Chapter 5: Colour Processing

Andre L. C. Barczak

Computer Science
Massey University, Albany

Contents

Contents

Introduction

- Colours present additional cues
- What are the “correct” colours?
- Mixing colours have the same effect in different devices?
- *Additive* X *Subtractive* systems
- “primary” colour somewhat different from art (R,B,Y)

Colours in three components

- Humans can see between 400 nm to 700 nm
- under 400 nm: ultra-violet
- above 700 nm: infra-red
- Humans have three main sensors, R G B
- Spectral sensitivity is approximately shown in figure.

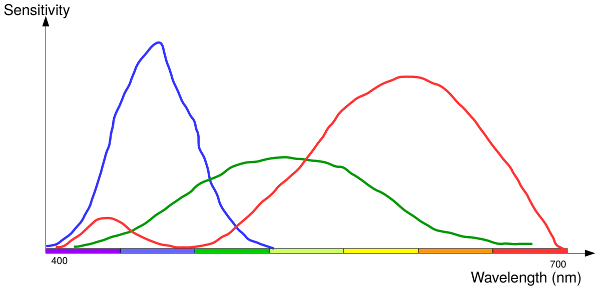


Figure 1: Spectral sensitivity.

Bayer Pattern

- Cameras are designed to be similar to Human vision
- Difficult to make the sensors to behave the same
- in practise, G is much less sensitive than B and R
- Photosensitive cells are built with the Bayer pattern (below)

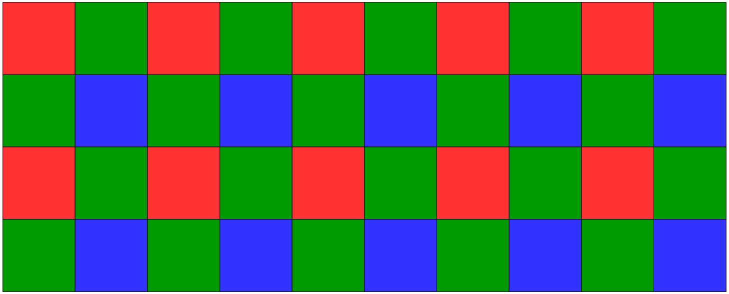


Figure 2: Bayer pattern.

Additive X Subtractive

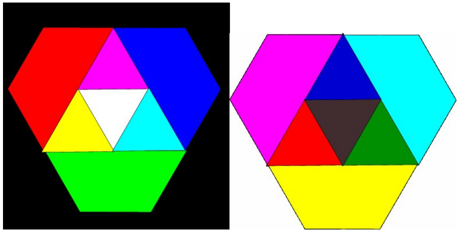


Figure 3: RGB space and YMC space. Notice the "brownish" colour in the centre of the YMC space.

Introduction (cont.)

Sensors and frequencies

- Three sensors systems: RGB, others?
- Different intensities of Red, Green and Blue, e.g., from 0 to 255
- Printing: opposite idea, fill up a white surface with ink
- Magenta, Yellow and Cyan
- No colour system is able to produce all the natural colours...

Introduction (cont.)

some definitions

- Luminance: amount of energy perceived (sensors)
- Radiance: amount of energy given out (objects)
- Brightness: sensation of the strenght of a particular wavelength
- Hue: dominance of a particular wavelength
- Saturation: purity of a particular wavelength (degree of mixing with white)

Introduction (cont.)

insights to the definitions

- Some of these definitions are too subjective
- Others are exact, but difficult to implement (e.g., how to correlate energy to a variable type int?)
- Advice: always check how these are defined with the library you are using...
- Example: Hue can be defined as an angle from 0 to 360.
- Question: if H is represented by an uchar, what is the accuracy of H?
- Question: does the implementation use the entire byte?

Contents

Colour Spaces

there are too many colour spaces

- Some of the standard colour spaces are:
- CIE XYZ, CIELUV, CIEUVW, CIELAB, CMYK...
- HSL, HSV, HSI,
- RGB, RGBA (alpha channel)
- YIQ, YUV, YPbPr, YCbCr, xvYCC
- for us, it suffices to describe RGB, HSV and YUV...

RGB

- usually one byte per colour
- 24 bits amounts to 2^{24} colours
- we cannot perceive them all
- some natural colours are missing, e.g., gold
- RGB model emulates human vision well
- RGB or BGR etc...

RGB

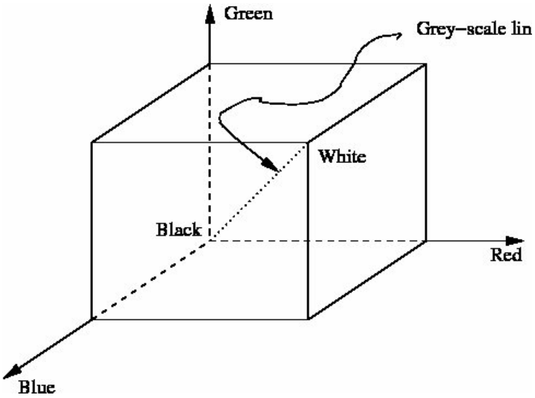


Figure 4: RGB space.

HSV

- Hue, Saturation and Value (alternatively, Intensity)
- different ways of expressing H, S and I
- e.g., H can be an angle, between red (0) and violet (359)
- I or V (there is a difference, but in practise...)
- S and I represented in similar way as R,G and B.

