

# Graphic/Image File Formats

Common graphics and image file formats:

- <http://www.dcs.ed.ac.uk/home/mxr/gfx/> — comprehensive listing of various formats.
- See [Encyclopedia of Graphics File Formats](#) book in library
- Most formats incorporate *compression*
- Graphics, video and audio compression techniques in next Chapter.



Back

Close

# Graphic/Image Data Structures

**“A picture is worth a thousand words, but it uses up three thousand times the memory.”**

- A digital image consists of many picture elements, termed **pixels**.
- The number of pixels determine the quality of the image (**resolution**).
- Higher resolution always yields better quality.
- A *bit-map* representation stores the graphic/image data in the same manner that the computer monitor contents are stored in video memory.



Back

Close

## Monochrome/Bit-Map Images



Figure 5: Sample Monochrome Bit-Map Image

- Each pixel is stored as a single bit (0 or 1)
- A 640 x 480 monochrome image requires 37.5 KB of storage.
- *Dithering* is often used for displaying monochrome images

# Gray-scale Images



Figure 6: Example of a Gray-scale Bit-map Image

- Each pixel is usually stored as a byte (value between 0 to 255)
- A 640 x 480 greyscale image requires over 300 KB of storage.

## 8-bit Colour Images



Figure 7: Example of 8-Bit Colour Image

- One byte for each pixel
- Supports 256 out of the millions possible, acceptable colour quality
- Requires Colour Look-Up Tables (LUTs)
- A 640 x 480 8-bit colour image requires 307.2 KB of storage (the same as 8-bit greyscale)

## 24-bit Colour Images



Figure 8: Example of 24-Bit Colour Image

- Each pixel is represented by three bytes (e.g., RGB)
- Supports  $256 \times 256 \times 256$  possible combined colours (16,777,216)
- A  $640 \times 480$  24-bit colour image would require 921.6 KB of storage
- Most 24-bit images are 32-bit images,
  - the extra byte of data for each pixel is used to store an *alpha* value representing special effect information

# Standard System Independent Formats

## GIF (GIF87a, GIF89a)

- Graphics Interchange Format (GIF) devised by the UNISYS Corp. and CompuServe, initially for transmitting graphical images over phone lines via modems
- Uses the Lempel-Ziv Welch algorithm (a form of Huffman Coding), modified slightly for image scan line packets (line grouping of pixels) — [Algorithm Soon](#)
- Limited to only 8-bit (256) colour images, suitable for images with few distinctive colours (e.g., graphics drawing)
- Supports *interlacing*



Back

Close

# JPEG

- A standard for photographic image compression created by the Joint Photographic Experts Group
- Takes advantage of limitations in the human vision system to achieve high rates of compression
- Lossy compression which allows user to set the desired level of quality/compression
- **Algorithm Soon** — Detailed discussions in next chapter on compression.



Back

Close



# TIFF

- Tagged Image File Format (TIFF), stores many different types of images (e.g., monochrome, greyscale, 8-bit & 24-bit RGB, etc.) → tagged
- Developed by the Aldus Corp. in the 1980's and later supported by the Microsoft
- TIFF is a lossless format (when not utilizing the new JPEG tag which allows for JPEG compression)
- It does not provide any major advantages over JPEG and is not as user-controllable it appears to be declining in popularity



## Postscript/Encapsulated Postscript

- A typesetting language which includes text as well as vector/structured graphics and bit-mapped images
- Used in several popular graphics programs (Illustrator, FreeHand)
- Does not provide compression, files are often large
- Although Able to link to external compression applications



Back

Close

# System Dependent Formats

## Microsoft Windows: BMP

- A system standard graphics file format for Microsoft Windows
- Used in Many PC Graphics programs, Cross-platform support
- It is capable of storing 24-bit bitmap images



Back

Close

## Macintosh: PAINT and PICT

- PAINT was originally used in MacPaint program, initially only for 1-bit monochrome images.
- PICT format was originally used in MacDraw (a vector based drawing program) for storing structured graphics
- Still an underlying Mac format (although PDF on OS X)



## X-windows: XBM

- Primary graphics format for the X Window system
- Supports 24-bit colour bitmap
- Many public domain graphic editors, e.g., *xv*
- Used in X Windows for storing icons, pixmaps, backdrops, etc.



