EECE 4353 Image Processing

Lecture Notes: JPEG Image Compression

Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2016

From the Article by Wallace:

Title:	The JPEG Still Picture Compression Standard
Source:	Communications of the ACM archive Volume 34, Issue 4 (April 1991) Special issue on digital multimedia systems
Pages:	30-44
Year:	1991
ISSN:	0001-0782
Author:	Gregory K. Wallace Digital Equipment Corp., Maynard, MA
Publisher:	ACM Press New York, NY, USA

The JPEG Still Picture Compression Standard

- Joint Photographic Experts Group
- A standards committee set up by the CCITT and the ISO.
- Tasked in the late 1980's to generate a general-purpose standard for compression of almost all continuous tone and still-image applications.
- Published Standard: "Digital Compression and Coding of Continuous-tone Still Images – Requirements and Guidelines," ISO/IEC 10918-1:1993(E)

JPEG's Goals for the Standard

- Be at or near the state of the art in compression rate and image fidelity.
- User decides on the trade-off between image fidelity and compression ratio.
- Be applicable to any kind of continuous-tone digital image source (unrestricted with respect to content, complexity, color-range, statistics, etc.)
- Have tractable computational complexity for implementation on a wide range of computational hardware.

JPEG's Modes of Operation

1. **Sequential encoding:** each component (band) encoded in a single raster scan
2. **Progressive encoding:** progressive coarse–to-fine encoding and decoding of entire image
3. **Lossless encoding:** exact recovery of original image
4. **Hierarchical encoding:** multiresolution compression with independently retrievable lower-resolution versions

JPEG Codec

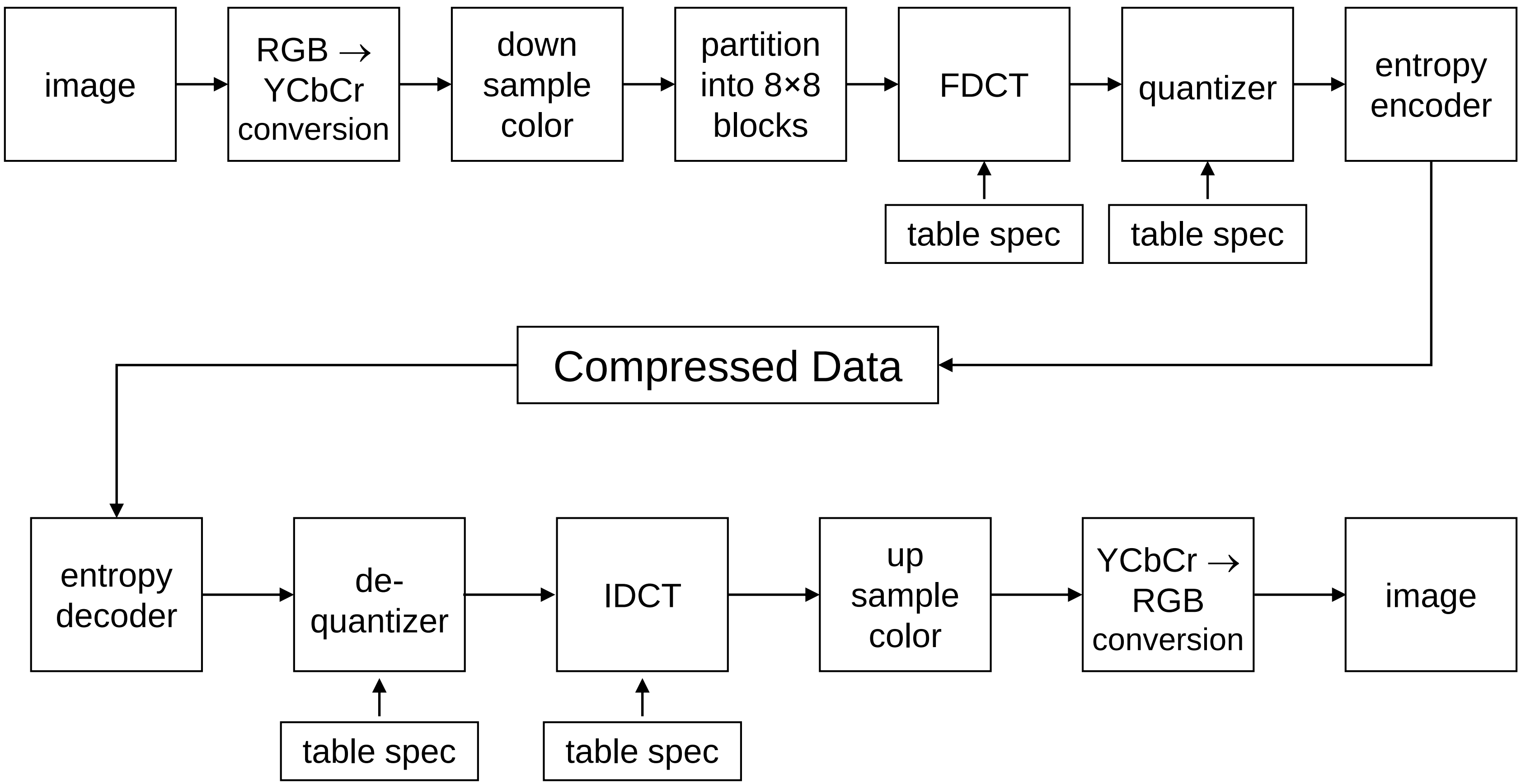


image
RGB → YCbCr conversion
down sample color
partition into 8×8 blocks
FDCT
quantizer
entropy encoder
table spec
table spec
Compressed Data
entropy decoder
de-quantizer
IDCT
up sample color
YCbCr → RGB conversion
image
table spec
table spec

JPEG Baseline Process

- DCT-based process
- Source image: 8-bit samples within each component
- Sequential
- Huffman coding: 2 AC and 2 DC tables
- Decoders process scans with 1, 2, 3, or 4 components
- Interleaved and non-interleaved scans

Entropy Encoding

- Set $A = \{s_1, \ldots, s_N\}$ is called an **alphabet**.
- Each $s_i \in A$ is called a **symbol**.
- A **string** over the alphabet is a sequence of symbols, *e.g.*
- $s_{11}s_4s_{105}s_{105}s_{47}s_{86}s_{86}s_3\cdots s_{30}$.
- A **codeword** is a sequence of bits that represents a symbol or string. *e.g.* 101100001011011101.
- $p_i = p(s_i)$ is the **probability of occurrence** of symbol s_i in a string.
- L_i is the **length of the codeword** for s_i in bits.

In the context of **Information Theory** the entropy is the number of bits on the average need to represent a symbol in set A.

Information Content of a Symbol

If in a signal or message, the symbol s_i has occurrence probability $\mathrm{p}(s_i)$ then a measure of the extraordinariness of the event "s_i occurs" is $1/\mathrm{p}(s_i)$ – the less likely the event, the more extraordinary it is. The information content of the event is defined as the number of bits required to code the measure of its extraordinariness. That is

$$\mathrm{I}(s_i) = \log_2\left(\frac{1}{\mathrm{p}(s_i)}\right) = -\log_2 \mathrm{p}(s_i).$$

is the *information content* of symbol s_i, — the number of bits needed to code it. $\mathrm{I}(s_i)$ is a real number, not necessarily an integer.

Entropy Encoding

Given a finite set of symbols with associated probabilities of occurrence,

$$A = \{ s_i , \mathrm{p}(s_i) \}_{i=1}^{N} ,$$

such that

$$\sum_{i=1}^{N} \mathrm{p}(s_i) = 1.$$

Then the entropy of A is the expected value

$$\begin{aligned} \mathrm{H}(A) &= E\left[\left\{ \mathrm{I}(s_i) \right\}_{i=1}^{n} \right] \\ &= \sum_{i=1}^{N} \mathrm{I}(s_i)\,\mathrm{p}(s_i) \\ &= \sum_{i=1}^{N} \left(-\log_2 \mathrm{p}(s_i) \right) \mathrm{p}(s_i) \\ &= -\sum_{i=1}^{N} \mathrm{p}(s_i) \log_2 \mathrm{p}(s_i), \end{aligned}$$

the mean information content of the symbols.

Entropy Examples

1. $A = \{a,b\}$; $\mathrm{p}(a) = 0.5$; $\mathrm{p}(b) = 0.5$.

$$\begin{aligned}
\mathrm{H}(a,b) &= -\mathrm{p}(a)\log_2 \mathrm{p}(a) - \mathrm{p}(b)\log_2 \mathrm{p}(b) \\
&= -0.5\log_2 0.5 - 0.5\log_2 0.5 \\
&= -0.5(-1) - 0.5(-1) \\
&= 1
\end{aligned}$$

2. $A = \{a,b\}$; $\mathrm{p}(a) = 0.8$; $\mathrm{p}(b) = 0.2$.

$$\begin{aligned}
\mathrm{H}(a,b) &= -\mathrm{p}(a)\log_2 \mathrm{p}(a) - \mathrm{p}(b)\log_2 \mathrm{p}(b) \\
&= -0.8\log_2 0.8 - 0.2\log_2 0.2 \\
&= -0.8(-0.3219) - 0.2(-2.3219) \\
&= 0.7219
\end{aligned}$$

Entropy Examples

3. $A = \{a,b,c,d\}$; p(a) = 0.25; p(b) = 0.125; p(c) = 0.2917; p(d) = 0.3333.

$$\begin{aligned}
\mathrm{H}(a,b,c,d) &= -\mathrm{p}(a)\log_2 \mathrm{p}(a) - \mathrm{p}(b)\log_2 \mathrm{p}(b) - \mathrm{p}(c)\log_2 \mathrm{p}(c) - \mathrm{p}(d)\log_2 \mathrm{p}(d) \\
&= -0.25\log_2 0.25 - 0.125\log_2 0.125 - 0.2917\log_2 0.2917 - 0.3333\log_2 0.3333 \\
&= -0.25(-2) - 0.125(-3) - 0.2917(-1.7776) - 0.3333(-1.585) \\
&= 1.9218
\end{aligned}$$

The entropy is usually not integral. How, then can it be coded?

