VIP: Color Image Analysis

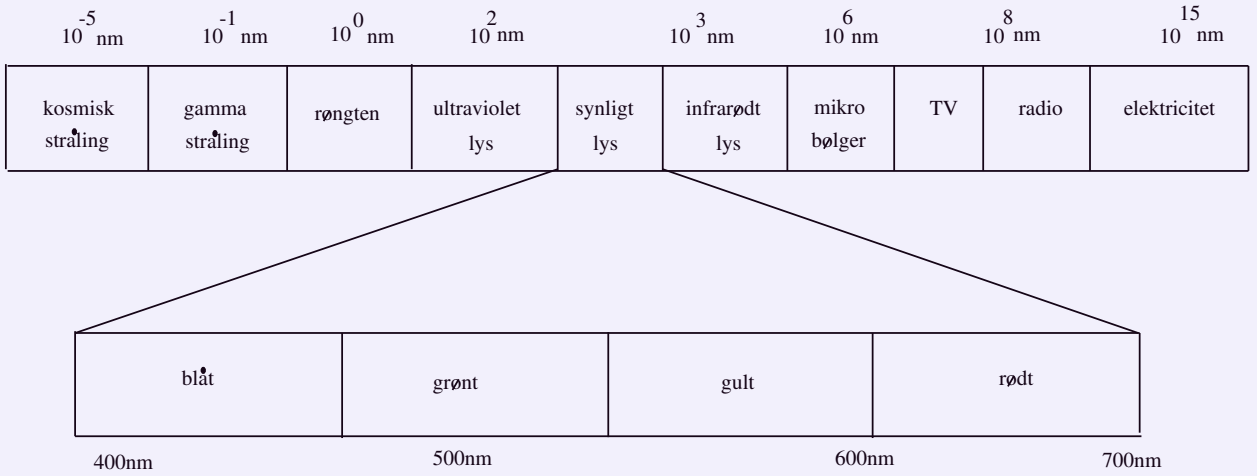
Søren Olsen

Department of Computer Science
University of Copenhagen

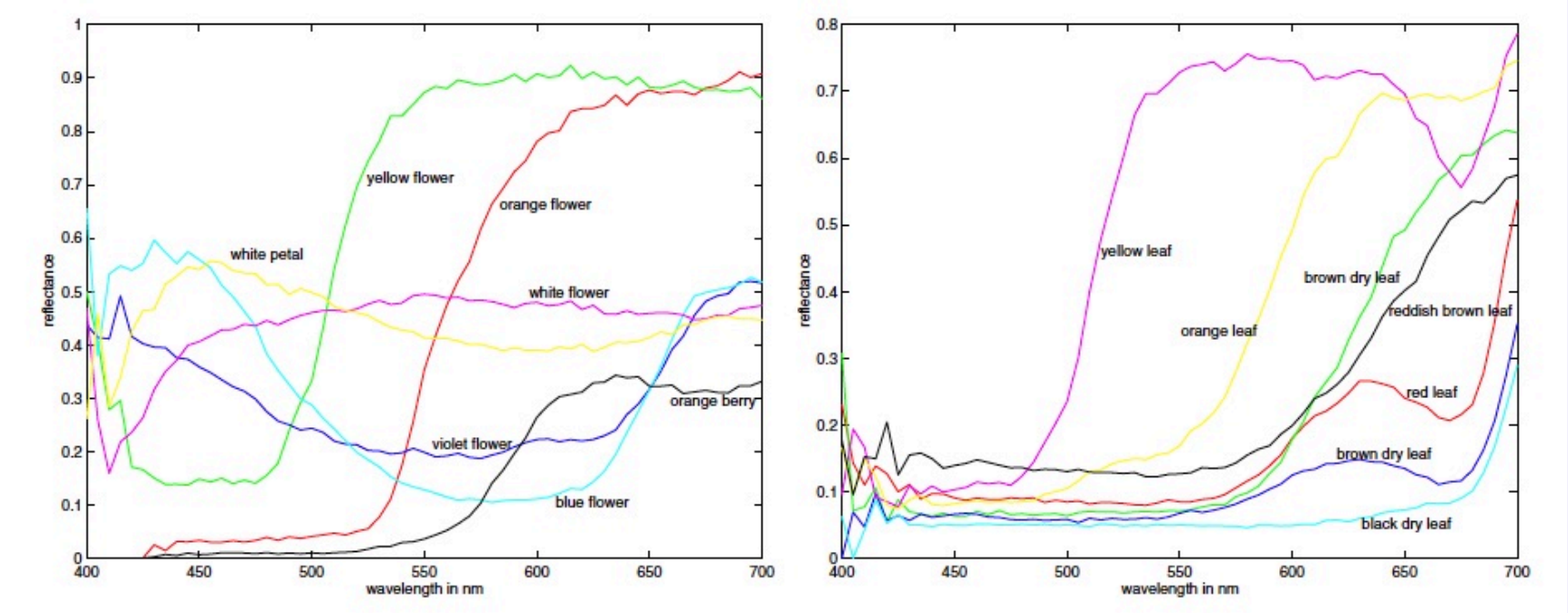
VIP: Color Image Analysis

Colors don’t exist!

