Colour

Colour

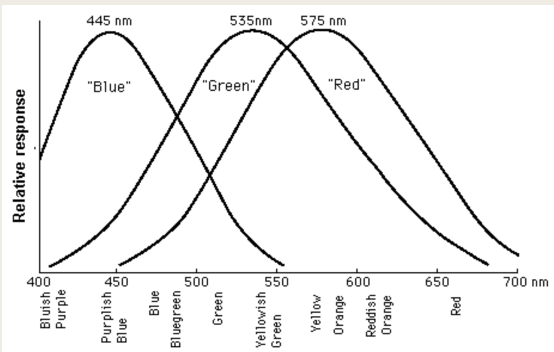
- Look at the meaning of colour – eyes response.
- Colour in the computer.
- Properties of colour. Hue, Saturation and Luminance.
- Limitations of luminance algorithms in computers.
- Colour in Matlab

What is colour?

- Visible light is broken up into wavelengths ranging from Red to Violet. (rainbow)
- This is called the visible spectrum.
- We perceive colours through the stimulation of three types of receptors in our eye called cones.

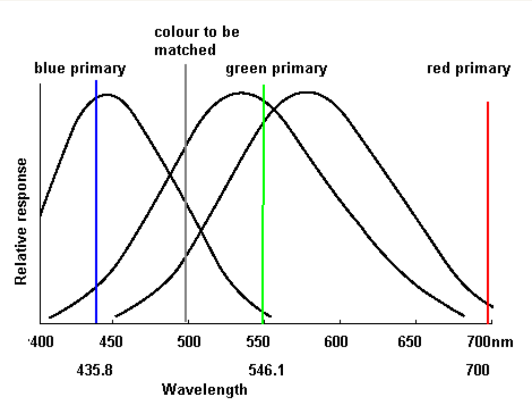
The eye's response to colour

- One cone is more responsive to red, one to green and one to blue.
- But there is overlap.
- Any single wavelength may stimulate more than one type of receptor.



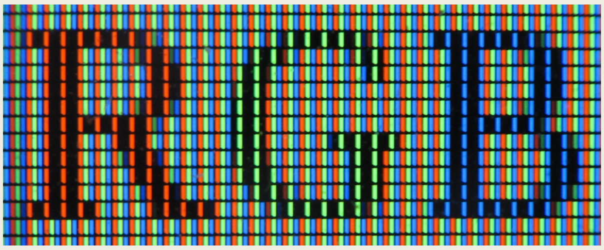
The eye's response to colour

- If we can artificially stimulate the receptors to the same degree as a naturally occurring colour, then the eyes will perceive that colour.
- We therefore try three colours to stimulate the receptors.

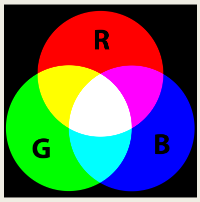


Colour in the computer

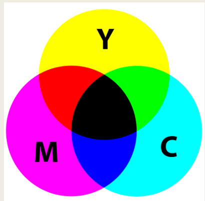
- The computer monitor only emits red, green and blue light.
- It is the combination of these lights which give the perception of colour.
- We fool the eye to “see” other colours.
- However we cannot simulate all colours in this way.



Mixing of colours

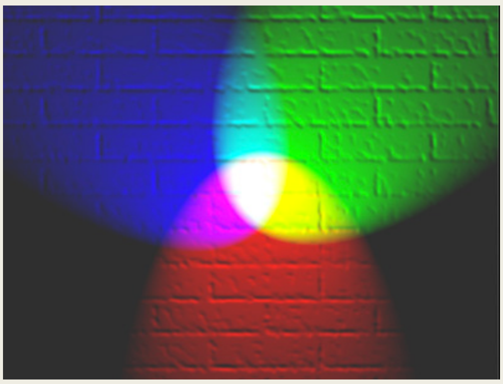


- They are mixed together and the system is arranged so that mixing the maximum (and equal) values of red, green and blue produce a nominal white colour.
- This is called "additive mixing"
- Do not confuse with "subtractive mixing" (dyes) which subtract to give black, this mainly used with printers.



