# **Lectures on Digital Color, M. Goldman** Physics 1230, Light and Color, Fall 2007

## **I** Digital information - bits, bytes and pixels.

## a Bit

- 1 A bit is an irreducible discrete unit of information used by computers.
  - It can have two different values or "settings."
  - It be thought of as an on-off switch.
    - In this case, the two possible settings are "on" or "off".
  - It can also be thought of as a yes-no instruction.
    - In this case, the two possible values are "yes" or "no".
  - It can be represented by the digits 0 or 1
    - In this case, the two possible values are 1 or 0
- 2 A set or ordered "stream" of bits can be used to carry complex information

## b Binary numbers

- A set of bits can be thought of as a representation of a "binary" number, which is another way of representing familiar (base 10) numbers, *as discussed in class*. Examples:
  - 00 (binary number for 0)
  - 01 (binary number for 1)
  - 10 (binary number for 2)
  - 11 (binary number for 3)

## c Byte

- 1 A byte is an ordered set of 8 bits having given values
- 2 A byte can have any one of 256 "values," depending on the values (e.g., 0 or 1) of each of its 8 bits.
  - Examples of different values of a byte are:
    - 00000000 (binary number for 0)

- 00000001 (binary number for 1)
- 00000010 (binary number for 2)
- .....
- 11001000 (binary for 208)
- .....
- 11111111 (binary number for 255)
- The reason there are 256 values is that there are 256 different 8 digit "binary numbers" made up out of 0's and 1's only. Mathematically,
  - 256 = (# of values of each bit) raised to the power equal to the number of bits = 2 raised to the 8th power
    = 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2 x 2.
- $\circ$  The value of a byte, therefore, can be ordered from 0 to 255.
- 3 A Megabyte is a million bytes, a gigabyte is a billion bytes, a terabyte is a trillion bytes
  - Each of the bytes can have any one of the 256 values.
  - Computer memory and information storage capability is measured in gigabytes or terabytes

## d Discrete vs. Continuous

- 1 A byte can have one of 256 values, it is known as a discrete unit of information
- 2 Many pieces of digital information can have only discrete values
- 3 The opposite of discrete is continuous. A continuous unit of information can have any value. The speedometer in your car can read any speed value within its range, it is not limited to a specific set of possible values.

## e Pixel

- 1 A pixel is an irreducible discrete unit of information in an image or picture.
  - Pixel stands for "picture element."
  - In a "mosaic," a picture is made up of discrete colored tiles. Each tile is a pixel. The *value* of each pixel is its color.
  - The "pointillist" artist, George Seurat, used dots of color as pixels. It is often convenient to think of pixels as dots of color (or grey-scale values).
  - TV screens examined closely will be found to consist of dots of color (pixels).
  - All images can be "digitized" into pixels by using digitizing instruments, such as scanners.

2 The *location* of a pixel is given by an x-y coordinate on a graph.

#### f Bitmap images and color digital photos.

- 1 Any *bitmapped* image can be represented by giving the locations and values of all its pixels.
  - A "map" of locations and values of pixels is called a bitmap.
  - An image described by a bitmap is called a bitmapped image.
  - A digital photo is an example of a bitmapped image.
- 2 The *depth* of a pixel in a digital photo seen on a computer screen is determined by the number of *bits* of information it contains.
  - The appearance and quality of a digital photo seen on the computer screen is determined by the *depth* of the pixels.
  - A 1 bit pixel can take on  $2^1 = 2$  different values, 0 or 1 (dot is off or on)
    - Usually the dot is white, so this means the color of a 1 bit pixel is black or white
    - Note, in Photoshop language, a bitmapped image is alway made up of 1 bit pixels, whereas in more general usage a bitmapped image can be made up of pixels of arbitrary depth
    - Show using Photoshop mode how a color image looks in 1 bit mode. Show pixels (simulated)
  - A 2 bit pixel can take on  $2^2 = 4$  different values corresponding to a two digit binary number.
    - These values are 00, 01, 10, and 11.
    - For example, these values can trigger black, white, and two shades of gray.
  - A 3 bit pixel can take on  $2^3 = 8$  different values.
    - For example, the values can trigger black, white and 6 different colors.
  - An n-bit pixel can take on  $2^n$  different values.
    - In simple language, multiply 2 times itself a total of n times: 2 x 2 x 2 x ... (n factors of 2)
  - A computer set to show 8-bit color has 8-bit pixels, and can display images with  $2^8 = 256$  colors. A color scheme using 256 colors is sometimes called *in dexed color*.
    - Later we will show what a color image looks like in 8 bit color using the Photoshop mode *indexed color*.
  - A computer set to show *16-bit color*, has 16-bit pixels and displays images with  $2^{16} = 32,768$  colors