Entropy encoding finds a mapping from the alphabet to the code words such that the average number of bits per symbol (bps) approaches the entropy as the number of symbols increases without bound.

$$\lim_{N\to\infty} \mathrm{bps} = \mathrm{H}(A)$$

bps = (size of encoded message)/(size of original message).

Code Types

Fixed length: all codewords have the same length (number of bits). *E.g.* letters A through H encoded as 3-bit binary numbers.

$$A \leftrightarrow 000,\ B \leftrightarrow 001,\ C \leftrightarrow 010,\ \ldots,\ H \leftrightarrow 111$$

Variable length: different codewords may have different lengths depending on, for example, their probability of occurrence.

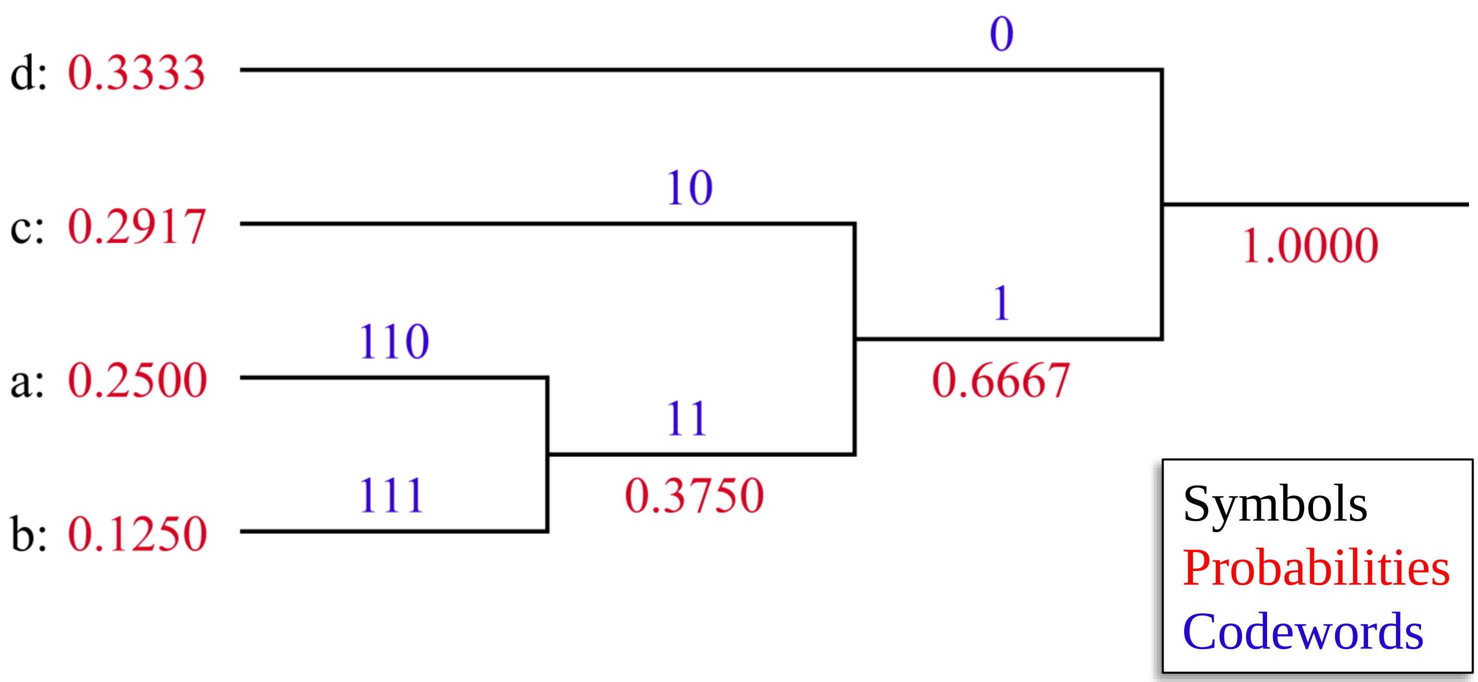
$$A \leftrightarrow 0,\ B \leftrightarrow 10,\ C \leftrightarrow 110,\ \ldots,\ H \leftrightarrow 11111110$$

That variable length code is an example of a prefix code, no code is the prefix of another code.

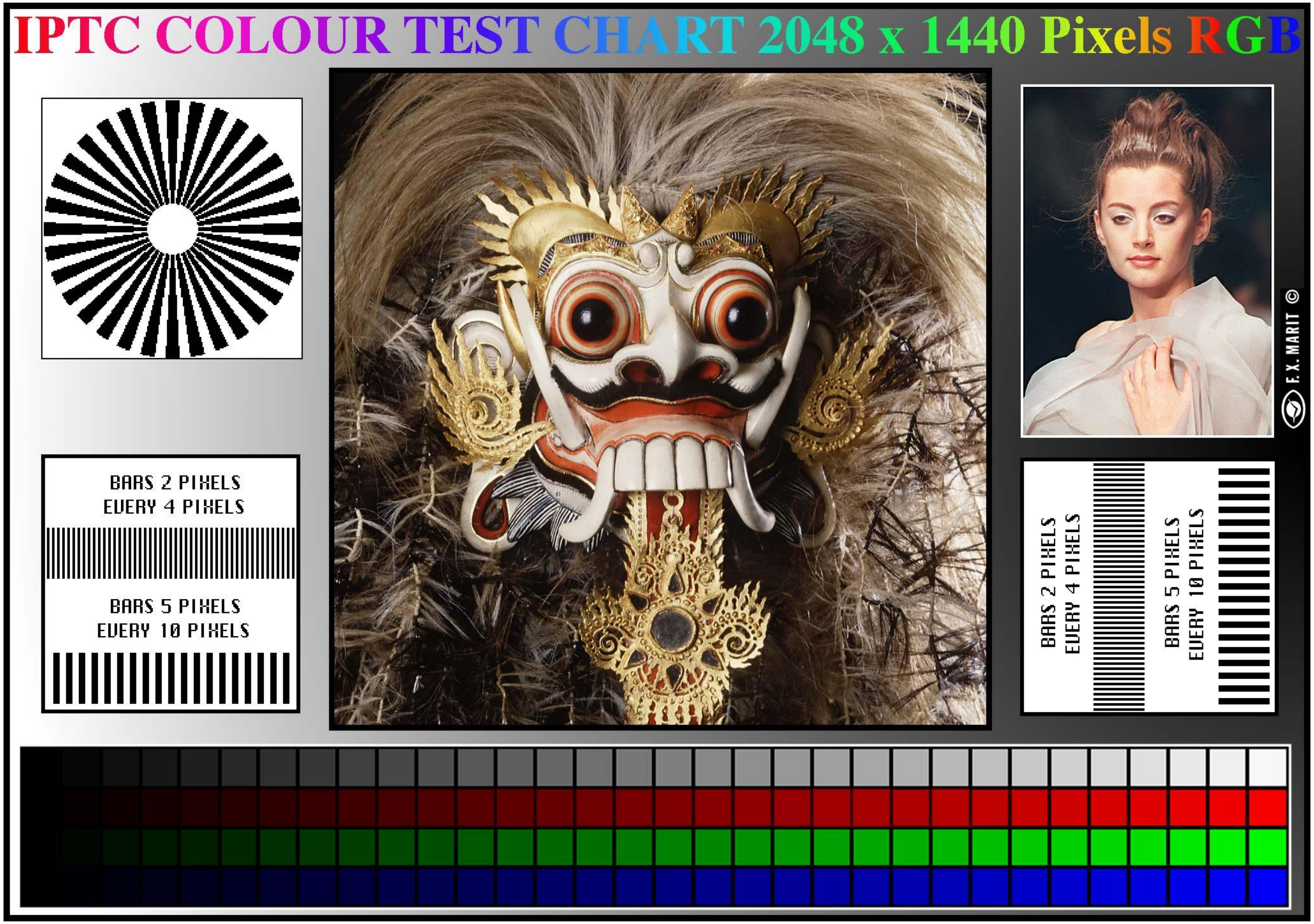
Huffman Coding

- Each symbol is assigned a specific code with a length dependent on its frequency of occurrence.
- There is a prefix code – no codeword is a prefix of any other codeword.
- Is uniquely decodable – only one possible source message.
- A Huffman code is the optimal prefix+variable-length code given the probability of occurrence of the symbols inb the message.
- Generated with a Huffman tree.

Huffman Tree



A Standard Color Test Chart



We’ll use the image in some of the following slides.