Primary & Secondary colours

- Because Red, Green and Blue are used to produce all the other colours on the computer monitor, they are called **primary colours**.
- If equal amounts of all primary colours give white, what do equal amounts of any two of them give?
  - *Red + Blue gives magenta*
  - *Green + Blue gives cyan*
  - *Red + Green gives yellow*
- These important results are called **secondary colours** and are easily generated.

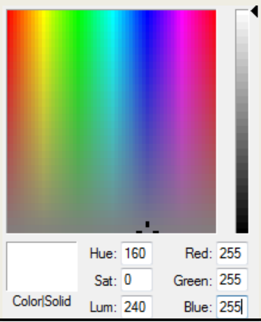


24 bit colour

- We will consider “true colour” or 24 bit colour in the computer.
- These 24 bits are divided into three groups of 8 bits.
- This gives a range of 0-255 for each colour.
- Each group controls the intensity of the primary colours: red, green and blue.

White or neutral colours

- As mentioned above the maximum equal values of red, green and blue produce white light on the computer monitor.
- That is when red=255, green= 255 and blue=255.
- We can describe this 24 bit colour (in hex) as FFFFFF.
- 3 byte representation



White or neutral colours

- Lesser equal amounts of the three primary colours produce neutral colours (shade of grey).
- Put another way, these combinations have no colour cast.
- Thus the colours:
  - *000000 (0, 0, 0) (black),*
  - *404040 (64, 64, 64)*
  - *and F0F0F0 (240, 240, 240) (nearly white)*
- are all examples of neutral colours.
- They can be considered as proportions of white.

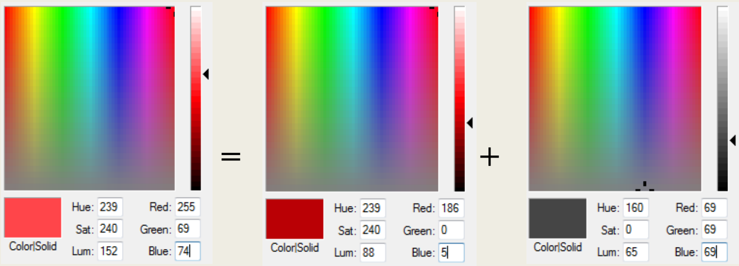
Gray Shades	HEX	RGB
	#000000	rgb(0,0,0)
	#080808	rgb(8,8,8)
	#101010	rgb(16,16,16)
	#181818	rgb(24,24,24)
	#202020	rgb(32,32,32)
	#282828	rgb(40,40,40)
	#303030	rgb(48,48,48)
	#383838	rgb(56,56,56)
	#404040	rgb(64,64,64)
	#484848	rgb(72,72,72)
	#505050	rgb(80,80,80)
	#585858	rgb(88,88,88)
	#606060	rgb(96,96,96)
	#686868	rgb(104,104,104)
	#707070	rgb(112,112,112)
	#787878	rgb(120,120,120)
	#808080	rgb(128,128,128)
	#888888	rgb(136,136,136)
	#909090	rgb(144,144,144)
	#989898	rgb(152,152,152)
	#A0A0A0	rgb(160,160,160)
	#A8A8A8	rgb(168,168,168)
	#B0B0B0	rgb(176,176,176)
	#B8B8B8	rgb(184,184,184)
	#C0C0C0	rgb(192,192,192)
	#C8C8C8	rgb(200,200,200)
	#D0D0D0	rgb(208,208,208)
	#D8D8D8	rgb(216,216,216)
	#E0E0E0	rgb(224,224,224)
	#E8E8E8	rgb(232,232,232)
	#F0F0F0	rgb(240,240,240)
	#F8F8F8	rgb(248,248,248)
	#FFFFFF	rgb(255,255,255)

Saturation

- If a colour contains only one or two primary colours, it is said to be fully (100%) saturated.
- In other words the colour is as "strong" (saturated) as it can be. (given the three primary colours).