Back

Close

# Basics of Colour: Image and Video

## Light and Spectra

- Visible light is an electromagnetic wave in the 400nm - 700 nm range.
- Most light we see is not one wavelength, it's a combination of many wavelengths (Fig. 9).

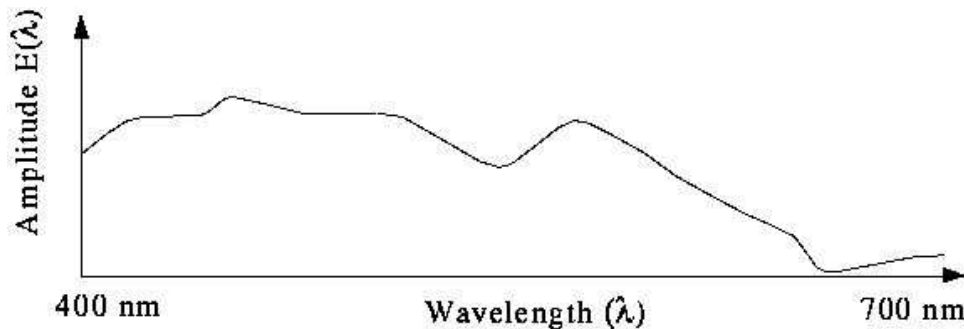


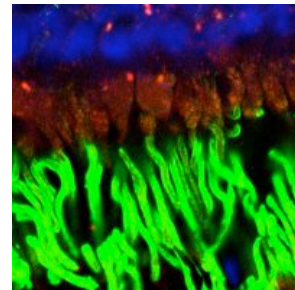
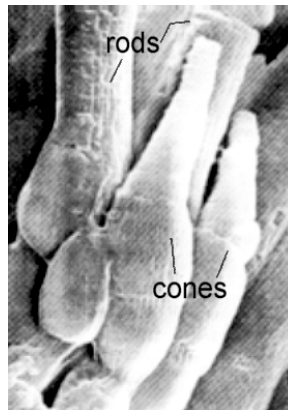
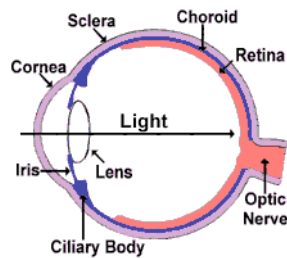
Figure 9: Light Wavelengths

- The profile above is called a *spectra*.



# The Human Eye

- The eye is basically similar to a camera
- It has a lens to focus light onto the Retina of eye
- Retina full of **neurons**
- Each neuron is either a *rod* or a *cone*.
- Rods are not sensitive to colour.



Back

Close

## Cones and Perception

- Cones come in 3 types: red, green and blue. Each responds differently to various frequencies of light. The following figure shows the spectral-response functions of the cones and the luminous-efficiency function of the human eye.