Color Conversion: RGB → YCbCr

RGB

→

Y

+

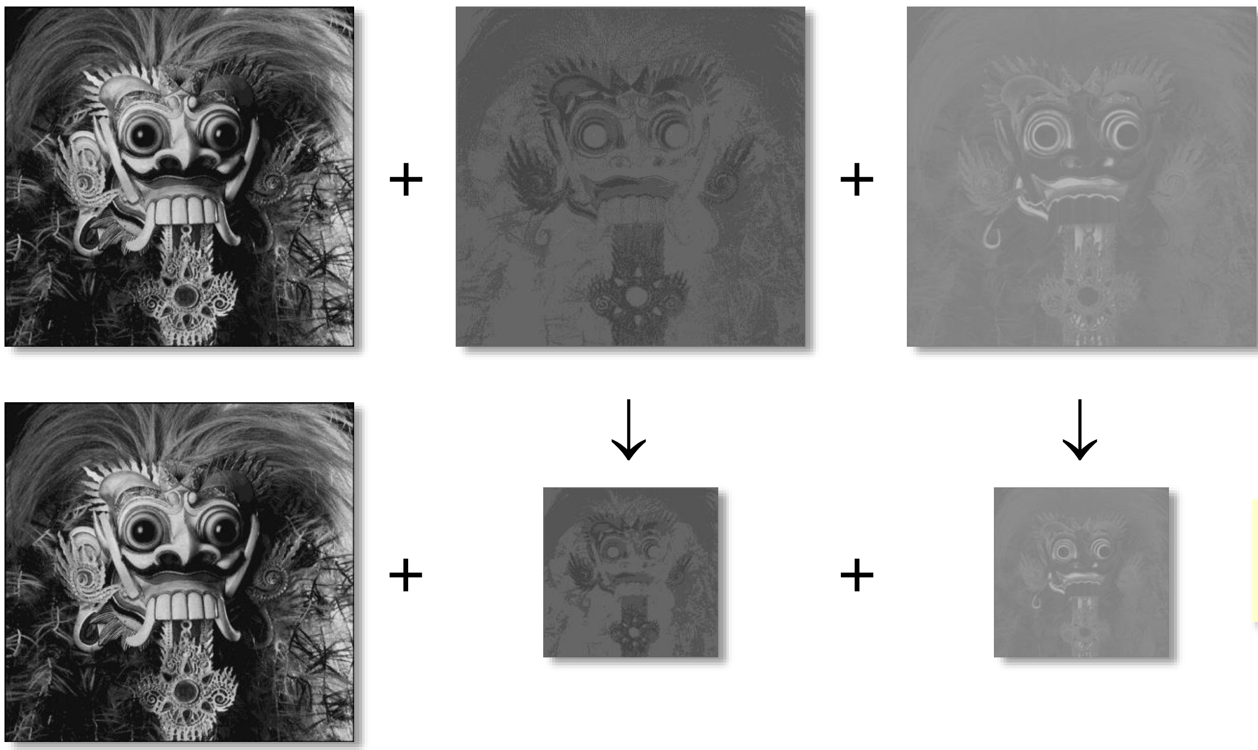
Cb

+

Cr

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.25678824 & 0.50412941 & 0.09790588 \\ -0.14822290 & -0.29099279 & 0.43921596 \\ 0.43921569 & -0.36778831 & -0.07142737 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

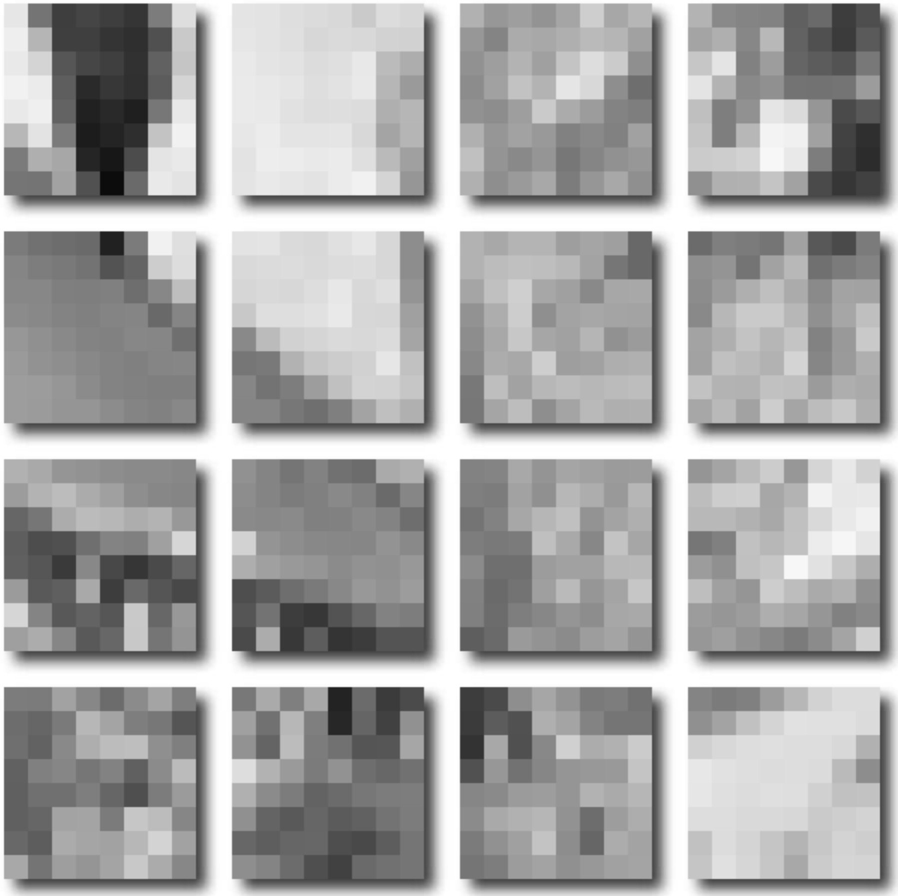
Downsample Color Bands



Y Cb Cr

by a factor of 2

Partition Each Band into 8×8 Blocks



Prep for Forward Discrete Cosine Transform

- Each image band is operated on independently.
- Each 8×8 image block is operated on independently.
- Shift all image values from $I \in [0, 2^{P}-1]$ to $I \in [-2^{P-1}, 2^{P-1}-1]$. E.g. $[0, 255] \rightarrow [-128, 127]$.
- The FDCT decomposes each block into a set of coefficients with respect to the 64 orthogonal basis functions shown on the next slide.

DCT Functions and Variables

$\mathfrak{F}\{\mathbf{I}\}(v,u\,;r,c)$ - DCT of 8×8 block from $\mathbf{I}$ starting at (r,c)

$\Lambda(\xi)$ - Normalization Factor

$\phi(v,u\,;\rho,\chi)$ - 2D Cosine Basis Function, (v,u)

r - image row index (vertical, increasing down)

c - image column index (horizontal, increasing right)

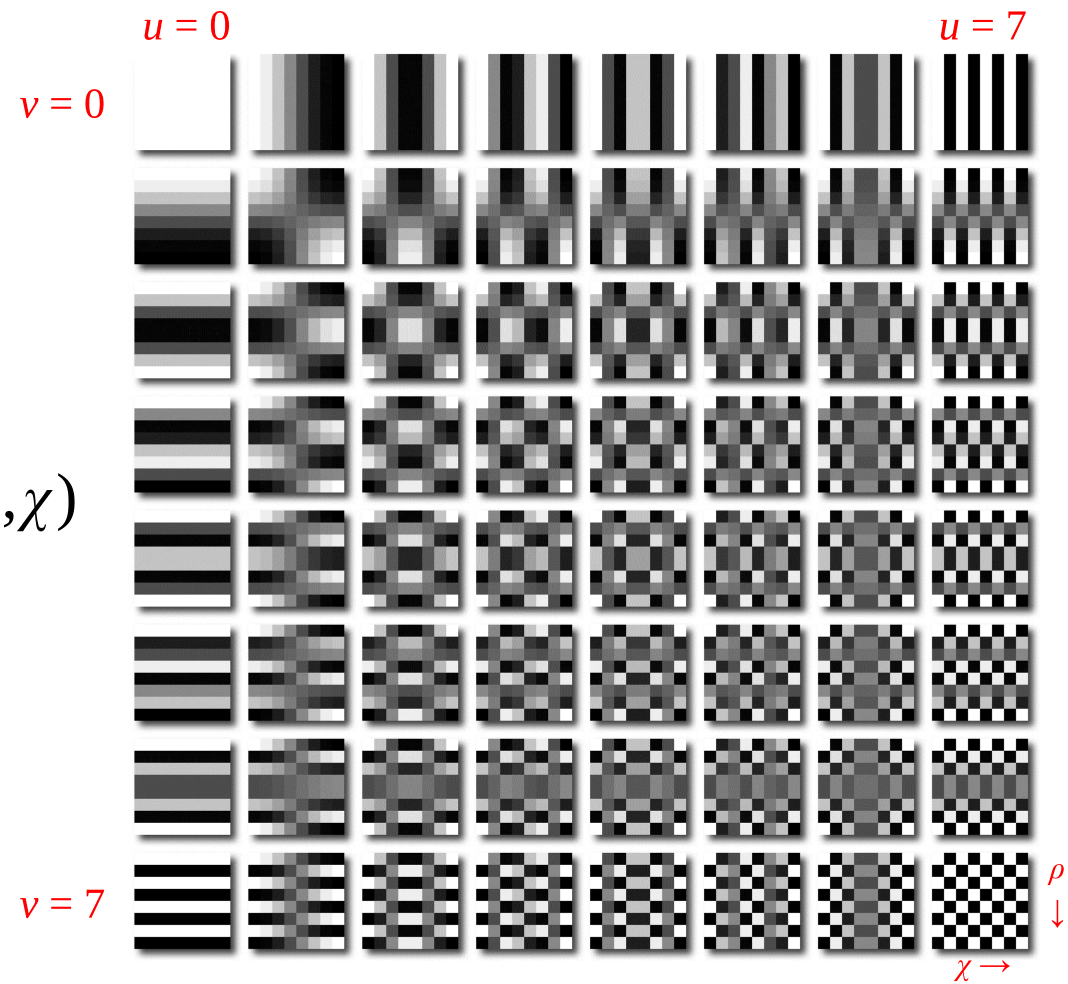
ρ - DCT row index (horizontal wave fronts, vertical propagation down)

χ - DCT column index (vertical wave fronts, horizontal propagation right)

v - DCT row frequency index (vertical, increasing down)

u - DCT column frequency index (horizontal, increasing right)