- These days, most laptops can show 24-bit color (also known as RGB color, with 8 bits (1 byte) describing the *brightness* of each primary)
  - An image in 24-bit color can show  $2^{24} =$  (over 16 million) different colors. (Check with your calculators)
  - See the section of <u>Physics 2000</u> under Color TV's dealing with partitive mixing to see one way this works.

## 11 How does a computer screen display colored images such as digital photos?

## a A computer monitor screen consists of a large array of *pixels*.

- 1 A low cost color computer monitor screen might contain 823 x 624 = over 500 thousand pixels. A more expensive one 1280 x 1024 = over 1.3 million pixels. A full-HD display is 1920 x 1080 pixels, or approximately 2.1 million pixels. A 15" Apple Retina display is 2880 x 1800 pixels, approximately 5.2 million pixels.
- 2 A *colored image* is produced when the pixels on the screen each take on appropriate colors.
  - Every different image can therefore also be thought of as a large array of pixels which can take on different colors.
  - $\circ$  These colors are represented by numerical values in the video buffer .

## b The Red, Blue and Green "parts" of each pixel determine the color of that pixel.

1 Each pixel on an active color computer screen is composed of 3 glowing *phosphors* : one red, one green and one blue:



- 2 A computer monitor (or TV screen) makes each of the 3 phosphors in each pixel glow with a different (adjustable) *brightness*.
  - $\circ~$  The brightness of a given phosphor in the color monitor or TV at a given instant is controlled by the voltage on the cell
    - Flat screen displays come in two main types, plasma and liquid crystal display (LCD)

- Plasma display pixels are essentially tiny neon or fluorescent lights, generating UV light by applying voltage to a gas and illuminating a phosphor to generate the red, green and blue light
- LCD pixels are illuminated from behind (backlit) by either a compact fluorescent or LEDs, and the voltage applied to the cell controls how much of the backlight gets through the cell and the colored filter at the front determines the color (red, green, or blue) See <u>Physics 2000</u> for a more detailed description of how LCD screens work.
- 3 When viewed from a distance, the separate glowing phosphors cannot be seen as distinct, but together make the pixel appear to be a single "new' color.
  - This occurs by *partitive mixing* (which we have already studied) of the three different phosphors in each pixel.
    - The colors of the 3 different phosphers are the *additive primaries*, R, G and B.
  - For example, red and green glowing phophors in one pixel make that pixel appear yellow from a distance.
- 4 Our perception of the overall color of each pixel depends on the brightness of *each* of the 3 phosphors.
  - In computer applications the brightness is measured by numbers which can only take on certain integer values.
  - The brightness is <u>not</u> *continuously* adjustable, as with a dimmer switch, but can only take on a set of *discrete* values determined by these integers.
  - See section of <u>Physics-2000</u> Website dealing with how TV screens work

## c 24 bit or RGB color makes the most colors available to pixels

- 1 The widest assortment of pixel colors is achieved with 24 bit pixels
  - In 24 bit color, the red, green and blue phosphor brightness are each separately determined by 8 bit (1 byte) values
    - For example, 00000000 = brightness level 0 is the darkest, followed by 00000001, 00000010, 00000011, 00000100, ... etc, up to 11111111 = level 255, the brightest.
  - Thus the red, green and blue phosphor brightnesses can each separately take on any of 256 different brightness levels.
  - The total number of different colors each pixel can exibit in 24 bit color is therefore =  $256 \times 256 \times 256 \approx 16.8$  million colors.
    - Each of the different colors is defined by a different set of 24 bits. For example, bright yellow is 1111111 1111111 00000000, meaning full brightness red (1111111), full brightness green (1111111) and no blue (00000000).