HSI space in GIMP

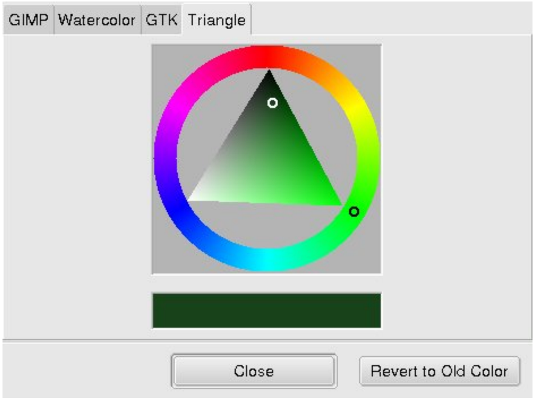


Figure 5: HSI space.

YUV

- Similar to HSV or HSI, but uses different sizes for each component
- Humans are more sensitive to Y (luminance)
- YUV uses Y with full resolution
- U and V chrominance values are stored in low resolution
- Most common YUV411 (also YUV422 etc)

YUV411

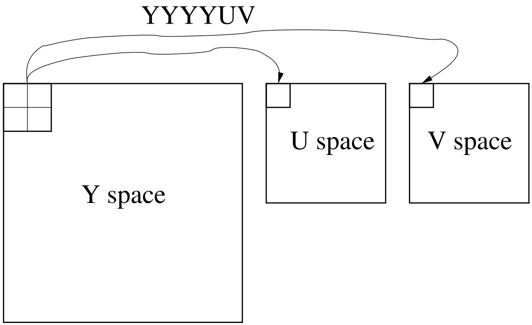


Figure 6: YUV411 space.

Converting colours

- Conversion depends on implementation
- e.g., one component does not use 8 bits
- or the definition does not fit in 8 bits
- e.g., H varies from 0 to 360
- rounding up values lose info

Converting colours in OpenCV

- `cvtColor(InputArray src, OutputArray dst, int code, int dstCn=0 );`
- e.g., `cvtColor(frame,greyimage, CV_BGR2GRAY);`
- careful: there are CV_BGR2GRAY and CV_RGB2GRAY transforms
- In OpenCV: $graypixel = 0.299R + 0.587G + 0.114B$

RGB to HSV

Converting RGB to HSV (pseudo-code):

```
$V = max(R, G, B);$
$if(V! = 0)\ then\ S = (V - min(R, G, B)) * 255/V;$
$else\ S = 0;$
$if(V = R)\ then\ H = (G - B) * 60/S;$
$if(V = G)\ then\ H = 180 + (B - R) * 60/S;$
$if(V = B)\ then\ H = 240 + (R - G) * 60/S;$
$if(H < 0) then\ H = H + 360;$
```

Converting colours in OpenCV

- $0 < H < 360$, in OpenCV H is restricted by 256 or 65536
- for **8 bit** images:
  $H = H/2$ (to fit into 255),
  $V = 255\,V$ and $S = 255\,S$
- for **16 bit** images:
  $H = H$ (uses only 360 out of 65535),
  $V = 65535\,V$ and $S = 65535\,S$
- for **32 bit** image: left as they are computed by the previous algorithm.

Converting colours

```
void ConvertRGBtoHSV(){
    min = MIN(r,g,b);
    max = MAX(r,g,b);
    v = max;
    if(max != 0) s=(max-min)/max;
    else s=0;
    if(r==max) h = (g-b)/(max-min);
    else{
        if(g==max) h=2+(b-r)/(max-min);
        else h=4+(r-g)/(max-min);
    }
    h=h*60;
}
```

Contents

Colour Segmentation

- Use 2 threshold, max and min
- This only defines a cubic or an oblong region
- For more sophisticate classification $\Rightarrow$ AI

Colour Segmentation

In RGB space

- RGB do not separate illumination from colours
- Oblong region may be good for some colours
- For other colours, results depend on constant illumination
- E.g., skin colours are not contained in oblong regions in RGB space

Colour Segmentation

In HSV space

- Split illumination from H (hue) and S (saturation)
- Oblong region may be just enought for most cases
- Allow for some variation in illumination
- More robust than with RGB, but...
- Sometimes this is not enough and requires other approch (see Fuzzy Colour Contrast)

Colour Segmentation

Demonstration

- Watch the demonstration: RGB X HSV

Contents

High Dynamic Range

- Colour classification has important applications.
- However, changing illumination conditions is an issue.
- Many solutions, from simple RGB to HSV conversion to sophisticated Fuzzy approaches that can overcome colour migration.
- Colour constancy algorithms tend to work well, but not under extreme illuminations.
- Most algorithms are limited by the sensors: extreme illuminations tend to merge colours into either low or high RGB values.

Multiple Exposures

Gustave LeGray, a painter and photographer, used a combination of pictures with different exposures as early as 1850s.

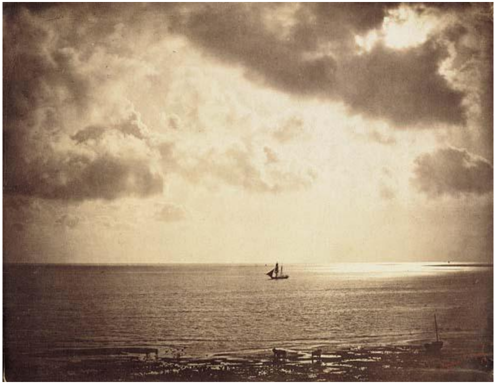


Figure 7: First HDR to LDR approach.