DCT Basis Functions



$\phi(v,u;\rho,\chi)$

white = 1
black = -1

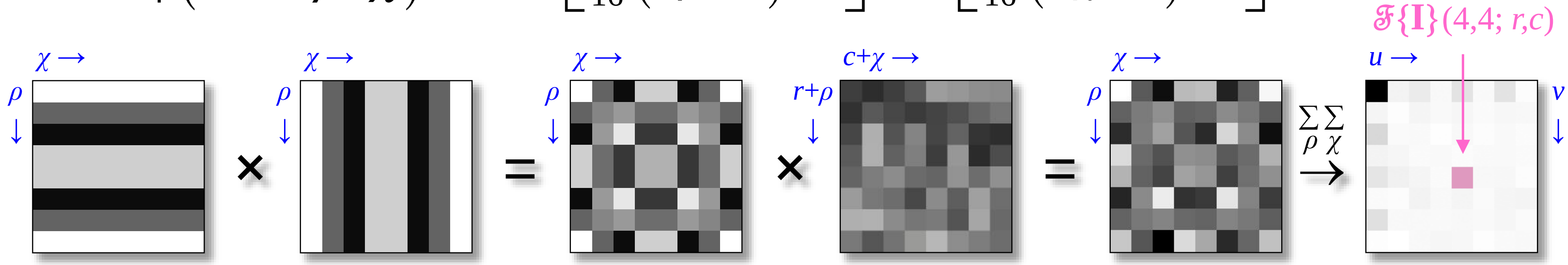
Forward Discrete Cosine Transform

$$\mathfrak{F}\{\mathbf{I}\}(v,u\,;r,c) = \sum_{\rho=0}^{7}\sum_{\chi=0}^{7} \tfrac{1}{4}\Lambda(v)\Lambda(u)\phi(v,u\,;\rho,\chi)\,\mathbf{I}(r+\rho,c+\chi)$$

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}$$

$(r,c) \in \{0,8,16,\ldots,R\} \times \{0,8,16,\ldots,C\}$,
(v,u), $(\rho,\chi) \in \{0,\ldots,7\} \times \{0,\ldots,7\}$.

$$\phi(v,u\,;\rho,\chi) = \cos\left[\tfrac{1}{16}(2\rho+1)\pi v\right]\cos\left[\tfrac{1}{16}(2\chi+1)\pi u\right]$$

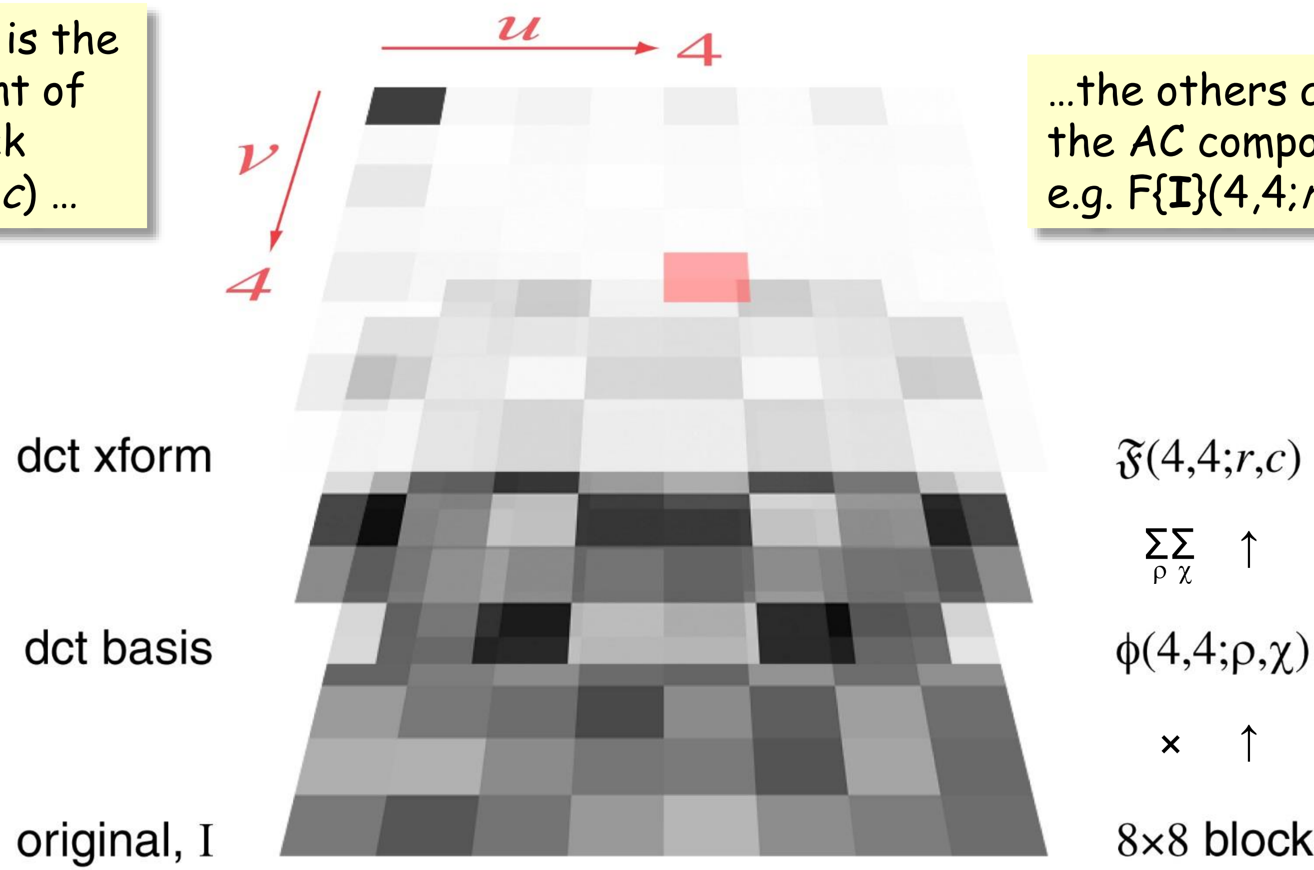


row cos: (v=4, ρ) col cos: (u=4, χ) Φ = col cos × row cos 8x8 block of $\mathbf{I}$ @ (r,c) p.w. product $\mathfrak{F}(v,u;r,c)$

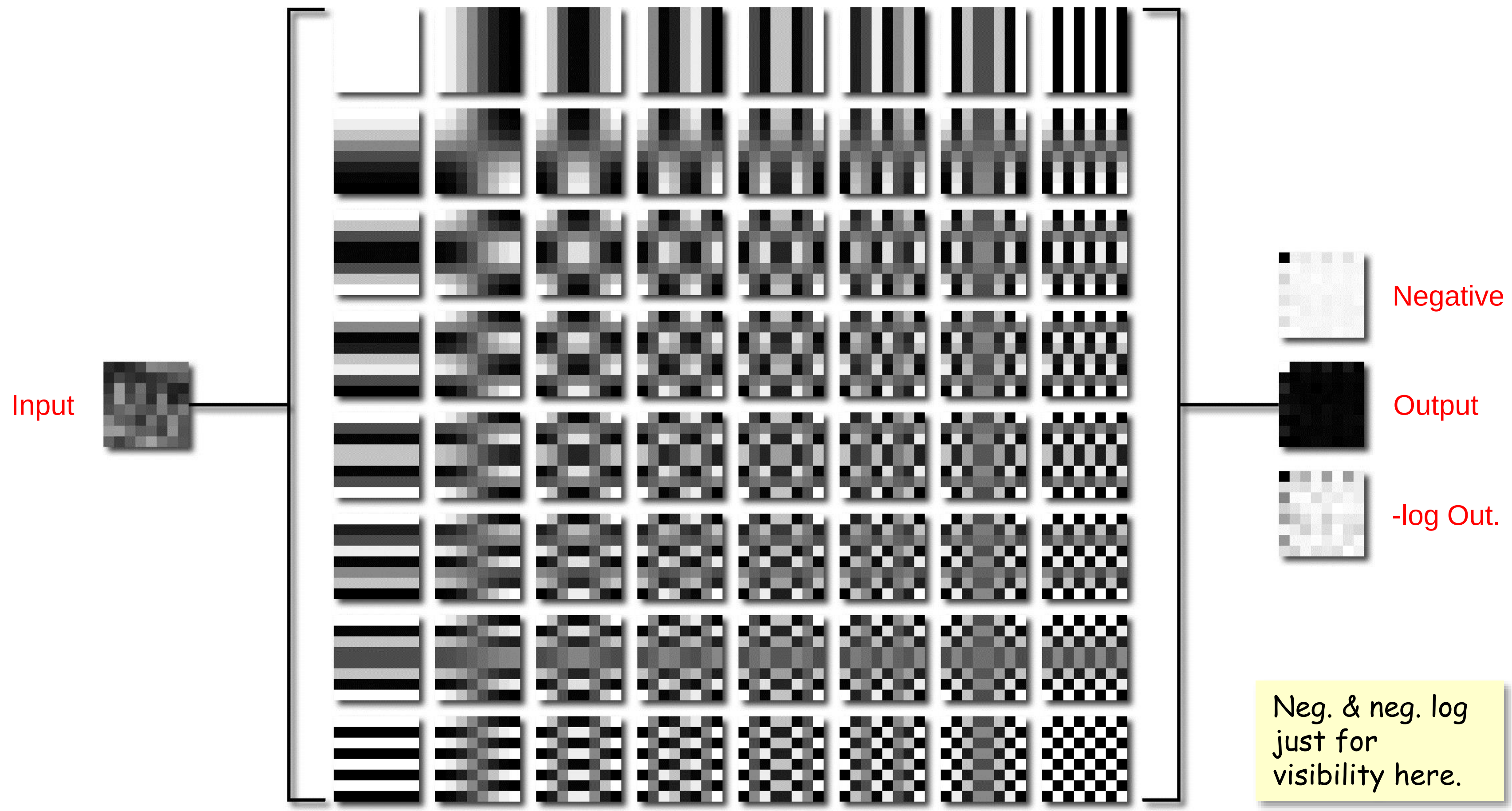
Forward Discrete Cosine Transform

F{**I**}(0,0;*r*,*c*) is the DC component of the 8x8 block from **I** at (*r*,*c*) …

…the others are the AC components, e.g. F{**I**}(4,4;*r*,*c*).

