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*, including lossless or lossy
- Graphics, video and audio compression techniques in next Chapter.

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.

Monochrome/Bit-Map Images

Figure 1: 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 2: Example of a Gray-scale Bit-map Image

- Each pixel is usually stored as a byte (value between 0 to 255)
- A dark pixel may have a value of 10; a bright one may be 240
- A 640 x 480 greyscale image requires over 300 KB of storage.

Dithering

- Dithering is often used when converting greyscale images to monochrome ones e.g. for printing
- The main strategy is to replace a pixel value (from 0 to 255) by a larger pattern (e.g. 4×4) such that the number of printed dots approximates the greyscale level of the original image
- If a pixel is replaced by a 4×4 array of dots, the intensities it can approximate from 0 (no dots) to 16 (full dots).
- Given a 4×4 dither matrix e.g.

$$\begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

we can re-map pixel values from 0–255 to a new range 0–16 by dividing the value by (256/17) (and rounding down to integer).

Dithering (cont.)

- A simple approach: replace each pixel by a 4×4 dots (monochrome pixels). If the remapped intensity is $>$ the dither matrix entry, put a dot at the position (set to 1) otherwise set to 0.
- Note that the size of the dithered image may be much larger. Since each pixel is replaced by 4×4 array of dots, the image becomes 16 times as large.
- To keep the image size: an ordered dither produces an output pixel with value 1 iff the remapped intensity level just at the pixel position is greater than the corresponding matrix entry.

Back
Close

24-bit Colour Images

Figure 3: Example of 24-Bit Colour Image

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

8-bit Colour Images

Figure 4: 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)

Colour Look-Up Tables (LUTs)

- Store only the index of the colour LUT for each pixel
- Look up the table to find the colour (RGB) for the index
- LUT needs to be built when converting 24-bit colour images to 8-bit: grouping similar colours (each group assigned a colour entry)
- Possible for palette animation by changing the colour map

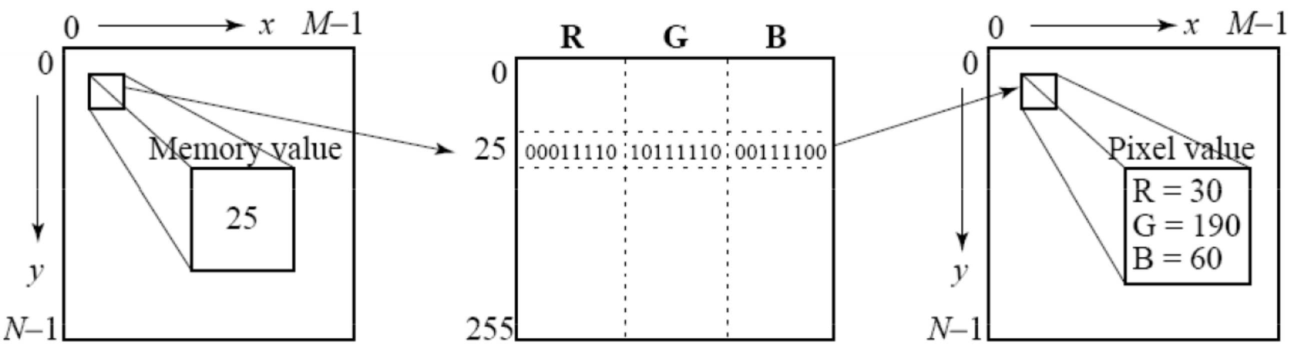


Figure 5: Colour LUT for 8-bit Colour Images

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) so lossless — **Algorithm Soon**
- Limited to only 8-bit (256) colour images, suitable for images with few distinctive colours (e.g., graphics drawing)
- Supports *interlacing*
- GIF89a: supports simple animation, transparency index etc.

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.

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 utilising 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

PNG

- PNG stands for Portable Network Graphics, meant to supersede GIF standard
- Features of PNG
  - Support up to 48 bits per pixel – more accurate colours
  - Support description of gamma-correction and alpha-channel for controls such as transparency
  - Support progress display through 8×8 blocks.

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

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

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

X-window: 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.

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. 6).