Dynamic Range

- Luminance range, i.e., difference between brightest to darkest
- High Dynamic Range (HDR)
- Standard Dynamic Range (SDR)
- Low Dynamic Range (LDR)

In the real world...

Luminance (candela/m2)

Star	0.001
Moon light	0.1
Indoors	50
Sunny day outside	100000

Devices vs. human eye

from brightest to darkest

Human Eye	1,000,000:1
Film (in the old school sense)	2000:1
Monitor/TV	500:1
Good Digital Camera	300:1
Printing (good quality)	200:1
Cheap digital camera	100:1

Using Multiple LDR Cameras

- Cheap digital cameras offer only LDR frames.
- Using multiple cameras, use different exposures, take simultaneous frames, combine them somehow (or not...) and then classify.
- Several issues:
  - pixel alignment (geometric calibration): not covered in this talk.
  - movement of objects or cameras: constrained to same plan.
  - colour calibration (spectrometric calibration).

HDR

Typical Approaches

- Multiple exposures using a single camera
    - Available on modern cameras
    - Automatic exposure bracketing, aka AEB
- Dual ccd cameras
    - With a prism to split the light.
    - This model launched as a commercial camera recently.

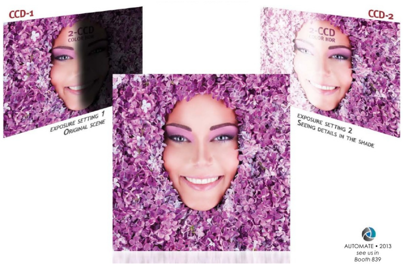


Figure 8: Dual CCD approach. (source: Vision Systems Design magazine, Jan/2013)

Single Camera, Multiple exposures

- Disadvantages:
  - Difficult to align the pixels (image registration)
  - Objects move
  - Camera itself moves
  - Light changes
- Advantages:
  - Easier to carry out spectrometric calibration (single CCD)
  - Cheap, simple devices

Dual CCD with optical splitter

- Disadvantages:
  - Expensive, depends on good optics (the prism is expensive)
  - The splitter can introduce noise, signal is weaker
  - Only two cameras are viable, so limited set of exposures
- Advantages:
  - Potential to combine techniques (dual CCD multiple exposures)
  - No pixel alignment issues (at least for the user)

Multiple CCDs

- N Cameras side by side.
- Disadvantages:
  - Geometric calibration can be an issue, difficult to align pixels in 3D.
  - Easy to implement with a fixed geometry (e.g., the frames are gathered from a fixed plan).
- Advantages:
  - Availability of multiple frames, easy to eliminate over-saturated or noisy pixels.
  - More data as input to a classifier.

Multiple LDR to HDR?

$N * LDR \rightarrow HDR$: need to combine multiple frames into one HDR. Plenty of methods:

- Parametric, pixel-based:
  - Mitsunaga and Nayar (1999)
  - Mann and Piccard (1995)
- Parametric, histogram-based:
  - Grossberg and Nayar (2003)
  - Nayar (2004)
- Non-parametric, pixel-based:
  - Devlin et al. (2002)
  - Robertson et al. (1999)
  - Debevec and Malik (1997)

HDR to LDR?

- Why *HDR* $\rightarrow$ *LDR*? To show on devices with LDR capabilities.
- Plenty of algorithms to convert HDR into LDR.
- "Tone mapping" is a common term to describe mapping.
- Paper (printers) and monitors are LDR.
  - Converting HDR to LDR through tone mapping is visually appealing.

  - But will it improve colour segmentation?
  - Lost information when saving LDR.
  - What to do with saturated pixels? Noisy low value pixels?

Tone mapping example

This is a counter example: the exposures are not known accurately, and no colour calibration was carried out. Note the noise and the colour migration, particularly on the darker parts of the image.

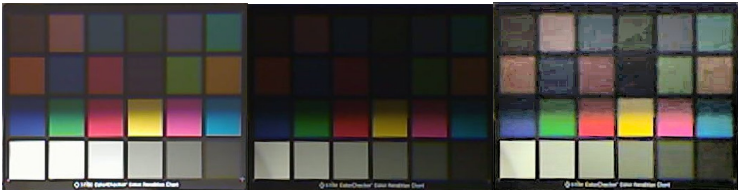


Figure 9: Tone mapping example.

Which one? HDR or LDR?

- Either: $N * LDR \rightarrow HDR \rightarrow classify$
- Or: $N * LDR \rightarrow HDR \rightarrow LDR(after\ tone\ mapping) \rightarrow classify$
- A third alternative (here is the novelty):
  - No conversion, and use all the data available directly
  - Train a classifier using N*3 dimensions (easy with AdaBoost)
  - Advantages:
    - No need for colour calibration (in fact, the training is doing that).
    - No linear-response requirements (the CCD behaves linearly, but who knows what happens when we finally get pixels values...).
    - The training requires the same effort, without adding noise.
    - Colour class can be redefined arbitrarily (important!).
  - Disadvantage:
    - Getting enough training examples is hard...