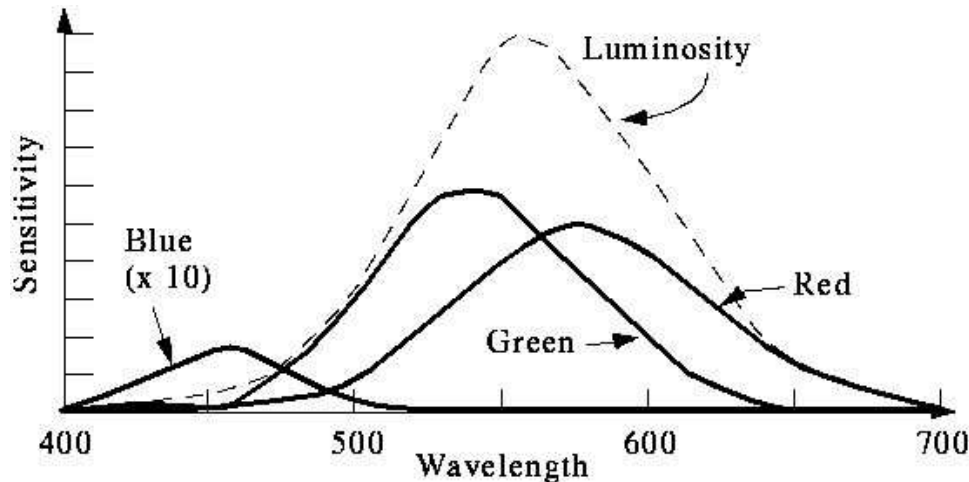


Figure 10: Cones and Luminous-efficiency Function of the Human Eye

- The profile above is called a *spectra*.



Back

Close



# RGB Colour Space



Figure 11: Original Color Image

- Colour Space is made up of Red, Green and Blue intensity components



Back

Close

# Red, Green, Blue (RGB) Image Space



Red, Green, Blue (RGB) Respective Intensities

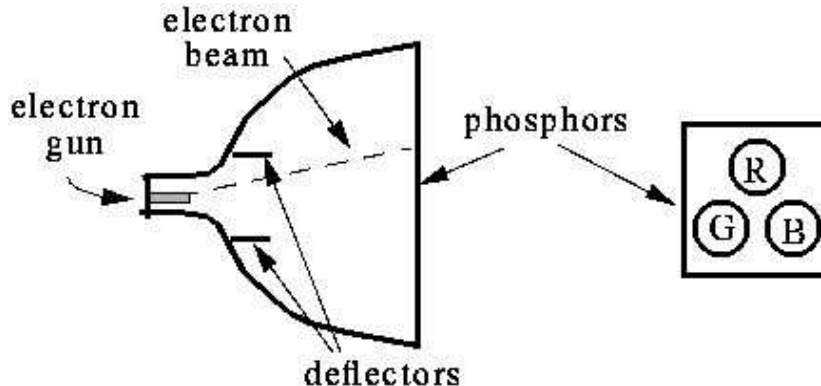


Back

Close

## CRT Displays

- CRT displays have three phosphors (RGB) which produce a combination of wavelengths when excited with electrons.



- The *gamut* of colours is all colours that can be reproduced using the three primaries
- The gamut of a colour monitor is smaller than that of color models, E.g. CIE (LAB) Model — [see later](#).

# CIE Chromaticity Diagram

Does a set of primaries exist that span the space with only positive coefficients?

- Yes, but not the pure colours.
- In 1931, the CIE defined three standard primaries ( $X, Y, Z$ ). The  $Y$  primary was intentionally chosen to be identical to the luminous-efficiency function of human eyes.
- All visible colours are in a *horseshoe* shaped cone in the  $X$ - $Y$ - $Z$  space. Consider the plane  $X+Y+Z=1$  and project it onto the  $X$ - $Y$  plane, we get the *CIE chromaticity diagram* as shown overleaf.