Humans sense electro magnetic radiance with wavelength λ between 400 nm and 700 nm as colored light.

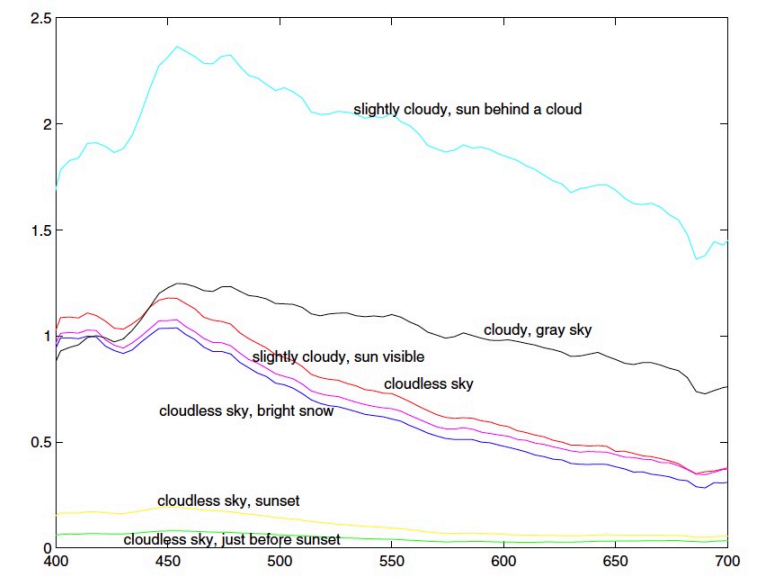


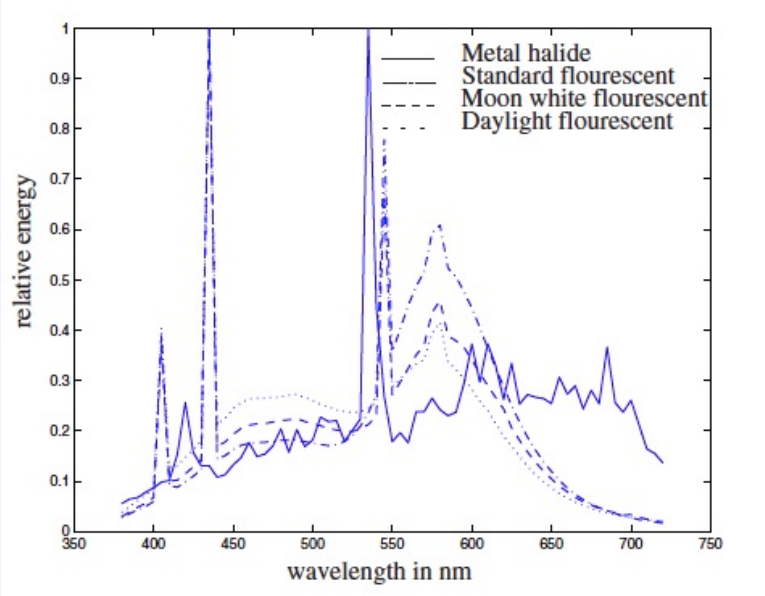
Colors of surfaces



From Forsyth and Ponce.