Experimental Setup

- Three cheap web cameras (Logitech 9000)
- Simplistic geometric calibration (fixed plan, fixed cameras)

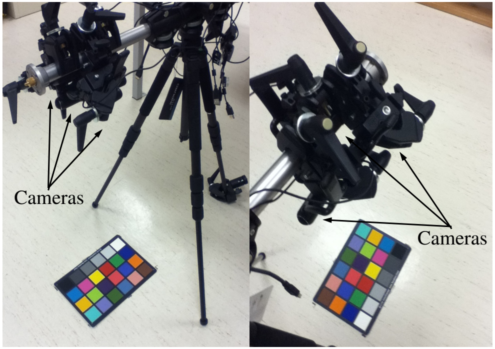


Figure 10: Three cameras approach.

Setup

- 24 colours chart (standard colours).
- classify only 8 colour at this stage: red, green, blue, white, black, magenta, yellow, orange.
- 7 levels of illumination.
- Looking at the chart: some pixels are saturated, others are too dark.

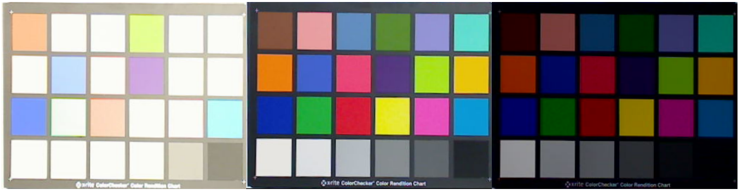


Figure 11: The brightest illumination for each camera.

Setup

- Keep the original exposure in each camera, take frames from the 3 cameras.
- Note that most pixels are too dark, except for camera 1.
- Let's look at the data for the RGB tuples. (take pixels within a mask).

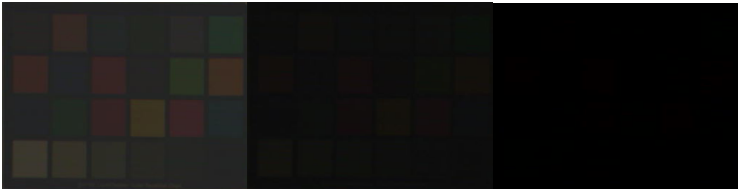
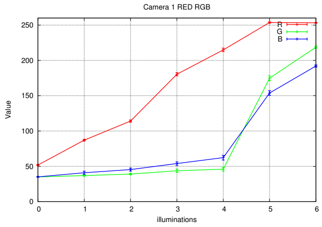
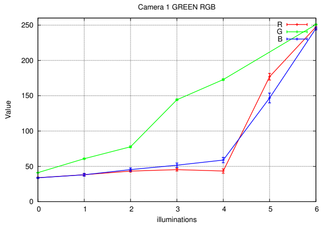


Figure 12: The darkest illumination for the 3 cameras.