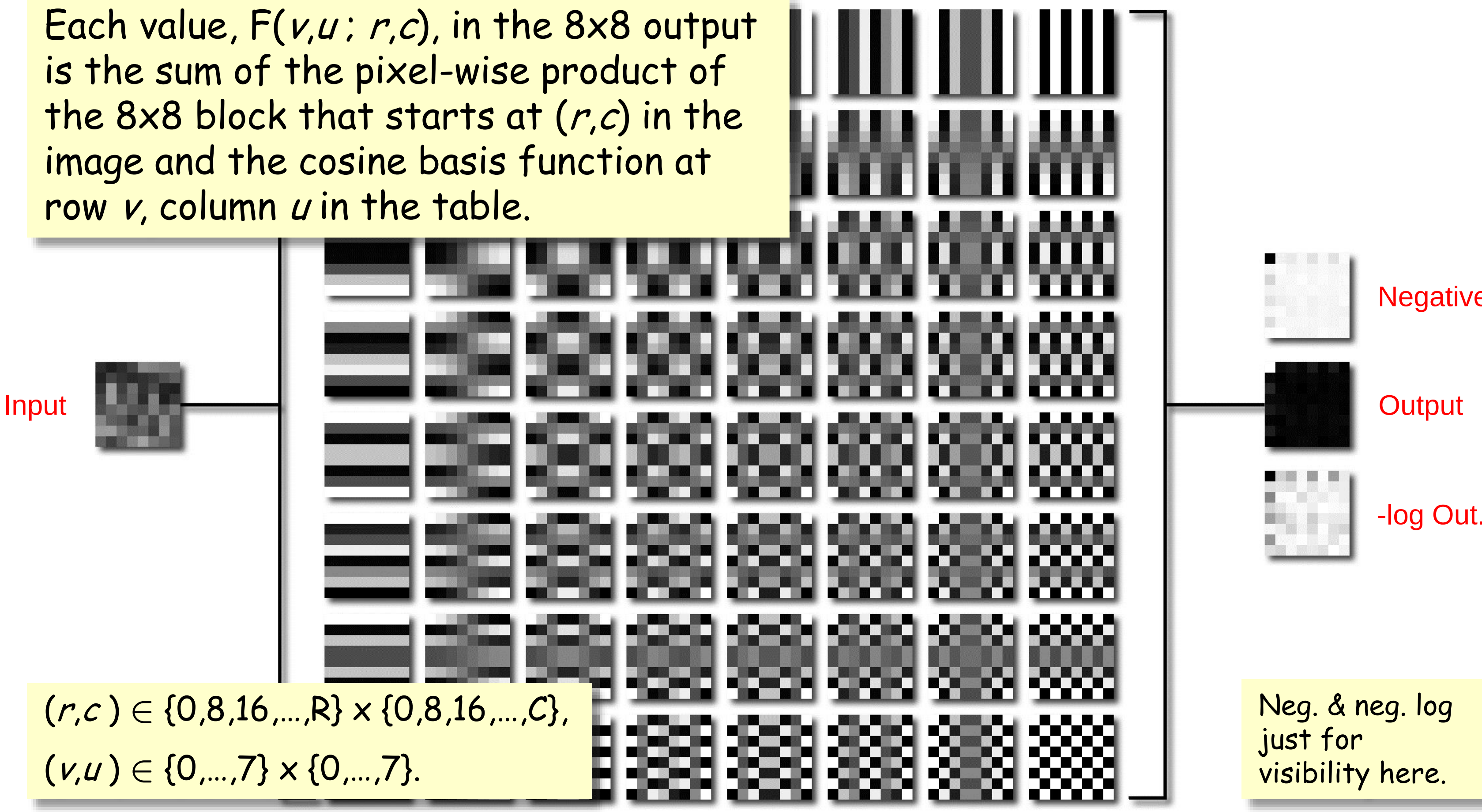


Forward Discrete Cosine Transform



Forward Discrete Cosine Transform

Each value, F(*v*,*u* ; *r*,*c*), in the 8x8 output is the sum of the pixel-wise product of the 8x8 block that starts at (*r*,*c*) in the image and the cosine basis function at row *v*, column *u* in the table.



$(r,c) \in \{0,8,16,\ldots,R\} \times \{0,8,16,\ldots,C\}$,

$(v,u) \in \{0,\ldots,7\} \times \{0,\ldots,7\}$.

Neg. & neg. log just for visibility here.

Example FDCT Quantization Tables

Luminance Quantization Table

2	2	3	4	5	6	8	11
2	2	2	4	5	7	9	11
3	2	3	5	7	9	11	12
4	4	5	7	9	11	12	12
5	5	7	9	11	12	12	12
6	7	9	11	12	12	12	12
8	9	11	12	12	12	12	12
11	11	12	12	12	12	12	12

Precision: 8 bits
Approximate quality factor: 91.64
Scaling: 16.71 Variance: 22.54

Chrominance Quantization Table

3	3	7	13	15	15	15	15
3	4	7	13	14	12	12	12
7	7	13	14	12	12	12	12
13	13	14	12	12	12	12	12
15	14	12	12	12	12	12	12
15	12	12	12	12	12	12	12
15	12	12	12	12	12	12	12
15	12	12	12	12	12	12	12

Precision: 8 bits
Approximate quality factor: 92.57
Scaling: 14.85 Variance: 23.00

FDCT Quantization Procedure

255	23	34	13	44	11	44	6
19	4	19	12	18	9	16	15
58	10	11	2	14	6	12	7
22	18	9	12	7	11	12	9
42	26	19	9	23	15	16	6
9	6	23	14	20	10	19	21
48	13	13	11	15	12	18	10
3	0	13	18	16	4	11	14

Output of FDCT, $\mathscr{F}(u,v)$, at image pixel location (r,c).

1	1	1	2	3	6	8	10
1	1	2	3	4	8	9	8
2	2	2	3	6	8	10	8
2	2	3	4	7	12	11	9
3	3	8	11	10	16	15	11
3	5	8	10	12	15	16	13
7	10	11	12	15	17	17	14
14	13	13	15	15	14	14	14

Quantization Table, $Q(u,v)$

255	23	34	7	15	2	6	1
19	4	10	4	5	1	2	2
29	5	6	1	2	1	1	1
11	9	3	3	1	1	1	1
14	9	2	1	2	1	1	1
3	1	3	1	2	1	1	2
7	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1

Quantized Result, $\mathscr{F}^Q(u,v)$ at image pixel location (r,c).

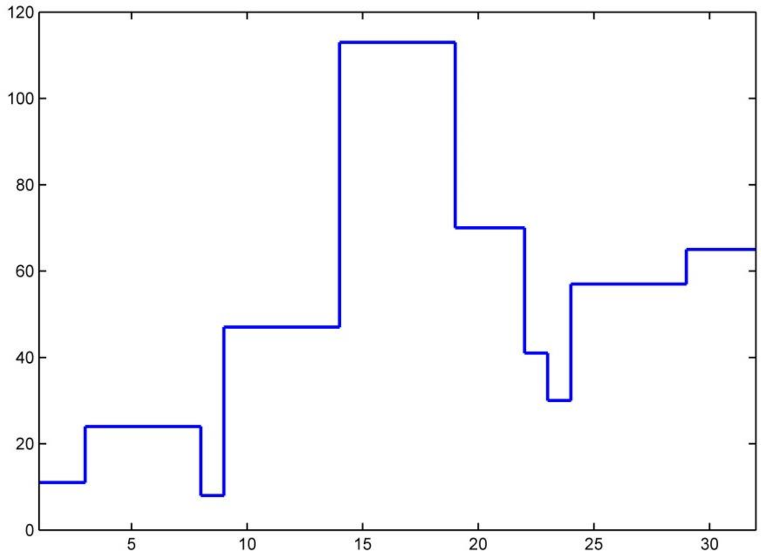
$$\mathscr{F}^Q(u,v) = \text{round}\left(\frac{\mathscr{F}(u,v)}{Q(u,v)}\right)$$

$(r,c) \in$
$\{0,8,16,\dots,R\} \times \{0,8,16,\dots,C\}$,

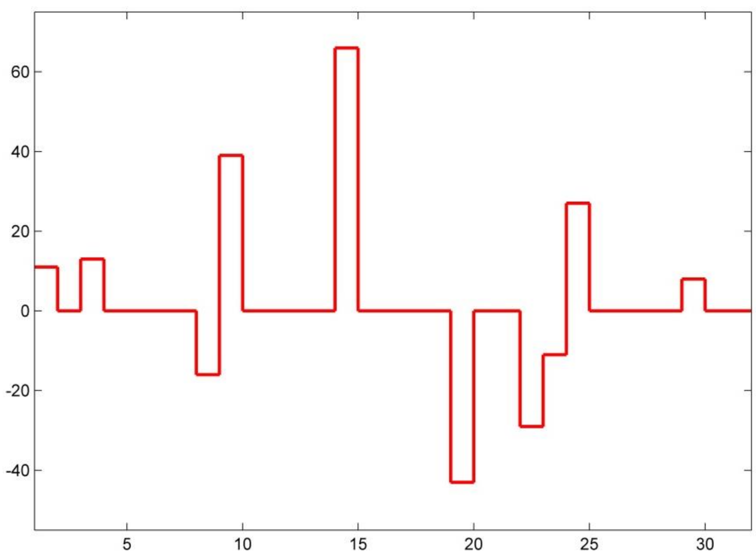
This Q(u,v) is yet another quantization table.