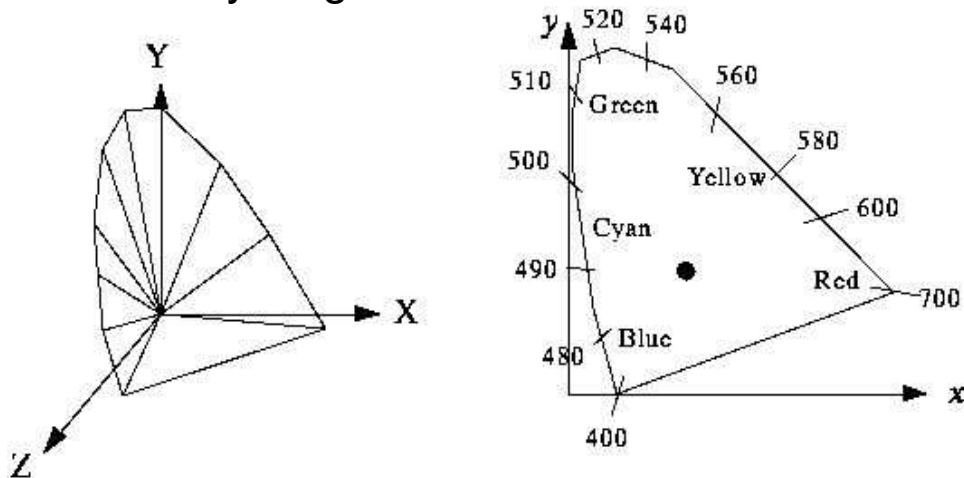


Back

Close

## CIE Chromaticity Diagram (Cont.)

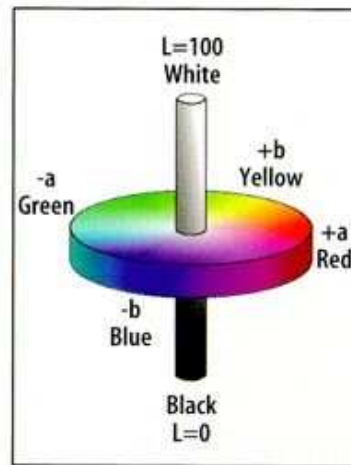
- CIE chromaticity diagram:



- The edges represent the *pure* colours (sine waves at the appropriate frequency)
- White (a blackbody radiating at 6447 kelvin) is at the *dot*
- When added, any two colours (points on the CIE diagram) produce a point on the line between them.

## L\*a\*b (Lab) Colour Model

- A refined CIE model, named CIE L\*a\*b in 1976
- **Luminance:** **L**
- **Chrominance:** **a** – ranges from green to red, **b** – ranges from blue to yellow
- Used by *Photoshop*



Lab model



Back

Close

# Lab Image Space



Original Color Image



L, A, B Image Intensities

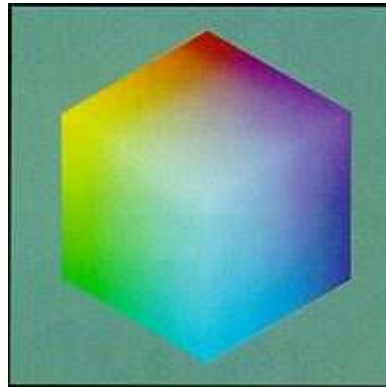


Back

Close

# Colour Image and Video Representations

- Recap: A black and white image is a 2-D array of integers.
- Recap: A colour image is a 2-D array of (R,G,B) integer triplets. These triplets encode how much the corresponding phosphor should be excited in devices such as a monitor.
- Example is shown:



Beside the RGB representation, YIQ and YUV are the two commonly used in video.



Back

Close



## YIQ Colour Model

- YIQ is used in colour TV broadcasting, it is downward compatible with B/W TV.

- Y (luminance) is the CIE Y primary.

$$Y = 0.299R + 0.587G + 0.114B$$

- the other two vectors:

$$I = 0.596R - 0.275G - 0.321B \quad Q = 0.212R - 0.528G + 0.311B$$

- The YIQ transform:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

- I is red-orange axis, Q is roughly orthogonal to I.
- Eye is most sensitive to Y, next to I, next to Q. In NTSC, 4 MHz is allocated to Y, 1.5 MHz to I, 0.6 MHz to Q.