Camera 1

Red, Blue and Green





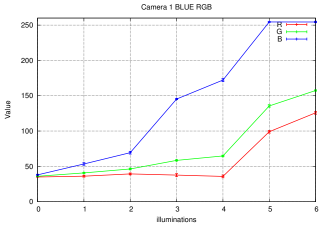


Figure 13: RGB TESTS camera 1

Camera 1

black, white, yellow, orange and magenta

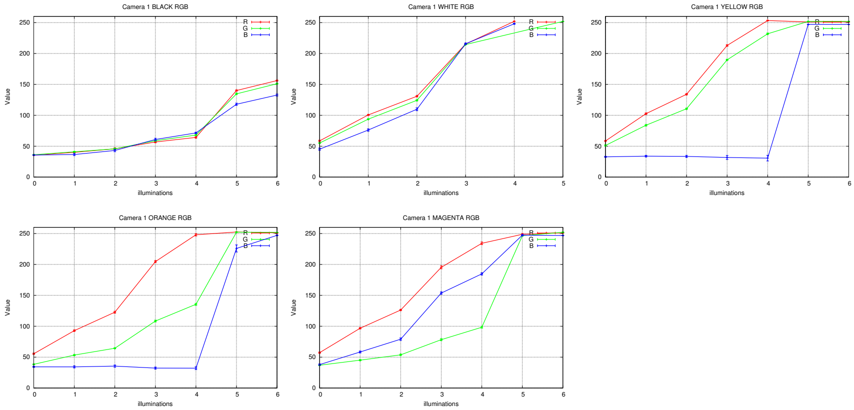


Figure 14: RGB TESTS camera 1

Camera 2

Red, Blue and Green

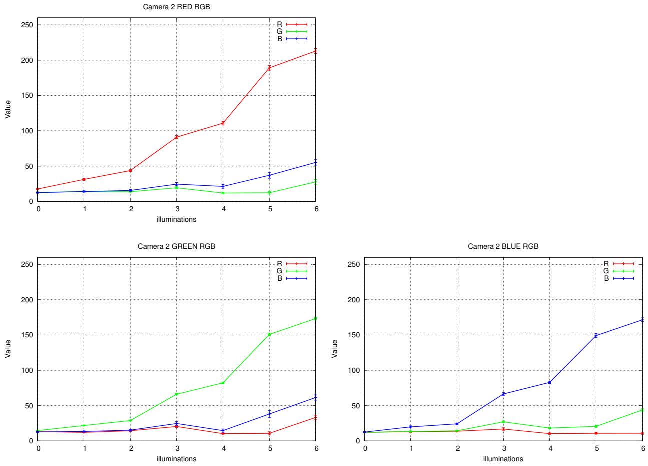


Figure 15: RGB TESTS camera 2

Camera 2

black, white, yellow, orange and magenta

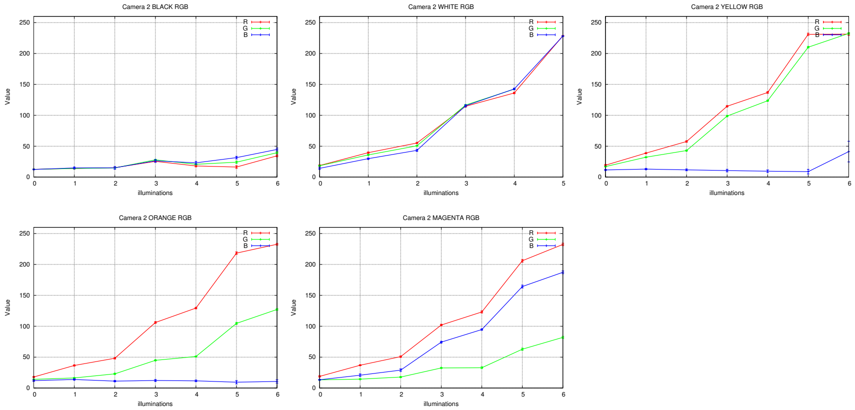


Figure 16: RGB TESTS camera 2

Camera 3

Red, Blue and Green

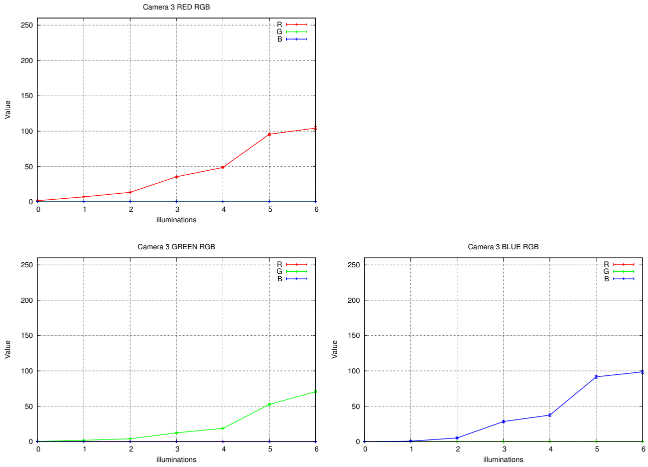


Figure 17: RGB TESTS camera 3

Camera 3

black, white, yellow, orange and magenta

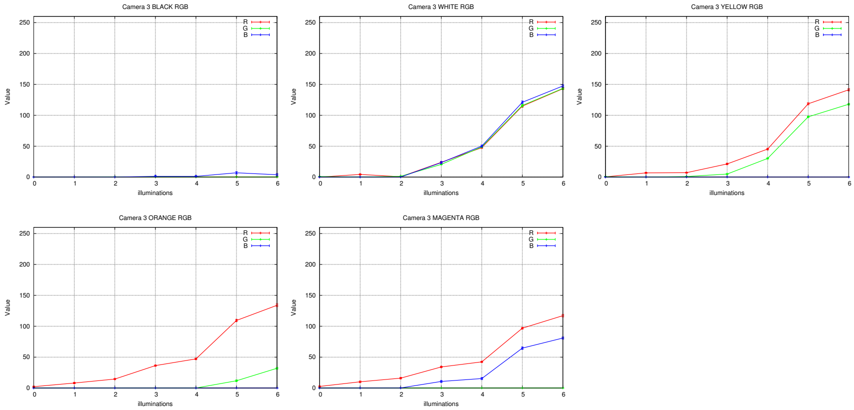


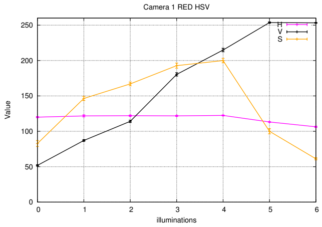
Figure 18: RGB TESTS camera 3

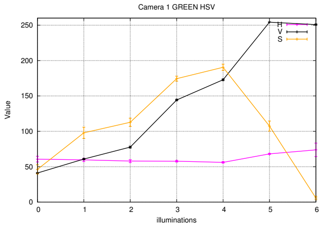
RGB tuples

- Linearity with illumination?
- Clearly a lot of noise on the low illumination
- No clear rule when the pixels saturate