Image DC Components (8×8 constant blocks)

DC Component Coding (DPCM)



DC components from 32 blocks

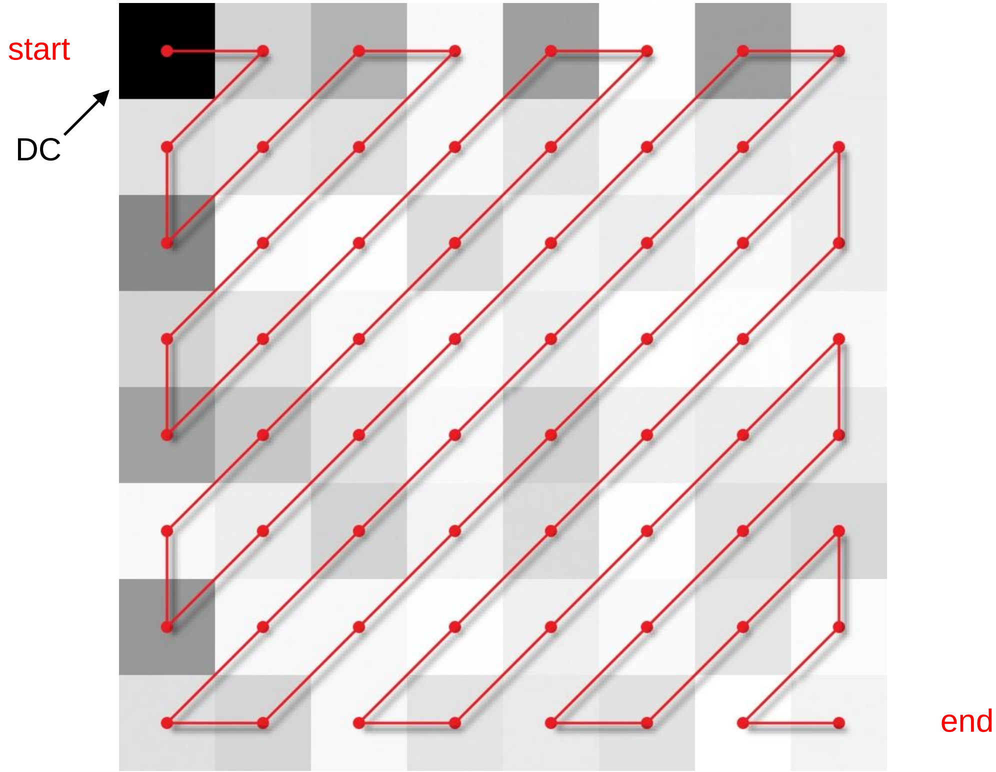


Differential PCM coding of same

DC components are coded separately from the AC components.

Quantized coefficient encoding order

$\boxed{255}$	23	34	7	15	2	6	1
19	4	10	4	5	1	2	2
29	5	6	1	2	1	1	1
11	9	3	3	1	1	1	1
14	9	2	1	2	1	1	1
3	1	3	1	2	1	1	2
7	1	1	1	1	1	1	1
0	0	1	1	1	0	1	1



AC: 23 19 29 4 34 7 10 5 11 14 9 6 4 15 2 5 1 3 9 3 7 1 2 3 2 1 6 1 2 1 1 1 3 1 0 0 1 1 2 1 1 2 1 1 1 2 1 1 1 1 1

Entropy Encoding of AC Coefficients

Two step process:

1. Convert zigzag sequence of quantized coefficients into a sequence of symbols.
2. Convert symbols to a data stream of variable length codes.

Symbol 1 = (Runlength, Size)
Symbol 2 = Amplitude

Entropy Encoding of Coefficients

Symbol 1: (`Runlength`, `Size`);

Symbol 2: `Amplitude`

AC:

`Runlength`: number of consecutive zero values (0-15)

`Size`: number of bits used to encode amplitude (1-10)

`Amplitude`: quantized value

DC:

`Runlength`: not included

`Size`: number of bits used to encode amplitude (1-11)

`Amplitude`: quantized value

Variable Length Entropy Encoding

size	amplitude
1	-1,1
2	-3,-2,2,3
3	-7…-4,4…7
4	-15…-8,8…15
5	-31…-16,16…31
6	-63…-32,32…63
7	-127…-64,64…128
8	-255…-128,128…255
9	-511…-256,256…511
10	-1023…-512,512…1023
DC only 11	-2047…-1024,1024…2047

Symbol 2 Coding

Effects of Quality Settings

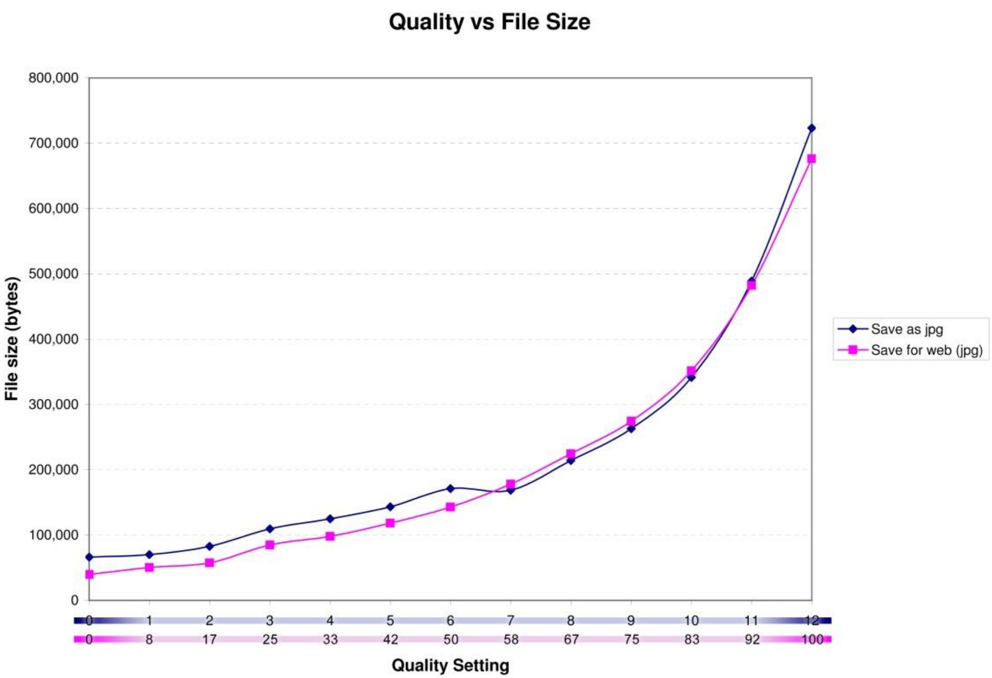


Image: `http://en.wikipedia.org/wiki/File:Quality_comparison_jpg_vs_saveforweb.jpg`

JPEG Compression

Original image is 5244w x 4716h @ 1200 ppi: 127MBytes

Yoyogi Park, Tokyo, October 1999. Photo by Alan Peters.

BMP

Original Image

File size in bytes
3,145,784

JPEG

Matlab's quality 100

File size in bytes
632,210

JPEG

Matlab's quality 50

File size in bytes
56,541

JPEG

Matlab's quality 10

File size in bytes
26,262

JPEG

Matlab's quality 0

File size in bytes
19,117

Quality numbers are different between programs…

JPEG

Photoshop's quality 0

File size in bytes
47,800

Photoshop's quality zero is similar to Matlab's quality 40…

JPEG

Matlab’s quality 40

File size in bytes
49,115

Does not matter because the quantization tables are in the file with the data. Each program uses that table when decoding the data.

JPEG

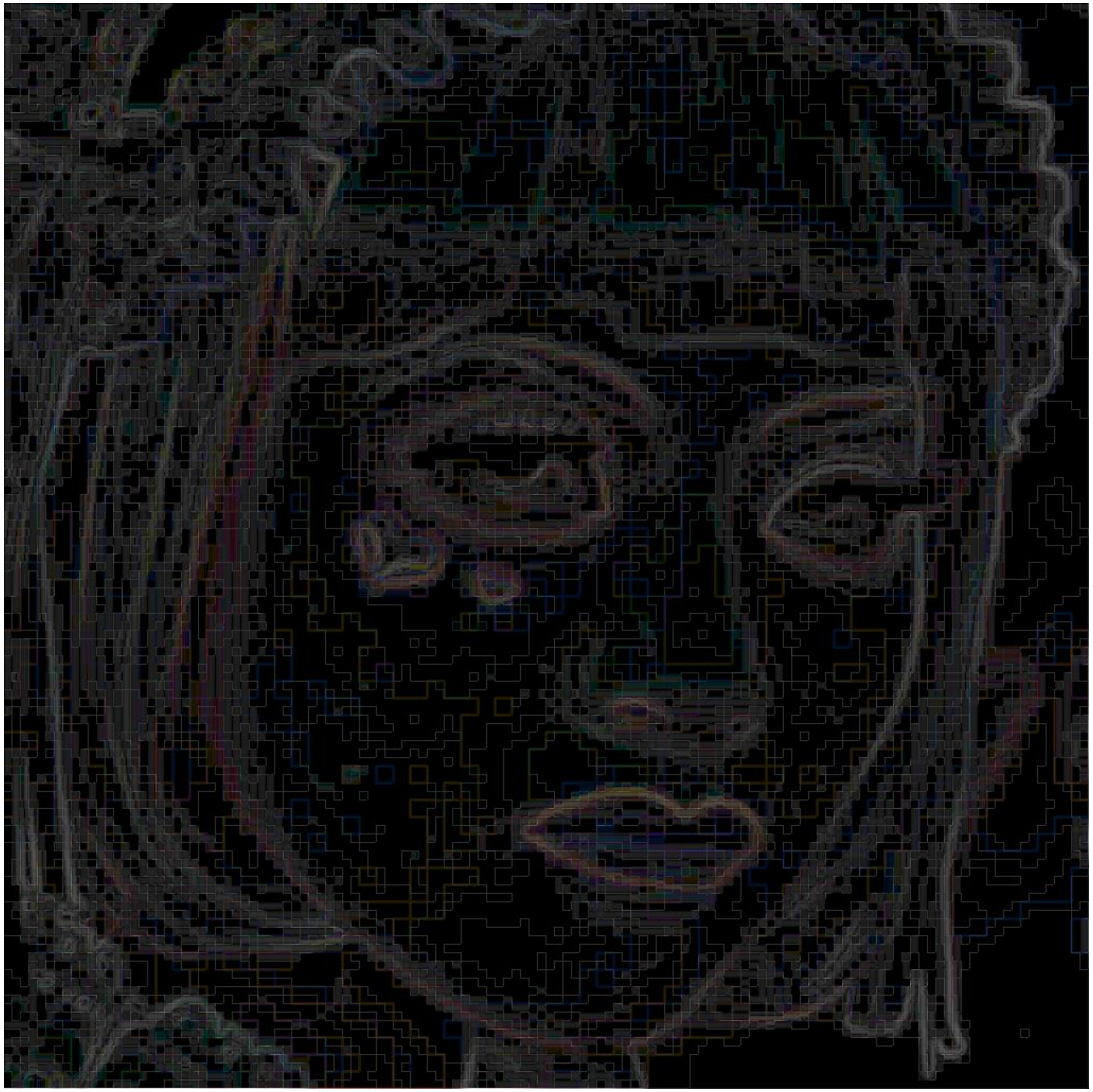
Magnitude of Gradient of Original image

JPEG

Magnitude of gradient of jpeg quality 100 image

JPEG

Magnitude of gradient of jpeg quality 50 image



JPEG

Magnitude of gradient of jpeg quality 10 image

JPEG

Magnitude of gradient of jpeg quality 0 image

JPEG

Gradient enhanced original image

JPEG

Gradient enhanced quality 100 image

JPEG

Gradient enhanced quality 50 image

JPEG

Gradient enhanced quality 10 image

JPEG

Gradient enhanced quality 0 image

Image Compression: JPEG

Image Compression: JPEG

JPEG quality level

No Compr. 197kB

JPEGQ: 12 76kB

JPEGQ: 6 21kB

JPGEQ: 0 12kB

File size in bytes

JFIF Information Display Program

JPEGsnoop - JPEG File Decoding Utility
by Calvin Hass © 2010

From the web page:

Every digital photo contains a wealth of hidden information -- JPEGsnoop was written to expose these details to those who are curious. Not only can one determine the various settings that were used in the digital camera in taking the photo (EXIF metadata, IPTC), but one can also extract information that indicates the quality and nature of the JPEG image compression used by the camera in saving the file. Digital cameras specify compression quality levels, many of them wildly different, leading to the fact that some cameras produce far better JPEG images than others.

`http://www.impulseadventure.com/photo/`

Example: JPEG/JFIF Encoded Image

Keith Hamshere – Soairse Ronan – 2007

Data Markers from the JPEG File Interchange Format (JFIF)

Short name	Code	Payload	Name	Comments
SOI	0xFFD8	none	Start Of Image	
SOF0	0xFFC0	variable size	Start Of Frame (Baseline)	Indicates that this is a baseline DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
SOF2	0xFFC2	variable size	Start Of Frame (Progressive)	Indicates that this is a progressive DCT-based JPEG, and specifies the width, height, number of components, and component subsampling (e.g., 4:2:0).
DHT	0xFFC4	variable size	Define Huffman Table(s)	Specifies one or more Huffman tables.
DQT	0xFFDB	variable size	Define Quantization Table(s)	Specifies one or more quantization tables.
DRI	0xFFDD	2 bytes	Define Restart Interval	Specifies the interval between RSTn markers, in macroblocks. This marker is followed by two bytes indicating the fixed size so it can be treated like any other variable size segment.
SOS	0xFFDA	variable size	Start Of Scan	Begins a top-to-bottom scan of the image. In baseline DCT JPEG images, there is generally a single scan. Progressive DCT JPEG images usually contain multiple scans. This marker specifies which slice of data it will contain, and is immediately followed by entropy-coded data.
RSTn	0xFFD0…D7	none	Restart	Inserted every r macroblocks, where r is the restart interval set by a DRI marker. Not used if there was no DRI marker. The low 3 bits of the marker code cycle in value from 0 to 7.
APPn	0xFFEn	variable size	Application-specific	For example, an Exif JPEG file uses an APP1 marker to store metadata, laid out in a structure based closely on TIFF.
COM	0xFFFE	variable size	Comment	Contains a text comment.
EOI	0xFFD9	none	End Of Image	

Source: ISO/IEC 10918-1:1994, p.36, via https://en.wikipedia.org/wiki/JPEG

JPEG / JFIF Information (via JPEGsnoop)

```
Filename: [Saoirse Ronan - Ember.jpg]
Filesize: [3156857] Bytes

Start Offset: 0x00000000
*** Marker: SOI (xFFD8) ***
  OFFSET: 0x00000000

*** Marker: APP13 (xFFED) ***
  OFFSET: 0x00000002
  length          = 28
  Identifier      = [Photoshop 3.0]
    8BIM: [0x0404] Name=[] Len=[0x0000]
    IPTC [0xFFE1:002] ? size=12392

*** Marker: APP1 (xFFE1) ***
  OFFSET: 0x00000020
  length      = 560
  Identifier   = [http://ns.adobe.com/xap/1.0/]
    XMP =

*** Marker: APP0 (xFFE0) ***
  OFFSET: 0x00000252
  length    = 16
  identifier = [JFIF]
  version   = [1.2]
  density   = 300 x 300 DPI (dots per inch)
  thumbnail  = 0 x 0
```

```
*** Marker: APP2 (xFFE2) ***
  OFFSET: 0x00000264
  length        = 576
  Identifier     = [ICC_PROFILE]
    ICC Profile:
      Marker Number = 1 of 1
        Profile Size  : 560 bytes
        Preferred CMM Type          : 'ADBE' (0x41444245)
        Profile Version             : 0.2.1.0 (0x02100000)
        Profile Device/Class        : Display Device profile ('mntr' (0x6D6E7472))
        Data Colour Space           : rgbData ('RGB ' (0x52474220))
        Profile connection space (PCS)  : 'XYZ ' (0x58595A20)
        Profile creation date       : 1999-06-03 00:00:00
        Profile file signature      : 'acsp' (0x61637370)
        Primary platform            : Apple Computer, Inc. ('APPL' (0x4150504C))
        Profile flags               : 0x00000000
        Profile flags               > Profile not embedded
        Profile flags               > Profile can't be used independently of embedded
        Device Manufacturer         : 'none' (0x6E6F6E65)
        Device Model                : '....' (0x00000000)
        Device attributes           : 0x00000000_00000000
        Device attributes           > Reflective
        Device attributes           > Glossy
        Device attributes           > Media polarity = negative
        Device attributes           > Black & white media
        Rendering intent            : Perceptual
        Profile creator             : 'ADBE' (0x41444245)
        Profile ID                  : 0x00000000_00000000_00000000
```

JPEG / JFIF Information (via JPEGsnoop)

```
*** Marker: DQT (xFFDB) ***
  Define a Quantization Table.
  OFFSET: 0x000004B6
  Table length = 132
  ----
  Precision=8 bits
  Destination ID=0 (Luminance)
    DQT, Row #0:   2   2   3   4   5   6   8  11
    DQT, Row #1:   2   2   2   4   5   7   9  11
    DQT, Row #2:   3   2   3   5   7   9  11  12
    DQT, Row #3:   4   4   5   7   9  11  12  12
    DQT, Row #4:   5   5   7   9  11  12  12  12
    DQT, Row #5:   6   7   9  11  12  12  12  12
    DQT, Row #6:   8   9  11  12  12  12  12  12
    DQT, Row #7:  11  11  12  12  12  12  12  12
   Approx quality factor = 91.64 (scaling=16.71 variance=22.54)
  - ---
  Precision=8 bits
  Destination ID=1 (Chrominance)
    DQT, Row #0:   3   3   7  13  15  15  15  15
    DQT, Row #1:   3   4   7  13  14  12  12  12
    DQT, Row #2:   7   7  13  14  12  12  12  12
    DQT, Row #3:  13  13  14  12  12  12  12  12
    DQT, Row #4:  15  14  12  12  12  12  12  12
    DQT, Row #5:  15  12  12  12  12  12  12  12
    DQT, Row #6:  15  12  12  12  12  12  12  12
    DQT, Row #7:  15  12  12  12  12  12  12  12
    Approx quality factor = 92.57 (scaling=14.85 variance=23.00)
```

```
*** Marker: SOF0 (Baseline DCT) (xFFC0) ***
  OFFSET: 0x0000053C
  Frame header length = 17
  Precision = 8
  Number of Lines = 2592
  Samples per Line = 3872
  Image Size = 3872 x 2592
  Raw Image Orientation = Landscape
  Number of Img components = 3
    Comp[1]: ID=0x01, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x00 (Lum:   Y)
    Comp[2]: ID=0x02, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x01 (Chrom: Cb)
    Comp[3]: ID=0x03, Samp Fac=0x11 (Subsamp 1 x 1), Quant Tbl Sel=0x01 (Chrom: Cr)

*** Marker: DRI (Restart Interval) (xFFDD) ***
  OFFSET: 0x0000054F
  length     = 4
  interval   = 484
```

JPEG / JFIF Information (via JPEGsnoop)

```
*** Marker: DHT (Define Huffman Table) (xFFC4) ***
  OFFSET: 0x00000555
  Huffman table length = 418
  ----
  Destination ID = 0
  Class = 0 (DC / Lossless Table)
    Codes of length 01 bits (000 total):
    Codes of length 02 bits (000 total):
    Codes of length 03 bits (007 total): 04 05 03 02 06 01 00
    Codes of length 04 bits (001 total): 07
    Codes of length 05 bits (001 total): 08
    Codes of length 06 bits (001 total): 09
    Codes of length 07 bits (001 total): 0A
    Codes of length 08 bits (001 total): 0B
    Codes of length 09 bits (000 total):
    Codes of length 10 bits (000 total):
    Codes of length 11 bits (000 total):
    Codes of length 12 bits (000 total):
    Codes of length 13 bits (000 total):
    Codes of length 14 bits (000 total):
    Codes of length 15 bits (000 total):
    Codes of length 16 bits (000 total):
    Total number of codes: 012
```

```
  ----
  Destination ID = 1
  Class = 0 (DC / Lossless Table)
    Codes of length 01 bits (000 total):
    Codes of length 02 bits (002 total): 01 00
    Codes of length 03 bits (002 total): 02 03
    Codes of length 04 bits (003 total): 04 05 06
    Codes of length 05 bits (001 total): 07
    Codes of length 06 bits (001 total): 08
    Codes of length 07 bits (001 total): 09
    Codes of length 08 bits (001 total): 0A
    Codes of length 09 bits (001 total): 0B
    Codes of length 10 bits (000 total):
    Codes of length 11 bits (000 total):
    Codes of length 12 bits (000 total):
    Codes of length 13 bits (000 total):
    Codes of length 14 bits (000 total):
    Codes of length 15 bits (000 total):
    Codes of length 16 bits (000 total):
    Total number of codes: 012
```