# YIQ Colour Space



Original Color Image



Y, I, Q Image Intensities

## YUV (CCIR 601 or YCrCb) Color Model

- Established in 1982 to build digital video standard
- Video is represented by a sequence of fields (odd and even lines). Two fields make a frame.
- Works in PAL (50 fields/sec) or NTSC (60 fields/sec)
- Uses the Y, Cr, Cb colour space (also called YUV)  
 $Y = 0.299R + 0.587G + 0.114B$   $Cr = R - Y$   $Cb = B - Y$
- The YCrCb (YUV) Transform:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

# YIQ Colour Space



Original Color Image



Y, I, Q Imag Intensities



Back

Close

## The CMY Colour Model

- Cyan, Magenta, and Yellow (CMY) are complementary colours of RGB (Fig. 12). They can be used as *Subtractive Primaries*.
- CMY model is mostly used in printing devices where the colour pigments on the paper absorb certain colours (e.g., no red light reflected from cyan ink).

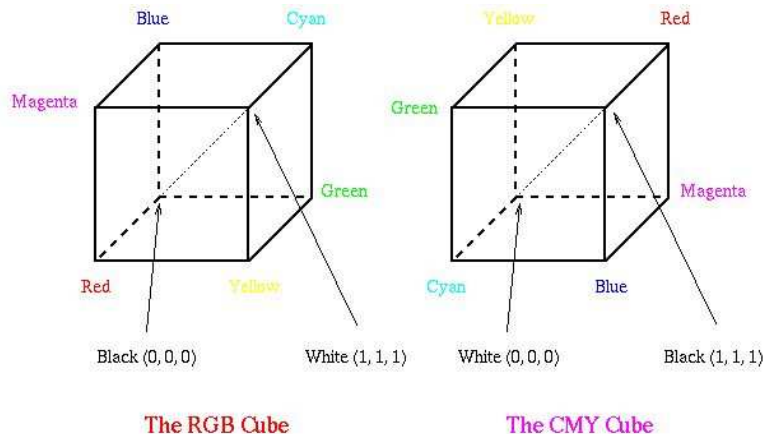


Figure 12: The RGB and CMY Cubes

## Conversion between RGB and CMY

E.g., convert **White** from (1, 1, 1) in RGB to (0, 0, 0) in CMY.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$



# CMYK Color Model

- Sometimes, an alternative CMYK model (K stands for *Black*) is used in colour printing (e.g., to produce darker black than simply mixing CMY). where

$$K = \min(C, M, Y),$$

$$C = C - K,$$

$$M = M - K,$$

$$Y = Y - K.$$

[Back](#)[Close](#)

# YIQ Colour Space



Original Color Image



C, M, Y, K image Intensities



Back

Close



# Summary of Colour

- Colour images are encoded as triplets of values.
- Three common systems of encoding in video are RGB, YIQ, and YCrCb.
- Besides the hardware-oriented colour models (i.e., RGB, CMY, YIQ, YUV), HSB (Hue, Saturation, and Brightness, e.g., used in Photoshop) and HLS (Hue, Lightness, and Saturation) are also commonly used.
- YIQ uses properties of the human eye to prioritise information. Y is the black and white (luminance) image, I and Q are the colour (chrominance) images. YUV uses similar idea.
- YUV is a standard for digital video that specifies image size, and decimates the chrominance images (for 4:2:2 video) — [more soon](#).



Back

Close

# Basics of Video

## Types of Colour Video Signals

- **Component video** – each primary is sent as a separate video signal.
  - The primaries can either be RGB or a luminance-chrominance transformation of them (e.g., YIQ, YUV).
  - Best colour reproduction
  - Requires more bandwidth and good synchronization of the three components
- **Composite video** – colour (chrominance) and luminance signals are mixed into a single carrier wave. Some interference between the two signals is inevitable.
- **S-Video** (Separated video, e.g., in S-VHS) – a compromise between component analog video and the composite video. It uses two lines, one for luminance and another for composite chrominance signal.



