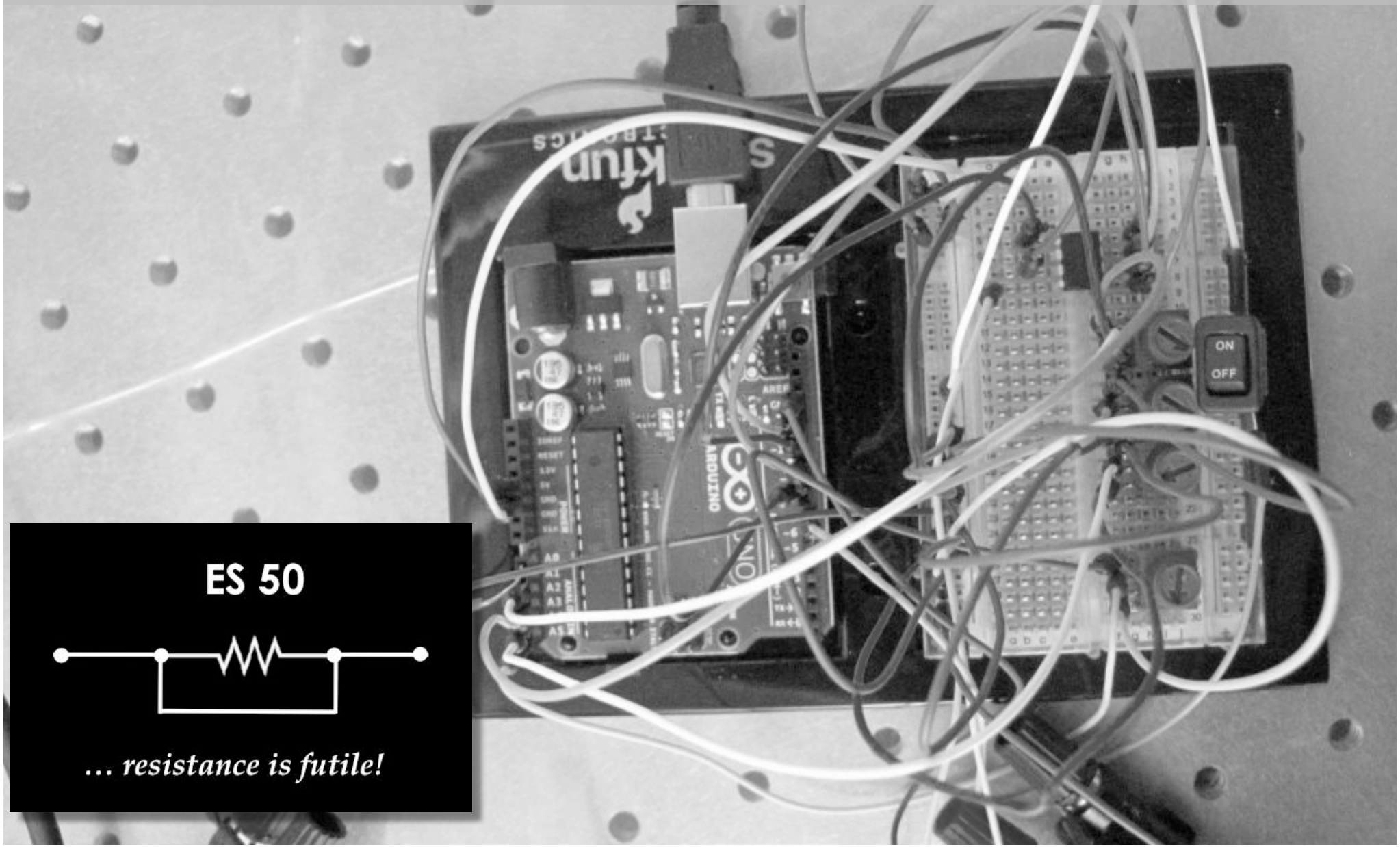
ES 50: Introduction to Electrical Eng.

Explore EE, satisfy a Gen Ed requirement & have fun



ES 50

… resistance is futile!

Lecture 16: Images Processing, Color, & Compression

November 11, 2015

HARVARD

From last time

From Audio to Visual

- Audio = '1 dimensional', vector of amplitudes (versus time)
- Visual (aka images) = MATRIX of PIXELS

Commonality of signal processing

- Sampling, quantization, 'aliasing'
- The importance of frequency components

Gray scale images

- An image in which each pixel carries only the intensity information (no color associated with it)
- Represented by a matrix with each element (pixel) being an intensity value
- Typically, we use a BYTE (8 bits), with 256 shades of gray
  - A brightness step of 1/256 is less than the eye can perceive!*

Image processing: binary (black & white)

HARVARD

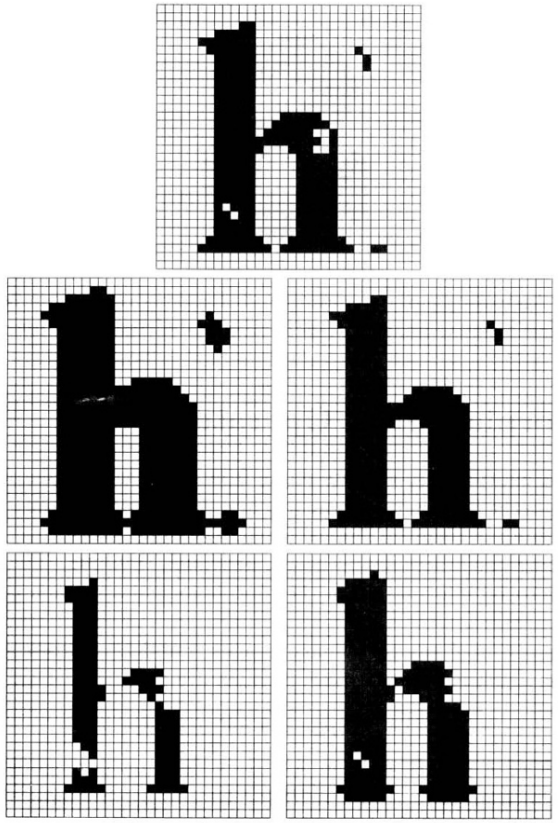
** anticipating: how much information is 'good enough'?*

‘Filtering’ the image:

Close, but ‘no cigar’…..

Try to expand the region to fill the gaps

Try to shrink the region to remove noise



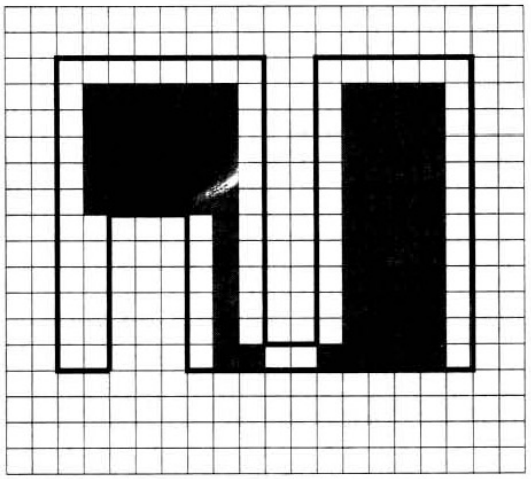
noisy image

expanding followed by shrinking filled holes but did not eliminate noise

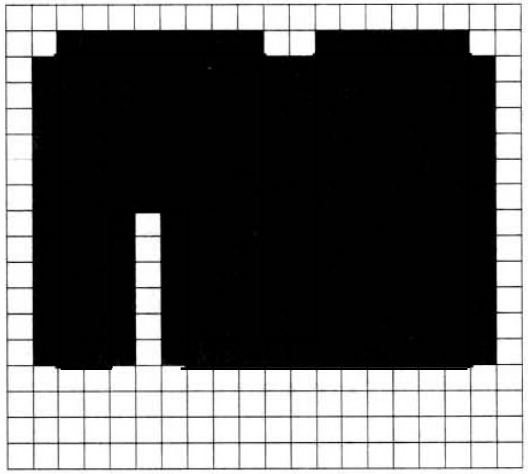
shrinking followed by expanding eliminated noise but did not fill the holes

HARVARD

Erosion and Dilation



the erosion of the original image with the *SE*
the bold line shows the border of the original image



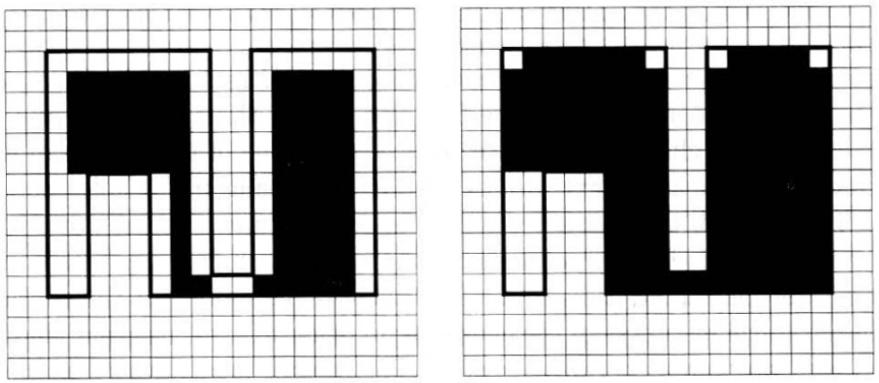
the dilation of the original image with the *SE*
the bold line shows the border of the original image

HARVARD

Opening and Closing

opening

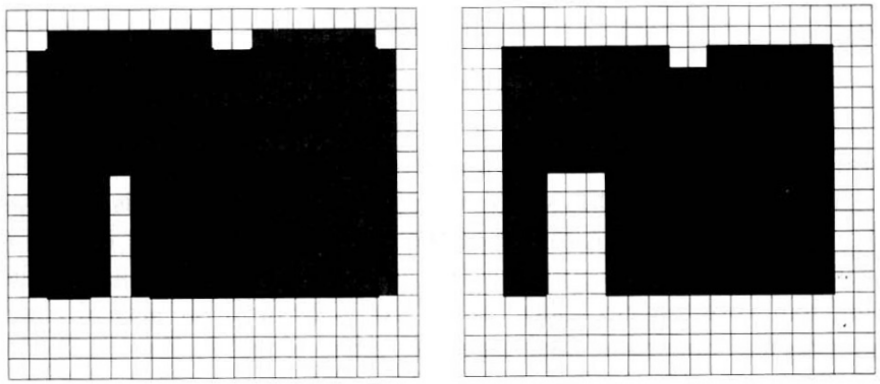
initial erosion



succeeding dilation

closing

initial dilation



succeeding erosion

HARVARD

Image Processing (using black & white images)

The transformation of one 2D image to another with the purpose of extracting information, image enhancement, artistic editing, etc…

We introduced following image processing techniques

- ***Thresholding*** $J(x,y) = 1$, If $I(x,y) \geq T$, otherwise, $J(x,y) = 0$, If $I(x,y) < T$
- ***Histogram*** of an image *H(r) = # of pixels with intensity level r* in image I. Histogram is used to determine appropriate threshold T
- ***Background subtraction*** $J(x,y) = |I(x,y) - B(x,y)|$, where B$(x,y)$ is background image
- ***Background addition***
- ***Dilation*** operation that grows objects in an image, according to the the *structuring element* used (used to fill in the gaps, etc)
- ***Erosion*** operation that shrinks objects in an image, according to the the *structuring element* used (used to remove noise specks, etc)
- ***Opening*** erosion followed by dilation
- ***Closing*** dilation followed by erosion

HARVARD

Opening and Closing

OPENING: erosion followed by dilation with the same SE

- Smooth boundaries are preserved
- Jagged boundaries are smoothed
- Thin connections are broken
- Small objects and protrusions are removed

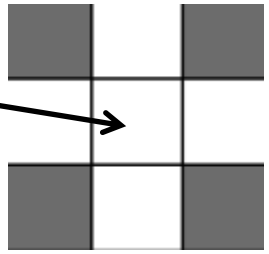
CLOSING: dilation followed by erosion with the same SE