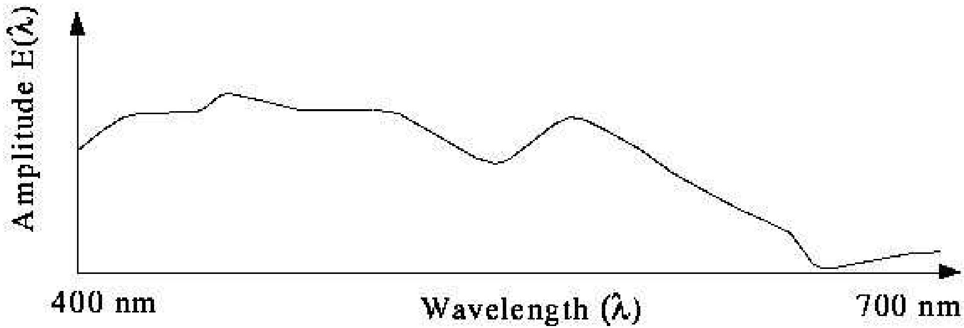
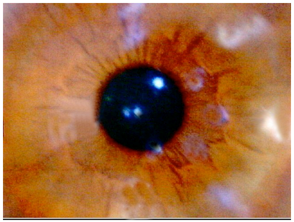


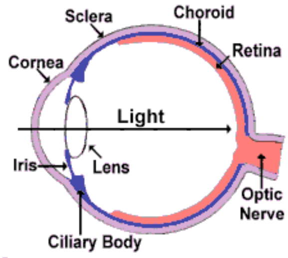
Figure 6: Light Wavelengths

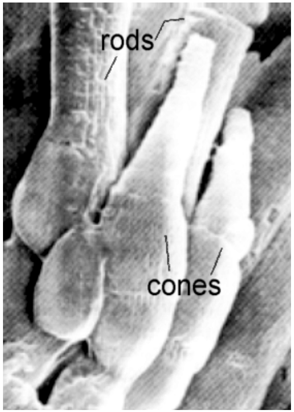
- The profile above is called a *spectra*.

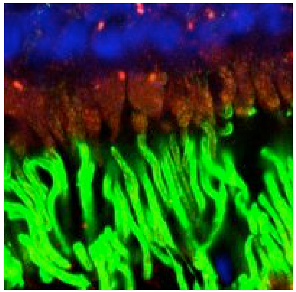
The Human Eye

- 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.









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.

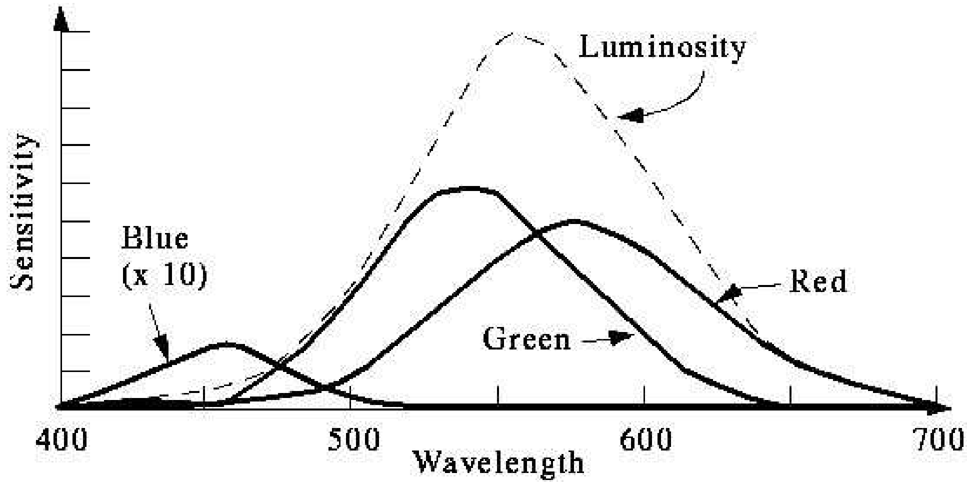


Figure 7: Cones & Luminous-efficiency Function of the Human Eye

- Human eyes can't tell lights with different spectra as long as they have the same stimuli.

RGB Colour Space

Figure 8: Original Color Image

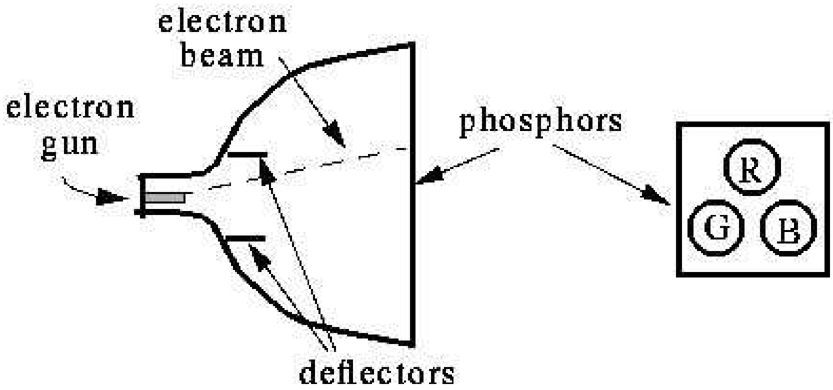
- Colour Space is made up of Red, Green and Blue intensity components