Saturation

- If we add the missing colour(s) we **desaturate the colour.**
- We are effectively adding some white.
- Take a red colour FF0000 (255,0,0)
- Red and green are missing; so add some.
- FF454A = BA0005 + 454545
- (255, 69, 74) = (186,0,5)+ (69, 69, 69)
- 454545 is the proportion of white we have added.



Saturation

- Adding more white (equal amounts of red, green and blue) desaturates the colour.
- "Pastel" colours are desaturated colours.
- White (and greys) are totally desaturated. (0% saturation)
- For example pink is desaturated red.

Saturation

- From the above description the equation for saturation is intuitive:

$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)} \times 100\%$$

- So if any primary colour is missing then min=0 and saturation=100%
- If all primary colours are present in equal amounts, then max=min and saturation =0%.

Microsoft Saturation

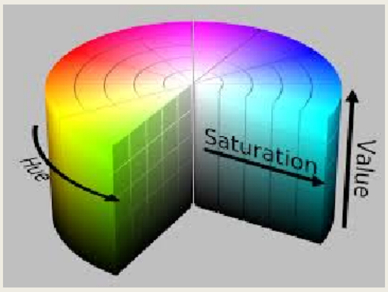
- Microsoft have a different idea about saturation. It is a variation of equation.

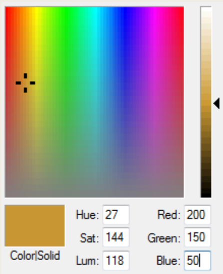
$$Saturation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue) + \min(red, green, blue)} \times 240$$

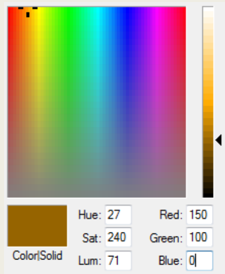
- But it is modified according to the brightness values.
- Microsoft set the maximum saturation as 240, that is why we multiply by 240.

Saturation & Hue

- We can make a desaturated colour saturated by removing the min(r,g,b) from all colours.
- Then the smallest colour primary will be missing.
- But one property of colour remains the same.
- The colour hue.

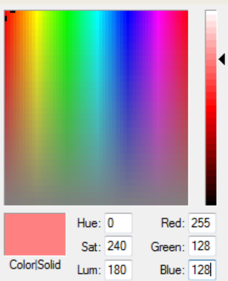


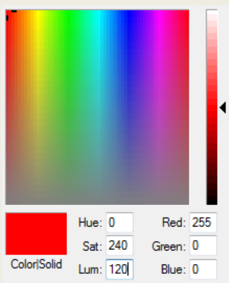


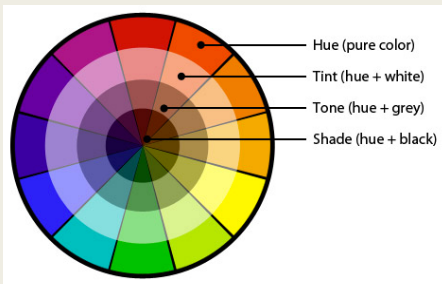


Hue

- Hue is the attribute of a visual sensation according to which an area appears to be similar to one of the perceived colours (e.g., red, green, blue, etc.)
- So pink FF8080 (255,128,128) has the same hue as red.
- It is expressed in degrees around a colour circle.
- Colours from red to blue are arranged around the circle and the colour is specified in degrees.