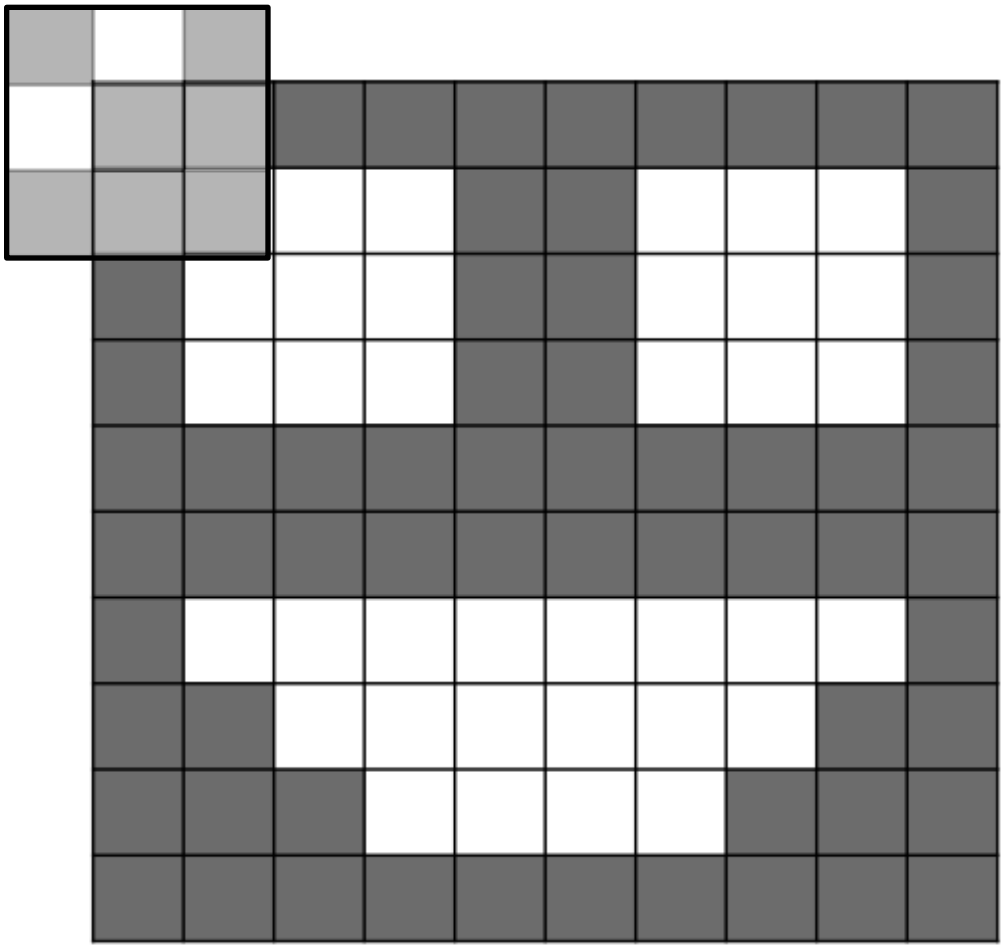
- Smooth boundaries are preserved
- Jagged boundaries are smoothed
- Narrow breaks are joined
- Small holes and long gulfs are filled

Let's do a simple example

Perform Opening

This is the center of the structuring element, THE ENTIRE SHAPE IS THE STRUCTURING ELEMENT





0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

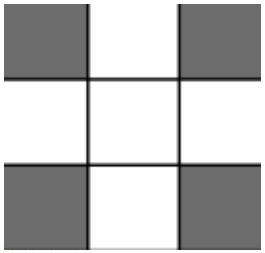
Erosion: center of SE at a pixel
Entire SE within region, then pixel value = 1
Otherwise, pixel value = 0

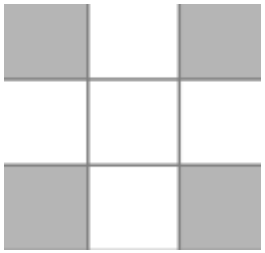
HARVARD

First Erosion

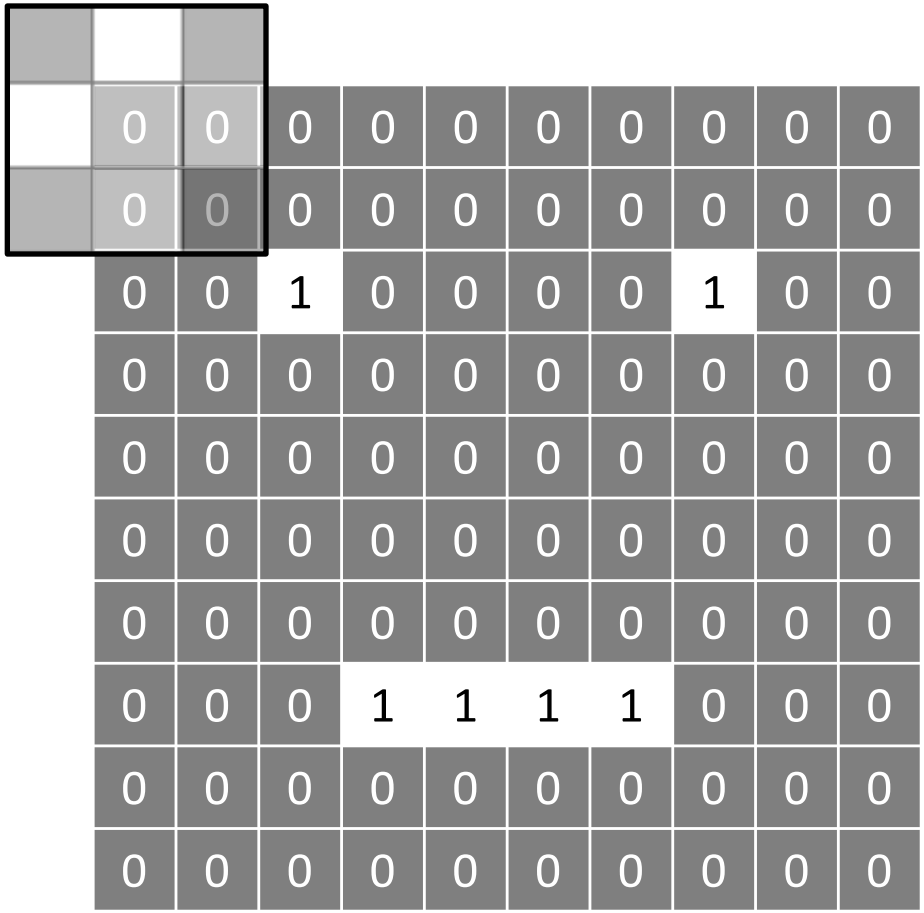
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

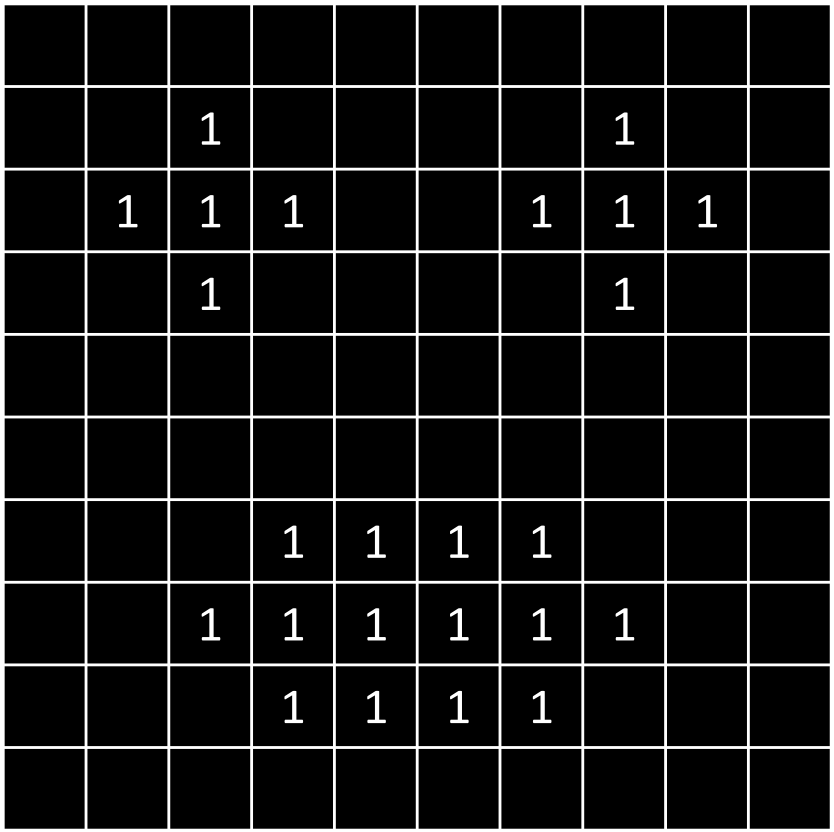
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

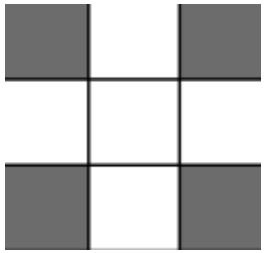




Then Dilation







Dilation: center of SE at a pixel

<u>If at least I pixel</u> of SE within region, then pixel value = 1

Otherwise, pixel value = 0

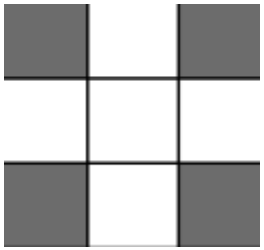
Then Dilation

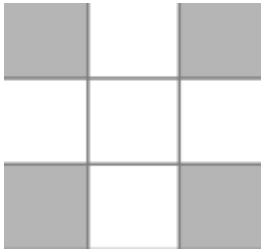
Erosion Only

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Erosion + Dilation

0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

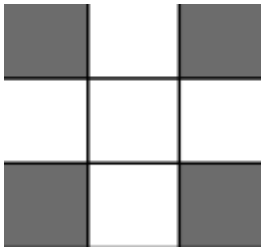


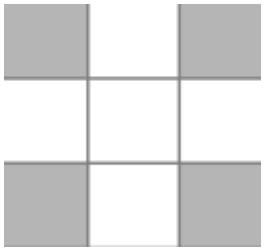


Post Dilation

0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0

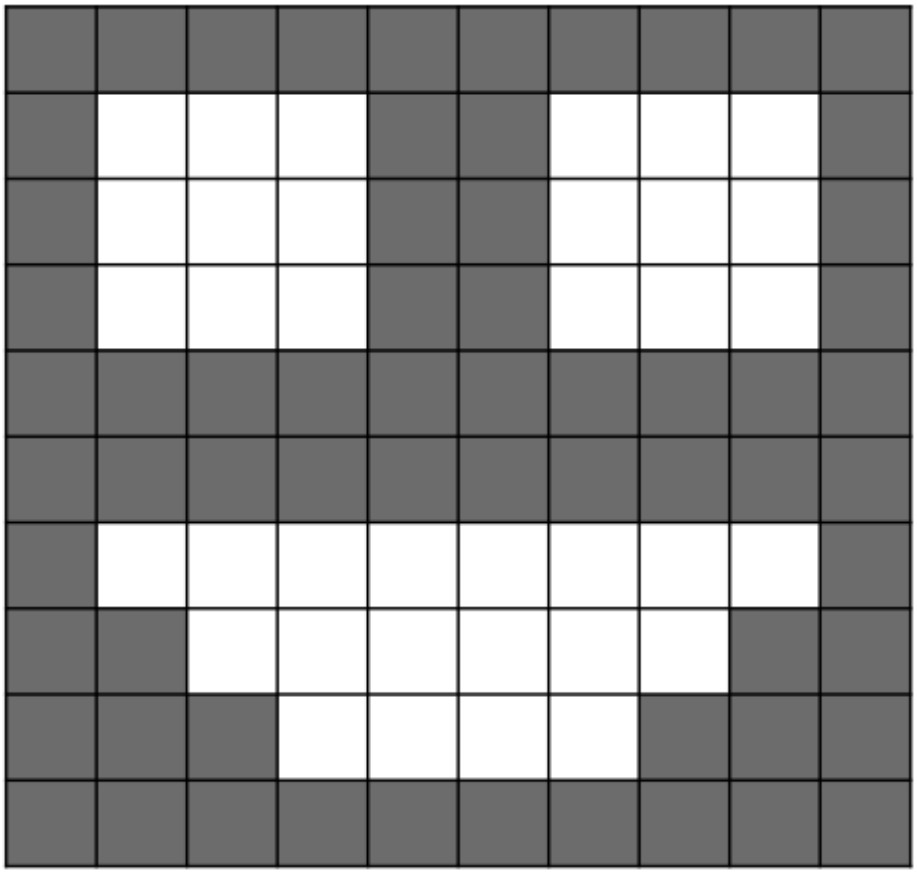
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0