Camera 1

Red, Blue and Green





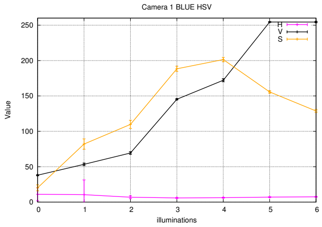


Figure 19: HSV TESTS camera 1

Camera 1

black, white, yellow, orange and magenta

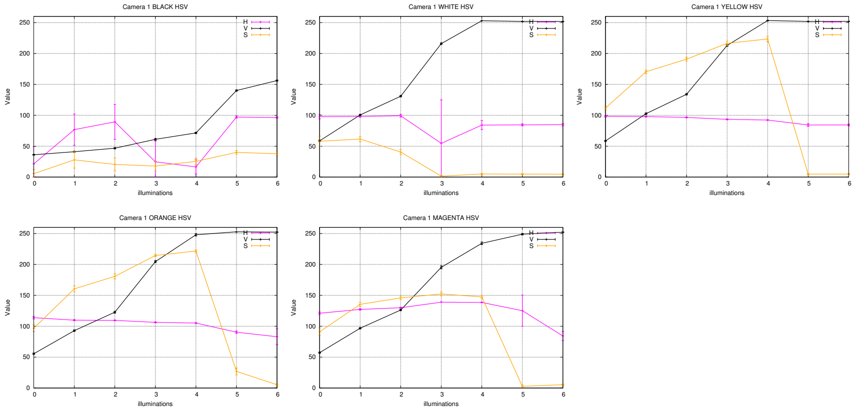
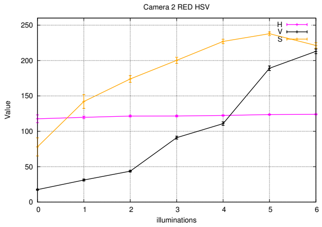
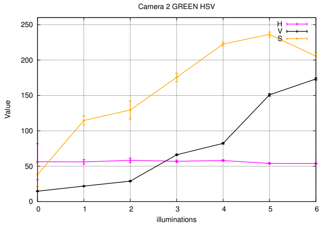


Figure 20: HSV TESTS camera 1

Camera 2

Red, Blue and Green





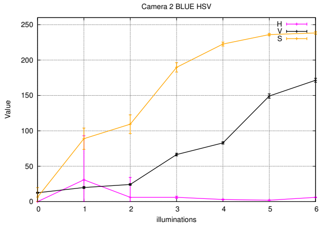


Figure 21: HSV TESTS camera 2

Camera 2

black, white, yellow, orange and magenta

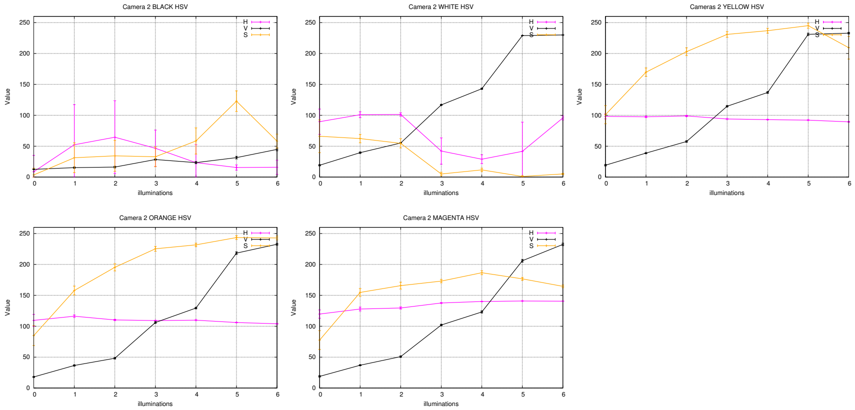


Figure 22: HSV TESTS camera 2

Camera 3

Red, Blue and Green

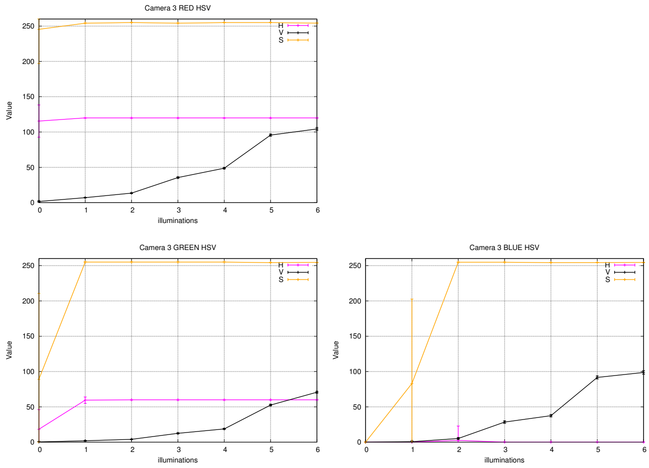


Figure 23: HSV TESTS camera 3

Camera 3

black, white, yellow, orange and magenta

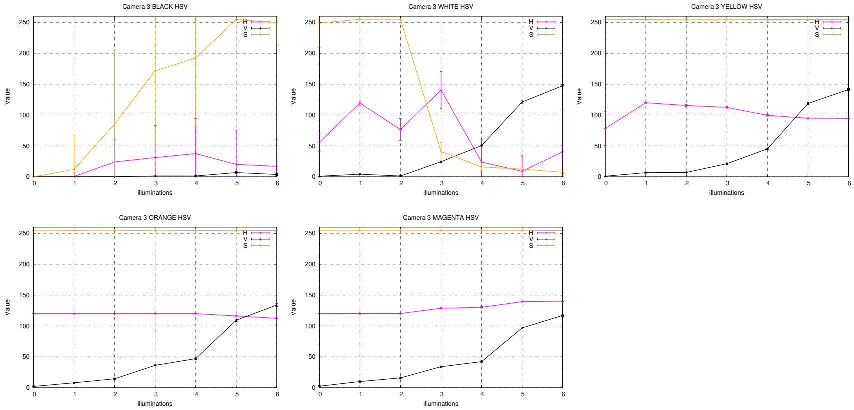


Figure 24: HSV TESTS camera 3

HSV tuples

- H should be independent of illumination (constant).
- V should be linear.
- They clearly are not, this is why we need the classifier in the first place!
- H is unstable for some hues (black and white).