## d How to think about an image such as a digital photo when it is <u>not</u> on screen, but stored in your computer: *bitmapped image files*

- 1 We can speak abstractly about the numerical values of pixels in a *bitmapped image* separately from the pixels on a computer screen.
  - This is called the bitmapped data
  - The bitmapped data is one part of the *bitmapped image file* a long stream of bits organized in a special way
  - $\circ~$  In this case there are no phosphors, but we can still talk about the image pixels in the file
    - The colors available to those pixels will depend on whether the color scheme is 24 bit color, 8 bit color or something else
    - For 24 bit color we can describe the relative amounts (intensities) of Red Green and Blue (RGB) in a pixel without reference to *phosphors*.
- e Hue, saturation and brightness are another way to describe the color of pixels in a digital picture
  - 1 Another way of describing our *perception* of the color of a pixel is by 3 properties: <u>Hue</u> (color name), <u>Saturation</u> (color deepness or purity) and <u>Brightness</u> (light or dark color ).
  - 2 This description is called the HSB model of color. Bitmapped computer images have discrete (countable) values of H, S and B.
    - (Do not confuse the brightness levels of the red, green and blue phosphors within each pixel with the brightness level of the resulting effective pixel color).
  - 3 Show examples of HSB, RGB, and other color descriptions in Photoshop (Choose Foreground Color).
    - Explain color *cube* in terms of sliding to different hue cross-sections. Like a loaf of bread cut into sandwich slices.



- Explain  $L^*a^*b$  color description in terms of lightness and psychological opposition primaries
  - "a" corresponds to the r-g channel, "b" corresponds to the y-b channel and L corresponds to lightness or darkness

## f Size and resolution of images such as digital photos

- 1 Show in Photoshop, using still life picture
- 2 The *size* of an image is the length and width (e.g. in inches) of the picture when *printed* 
  - The *size* of the image when viewed on the *screen* may be very different (larger or smaller than the *printed* size)
  - The *size* of the printed picture can be changed in Photoshop
- 3 Another measure commonly used for *size* is the *pixel dimension* of a picture.
  - The *pixel dimension* is the number of *bytes* needed to store the color information of all of the pixels in the picture
    - Usually this is given in thousands of bytes (Kilobytes, Kb or K) or millions of bytes (Megabytes, Mb or M)
  - The *pixel dimension* can be obtained from the number of pixels in the entire image as follows
    - The *total number of pixels* in the entire image is obtained by multiplying the number of pixels in the one row (along width of the picture) by the number of pixels in one column (along the height of the picture).
    - The *total number of pixels* must then be multiplied by 3 to get the *total number of bytes*, assuming there are 3 bytes of information stored in each

pixel (RGB 24 bit color uses 3 bytes for each color). This gives the *pixel dimension*.

- 4 Digital cameras are usually rated by the maximum number of pixels in their photos.
  - A *megapixel* is a million pixels and abbreviated Mp. The maximum number of *megabytes* in an RGB file for one photo is *three times* the number of pixels. Why?
  - Digital cameras capable of making pictures of size one or two Mp are fine for creating Web images or small prints (up to 4 x 6 inches)
  - If a picture is going to be enlarged to 8 x 10 inches or larger or *cropped* (only a small portion of it printed) then the digital camera should be rated at 3 Mp or higher
- 5 The resolution of an image is the number of pixels per inch
  - Computer *monitors* have a number of pixels (RGB phosphor trios) per inch called the *screen resolution* 
    - The screen resolution can be between 70 and 90 pixels per inch and often can be adjusted. (Show)
  - An *image* which has the <u>same</u> *resolution* as the *screen resolution* on the monitor on which it is viewed will show all of its pixels when the image is displayed at *full size* (100%). The size of such an image will be the size of the print when the image is sent to a printer.
    - However, in order for a digital color print to look good on paper the image usually needs to have a *higher* resolution than the *screen resolution*
    - It generally must have at least 100 pixels per inch and up to 300 pixels per inch for the richest color and sharpness when printed.
  - Images whose resolution is <u>more</u> than *screen resolution* can still be seen on the computer screen by displaying only a small portion of a larger on-screen image at full 100% size and screen resolution
    - This is the way digital photos are *opened* in Photoshop
    - The onscreen version of an image can also be displayed as smaller than the original image *reduced* in size by 50%, 33% or some other percent, with not all pixels showing onscreen, or
    - displayed as larger than the original image *enlarged* in size by 200%, 300%, etc, with pixels *added*
  - The resolution and size of a digital photo can be changed in Photoshop
    - Show

## III Images with *fewer* colors than in 24 bit color

a Displaying bitmapped images at lower pixel depth (fewer colors)