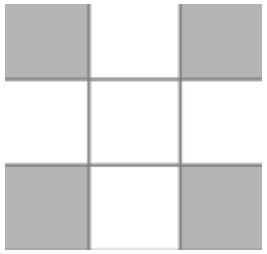
Opening, when everything is said and done ☺

Before

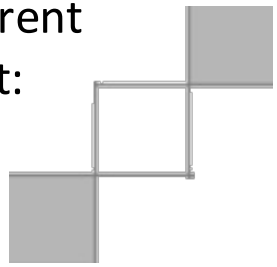


After

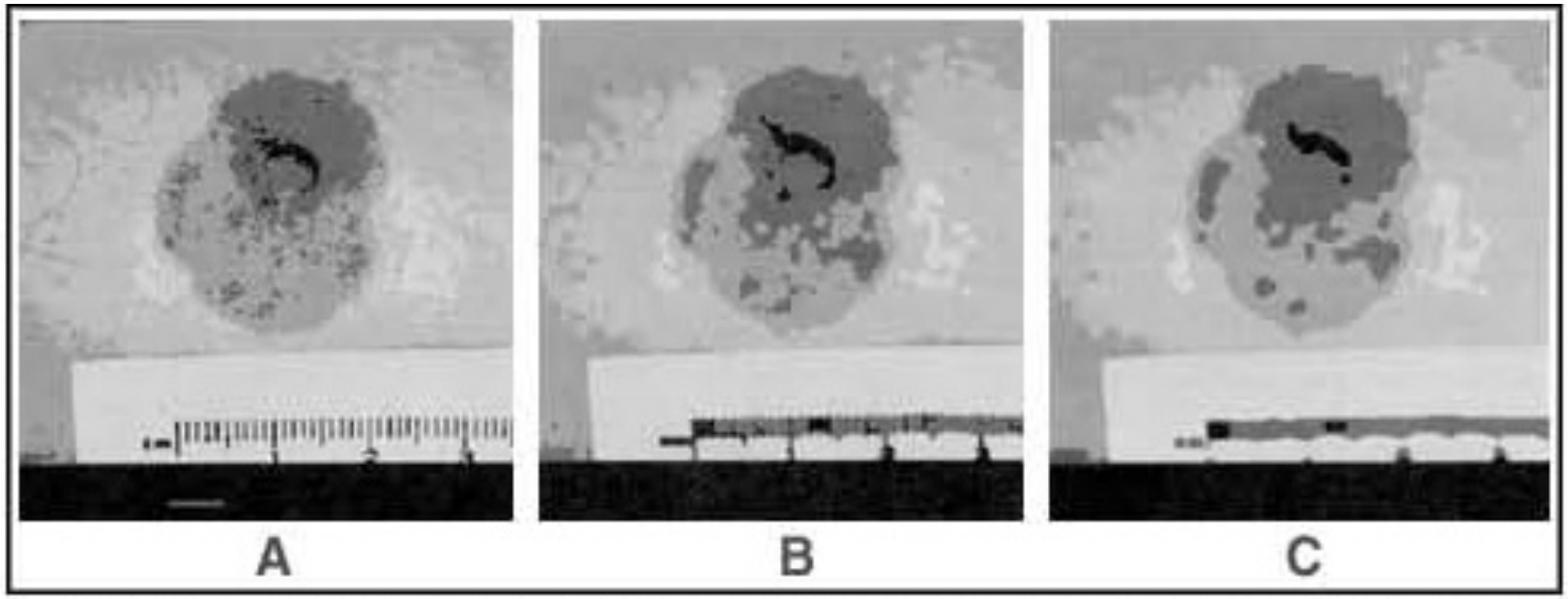
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1	0	0
0	1	1	1	0	0	1	1	1	0
0	0	1	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0



Try this with a different Structuring Element:



Opening and Closing of a Tumor Image



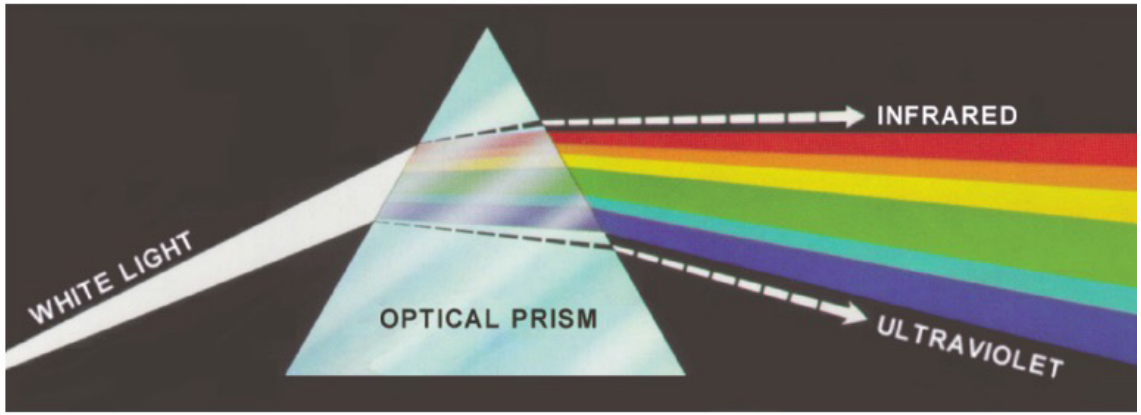
A) Original segmented tumor image
B) Image (A) after morphological opening using a 5x5 circular structuring element
C) Image (B) after morphological closing using a 5x5 circular structuring element

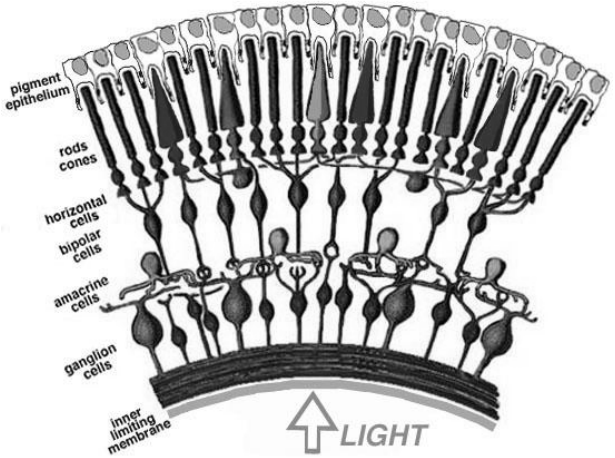
Opening: smooth contours, break narrow connections, eliminate small objects
Closing: fill in gaps, eliminate small holes

HARVARD

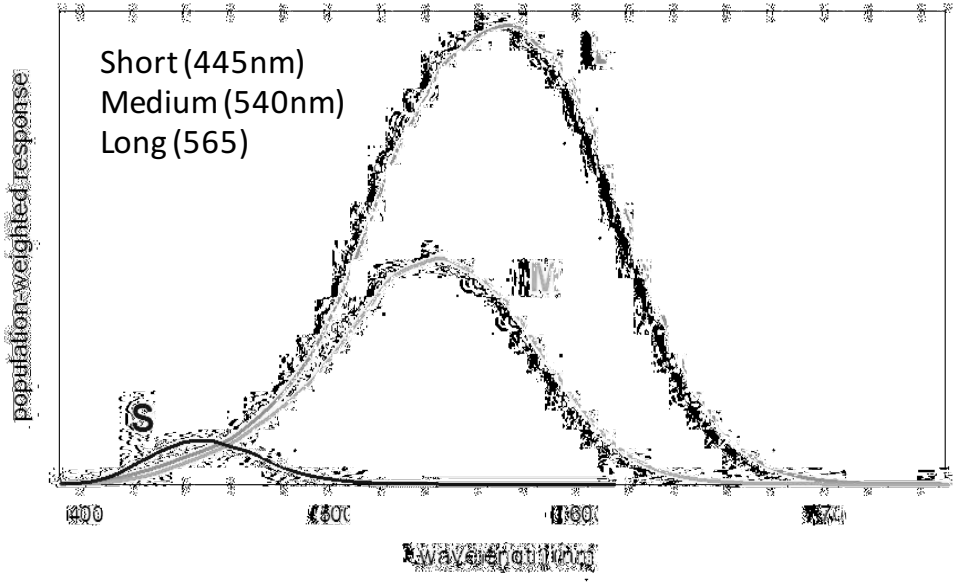
Color Images

RODS: low-light sensors
CONES: distinguish color, with higher incident light

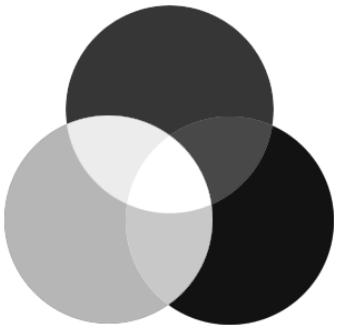




6 to 7 million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue.
65%: Red
33%: Green
2%: Blue (blue cones are the most sensitive)



RGB components

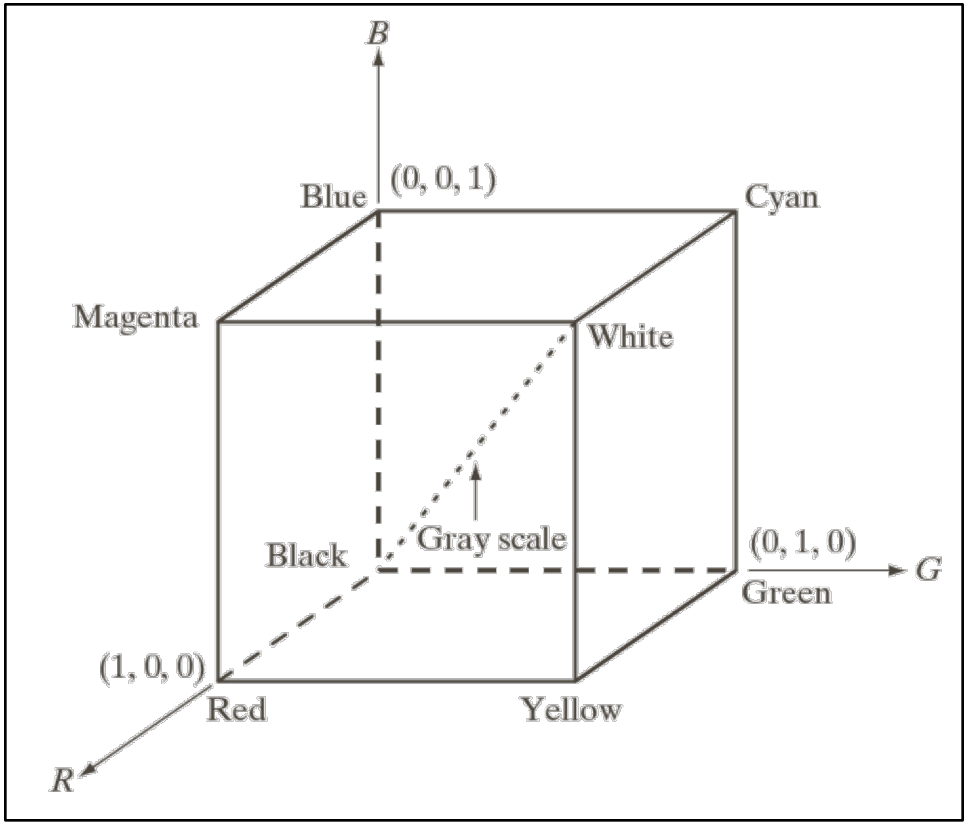


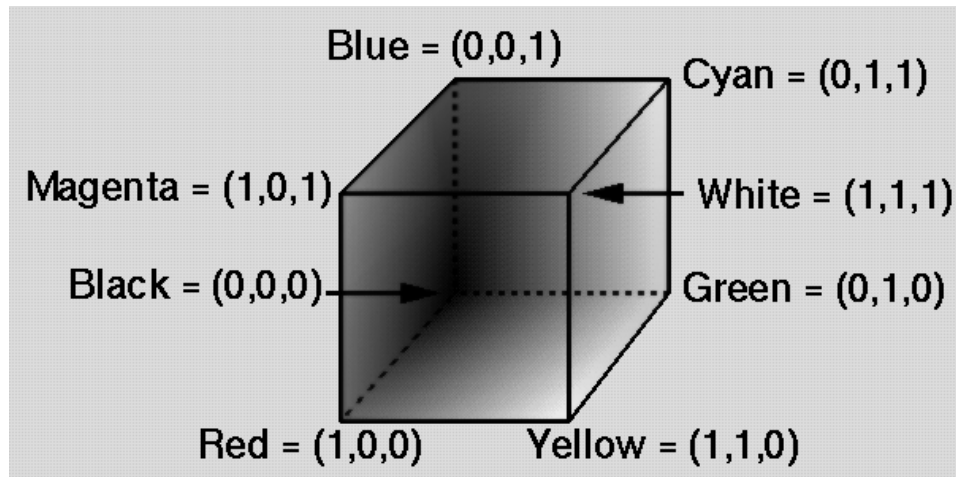
Absorption of light by the red, green and blue cones in the human eye as a function of wavelength