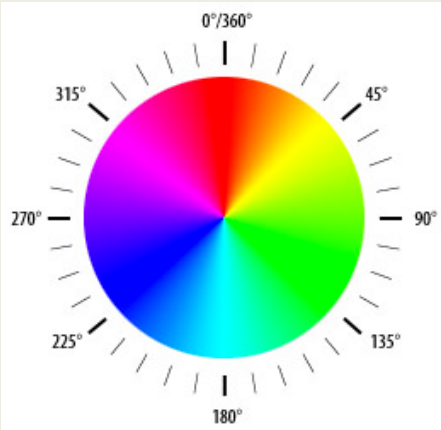


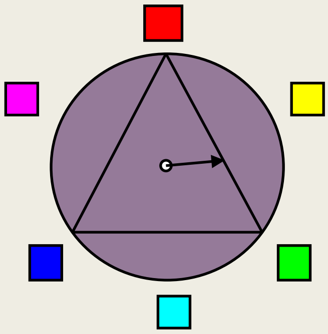




Hue

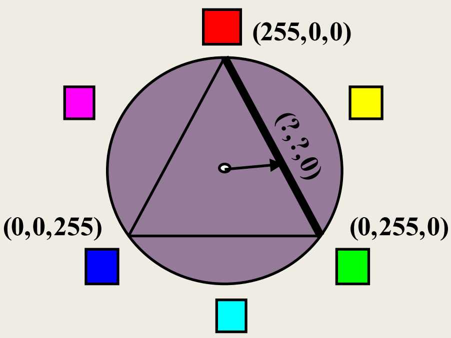
- Angular position of arrow determines the colour (hue).





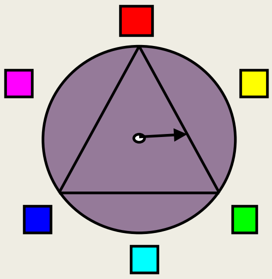
Hue

- Angular position of arrow is determined (and specified) by the distance along the line joining two of the three primary colours.
- All saturated colours lie along these lines.
- We can consider the centre of the circle/triangle as white (totally desaturated).



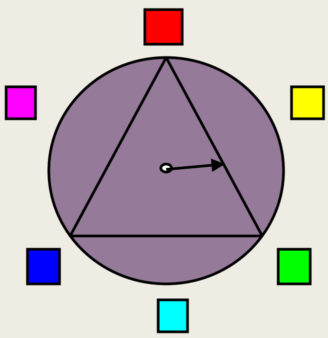
Hue

- **Each of the lines is divided into 120 steps**.
- The beginning and the end of the line correspond to 0,120, 240 degrees of angle.
- Mid points correspond to 60 degree intervals.
- But the points in between do not (quite).
- Nevertheless they are counted as degrees of colour.
- Easier for computation this way.



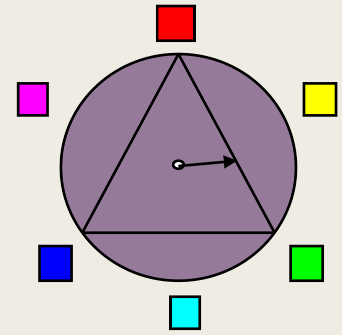
Hue

- So **for saturated colours** , the arrow will lie on one (and only one) of the lines .
  - *If a colour contains only one or two primary colours, it is fully saturated.*
- To **calculate the colour hue** we must find how far along the line the arrow lies.
- For desaturated colours it will point towards one line, unless the colour is neutral



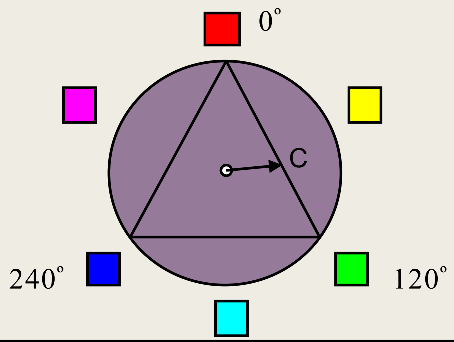
Hue

- In the diagram the arrow lies on the line red to green, but more towards the green primary.
- Max(r,g,b) will be equal to the green value in this case.
- We have to find how far along the line from red to green it lies.