- 1 The number of colors present in computer images is often much less than 16.8 million.
  - This makes the image file much smaller than an RGB file
  - This can greatly reduce the storage space normally needed by RGB images and the colors in the image are almost as good as in the RGB image
  - For example, an image in *8-bit color* can only show 256 different colors, compared to 16.8 million in 24-bit color, but the bitmapped image file is 1/3 the size
    - GIF images (*Graphics Interchange Format*, developed by CompuServe) are up to 8-bit (1 byte) are images which contain only 256 different colors.
- 2 In addition, an image file with many colors (even 16.8 million) is often viewed on a monitor which is set only to show a smaller total number of colors
- 3 Images in Photoshop can be viewed using various schemes for showing a smaller total numbers of pixel colors. These are usually called *indexed color* 
  - o 2-bit (4 colors), 4-bit (16 colors), 8-bit (256 colors), 16-bit (65,536 colors), etc
  - Show using Photoshop and mode set for *Indexed Color*.
- 4 Image quality is usually much more sensitive to the number of colors (pixel depth) than it is to the image *resolution*!
- 5 Dithering
  - One trick to make smaller palette images look realistic on the screen and in the printed version is called "*dithering*."
  - *Dithering* uses partitive mixing of *pixels* (rather than the phosphors within each pixel) to create new colors and desaturated versions of existing colors.
  - For example if we didn't have yellow pixels among the 256 available, we could make yellow from red and green pixels



• More realistically, we can make pink from red and white pixels juxtaposed (see Photoshop example).

• Dithering is often more effective when the pixels are arranged in patterns.

## b Converting *RGB* 24 bit images into *indexed color* (up to 8 bit) images using *color tables* and *palettes*

- 1 A *color coding table* for an *indexed color image* may consist of a total of 256 or fewer different colors used to display a *given* image.
  - Each pixel in the picture can take on one of the 256 different colors in the *color table*.
  - The 256 different colors in an 8-bit *color table* are each labeled by a different 8-bit binary number
    - Note, this is a different scheme from RGB color in which each color is represented by a different 24 bit binary number which gives the intensities of R, G and B.
  - $\circ$  Each of the colors in the table can still be a 24-bit color.
    - Thus, any 256 out of 16.8 million different colors can be put in the table.
  - There is an entirely *different* color table needed for each picture or digital photo.
    - The *color table* is part of the bitmapped image file for each picture.
  - In Photoshop, a color table for an image seen in *indexed color* (8-bit color or less) may be viewed
    - Demonstrate using Photoshop
- 2 Here is how *Photoshop* converts an RGB image into an *indexed color* image by constructing a *color table* 
  - A *color cube* containing all of the RGB colors is subdivided to construct a *color table* for a particular digital photograph.
    - The color cube contains all 16.7 million colors of a digitized RGB image
    - The x, y and z axes give 256 different brightness levels for each of the red, green and blue primaries
    - (The brightness levels range from a minimum = 0 to a maximum = 255, as in the demonstration of partitive mixing in <u>Physics 2000</u> under Color TV)
    - Think of the color cube as as a cube of cake containing an evenly- spaced 3-D array of poppyseeds each one representing a different RGB color. All together there are 256 x 256 x 256 = 16.8 million different colors (poppyseeds) in the cube.
    - However, not all of these colors (poppyseeds) are present in any given digital image. In a particular digital image, only certain of these colors will be present. Think of those as a much smaller number of *glowing* poppyseeds in the cube containing the evenly-spaced array of 16.8 million poppyseeds.



• The object now is to cut the cake (cube) into 256 smaller pieces (for 8-bit indexed color). Each piece will still contain many glowing poppyseeds (colors from the picture) but these will be very close in color, so we can take one glowing poppyseed (color) from each piece (to represent all the other glowing poppyseeds in that piece) and put it into the color table.



- The actual method of cutting the cake (subdividing the color cube) is to make each cut in such a way as to have an equal number of glowing poppyseeds (picture colors) in each piece, but these details are not important here and will not be discussed further)
- 3 The computer's *color palette* is filled using the table.
  - A program like Photoshop reads the table into the computer's display hardware *color palette* (e.g., into VRAM).
- 4 The color palette is used to color the screen pixels in the display of this image if the computer does not support 24-bit color.
  - Such a *color palette* is sometimes called a *Look Up Table*, or LUT.