RGB Tristimulus

Red, green, and blue are denoted X, Y, & Z, respectively. A color is defined by its trichromatic coefficients, defined as:

$$R = \frac{X}{X+Y+Z};\ G = \frac{Y}{X+Y+Z};\ B = \frac{Z}{X+Y+Z}$$

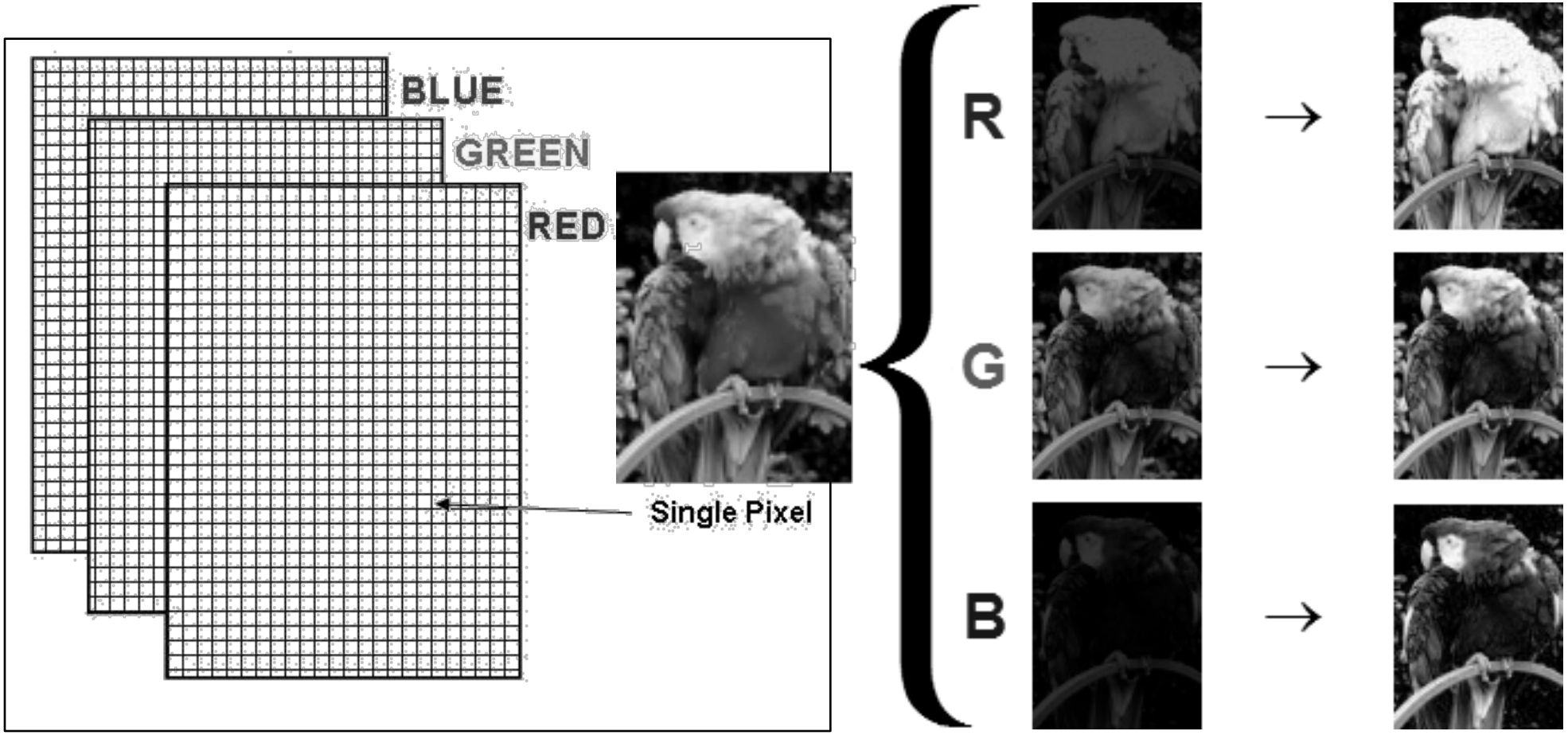




Schematic of the RGB color cube. Points along the main diagonal have gray values from black at the origin to white at point (1,1,1)

Choice of RGB primaries related to response of the human eye

RGB Color Images



To represent a color image we need <u>three matrices</u>: one for each red, green and blue channel;

red channel: *color_image(:, :, 1);*
green channel: *color_image(:, :, 2);*
blue channel: *color_image(:, :, 3);*

RGB Images: More information, but at a cost

MORE DATA, GREATER CHALLENGES

- 1 second of digital audio information ~ 8 kilobytes
- Gray-scale image: 1 byte/pixel
  - Assume 1024 x 768 pixels ➔ 800 kilobytes
- A 'true color' image: 3 bytes/pixel
  - For the same 1024 x 768 pixels ➔ 2.4 Megabytes!

A more limited representation: we don't need ALL RGB for *every pixel.*
Let's use a PALETTE of colors that would well represent our colored image.

Palette

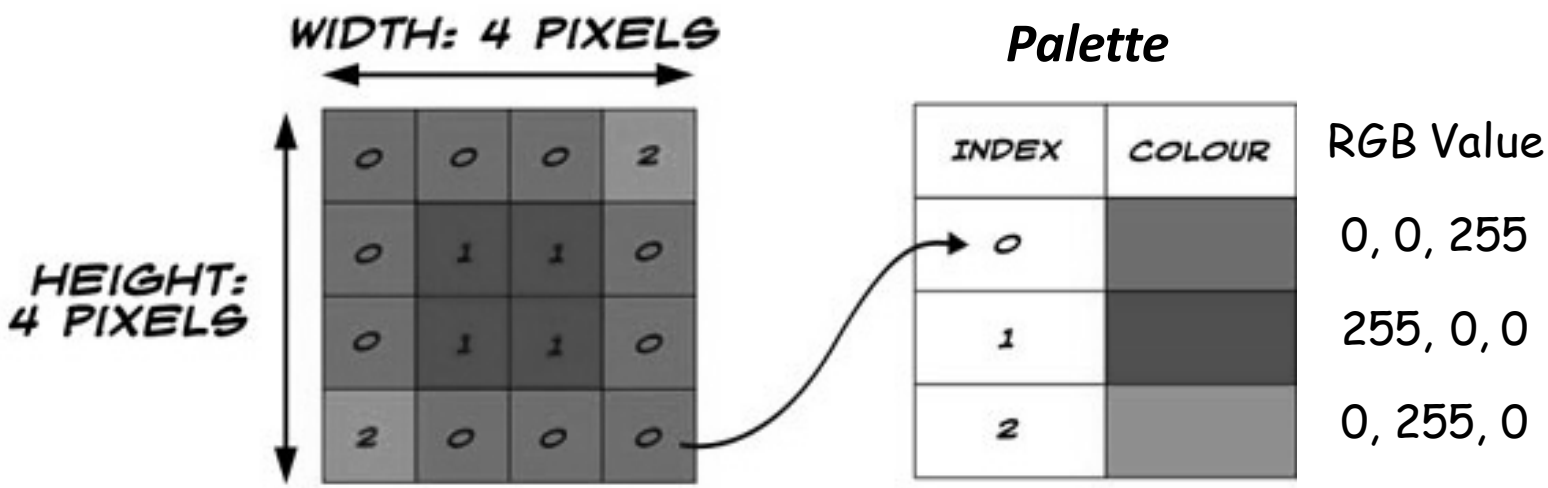
HARVARD

Indexed Color

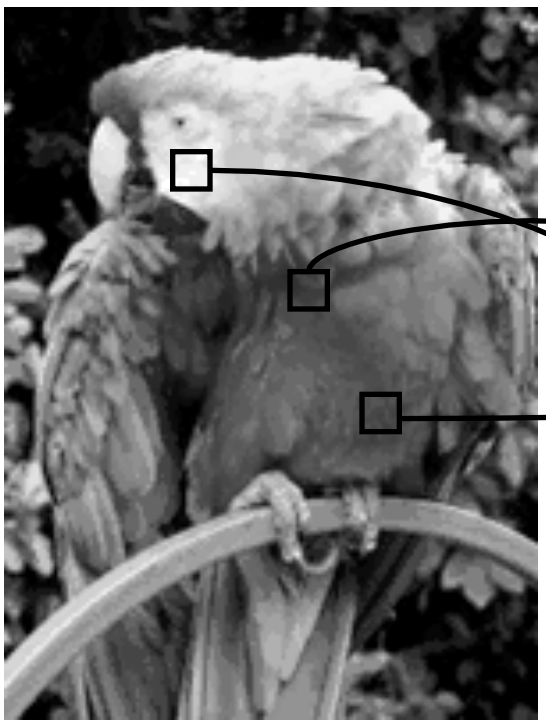
The color information is not directly carried by the image pixel, but it is stored into a separate set of data called a PALETTE.

- The PALETTE is an array in which every element represents a color. Common palettes have 256 colors.

Each pixel has a *number* which directs us to a *color* in the palette. This method of identifying color is sometimes called *pseudocolor* or *indirect color.*



Indexed Image



Note, <u>each</u> palette color has an R,G,B value

Index to Palette

	R	G	B	
				0
	255	**5**	**5**	
	255	**255**	**255**	
				255

Q: What is the size of the indexed image (for a palette with 256 colors)?