Hue

- If red corresponds to zero degrees and green corresponds to 120 degrees. How many degrees is point C?
- Well, if green is max(r,g,b) then:
  - *the **point lies between the mid point of the line 60 degrees** (red value =green value)*
  - *and the green end 120 degrees where the red value=0.*



Hue

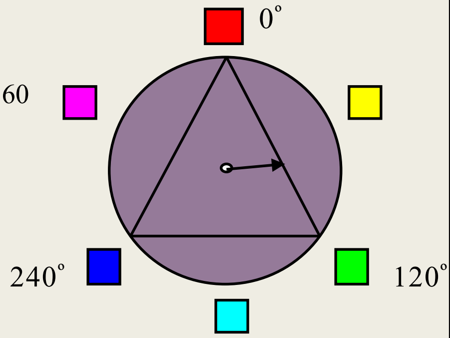
- An equation to express this would be:

$$Hue_{\max green} = 60 \times (1 - \frac{red\ value}{green\ value}) + 60$$

$$= 60 \times (2 - \frac{red\ value}{green\ value})$$

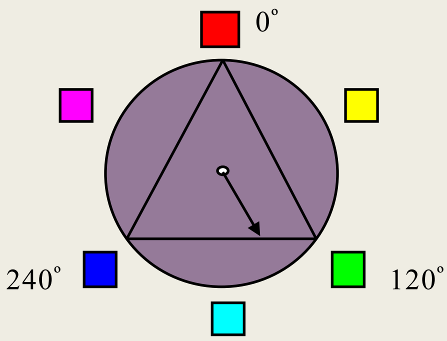
But if green value = max(r, g, b)

$$Hue = 60 \times (2 - \frac{red\ value}{\max(r,g,b)})$$



Hue

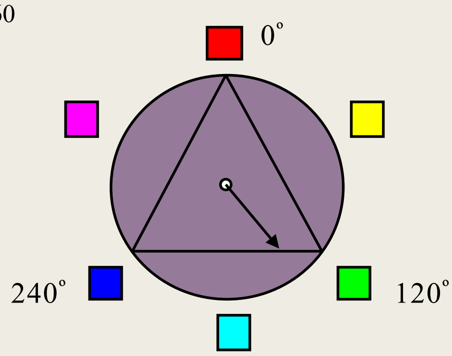
- However, if green is dominant and point C lies on the line between green and blue.
- Then the hue values will vary between green (120 degrees) and the midpoint of the green blue line.



Hue

- An equation to express this would be:

$$Hue_{\max green} = 60 \times (1 + \frac{blue\ value}{green\ value}) + 60$$
$$= 60 \times (2 + \frac{blue\ value}{green\ value})$$
$$\text{But if green value} = \max(r, g, b)$$
$$Hue = 60 \times (2 + \frac{blue\ value}{\max(r, g, b)})$$



Hue

- We have **not considered unsaturated** colours, when the third colour (red in the second case) is not zero.
- If we subtract the smallest colour (red) from all of the others. The colour will be saturated, but of the same colour hue.
- The resultant arrow will then **lie on one of the adjoining lines** and we can use the equations as before.

$$Hue_{\max green} = 60 \times (1 + \frac{blue\ value}{green\ value}) + 60$$
$$= 60 \times (2 + \frac{blue\ value}{green\ value})$$
$$\text{But if green value} = \max(r, g, b)$$
$$Hue = 60 \times (2 + \frac{blue\ value}{\max(r, g, b)})$$

Hue

- Subtracting the minimum value (red) in the second case gives

$$Hue = 60 \times (2 + \frac{blue\ value - red\ value}{\max(r,g,b) - \min(r,g,b)})$$

Hue

- And if we do this while considering all primary colours at maximum, we get a set of equations. One for each case.