Red, Green, Blue (RGB) Image Space

Red, Green, Blue (RGB) Respective Intensities

CRT Displays

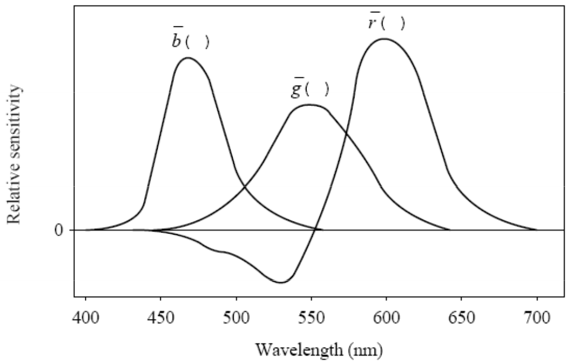
- CRT displays have three phosphors (RGB) which produce a combination of wavelengths when excited with electrons.



- 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**.

Colour-Matching Functions

- To match a given colour with the three colour primaries, a subject is asked to separately adjust the brightness of the three primaries until the resulting light match most closely to the desired colour
- The amounts of R/G/B the subject selects to match each single-wavelength light forms the color-matching curves:



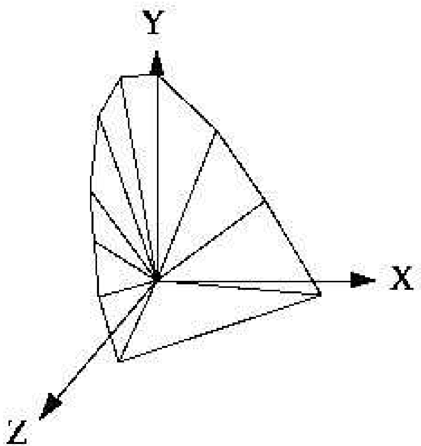
CIE Chromaticity Diagram

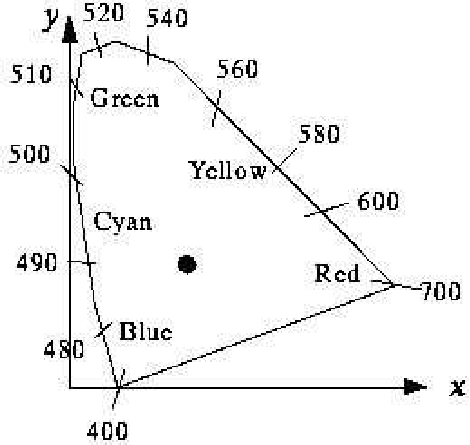
Does a set of primaries exist that span the space with only positive coefficients?

- 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.

CIE Chromaticity Diagram (Cont.)

- CIE chromaticity diagram:

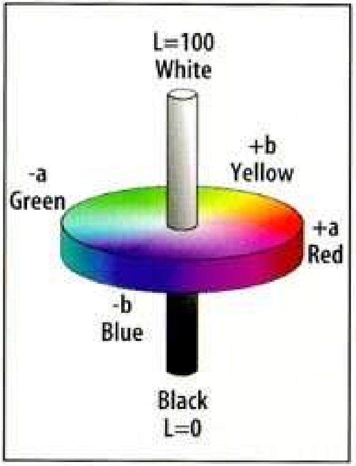




- The edges represent the *pure* colours (sine waves at the appropriate frequency)
- White (a blackbody radiating at 6447 kelvin) is at the *dot*
- When added, any two colours (points on the CIE diagram) produce a point on the line between them.

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

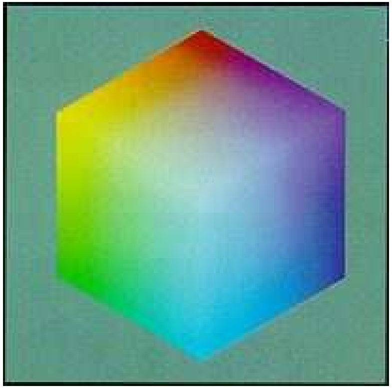
Lab Image Space

Original Color Image

L, A, B Image Intensities

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.