JPEG / JFIF Information (via JPEGsnoop)

```
----
Destination ID = 0
Class = 1 (AC Table)
  Codes of length 01 bits (000 total):
  Codes of length 02 bits (002 total): 01 02
  Codes of length 03 bits (001 total): 03
  Codes of length 04 bits (003 total): 11 04 00
  Codes of length 05 bits (003 total): 05 21 12
  Codes of length 06 bits (002 total): 31 41
  Codes of length 07 bits (004 total): 51 06 13 61
  Codes of length 08 bits (002 total): 22 71
  Codes of length 09 bits (006 total): 81 14 32 91 A1 07
  Codes of length 10 bits (007 total): 15 B1 42 23 C1 52 D1
  Codes of length 11 bits (003 total): E1 33 16
  Codes of length 12 bits (004 total): 62 F0 24 72
  Codes of length 13 bits (002 total): 82 F1
  Codes of length 14 bits (006 total): 25 43 34 53 92 A2
  Codes of length 15 bits (002 total): B2 63
  Codes of length 16 bits (115 total):
    73 C2 35 44 27 93 A3 B3 36 17 54 64 74 C3 D2 E2
    08 26 83 09 0A 18 19 84 94 45 46 A4 B4 56 D3 55
    28 1A F2 E3 F3 C4 D4 E4 F4 65 75 85 95 A5 B5 C5
    D5 E5 F5 66 76 86 96 A6 B6 C6 D6 E6 F6 37 47 57
    67 77 87 97 A7 B7 C7 D7 E7 F7 38 48 58 68 78 88
    98 A8 B8 C8 D8 E8 F8 29 39 49 59 69 79 89 99 A9
    B9 C9 D9 E9 F9 2A 3A 4A 5A 6A 7A 8A 9A AA BA CA
    DA EA FA
  Total number of codes: 162
```

```
----
Destination ID = 1
Class = 1 (AC Table)
  Codes of length 01 bits (000 total):
  Codes of length 02 bits (002 total): 01 00
  Codes of length 03 bits (002 total): 02 11
  Codes of length 04 bits (001 total): 03
  Codes of length 05 bits (002 total): 04 21
  Codes of length 06 bits (003 total): 12 31 41
  Codes of length 07 bits (005 total): 05 51 13 61 22
  Codes of length 08 bits (005 total): 06 71 81 91 32
  Codes of length 09 bits (004 total): A1 B1 F0 14
  Codes of length 10 bits (005 total): C1 D1 E1 23 42
  Codes of length 11 bits (006 total): 15 52 62 72 F1 33
  Codes of length 12 bits (004 total): 24 34 43 82
  Codes of length 13 bits (008 total): 16 92 53 25 A2 63 B2 C2
  Codes of length 14 bits (003 total): 07 73 D2
  Codes of length 15 bits (003 total): 35 E2 44
  Codes of length 16 bits (109 total):
    83 17 54 93 08 09 0A 18 19 26 36 45 1A 27 64 74
    55 37 F2 A3 B3 C3 28 29 D3 E3 F3 84 94 A4 B4 C4
    D4 E4 F4 65 75 85 95 A5 B5 C5 D5 E5 F5 46 56 66
    76 86 96 A6 B6 C6 D6 E6 F6 47 57 67 77 87 97 A7
    B7 C7 D7 E7 F7 38 48 58 68 78 88 98 A8 B8 C8 D8
    E8 F8 39 49 59 69 79 89 99 A9 B9 C9 D9 E9 F9 2A
    3A 4A 5A 6A 7A 8A 9A AA BA CA DA EA FA
  Total number of codes: 162
```

JPEG / JFIF Information (via JPEGsnoop)

```
*** Marker: SOS (Start of Scan) (xFFDA) ***
 OFFSET: 0x000006F9
 Scan header length = 12
 Number of img components = 3
    Component[1]: selector=0x01, table=0x00
    Component[2]: selector=0x02, table=0x11
    Component[3]: selector=0x03, table=0x11
 Spectral selection = 0 .. 63
 Successive approximation = 0x00

*** Decoding SCAN Data ***
 OFFSET: 0x00000707
 Scan Decode Mode: No IDCT (DC only)
    NOTE: Low-resolution DC component shown.

 Scan Data encountered marker
    0xFFD9 @ 0x00302B77.0

 Compression stats:
    Compression Ratio: 3028.43:1
    Bits per pixel:     0.01:1
```

```
Huffman code histogram stats:
 Huffman Table: (Dest ID: 0, Class: DC)
  # codes of length 01 bits:        0 (  0%)
  # codes of length 02 bits:        0 (  0%)
  # codes of length 03 bits:   142702 ( 91%)
  # codes of length 04 bits:     8490 (  5%)
  # codes of length 05 bits:     3974 (  3%)
  # codes of length 06 bits:     1649 (  1%)
  # codes of length 07 bits:        1 (  0%)
  # codes of length 08 bits:        0 (  0%)
  # codes of length 09 bits:        0 (  0%)
  # codes of length 10 bits:        0 (  0%)
  # codes of length 11 bits:        0 (  0%)
  # codes of length 12 bits:        0 (  0%)
  # codes of length 13 bits:        0 (  0%)
  # codes of length 14 bits:        0 (  0%)
  # codes of length 15 bits:        0 (  0%)
  # codes of length 16 bits:        0 (  0%)
```

```
 Huffman Table: (Dest ID: 1, Class: DC)
  # codes of length 01 bits:        0 (  0%)
  # codes of length 02 bits:   162056 ( 52%)
  # codes of length 03 bits:   128895 ( 41%)
  # codes of length 04 bits:    22673 (  7%)
  # codes of length 05 bits:        8 (  0%)
  # codes of length 06 bits:        0 (  0%)
  # codes of length 07 bits:        0 (  0%)
  # codes of length 08 bits:        0 (  0%)
  # codes of length 09 bits:        0 (  0%)
  # codes of length 10 bits:        0 (  0%)
  # codes of length 11 bits:        0 (  0%)
  # codes of length 12 bits:        0 (  0%)
  # codes of length 13 bits:        0 (  0%)
  # codes of length 14 bits:        0 (  0%)
  # codes of length 15 bits:        0 (  0%)
  # codes of length 16 bits:        0 (  0%)
```

JPEG / JFIF Information (via JPEGsnoop)

Huffman Table: (Dest ID: 0, Class: AC)

# codes of length 01 bits:	0 (0%)
# codes of length 02 bits:	2103874 (53%)
# codes of length 03 bits:	393151 (10%)
# codes of length 04 bits:	712367 (18%)
# codes of length 05 bits:	368952 (9%)
# codes of length 06 bits:	148211 (4%)
# codes of length 07 bits:	116694 (3%)
# codes of length 08 bits:	52460 (1%)
# codes of length 09 bits:	49772 (1%)
# codes of length 10 bits:	19533 (0%)
# codes of length 11 bits:	2118 (0%)
# codes of length 12 bits:	2566 (0%)
# codes of length 13 bits:	726 (0%)
# codes of length 14 bits:	367 (0%)
# codes of length 15 bits:	24 (0%)
# codes of length 16 bits:	1861 (0%)

Huffman Table: (Dest ID: 1, Class: AC)

# codes of length 01 bits:	0 (0%)
# codes of length 02 bits:	580346 (62%)
# codes of length 03 bits:	209085 (22%)
# codes of length 04 bits:	25344 (3%)
# codes of length 05 bits:	50682 (5%)
# codes of length 06 bits:	50395 (5%)
# codes of length 07 bits:	10862 (1%)
# codes of length 08 bits:	2482 (0%)
# codes of length 09 bits:	2378 (0%)
# codes of length 10 bits:	941 (0%)
# codes of length 11 bits:	273 (0%)
# codes of length 12 bits:	0 (0%)
# codes of length 13 bits:	1 (0%)
# codes of length 14 bits:	0 (0%)
# codes of length 15 bits:	0 (0%)
# codes of length 16 bits:	0 (0%)

YCC clipping in DC:
Y component: [<0= 0] [>255= 0]
Cb component: [<0= 0] [>255= 0]
Cr component: [<0= 0] [>255= 0]

RGB clipping in DC:
R component: [<0= 0] [>255= 0]
G component: [<0= 0] [>255= 0]
B component: [<0= 0] [>255= 0]

Average Pixel Luminance (Y):
Y=[86] (range: 0..255)

Brightest Pixel Search:
YCC=[1016, 0, 0]
RGB=[255,255,255] @ MCU[94, 1]

Finished Decoding SCAN Data
Number of RESTART markers decoded: 323
Next position in scan buffer:
Offset 0x00302B75.1

JPEG / JFIF Information (via JPEGsnoop)

```
*** Marker: EOI (End of Image) (xFFD9) ***
  OFFSET: 0x00302B77

*** Searching Compression Signatures ***

  Signature:                 0166B0BC0B82C8233430BF67FA31C829
  Signature (Rotated):       0166B0BC0B82C8233430BF67FA31C829
  File Offset:               0 bytes
  Chroma subsampling:        1x1
  EXIF Make/Model:           NONE
  EXIF Makernotes:           NONE
  EXIF Software:             NONE

    Searching Compression Signatures: (3327 built-in, 0 user(*) )

          EXIF.Make / Software        EXIF.Model                            Quality           Subsamp Match?
          -------------------------   -----------------------------------   ----------------  --------------
          SW :[Adobe Photoshop]                                             [Save As 10]

    Based on the analysis of compression characteristics and EXIF metadata:

    ASSESSMENT: Class 1 - Image is processed/edited

Position Marked @ MCU=[ 177,  52](0,0) YCC=[ -468,  -60,   30]
```