## **IV** Storing bitmapped image files (e.g., digital photos)

## a How IBM BMP files are stored

- 1 (All information is stored as bytes.)
- 2 File header: ,
  - Contains file type (e.g., BMP), file size, location of bitmap data
- 3 Information header
  - o Size of information header
  - o Image height
  - o Image width
  - Number of bits per pixel (pixel depth)
  - Compression method (to be discussed next)
  - o Resolution
  - Number of colors in image.
  - 0 .....
- 4 Color table
- 5 Bitmap data
  - Pixel values (colors) for each row of pixels in the image.

## v Compression of bitmapped images

#### a Image files which are not "compressed" can get very large

- Aside from headers, image file size is proportional to the number of pixels in the image and to the pixel depth (# of bits to represent each image.)
- A "true-color" (24-bit) image of size 1024 x 768 pixels contains over 2 megabytes of info.

#### b Image files can be made smaller by "compression" tricks.

- 1 Compression tricks can shrink a file by a factor of 5 or more.
- 2 Two classes of compression:
  - o <u>Lossless</u> compression

- Keeps all pixels, but "coding" is changed.
- Compression is modest.
- <u>Lossy</u> compression
  - Throws away some image information.
  - Compression can be more extreme.

## c RLE lossless compression (e.g, TIFF)

- 1 RLE = Run Length Encoding
- 2 TIFF files are RLE-compressed
- 3 Here is how it works for an image with 128 different colors, labeled from 0 to 127:
  - Consider a sequence of values of the first 18 pixels in the top row of pixels on the screen.
  - Each value is represented by a different number from 0 to 127, designating a specific color (say, in a color table). Thus, only 7 bits are needed to specify a color, since here are exactly  $2^7 = 128$  different 7 place binary numbers.
  - Since most images have certain areas of the <u>same</u> color, it is common to have some <u>repeating</u> pixel values.
  - A sequence of repeating pixel values is called a "run," and the number of repetitions is called the *run length*.
  - Below we give an example of RLE compression:
- 4 In the table below, there is a *run* of zeros of run length 5, a run of 128's of run length 4, and a run of 32's of run length 4.

0 0 0 0 0 32 84 128 128 128 128 96 74 56 32	2 32	32 32
---	------	-------

• This might, represent, for example, the following sequence of colors:



- An RLE compressor program scans pixel-values in the row of numbers from left to right, looking for repeated pixels.
- When 3 or more consecutive pixels of identical value are found, they are replaced by two values one specifying the run length, and the next specifying the value of the repeated pixel.

- The pixel carrying run length information is specified below by a grey cell with a value given by a red number for the run length. Such a value is called a *runlength token*..
  - A run-length token does not specify a color or correspond to a physical set of three phosphors on the screen, but is a <u>marker</u>, carrying *code*n for the number of repetitions of the next pixel value.
  - An <u>8th</u> bit with value one could be used to indicate that the number given by the first 7 bits is a run-length token rather than a code for a color. In reading the compressed file, the 8th bit would be ignored in the value of the next pixel, which gives the color value to be repeated.

5	0	32	84	4	128	96	74	56	4	32				

- An efficient scheme is for the next pixel *after* the pixel giving the repeated color value (0, in the example, below) to be another kind of marker or token this time, giving the number of pixels to follow which are <u>not</u> repeated.
- Below, we have represented this kind of token as a blue number (2) in a grey background. The value 2 means that 2 non-repeating pixels will follow.
  - This information can be coded into the 8th bit as a 0, rather than a one, indicating that the first 7 bits indicate the number of **non**-repeating pixels, rather than a color.
  - In the example below, the following pixels, would be read without paying attention to their 8th bit, followed by the next marker, which would have 1 as the 8th bit, indicating that the following value (128) is to be repeated 4 times.

5	0	2	32	84	4	128	3	96	74	56	4	32			
															1

- The above sequence of pixels represents an RLE compressed version of the original. In the original there were 18 pixels, while in the compressed version, there are only 13 pixels, carrying the *same* information.
  - Hence, this part of the image has been **compressed** by 5/18 = 28%.
- This process is repeated for each entire row (scan line) of pixels in the image
- In a graphics program, such as Photoshop, the sequence is re-expanded according to the coded information, and shown below. Note, we have color-coded the repeated pixels differently from the non-repeating pixels.

- No image information is lost, so this is lossless compression. The original (uncompressed) image is reproduced completely from the compressed file.
- For true-color (24-bit) images, the process of RLE compression is carried out separately for each of the Red, Blue and Green parts of the pixel.
- Can you describe how RLE compression might work for black and white (1-bit pixel) images? Would this kind of compression be efficient here?