A: $Size(Indexed) = H * W\ bytes + 256 * 3\ bytes$

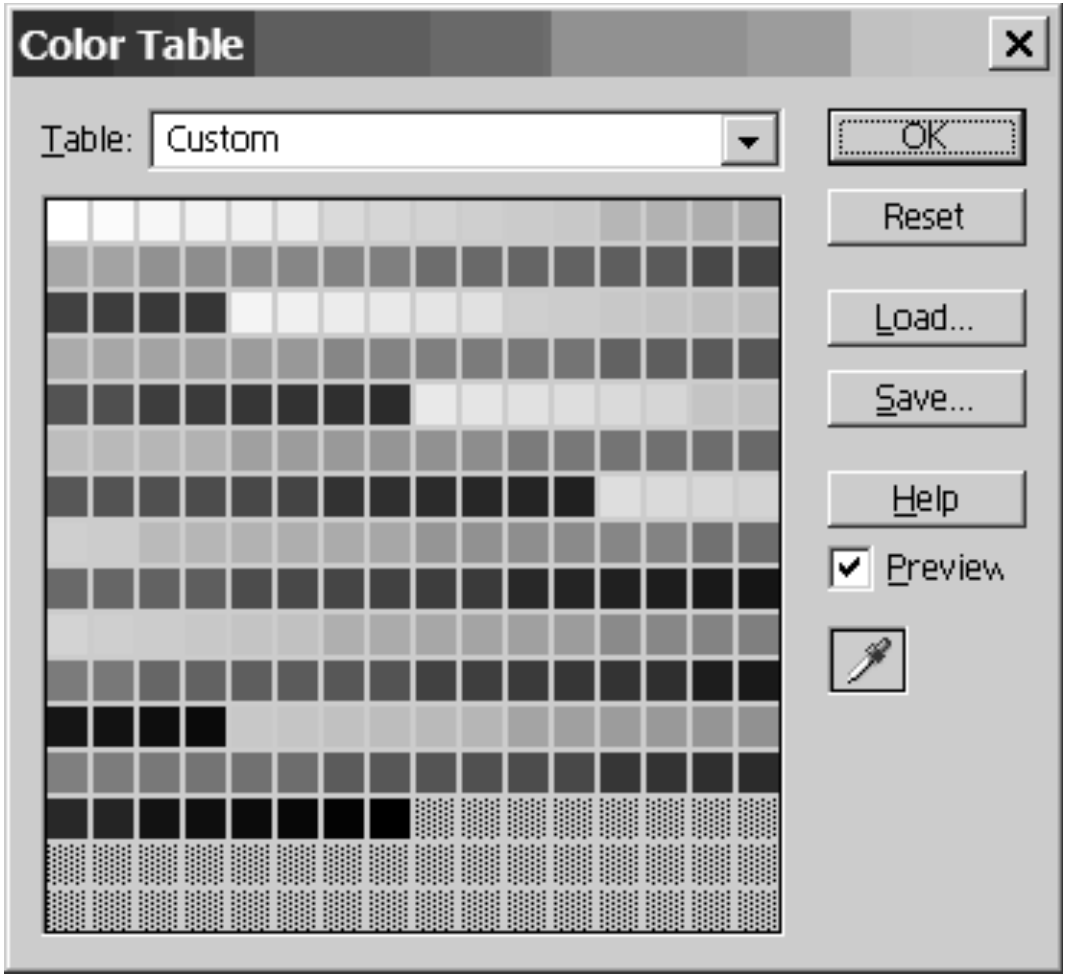
If H = 1024 pixels, and W = 768 pixels we get Size (Indexed) = 0.8 MB

Remember, true color image was ~2.4MB large ➔ compression ratio 3:1

Indexed images are good for graphics, logos, etc. In general, for non-natural images with well defined colors.

HARVARD

Standard Palette



216 colors equally spaced in RGB + 40 reserved for system colors

Compression and frequencies

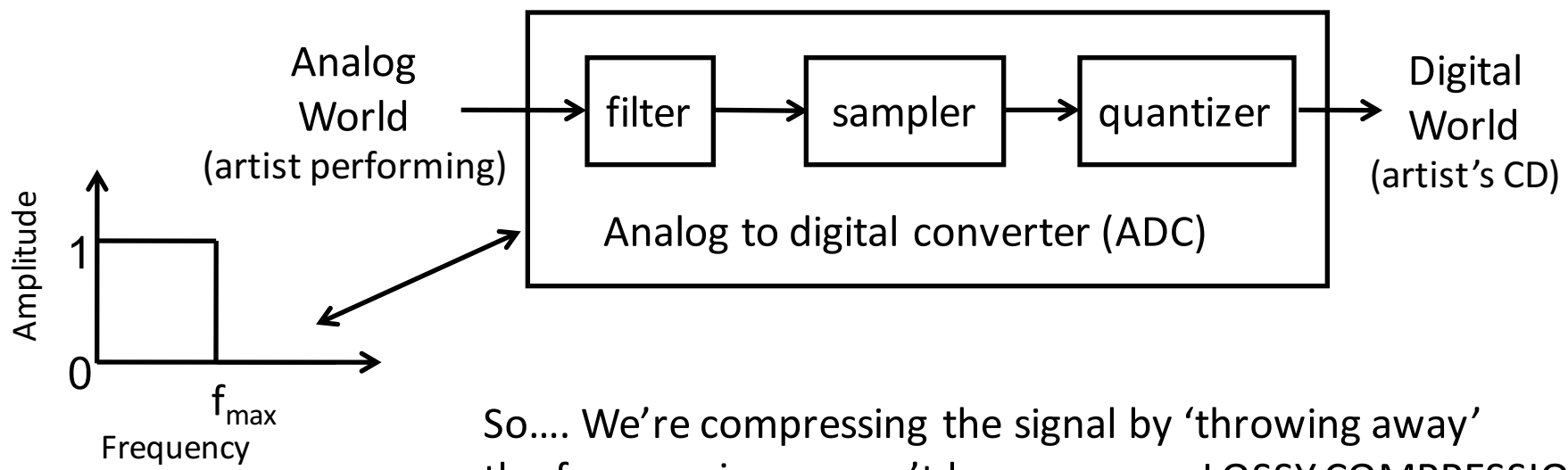
In audio signals, there are frequencies we cannot hear: eliminating those frequencies provides no loss to our 'enjoyment' of the signal

Example: Human ear cannot hear anything above 20kHz.

Q: Is there a need to store information with frequency above 20kHz?

A: low-pass filter with $f_{max} = 20\ kHz$ can be used to remove high frequency components and make it bandwidth limited!

low-pass: lets low frequencies ($f < f_{max}$) through, and rejects high frequencies ($f > f_{max}$).

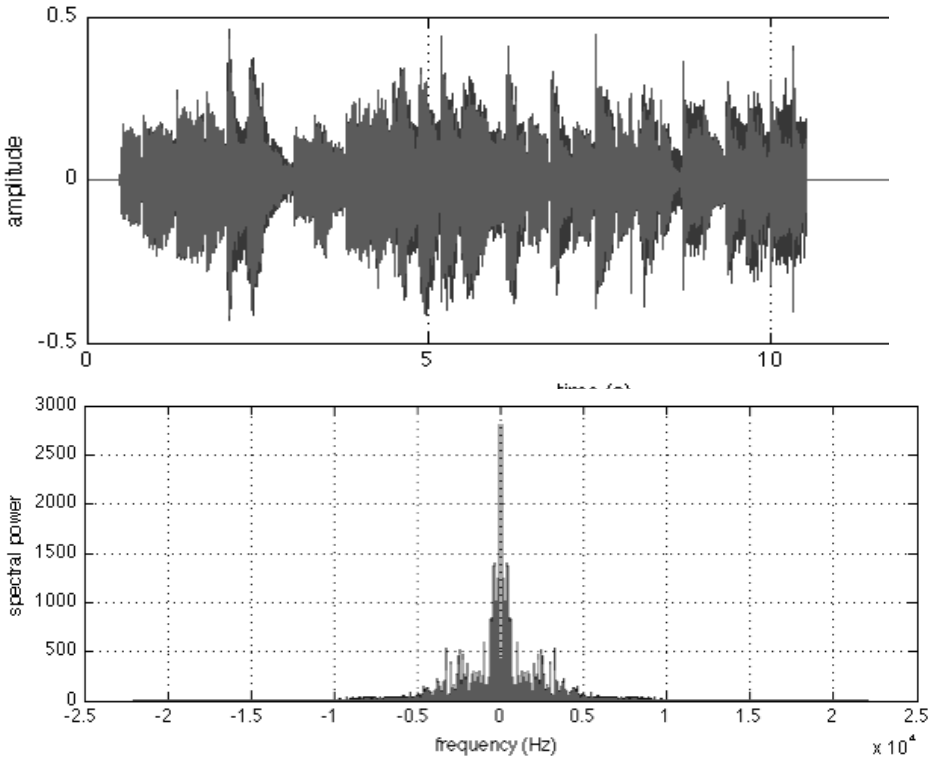


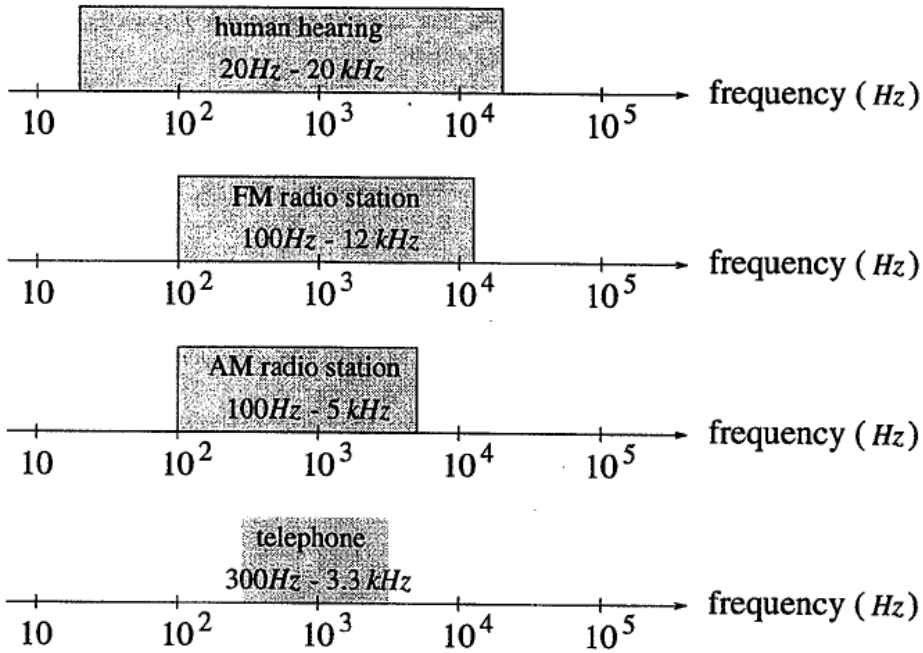
So…. We're compressing the signal by 'throwing away' the frequencies we can't hear anyway: <u>LOSSY COMPRESSION</u>

Reminder: Audio Signals

Understanding of *audio signals* in frequency domain as well as limitations of human hearing system, allowed us to manipulate audio signals effectively.

- To efficiently transmit the audio signals & to sample them and store them on the CD, we ignored high-frequency signals (in EE lingo we would say: “we filtered them out”).
- The quality of the received signal depends directly on how much of these high frequency waves (treble) we removed (ignored): for CD cutoff was at ~20kHz, for telephone it was ~3kHz.





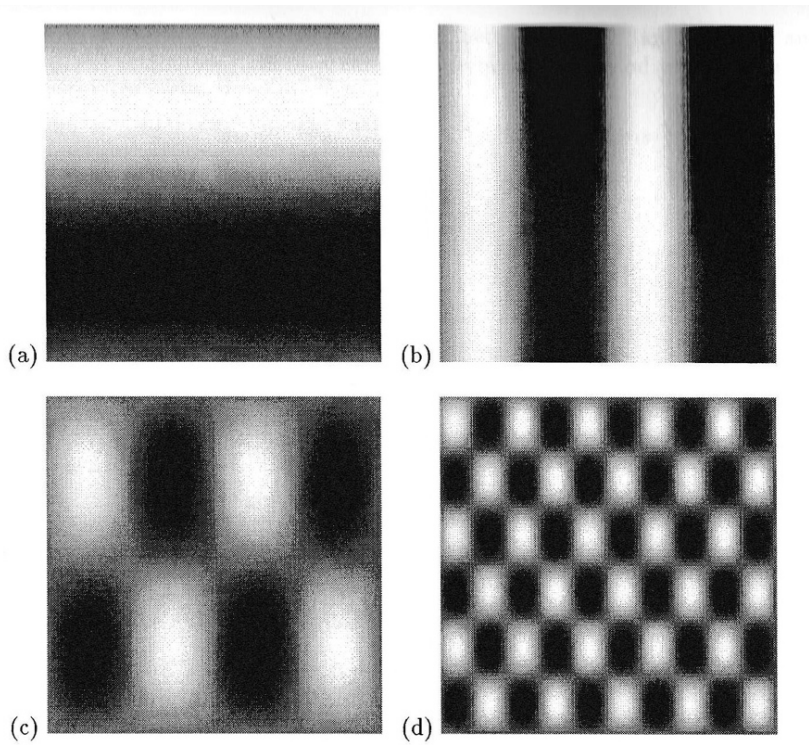
Compression and frequencies

In audio signals, there are frequencies we cannot hear: eliminating those frequencies provides no loss to our 'enjoyment' of the signal.

- LOSSY COMPRESSION

For visual images, there are similarly *spatial frequencies* that we are not sensitive to, and we can take away (lose) that information.