Back

Close

# NTSC Video

- 525 scan lines per frame, 30 frames per second (or be exact, 29.97 fps, 33.37 msec/frame)
- Aspect ratio 4:3
- Interlaced, each frame is divided into 2 fields, 262.5 lines/field
- 20 lines reserved for control information at the beginning of each field (Fig. ??)
  - So a maximum of 485 lines of visible data
  - Laser disc and S-VHS have actual resolution of  $\approx 420$  lines
  - Ordinary TV –  $\approx 320$  lines



Back

Close

# NTSC Video Colour and Analog Compression

- Colour representation:
  - NTSC uses YIQ colour model.
  - Composite =  $Y + I \cos(F_{sc} t) + Q \sin(F_{sc} t)$ ,  
where  $F_{sc}$  is the frequency of colour subcarrier
  - Basic Compression Idea

Eye is most sensitive to Y, next to I, next to Q.

- This is STILL Analog Compression:  
In NTSC,
  - \* 4 MHz is allocated to Y,
  - \* 1.5 MHz to I,
  - \* 0.6 MHz to Q.
- Similar (easier to work out) Compression (Part of ) in digital compression — **more soon**

# PAL Video

- 625 scan lines per frame, 25 frames per second (40 msec/frame)
- Aspect ratio 4:3
- Interlaced, each frame is divided into 2 fields, 312.5 lines/field
- Colour representation:
  - PAL uses YUV (YCrCb) colour model
  - composite =  

$$Y + 0.492 \times U \sin(F_{sc} t) + 0.877 \times V \cos(F_{sc} t)$$
  - In PAL, 5.5 MHz is allocated to Y, 1.8 MHz each to U and V.



Back

Close

# MATLAB Colour functions

Example MATLAB's image processing toolbox colour space functions:

## Colormap manipulation :

`colormap` — Set or get colour lookup table  
`rgbplot` — Plot RGB colourmap components  
`cmpermute` — Rearrange colours in colormap.

## Colour space conversions :

`hsv2rgb/rgb2hsv` — Convert HSV values/RGB colour space  
`lab2double/lab2uint16/lab2uint8` — Convert Lab colour values to double etc.  
`ntsc2rgb/rgb2ntsc` — Convert NTSC (YUV)/RGB colour values  
`ycbcr2rgb/rgb2ycbcr` — Convert YCbCr/RGB colour



Back

Close

# Chroma Subsampling

Chroma subsampling is a method that stores color information at lower resolution than intensity information.

Why is this done? — **COMPRESSION**

- Human visual system (HVS) more sensitive to variations in brightness than colour.
- So devote more bandwidth to Y than the color difference components Cr/I and Cb/Q.
  - HVS is less sensitive to the position and motion of color than luminance
  - Bandwidth can be optimized by storing more luminance detail than color detail.
- Reduction results in almost no perceivable visual difference.

[Back](#)[Close](#)

# How to Chroma Subsample?

Use **color difference components**. The signal is divided into:

**luma (Y)** component and

**Chroma** — two color difference components which we subsample in some way to reduce its bandwidth

## How to subsample for chrominance?

The subsampling scheme is commonly expressed as a three part ratio (e.g. 4:2:2):



Back

Close



# Chroma Subsample 3 Part Ratio Explained

Each part of the three part ratio is respectively:

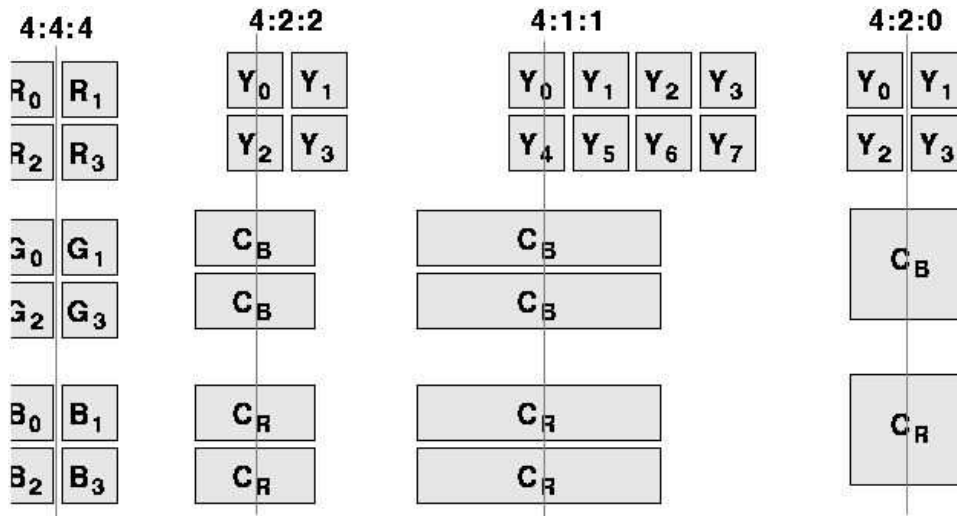
- 1: **Luma (Y) or Red (R)** — horizontal sampling reference  
(originally, as a multiple of 3.579 MHz in the NTSC analog television system — rounded to 4)
- 2: **Cr/I/G** — horizontal factor (relative to first digit)
- 3: **Cb/Q/B** – horizontal factor (relative to first digit), except when **zero**.
  - **Zero** indicates that Cb (Q/B) horizontal factor is equal to second digit, **and**,
  - Both Cr (I/G) and Cb (Qb) are subsampled 2:1 vertically.



Back

Close

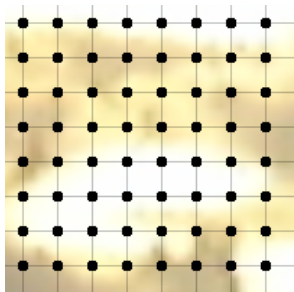
# Chroma Subsampling Examples



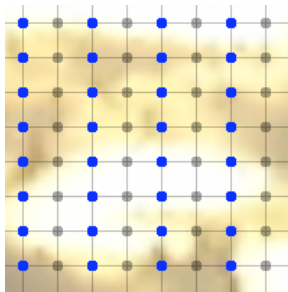
- 4:4:4 — no subsampling in any band — equal ratios.
- 4:2:2 → Two chroma components are sampled at half the sample rate of luma, horizontal chroma resolution halved.
- 4:1:1 → Horizontally subsampled by a factor of 4.
- 4:2:0 → Subsampled by a factor of 2 in both the horizontal and vertical axes

# Chroma Subsampling: How to Compute?

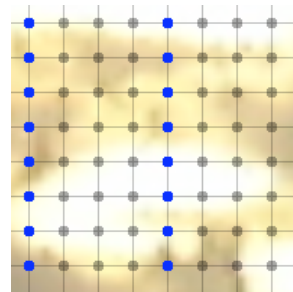
- Simply different frequency sampling of digitised signal
- Digital Subsampling: For 4:4:4, 4:2:2 and 4:1:1
  - Perform 2x2 (or 1x2, or 1x4) chroma subsampling
  - Subsample horizontal and, where applicable, vertical directions
  - *i.e.* Choose every second, fourth pixel value.



4:4:4



4:2:2

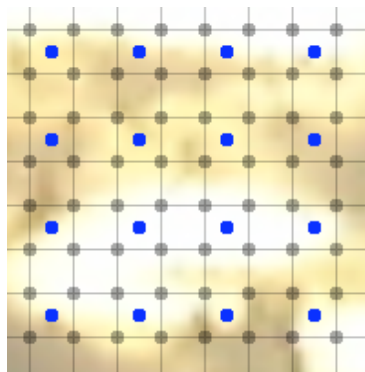


4:2:1

Subsampling

## Chroma Subsampling: How to Compute? (Cont.)

- For 4:2:0, Cb and Cr are effectively centered vertically halfway between image rows.:
  - Break the image into 2x2 pixel blocks and
  - Stores the **average** color information for each 2x2 pixel group.



4:2:0 Subampling

# Chroma Subsampling in MATLAB

The MATLAB function `imresize()` readily achieves all our subsampling needs:

`IMRESIZE` Resize image.