About AdaBoost

- Really good for binary problems.
- Multi-class needs a more sophisticated approach.
- AdaBoost ECC is the best option for this case, with a medium number of classes.
- Implementation uses cascades (ensemble of classifiers) to make it run faster.

What we expected to see

- No difference between RGB and HSV tuples when using a trained classifier
- Limited accuracy due to the limitation in training samples
- A reasonable correlation with colour (e.g., yellows close to oranges, greens close to blues etc..)

Training convergence

7 RGB classifiers

- Trained separate multi-class classifiers
- RGB with one, two and three cameras

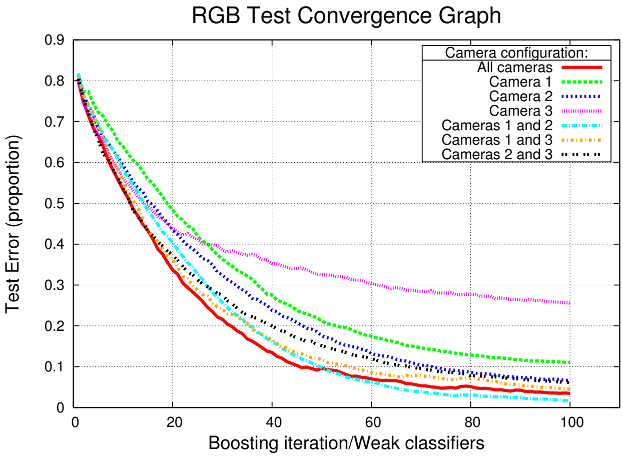
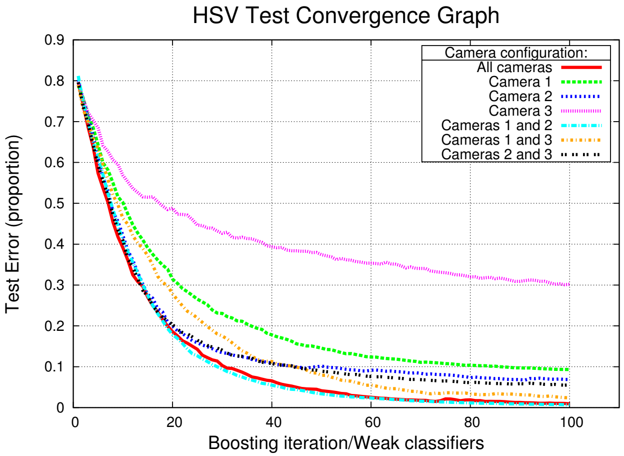


Figure 25: Classification errors for RGB data with different number of

Training convergence

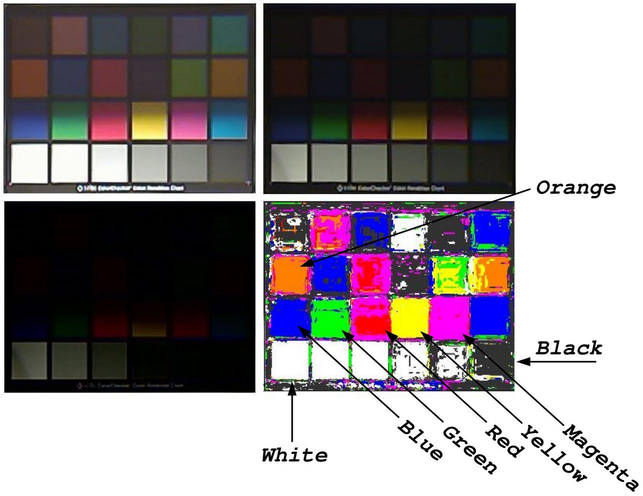
7 HSV classifiers

- Trained separate multi-class classifiers
- HSV with one, two and three cameras



Classification: getting test sets

- Get three frames, with mixed illumination (strong shades)
- Classify, using 1, or 2 or 3 cameras, RGB or HSV



Classification Summary

The worst RGB classification

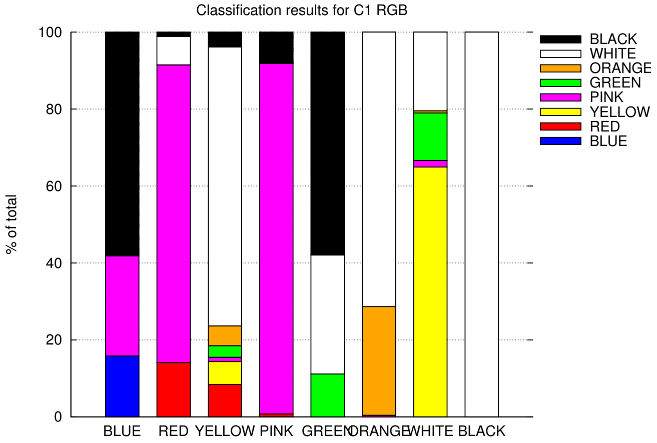


Figure 28: RGB with camera 1.