- When **red** is dominant

$$Hue = 60 \times (\frac{green\ value - blue\ value}{\max(r,g,b) - \min(r,g,b)})$$

- When **green** is dominant

$$Hue = 60 \times (2 + \frac{blue\ value - red\ value}{\max(r,g,b) - \min(r,g,b)})$$

- When **blue** is dominant

$$Hue = 60 \times (4 + \frac{red\ value - green\ value}{\max(r,g,b) - \min(r,g,b)})$$

Hue anomalies

- The first equation can give negative values (when red green is minimum. However since the answer is in degrees of a circle we can add 360 degrees and get a positive (valid) answer)

- **Microsoft have a values of 240 as maximum hue**,
  - *so to convert 360 degrees to 240 Microsoft units we must multiply by:*
- 240/360 = 2/3 = 0.667

Hue

- Lets get the hang of this with some exercises.
- Calculate the colour hue if (in decimal) green=255, blue = 45 and red = 50
  - *Max(50,**255**,45)=255 so **green** is dominant we use*

$$Hue = 60 \times (2 + \frac{blue\ value - red\ value}{\max(r,g,b) - \min(r,g,b)})$$

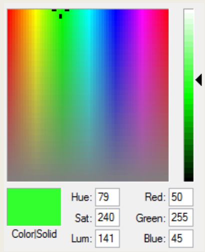
  - *Min(50,255,**45**)=45*
  - *So in the equation*

$$Hue = 60 \times (2 + \frac{blue\ value - red\ value}{\max(r,g,b) - \min(r,g,b)})$$

$$= 60 \times (2 + \frac{45 - 50}{255 - 45})$$

$$= 119^{o} \text{ (rounded)}$$

= 79 Microsoft units. Check it!



Brightness

- Brightness is a perception of the light emitted (or reflected) from an object.
- But our eyes are **more sensitive to green light** than it is for red and blue light.
- For the red, green and blue lights emitted by a computer monitor our eyes sensitivities are **30%**, **59%** and **11%** respectively.

Brightness/Value/Intensity/Luma

- Brightness, value, lightness, Intensity are terms used to loosely associate brightness with a colour.
- **HSV** stands for *hue*, *saturation*, and *value*. An alternative is **HSB** (*B* for *brightness*)
- In computer systems **no weight is given to the different colours.**
- "Value" is generally taken to be max(rgb)
- Brightness, lightness, "luma" as (max(r,g,b)-min(r,g,b))/2
  - *(Microsoft, put 240 max, so multiply by 240/255)*
- Intensity as (r + g +b) / 3

Hue, Saturation and Brightness

- Find the Hue ,Saturation and Brightness of the following colours.
  - *Red = 240, Green = 100, blue = 50*
  - *Red = 240, Green = 50, blue = 100*
  - *Red = 150, Green = 100, blue = 50*
  - *Red = 40, Green = 100, blue = 150*

Colour spaces

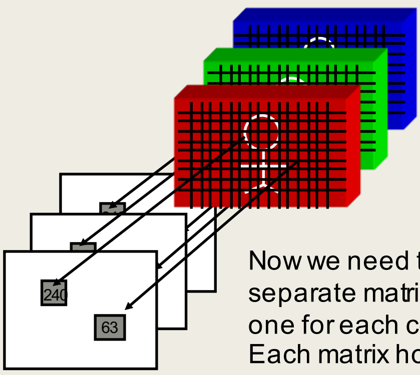
- RGB and HSB are called colour spaces
- This means that a colour can be described by suitable selection of the variables in the colour space.
- Other colour spaces exist
  - *YUV (one luma (Y') and two chrominance (UV) components)*
  - *Lab (L for lightness and a and b for the color-opponent dimensions)*

Production of a greyscale image

- Although a greyscale image can be produced from a colour image, by reducing the saturation in HSL colourspace.
- But better results would be obtained by taking the changes in sensitivity of the eye into account.
- So use from the YUV system

  $Y=0.3R + 0.59G + 0.11B$

Modifications to our bitmap structure for colour



Our matrix is now
Of size
640 x 480 x 3

Now we need three separate matrices, one for each colour. Each matrix holds values which represent the corresponding colour in the image