- LOSSY COMPRESSION



Let's represent our '2D images' by sines or cosines in 2D

(a) Horizontal $f_x = 0$, vertical $f_y = 1$ cycle/5 cm

(b) $f_x = 2\ cycles/5\ cm,\ f_y = 0$

(c) $f_x = 2\ cycles/5\ cm,\ f_y = 1\ cycle/5\ cm$

(d) $f_x = 2\ cycles/5\ cm,\ f_y = 3\ cycles/5\ cm$

From Kuc, The Digital Information Age

Image decomposition

The PREMISE

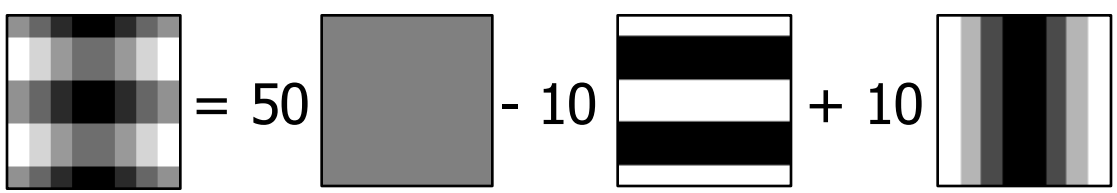
Every matrix can be decomposed into the weighted sum of orthogonal matrices.
For example, for 2 X 2 matrices we have:

$$\begin{bmatrix} 10 & 5 \\ 3 & 0 \end{bmatrix} = 10\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + 5\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + 3\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + 0\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

For 1D vectors, we have analogous situation: $\langle 4,5,8 \rangle = 4\langle 1,0,0 \rangle + 5\langle 0,1,0 \rangle + 8\langle 0,0,1 \rangle$

Well, images are matrices, therefore each image can be represented as sum of some standard matrices.
For example, consider following 8 x 8 image (ignore the coefficients for now):

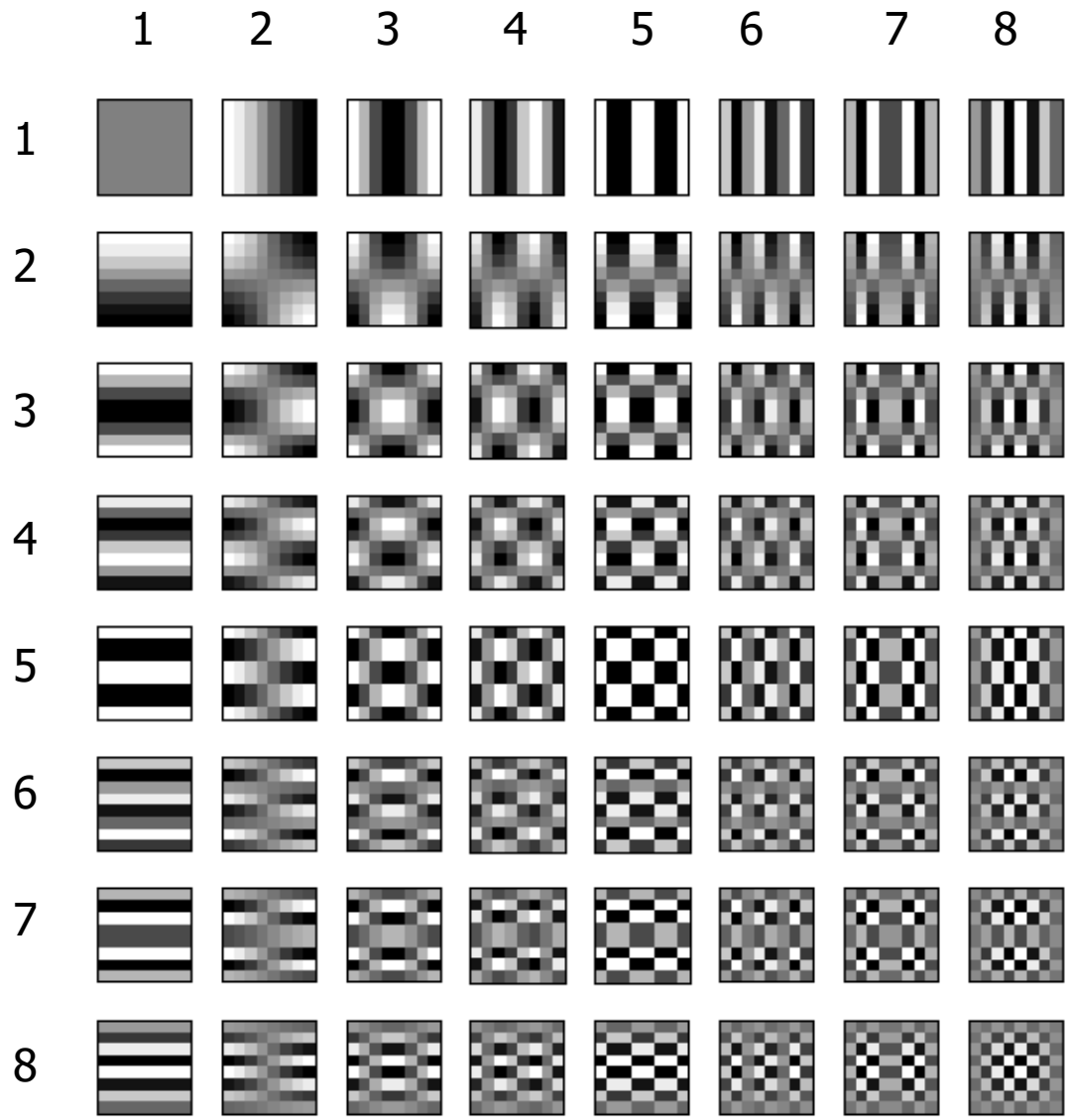
IMAGE



In fact, there is a set of 64 standard matrices that can be used to represent ANY 8 x 8 image

HARVARD

Standard set of matrices used for every image

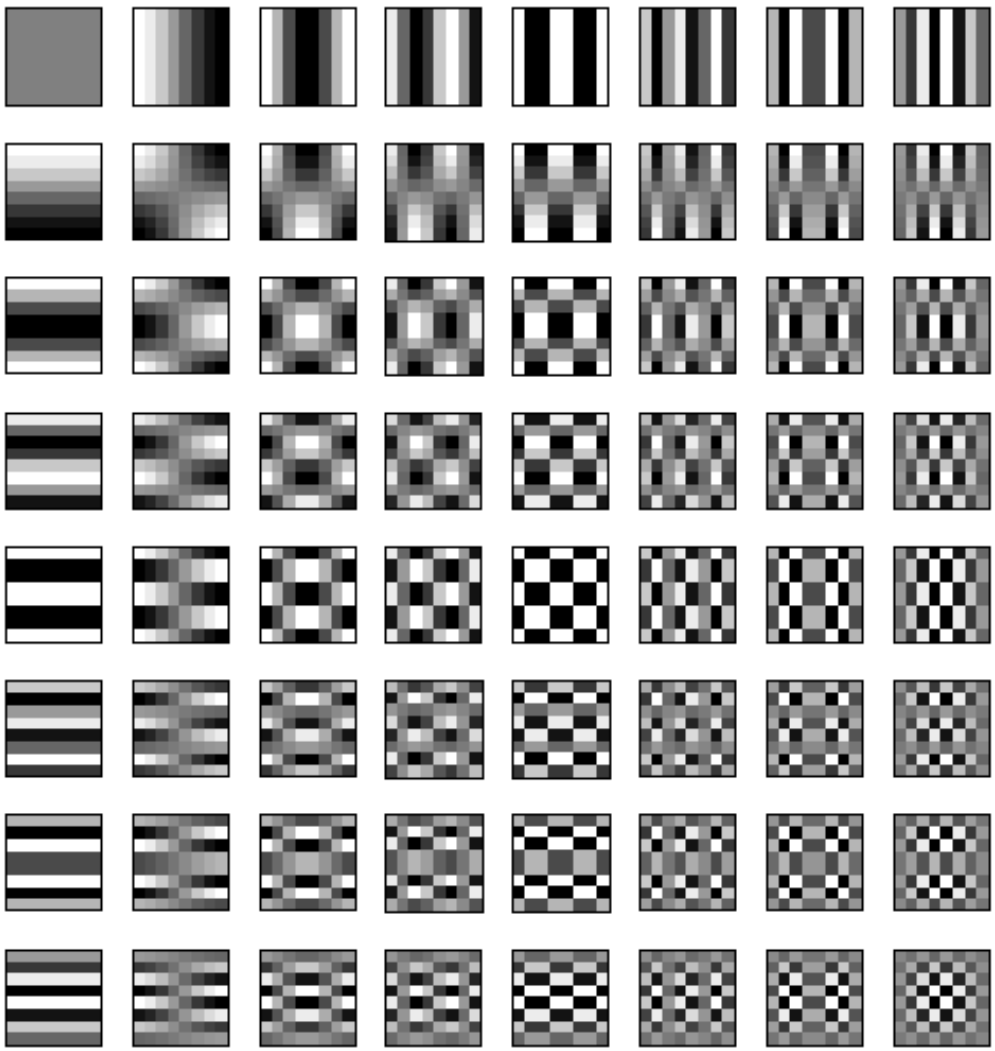


These 8 x 8 images represent the basis for the Discrete Cosine Transform (DCT)

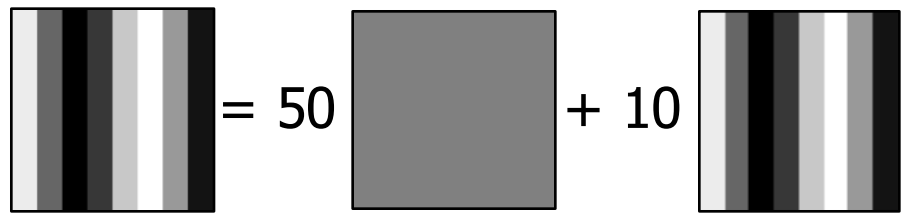
HARVARD

Example

Any image can be represented as weighted sum of DCT basis images



IMAGE

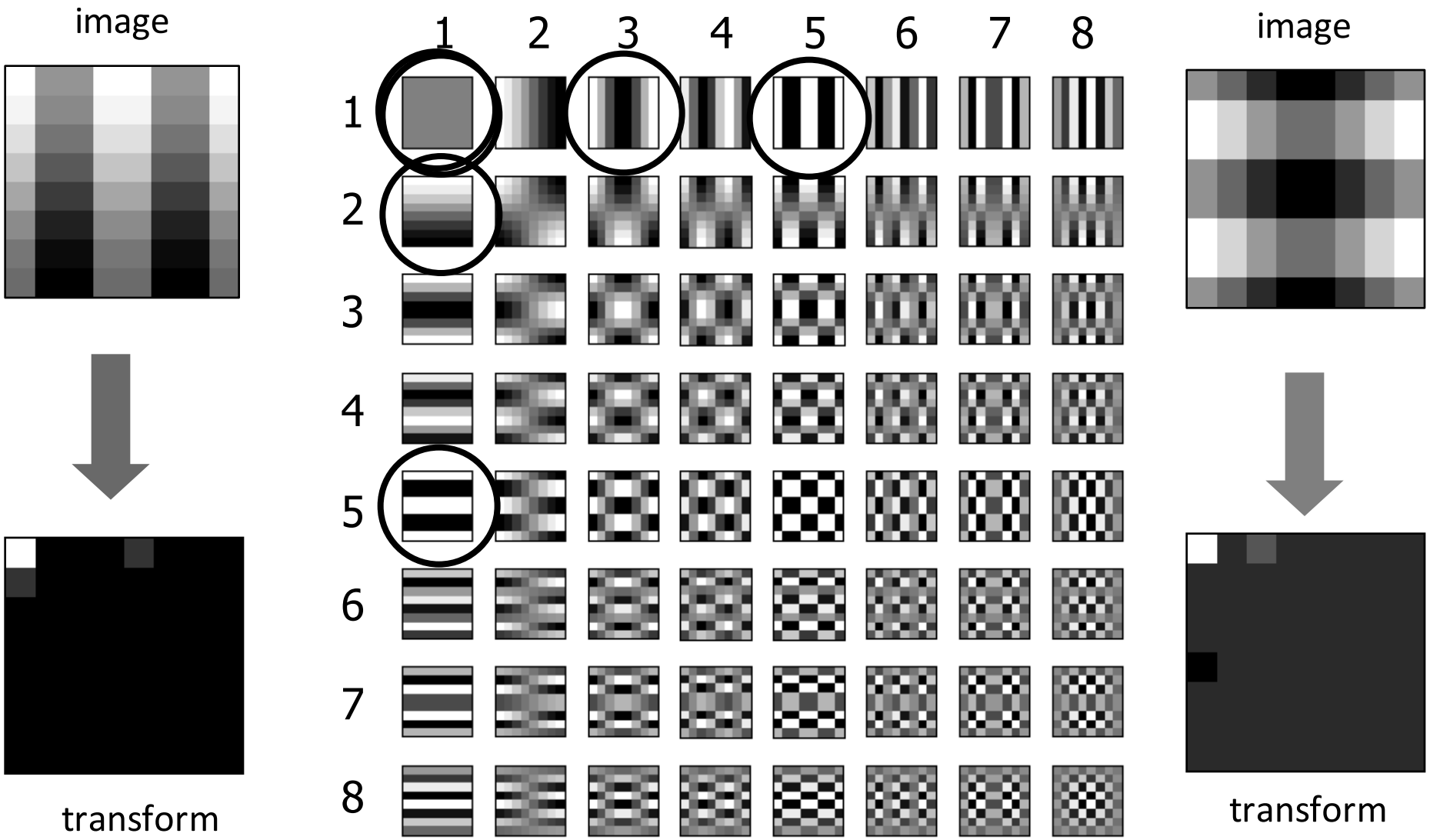


50	0	0	10	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT of an original image