Classification Summary

The best RGB classification

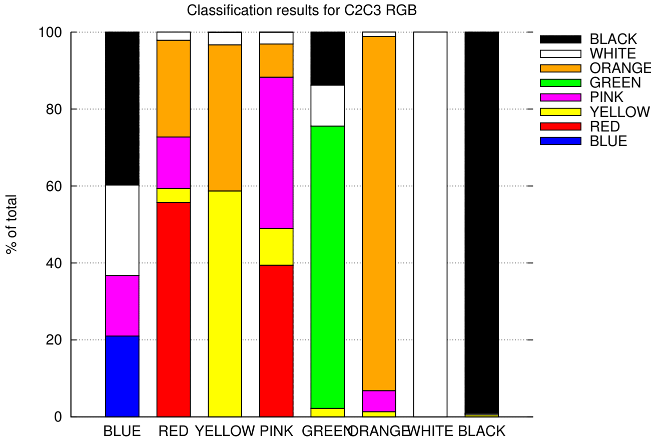


Figure 29: RGB with cameras 2 and 3.

Classification Summary

The worst HSV classification

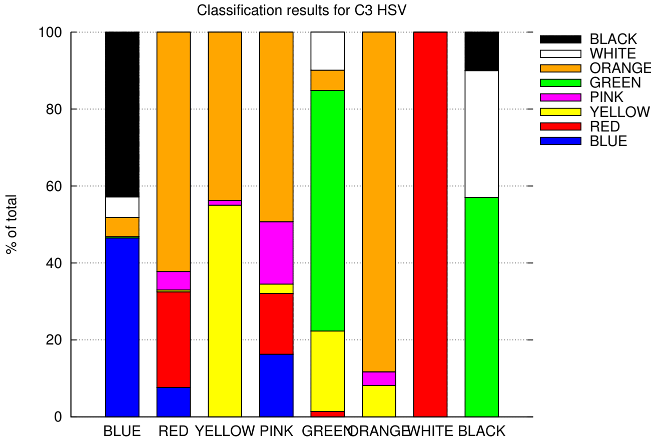


Figure 30: HSV with camera 3.

Classification Summary

The best HSV classification

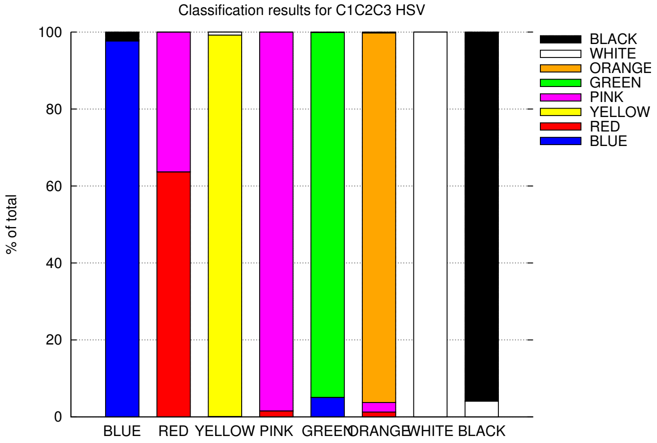


Figure 31: HSV with three cameras.

Overall classification accuracy %

The best HSV classification

Table 1: Overall classification accuracy results (%)

	RGB Hits	RGB False Pos.	HSV Hits	HSV False Pos.
C1C2C3	55	45	**93**	7
C1C2	61	39	86	24
C1C3	33	77	77	23
C2C3	**67**	23	73	27
C1	23	67	74	26
C2	50	50	75	25
C3	55	45	38	62

This HDR approach:

These experiments explored worst-case scenarios, with unknown cameras and a very limited knowledge about the illuminant.

- Improved colour classification using three cameras.
- No LDR to HDR conversion.
- No tone mapping.
- No spectrometric calibration.
- However, classification accuracy limited by lack of samples.

Contents

Exercises 1 and 2

- 1) Write a program that shows three greyscale images representing the three components R, G and B of a colour image. (hint: just create three greyscale images and copy each component to one of them).
- 2) Write a program that converts RGB to HSV and show the three components separately. Remember that the range of the HSV space is different than that for RGB. Adopt your own conversion table (e.g., for H, 0 is converted to 0, and 360 is converted to 255).

Exercises 3 and 4

- 3) Use the colour segmentation example programs to segment images by colour (using the camera). Modify and test the code using the images *peppers.png* and *peppers2.png* to try to separate the green peppers from the red ones.
- 4) Repeat the previous exercise trying to segment by skin colour using two different colour systems, RGB and HSV. Which colour space (RGB or HSV) achieves the best results?

Bibliography
(partial)

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2002.

D. Playne, V. Mehta, N. Reyes, and A. Barczak, “Hybrid fuzzy colour processing and learning,” in *ICONIP07*, (Hibikino, Japan), 2007.

N. H. Reyes and E. P. Dadios, “Dynamic color object recognition,” *Journal of Advanced Computational Intelligence*, vol. 8, no. 1, pp. 29–38, 2004.

T. Ross, *Fuzzy Logic with Engineering Applications*, Singapore: McGraw-Hill, Inc., 1997.

Barczak, A.L.C. and Susnjak, T. and Reyes, N.H. and Johnson, M.J,“Colour Segmentation for Multiple Low Dynamic Range Images using Boosted Cascaded Classifiers” *IVCNZ 2013 (Image and Vision Computing New Zealand)*, 2013.