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:

  $I = 0.596R - 0.275G - 0.321B$ $Q = 0.212R - 0.528G + 0.311B$
- 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 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, U, V colour space

  $Y = 0.299R + 0.587G + 0.114B$, $U = B - Y$ $V = R - Y$

- The YUV Transform:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.299 & -0.587 & 0.886 \\ 0.701 & -0.587 & -0.114 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Back
Close

YCrCb (CCIR 601) Colour Model

- Similar to YUV
- YUV is normalised by a scaling

$$Cb = (B - Y)/1.772$$

$$Cr = (R - Y)/1.402$$

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

YCrCb Colour Space

Original Color Image

Y, Cr, Cb Image Intensities

The CMY Colour Model

- Cyan, Magenta, and Yellow (CMY) are complementary colours of RGB (Fig. 9). 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).

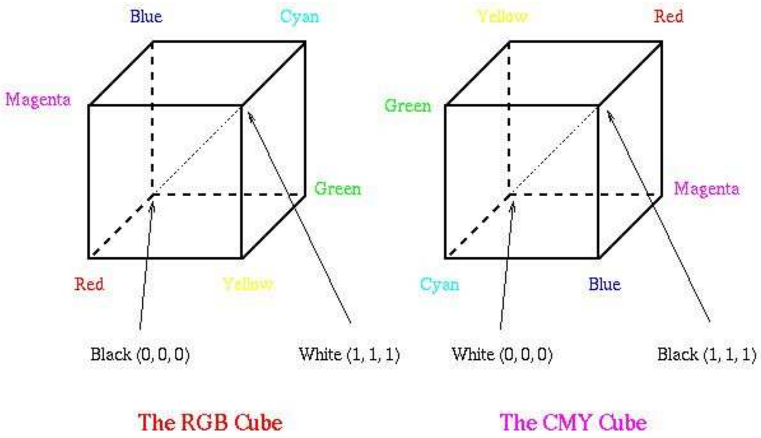
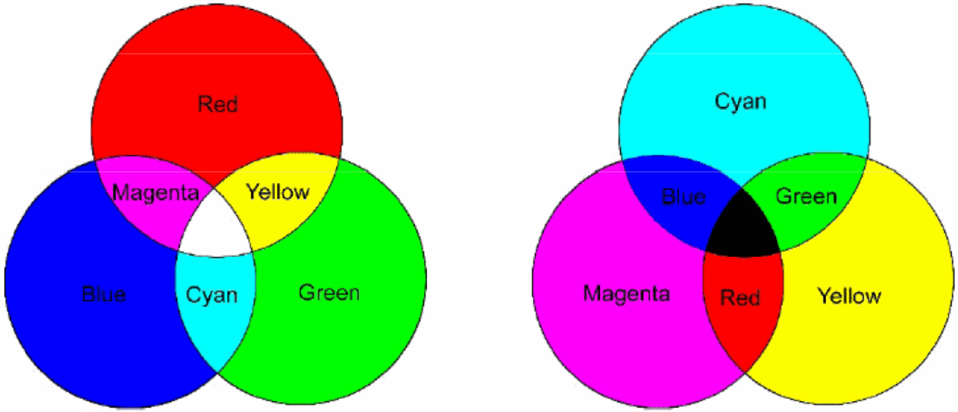


Figure 9: The RGB and CMY Cubes

The CMY Colour Model (cont.)

- Combinations of additive and subtractive colours.



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

$$
\begin{aligned}
K &= min(C, M, Y), \\
C &= C - K, \\
M &= M - K, \\
Y &= Y - K.
\end{aligned}
$$

CMYK Colour Space

Original Color Image

C, M, Y, K image Intensities

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/YCrCb is a standard for digital video that specifies image size, and decimates the chrominance images (for 4:2:2 video) — **more soon**.

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 synchronisation 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.

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
  - 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

NTSC Video Colour and Analog Compression

- Colour representation:
  - NTSC uses YIQ colour model.
  - Composite = Y + I cos(Fsc t) + Q sin(Fsc t), where Fsc 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 = $Y + 0.492 \times U\ sin(\mathrm{Fsc}\ t) + 0.877 \times V\ cos(\mathrm{Fsc}\ t)$
  - In PAL, 5.5 MHz is allocated to Y, 1.8 MHz each to U and V.

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 (YIQ)/RGB colour values
- `ycbcr2rgb`/ `rgb2ycbcr` — Convert YCbCr/RGB colour

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 optimised by storing more luminance detail than color detail.
- Reduction results in almost no perceivable visual difference.