`IMRESIZE` resizes an image of any type using the specified interpolation method. Supported interpolation methods include:

```
'nearest' --- (default) nearest neighbour interpolation
'bilinear' bilinear interpolation
```

`B = IMRESIZE(A,M,METHOD)` returns an image that is `M` times the size of `A`. If `M` is between 0 and 1.0, `B` is smaller than `A`. If `M` is greater than 1.0, `B` is larger than `A`.

`B = IMRESIZE(A,[MROWS MCOLS],METHOD)` returns an image of size `MROWS-by-MCOLS`.

After MATLAB colour conversion to YUV/YIQ:

- Use `nearest` for 4:2:2 and 4:2:1 and scale the `MROWS` `MCOLS` to half or quarter the size of the image.
- Use `bilinear` (to average) for 4:2:0 and set scale to half.

**See next Lab worksheet**



Back

Close

# Digital Chroma Subsampling Errors (1)

This sampling process introduces two kinds of errors:

1. A minor problem is that color is typically stored at only half the horizontal and vertical resolution as the original image — *subsampling*.

This is not a real problem:

- Recall: The human eye has lower resolving power for color than for intensity.
- Nearly all digital cameras have lower resolution for color than for intensity, so there is no high resolution color information present in digital camera images.



Back

Close

# Digital Chroma Subsampling Errors (2)

2. Another issue: The subsampling process demands two conversions of the image:

- From the original RGB representation to an intensity+color (YIQ/YUV) representation , and
- Then back again (YIQ/YUV  $\rightarrow$  RGB) when the image is displayed.
- Conversion is done in integer arithmetic — some round-off error is introduced.
  - This is a much smaller effect,
  - But (slightly) affects the color of (typically) one or two percent of the pixels in an image.

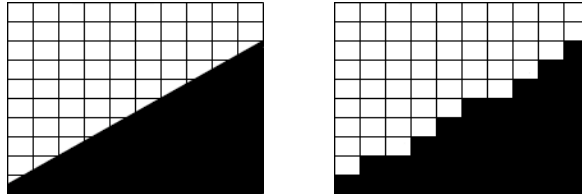


Back

Close

# Aliasing in Images

**Stair-stepping** — Stepped or jagged edges of angled lines, e.g., at the slanted edges of letters.



**Image Zooming** — changing resolution or not acquiring image in adequate resolution, e.g. digital zoom on cameras, digital scanning. (see [zoom\\_alias.m](http://zoom_alias.m))



**Explanation:** Simply Application of Nyquist's Sampling Theorem: Zooming in by a factor  $n$  divides the sample resolution by  $n$



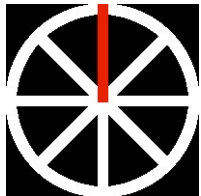
Back

Close

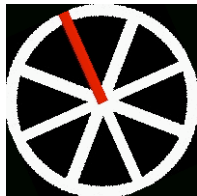


# Aliasing in Video

**Temporal aliasing** - e.g., rotating wagon wheel spokes apparently reversing direction, (see [aliasing\\_wheel.m](#) + [spokesR.gif](#)):



Frame 1



Frame2



Frame3



Frame4



Frame5

[Below Nyquist Video](#)   [At Nyquist Video](#)   [Above Nyquist Video](#)

**Raster scan aliasing** — e.g., twinkling or strobing effects on sharp horizontal lines, (see [raster\\_aliasing.m](#) + [barbara.gif](#)):

[Strobing Alias Video](#)   [Strobing Alias Frequency Distributions Video](#)

**Interlacing aliasing** — Some video is interlaced, this effectively halves the sampling frequency. e.g.: [Interlacing Aliasing effects](#)

**Image Aliasing** — Stair-stepping/Zooming aliasing effects as images.

**Explanation:** Simply Application of Nyquist's Sampling Theorem



Back

Close