#### d JPEG lossy compression

- 1 JPEG = Joint Photographic Experts Group.
- 2 This is a lossy compression, so when the image is reconstructed, information is lost
  - The method removes *small detail information* from the image
  - $\circ$  It merges colors that are close to each other into the same color.
  - With more and more compression, the image may appear more blurred or less rich in color or "pixelated."
- 3 The advantage, however, is that the compression can be much greater than RLE compression.
  - Compression of 100 to1 may be achieved!
  - The smaller the file size due to lossy compression the more the degradation of the image. Every time you save a JPG, compression is applied and colors merge together. Thus, every time you save the image looks worse
- 4 How to save a Photoshop image as a JPEG (YouTube Video)
- 5 Link to Webpage images showing JPG compression

# **VI** *Optional*: Here is a lengthy explanation of JPEG compression. (Details are mathematical.)

a It is easiest to understand the process by considering one row of pixels, again (the actual manipulations are carried out on a *matrix* — rows and columms — of pixels.
Also, color information and brightness are separated for each pixel.)

b We can visualize the discrete values in the row of pixels as a set of *heights* above the pixels. If we join those heights, we get a curve for that that row (shown in red, below):



- c There is an important theorem in mathematics which states that most curves can be synthesized by <u>adding together</u> with different amplitudes a special set of wiggly curve called cosine curves (or sine curves).
- d This is called *Fourier's theorem*.. It includes an exact mathematical description for finding the amplitudes of each of the different curves (called *components* of the original curve) which must be added together to get the original curve.
- e Each of these wiggly curves has a different wavelength and (usually) a different amplitude The results of adding the different curves together is the original curve (sometimes called a *waveform*)
- f Example of adding two waves using <u>Physics-2000</u> (Adding waves in The Atomic Lab)
- g Example of adding together a number of wiggly (cosine) curves of different amplitudes and frequencies to get a *step curve*.
  - This is what a step curve looks like:



- It might represent a bright part of an image on the left and a dark (0 pixel value) on the right.
- h We can build a curve which looks a lot like a step curve out of the special set of wiggly curves.
  - Let's take this low frequency large amplitude curve:



• Add to this a higher frequency, smaller amplitude wave:



• And add to this a still higher frequency, smaller amplitude wave:



• The result of adding these three curves together is the following curve:



- i Note, how just three wiggly curves of specified different amplitudes and frequencies can "make" a curve that looks like the step curve, but has a few extra wiggles and a transition from light to dark which is not quite so sharp.
- j If the step curve represented an <u>edge</u> or <u>image boundary</u>, the *approximate* step curve made out of three wiggly curves shows a *less abrupt* transition.
  - On the image, that would look less sharp and more blurry.
- k A better representation of the step curve would occur if we added more wiggly curves of higher frequency with appropriate amplitudes.
- 1 With only three wiggly curves we have *thrown away short wavelength* (*high spatial frequency*) *information in the picture.*
- m In order to get an abrupt or *sharp* curve transition, many *high frequency* curves need to be included.
  - Without them, the transition is more gradual.
- n If we think of the curve as representing a brightness transition, the original has a sharp boundary (say, from black to white), but the sum of wiggly curves which <u>omits</u> high frequencies has a more gradual boundary (shades of grey between the black and white).
- Leaving out high frequency component curves replaces a *sharp* part of the image with a more *blurry* one.

- p In our earlier example, above, if we think of the 8 pixel values as unjoined, there is a prescription for finding exactly 8 different wiggly sequences of pixels (i.e., 8 different frequencies) which, when added together, give the original sequence.
- **q** The central concept of JPEG compression is to *remove* the high-frequency information in an image. This leads to a description with fewer pixels, and therefore a smaller file.
  - Transform of a sequence:
    - For example, the amplitudes of the wiggly curves needed to make the pixel value sequence (16, 16, 12, 12, 10, 14, 10, 4), used above as an example, are [12, 4, -0.5, 2.4, -1.8, 0.5, -0.2, -1.1]. These amplitudes are given in order of wiggly curves with higher and higher frequencies (shorter and shorter wavelengths).
    - The new sequence is called the *transform* of the original sequence, and carries all the information needed to reconstruct the original pixel sequence. by taking the *inverse transform*.
  - Truncated sequence:
    - If we simply remove the *highest frequency amplitudes* from the transformed sequence, [12, 4, -0.5, 2.4, -1.8. 0.5, -0.2, -1.1],
    - and multiply every amplitude by 10, we get the sequence
    - [120, 40, -5, 24, -18, 0, 0, 0].
    - Note, that this "truncated" sequence has only 5 numbers to store instead of the original 8. By storing this modified transform sequence, we have compressed the image information (and lost information).
  - Inverse transform gives the image.
    - Now the mathematical process of taking the *inverse transform* of modified sequence, [120, 40, -5, 24, -18, 0, 0, 0], to get the pixel sequence representing the modified image corresponding to the compressed file yields the result,
    - (15, 14, 12, 13, 11, 12, 11, 8),
    - which, as we see below, has smoother transitions (image is less sharp).