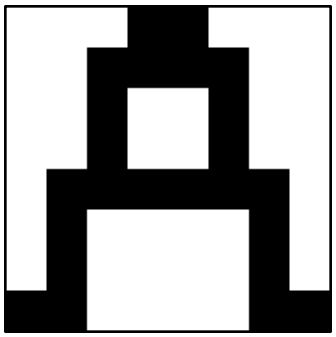
HARVARD

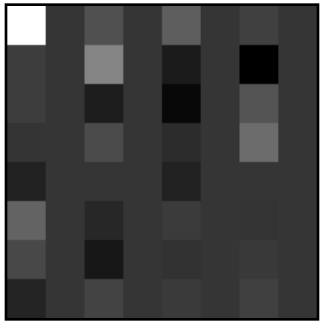
DCT itself can be displayed as an image



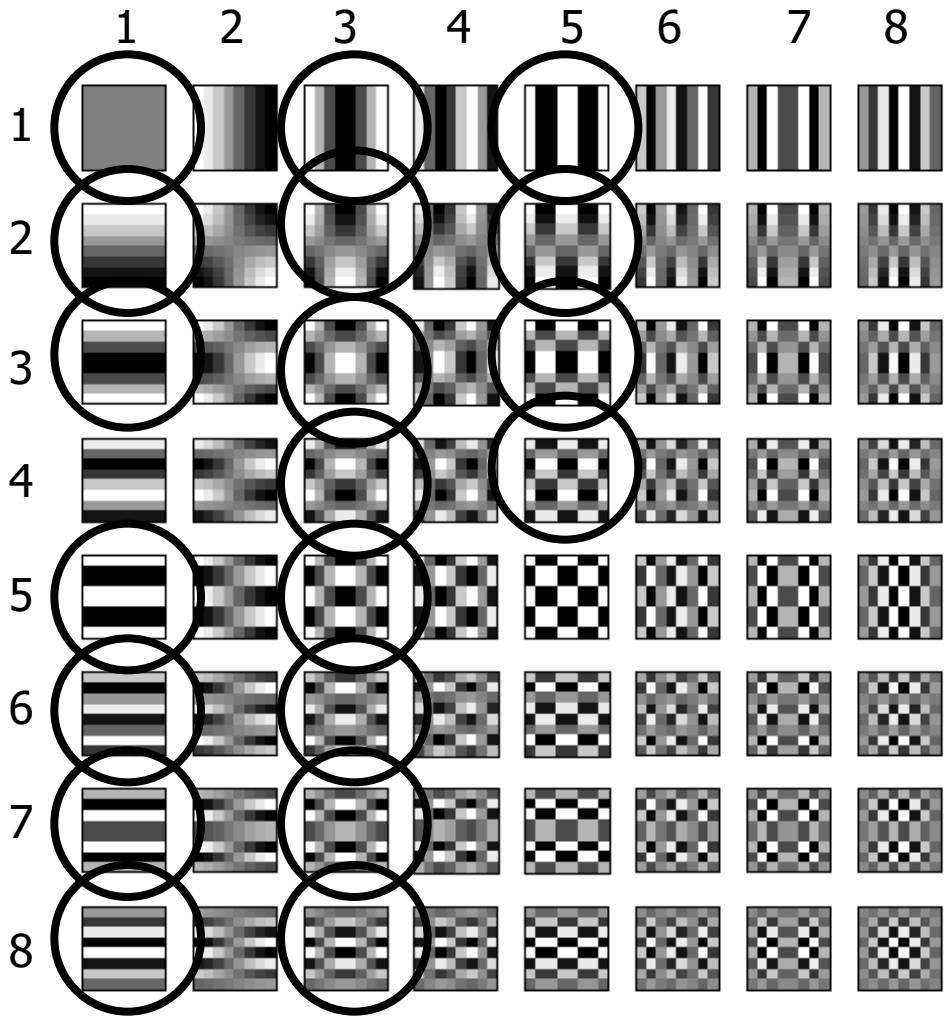
DCT Representation: Complex Images

image



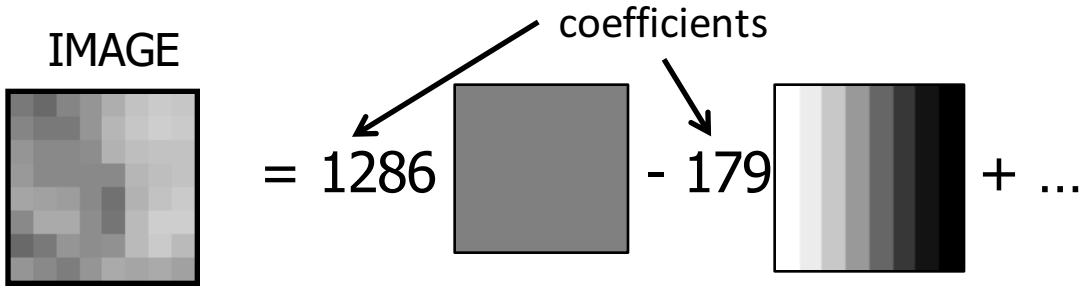


transform



http://cgjennings.ca/toybox/hjpeg/index.html

Discrete Cosine transform: summary



1286	-179	50	19	-29	19	13	-8
16	-42	-13	23	24	-3	3	7
-20	-31	-57	21	28	-3	-12	17
11	-14	19	-33	-38	11	-12	17
-14	25	9	8	17	6	3	-3
-14	-23	-16	-5	-7	5	9	6
14	21	5	-3	-4	-8	-2	2
0	1	-3	-1	0	1	2	8

DCT matrix

Image is divided into 8 x 8 pixel blocks, & each block is represented as a sum of 64 **standard 8 x 8 DCT** blocks that correspond to different spatial frequencies

- *Coefficients* for each of the DCT blocks are noted in a new 8x8 matrix
- At the end, the whole image is transformed (block by block) into a matrix of DCT coefficients (by blocks)

We can set a threshold for the coefficients & throw away the low value ones

A simple use of DCT with MATLAB

```
I = imread('cameraman.tif');
I = im2double(I);
T = dctmtx(8);
dct = @(block_struct) T * block_struct.data * T';
B = blockproc(I,[8 8],dct);
mask = [1  1  1  1  0  0  0  0
        1  1  1  0  0  0  0  0
        1  1  0  0  0  0  0  0
        1  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0];
B2 = blockproc(B,[8 8],@(block_struct) mask .* block_struct.data);
invdct = @(block_struct) T' * block_struct.data * T;
I2 = blockproc(B2,[8 8],invdct);
imshow(I), figure, imshow(I2)
```

Read in image (Tagged Image File Format)

'compress' image by using only 10 out of 64 of the basis matrices

Reconstruct image using basis matrices

original

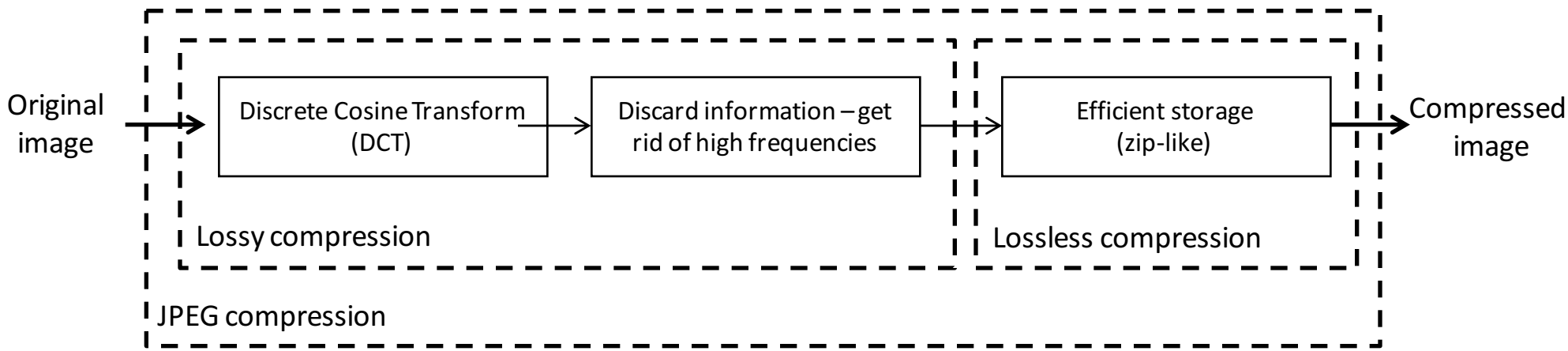
compressed

HARVARD

DCT is an important part of JPEG Compression

Joint Photo Experts Group

- Lossy compression, true color, used for photographic images
- Typical compression is 10:1 with little perceptible loss in image quality
- Good for natural images (smooth edges); *bad* for line drawings, textual or iconic graphics, with sharp contrasts between pixels



Other Popular Image Formats

TIFF (Tagged Image File Format) *introduced in the mid-1980's*

- In the beginning, TIFF was only a *binary image* format, grew to accommodate grayscale & color
- Industry standard for archival images; Portable across platforms (PC, MAX, UNIX)

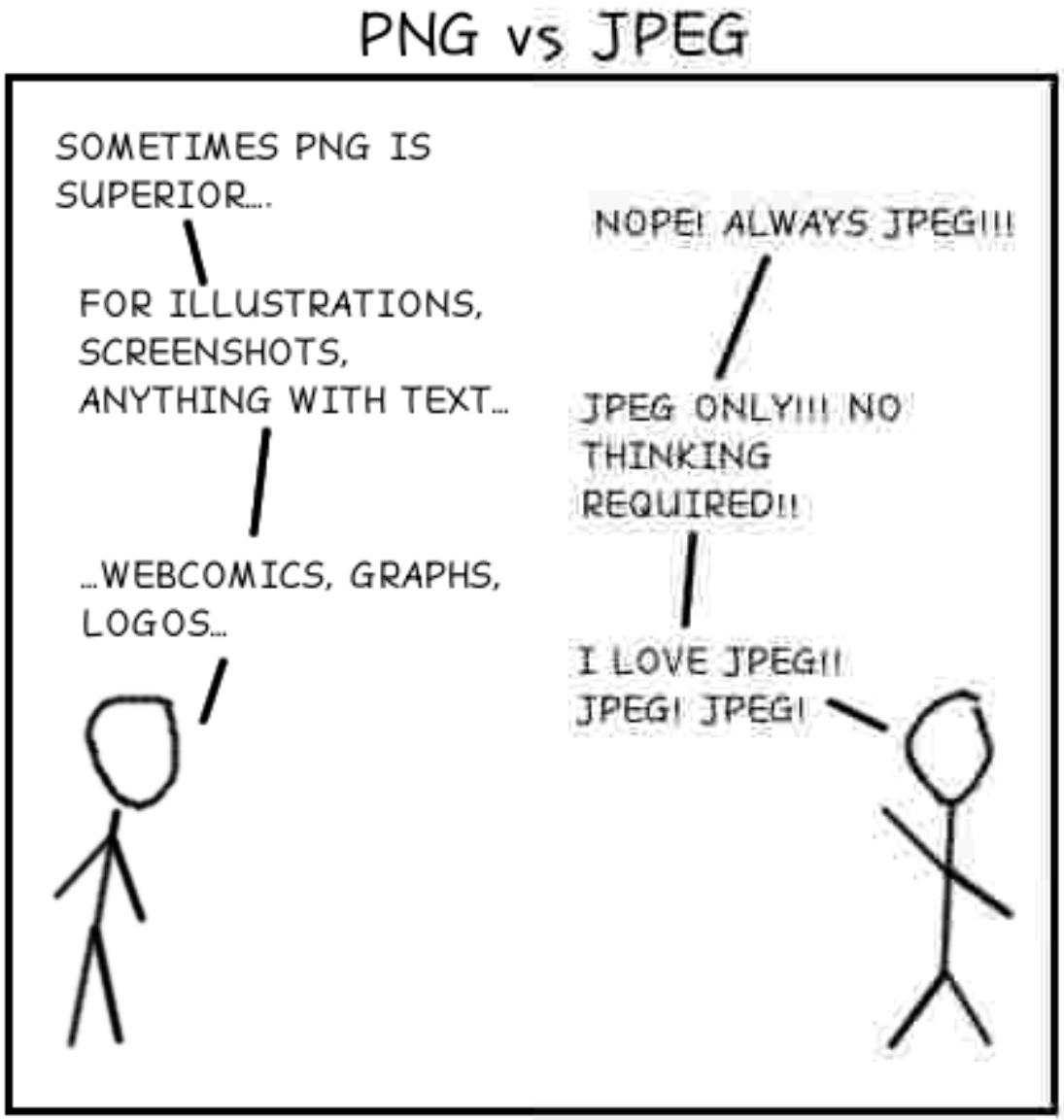
GIF (Graphics Interchange Format) *introduced by CompuServe in 1987*