Colour in Matlab

Exercises

- Matlab stores (as does 24 bit .bmp) colour images in a three dimensional arrays.
- You can think of the array as three colour (two dimensional) pictures/planes, each representing one of the colours red, green and blue, (indexed as 1,2,3 respectively.
- In Matlab syntax

P(row, column, colour) will select a single pixel

*So P(40, 50, **2**) will select the pixel on row 40, column 50 colour green (2)*

Primary colours in Matlab

Exercises

- If we create an 3D array which is filled with zeros to experiment with colour.

mycolarray=zeros(100, 180, 3) ;

Cast it so as to use byte representation:

mycolarray=uint8(mycolarray)

Fill the red plane with 255:

*mycolarray(: ,: ,**1**)=255*

And view it

image(mycolarray)

Primary colours in Matlab

Exercises

- *Reset the red plane to zeros*
  *mycolarray(: , : , **1**) = 0*

- *and repeat for green*
  *mycolarray(: , : , **2**)=255*

- *and then blue (exercise).*
- *Write a small function to calculate hue, saturation, etc.*
- *Are they as expected?*

White or neutral colours in Matlab

Exercises

- If you fill all the planes with 255 you will get a white image.
- Do this plane by plane as before. (it is possible to do it in one go).
- Change the values in all arrays between 0 and 255.
- The colours should be neutral.
- Save the image

imwrite(mycolarray, ‘myfile.bmp’,)

And inspect using “Paint”

- Look at the hue saturation and brightness values.
- Are they as expected?

Secondary colours in Matlab

Exercises

- You are now going to produce the secondary colours; yellow, cyan and magenta.
- Fill your array with 255 on two of the planes only.

  Fill blue and green you will get (yellow)

  Fill red and green you will get (cyan)

  Fill red and blue you will get (magenta)
- Look at HSL values as before.

The eyes differing sensitivity to colour

Exercises

- Make up an array with the first 60 columns red, the next 60 green and the last 60 blue.
- Use 255 for all values.

```
mycolarray(:,: ,:)=0
mycolarray(:,1:60 ,1)=255
mycolarray(:,61:120 ,2)=255
mycolarray(:,121:180 ,3)=255
```

- Look at it.
- What is the brightest colour?
- What is the darkest colour?
- Sketch how you think it would look in monochrome.

The eyes differing sensitivity to colour

Exercises

- Make a monochrome version using the “Microsoft” algorithm, setting red green and blue to equal values giving a neutral image.
  - *See Code 1 at end of presentation*

This would not be a good monochrome picture.

True Luminance

Exercises

- Make a monochrome version using the equation for the Y (Luma) component of YUV.

See Code 2 at end of presentation

- Inspect it
- Now the monochrome image reflects the eyes differing sensitivity to colour.
- Note that all of the band are less than white.
- This allows true white to be represented correctly.

Mix Colours

Exercises

- Create the colours that you used in the Hue exercises.
- Inspect using "Paint".
- Are the HSL values the same as you calculated.

True luma demo code

Exercises

Code 1

```
mycolarray=double(mycolarray)
mono_msoft(:,:,1)  =(max(mycolarray(:,:,3),max(mycolarray(:,:,1), mycolarray(:,:,2)))+ ...
min(mycolarray(:,:,3),min(mycolarray(:,:,1), mycolarray(:,:,2))))/2
mono_msoft(:,:,2)  =mono_msoft(:,:,1)
mono_msoft(:,:,3)  =mono_msoft(:,:,2)
image(uint8(mono_msoft))
```

Code 2

```
mycolarray=double(mycolarray)
mono_true_lum(:,:,1)  =0.3*mycolarray(:,:,1)+0.59*mycolarray(:,:,2)+0.11*mycolarray(:,:,3)
mono_true_lum(:,:,2)  = mono_true_lum(:,:,1)
mono_true_lum(:,:,3)  = mono_true_lum(:,:,2)
image(uint8(mono_true_lum))
```