• To summarize the procedure, here is what is done in JPEG compression:



 (In compression of a 2-dimensional image, the transform is of an 8 x 8 *matrix* (array) of pixel values, yielding another matrix, called a DCT, or Direct Cosine Transform)

## **VII** MPEG lossy compression

- 1 MPEG *movies* use the same compression methods as JPEG for each frame, but go further, to make the file size smaller:
  - MPEG compression looks for blocks of similar pixels in successive frames, even if they have moved slightly, and codes this information so that it doesn't have to get repeated in the file.
  - Only an average of 2 frames per second are normally sent in their entirety.
  - The rest are either encoded as differences from preceding frames, or as interpolations between frames.
- 2 MPEG compression approaches 200:1

## **VIII Processing bitmapped images**

- a Since a bitmapped image is represented by a large collection of numbers (the values of all the pixels), we can manipulate the image by manipulating the numbers.
  - 1 In *image processing*, the computer uses mathematical rules to systematically alter the numbers for each pixel.
    - We will show how to do image processing in Photoshop.
    - For example, we will explain how to sharpen, blur and perform other operations on images.
- b How to make an image sharper.
  - 1 Adobe Photoshop demo of sharpening and other filters.
    - o <u>You Tube Video</u>
    - Use photo of girl in hat.

- Enlarge section including one eye, hair and hat and view 8 x and show effect of each filter.
- 2 Any picture is a matrix (array) of pixel values, so image processing is number processing.
- 3 Sharpening is carried out using a <u>mathematical rule</u> or "filter" which changes the pixels to emphasize edges and increase contrast.
  - One-dimensional demonstration of a sharpening filter applied to a row of pixels (*Russ, Image Processng Handbook*)
  - Consider the following sequence of pixel values representing a sharp pixel transition:

	2	2	2	2	2	4	6	6	6	6	
--	---	---	---	---	---	---	---	---	---	---	--

• Now we will apply the following <u>filter</u> to this sequence

-1 2 -1

2

• The rule is, that whatever pixel value is centered above the red 2 is the pixel we are changing. The new value of this pixel is obtained by multiplying corresponding numbers and adding.

- For example, in the above position, the filter gives
   -1 x 2 + 2 x 2 + -1 x 2 = 0, so the filter in this position changes the green 2 into a 0
- 2 0 2 2 2 4 6 6 6 6 6
  - Next, we simply step the filter across the sequence of original pixel values, changing as we go along:

```
2 2 2 2 2 4 6 6 6 6 6
-1 2 -1
```

- This yields a new value for the third pixel: 2 0 2 2 4 6 6 6 6 6
- Stepping across the entire sequence gives the new set of pixel values:

0	0	0	0	-2	0	+2	0	0	0	0

• Adding these "enhancements" to the original sequence gives

	2	2	2	2	0	4	8	6	6	6	6
--	---	---	---	---	---	---	---	---	---	---	---

- This enhances the edge. (See picture).
- Two-dimensional <u>matrix</u> version of sharpening is best, because edges are not always vertical. We will not explain this here.
- 4 Art and optical illusions "prove" that we mainly pay attention to edges in our visual perception.
  - Picasso's Mother and Child and lateral inhibition demos.
  - Edge-cue illusions (cover edge and contrast disappears).
- 5 *Receptive fields* are our human "filter" for perceiving edges better on our retina.
  - Definition and examples of receptive fields. Retinal image light falling on <u>center</u> causes increase in nerve cell signal. Light falling on <u>surround</u> causes decreased signal.
  - Physiological basis. Neural network.
  - Similarity between matrix sharpening filter and receptive field on retina. We all have built in sharpening filters!