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):

Chroma Subsample 3 Part Ratio Explained

Each part of the three part ratio is respectively:

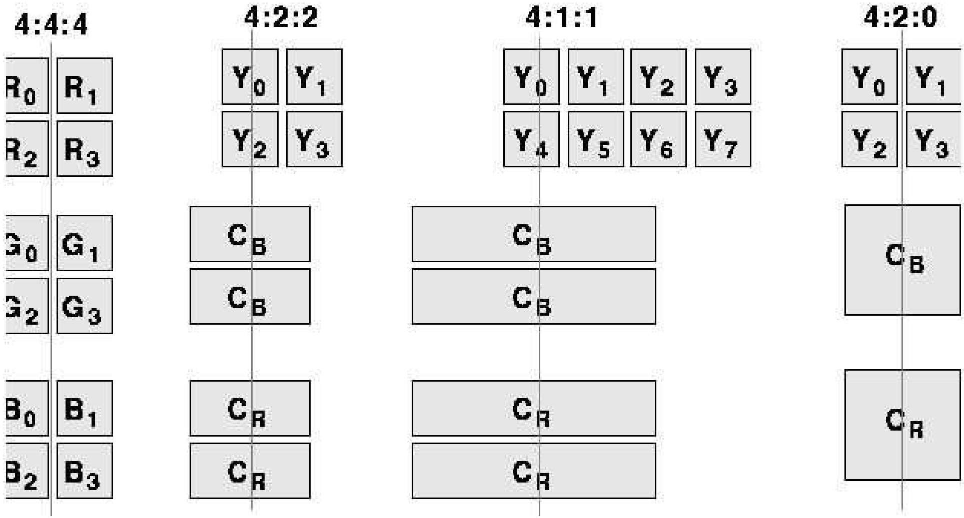
1: Luma (Y) or Red (R) — horizontal sampling reference (originally, as a multiple of 3.579 MHz in the NTSC analog television system — rounded to 4)

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.

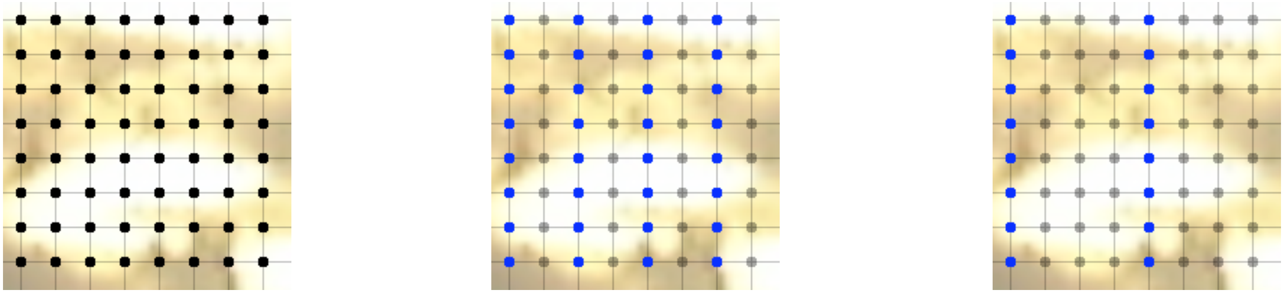
Chroma Subsampling Examples



- 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.

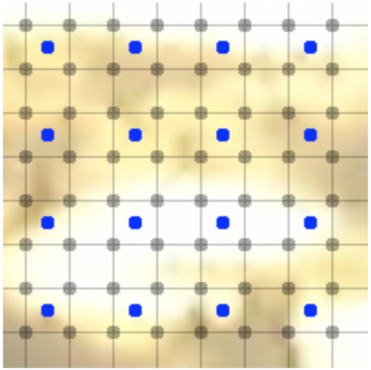


4:4:4 4:2:2 4:1:1

Subsampling

Chroma Subsampling: How to Compute? (Cont.)

- For 4:2:0, Cb and Cr are effectively centred 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 funtion `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 `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

Digital Chroma Subsampling Errors (1)

This sampling process introduces two kinds of errors:

1. A minor problem is that colour 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 colour than for intensity.
   - Nearly all digital cameras have lower resolution for colour than for intensity, so there is no high resolution colour information present in digital camera images.

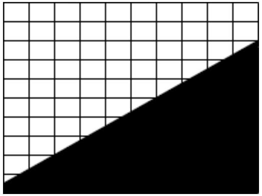
Digital Chroma Subsampling Errors (2)

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

   - From the original RGB representation to an intensity+colour (YIQ/YUV) representation , and
   - Then back again (YIQ/YUV –> 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 colour of (typically) one or two percent of the pixels in an image.

Aliasing in Images

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



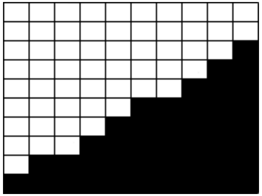


Image Zooming — changing resolution or not acquiring image in adequate resolution, *e.g.* digital zoom on cameras, digital scanning. (see zoom_alias.m)

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

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