- primarily for indexed colors: 1, 2, 4 and 8-bit palettes; supports transparent color;
- unsuitable for reproducing color photographs and other images with continuous color; used for graphics or logos;
- can store multiple images in one file - used on Web for simple animations.
- GIF images are compressed using the Lempel-Ziv-Welch (LZW) lossless data compression technique to reduce the file size without degrading the visual quality. (LZW is a way of efficiently encoding data: we will look at another LOSSLESS coding technique)

PNG (Portable Network Graphics)

- developed to improve upon and replace GIF as an image-file format not requiring a patent license. (The PNG acronym unofficially stands for "**P**NG's **N**ot **G**IF")
- PNG supports palette-based (24-bit or 32-bit) colors, grayscale
- PNG was designed for transferring images on the Internet, not professional graphics

File Format…Which is best?



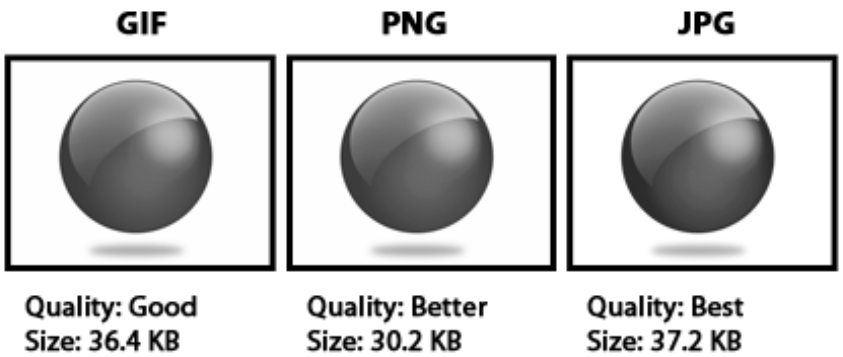
What file format would be best here?...and no this isn't XKCD

Choose the right file format for the type of image!!!

http://lbrandy.com/blog/2008/10/my-first-and-last-webcomic/

File Format Comparisons

A photo of a flower compressed with successively more lossy compression ratios from left to right.



http://www.practicalecommerce.com/articles/1821-Image-Formats-What-s-the-Difference-Between-JPG-GIF-PNG-

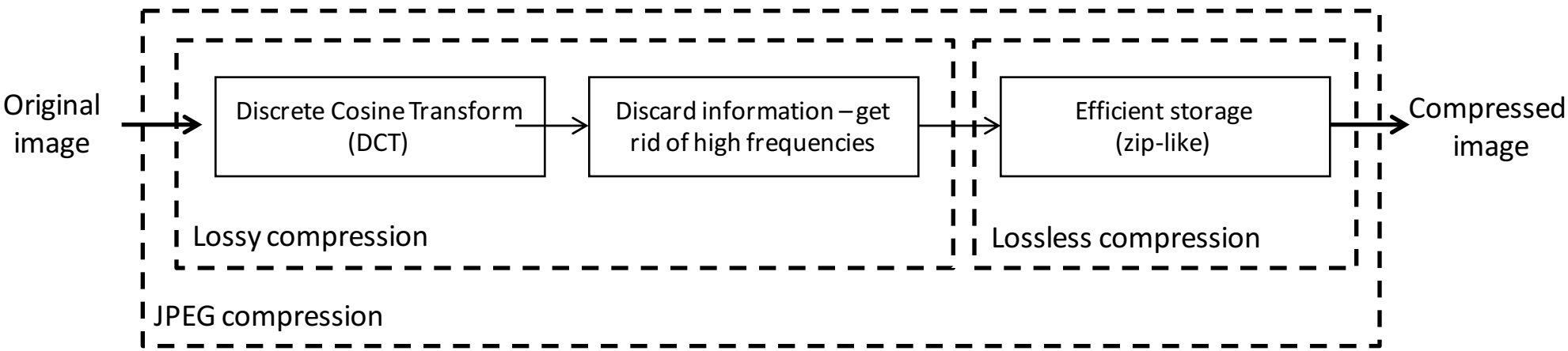
	Natural images 24-bit color, no sharp lines or edges	**Graphics, logos** solid colors (<256), sharp edges
Archival	TIF, PNG	PNG, GIF
Smallest	JPEG	GIF, PNG
Worst choice	GIF: limited color, larger than 24 bit JPG	JPEG: artifacts, smears text
Web	JPEG	GIF, PNG

http://www.scantips.com/basics09.html

DCT is an important part of JPEG Compression

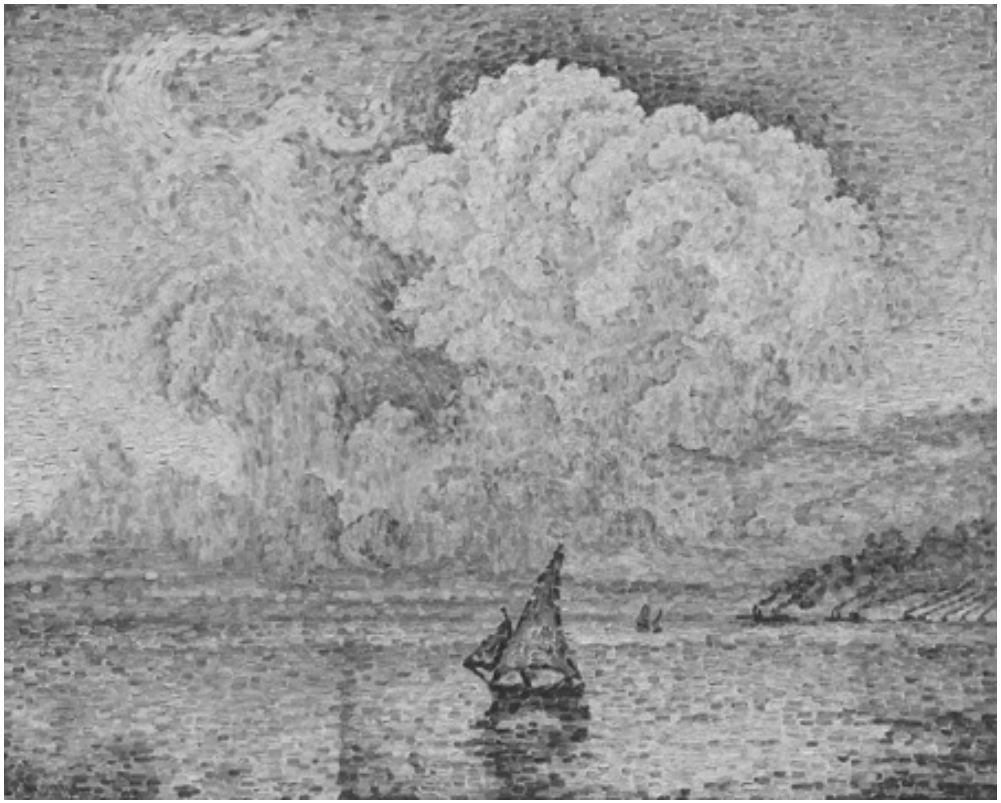
Joint Photo Experts Group

- Lossy compression, true color, used for photographic images
- Typical compression is 10:1 with little perceptible loss in image quality
- Good for natural images (smooth edges); *bad* for line drawings, textual or iconic graphics, with sharp contrasts between pixels



What about LOSSLESS COMPRESSION?

What is essential in representing an image?



Paul Signac, Antibes, The Pink Cloud

http://farm3.staticflickr.com/2689/4278721759_57e915edfe_o.jpg

If we must compress the information, how should we proceed?
What are the trade-offs?

- Do we want to SAVE ALL THE DATA, even as we 'compress' them? (LOSSLESS COMPRESSION)
- Or can we use a LOSSY COMPRESSION approach?

A photo compressed with successively more lossy compression ratios from left to right.

HARVARD

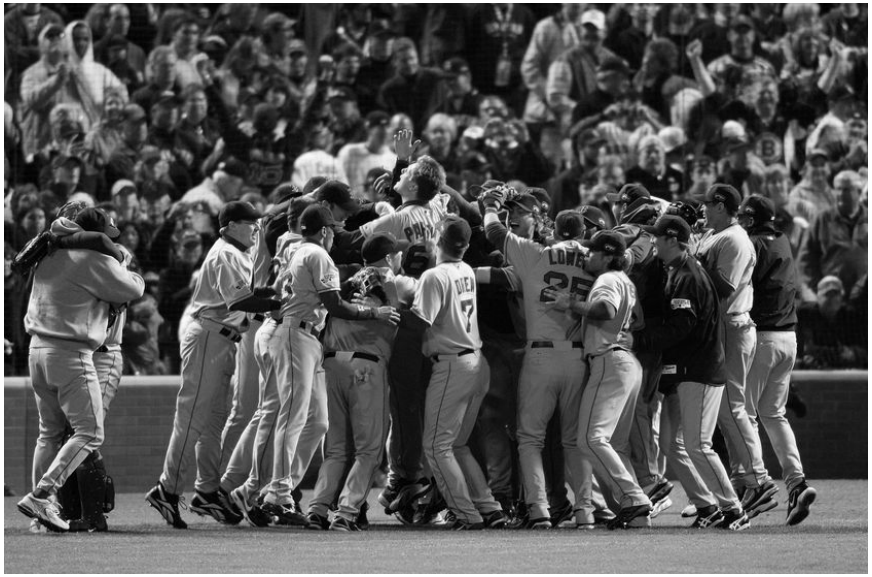
Compression and frequencies

In audio signals, there are frequencies we cannot hear: eliminating those frequencies provides no loss to our 'enjoyment' of the signal.

- For visual images, there are similarly *spatial frequencies* that we are not sensitive to, and we can take away (lose) that information.

We go back to ~~sines~~ & cosines : we use a DISCRETE COSINE TRANSFORM

- This approach is at the heart of the JPEG compression



(JPEG compressed) 24 bit RGB: 114kB

Joint Photo Experts Group

The ‘dimensions’ of information (from your phone)

- An all-text e-mail: 1- 10 kilobytes
- Email with high-resolution photo attachment: 2-5 Megabytes
- Uploading photo to social media site: 1 – 5 Megabytes
- Uploading a 3-minute video to a social media site: 5-15 Megabytes
- Making a 10 minute video call: 24 Megabytes
- Watching a 30-minute standard-definition video: 30-90 Megabytes

HARVARD

Source: S. Devadas, MIT, NY Times, March 23, 2014