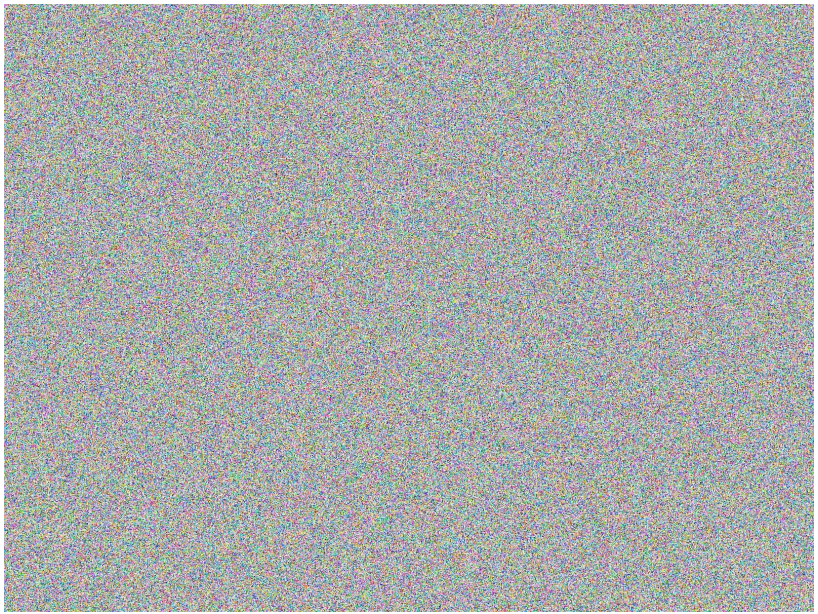
Encoded

Compression rate – Result

Test File	Raw Size	Encoded Size	Compression Ratio
1	1920000	1802687	6,11%
2	1920000	1272218	33,74%
3	1920000	1566242	18,42%
4	1920000	1488520	22,47%
5	1920000	1921055	−0,05%

Compression rate – Result(2)

File 2



File 5

Speed - Result

- Encoding - average of 400 ms
- Decoding - average of 300 ms

Comparison with 7zip

File	Raw Size	Adaptive Huffman	Ratio	7zip	Ratio
1	1920000	1802687	6,11%	919456	52,11%
2	1920000	1272218	33,74%	794661	58,61%
3	1920000	1566242	18,42%	850945	55,68%
4	1920000	1488520	22,47%	904077	52,91%
5	1920000	1921055	−0,05%	1946223	−1,37%

Questions?