Illuminant spectral power distribution





From Forsyth and Ponce.

The temperature of light

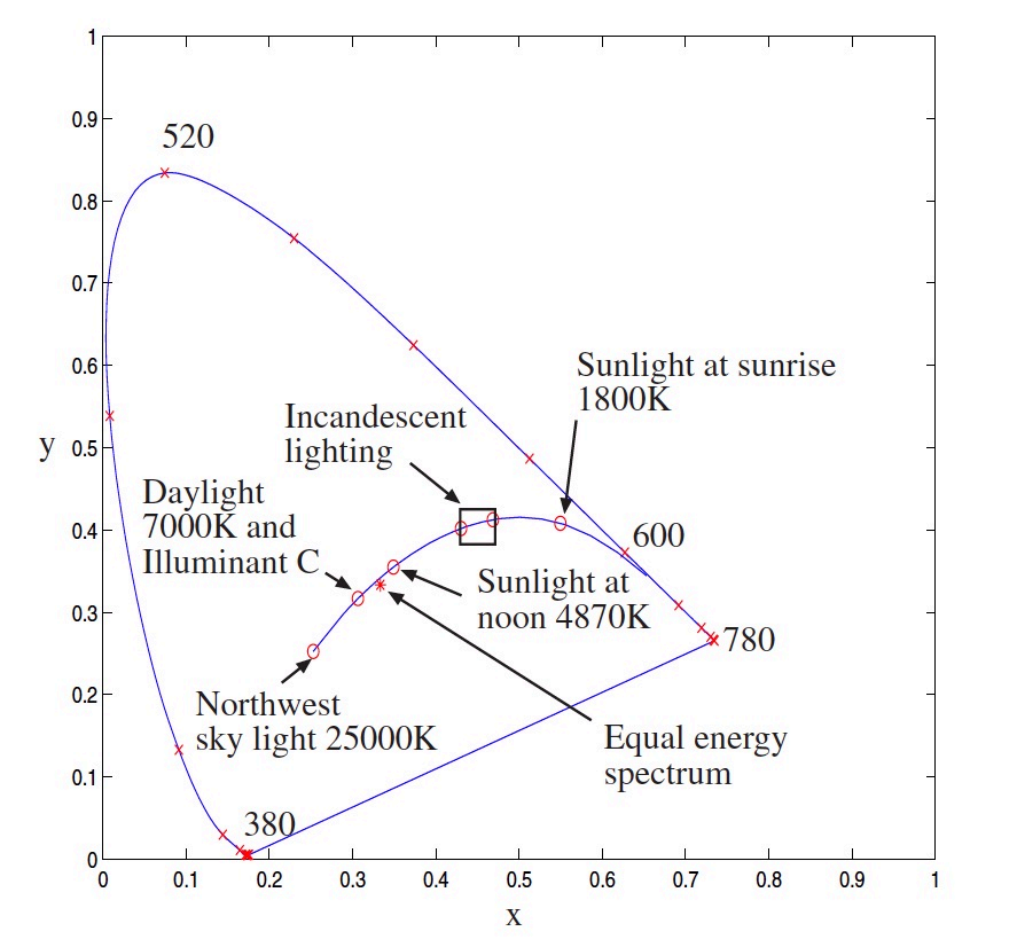
Electronic radiation often is modelled as emitted from a black body that does not reflect any other light. The emitted energy distribution depends in a simple way of only the temperature T of the body:

$$E(\lambda) = C \frac{1}{\lambda^5} \frac{1}{e^{hc/k\lambda T} - 1}$$

where

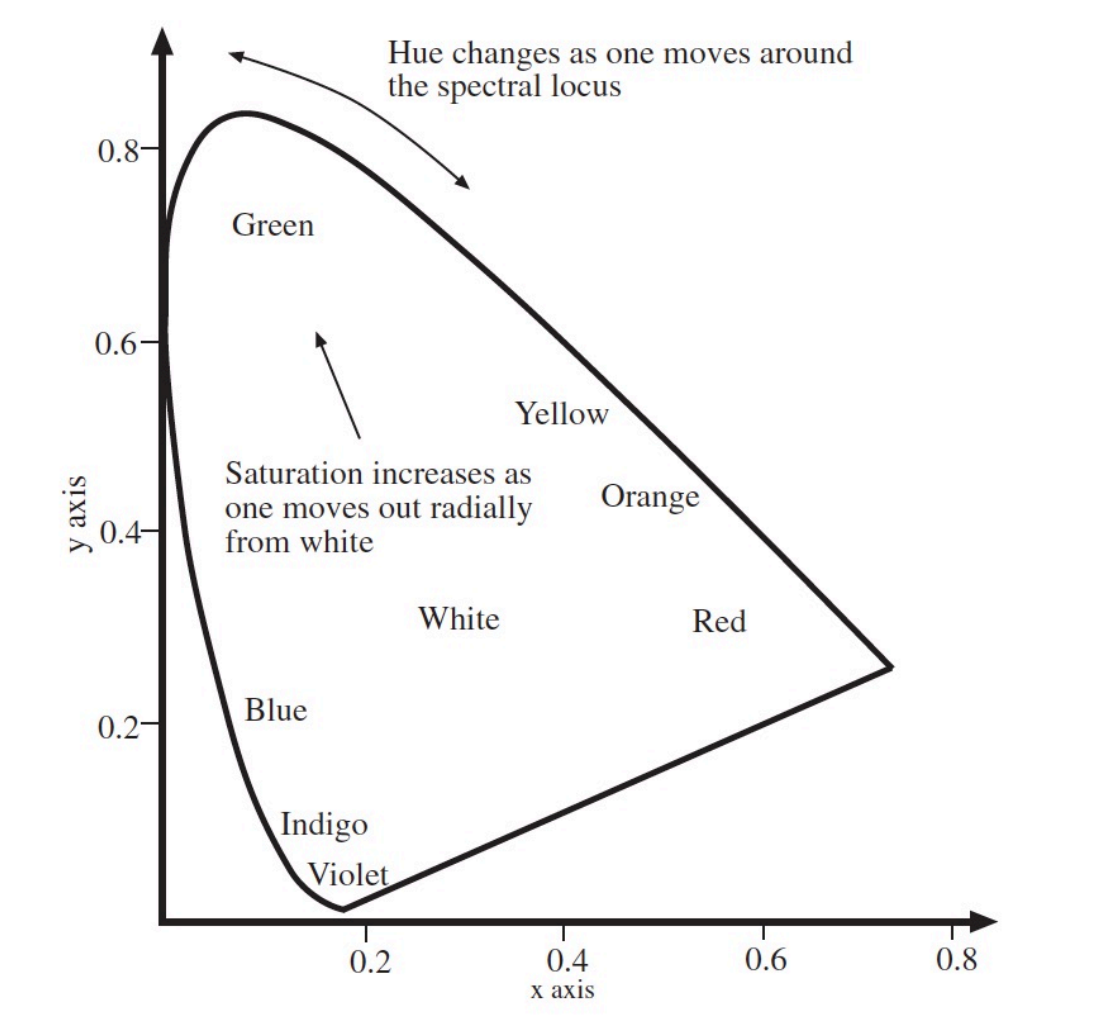
- T is the temperature in Kelvin
- c is the speed of light
- h is the Planck's constant
- k is the Boltzmann's constant
- C is a constant.

We will later today see how the relation between energy distribution and temperature may help in detecting and removing shadows in images.



From Forsyth and Ponce and using the CIE xy color space.

Color distribution



From Forsyth and Ponce.

Trichromacity

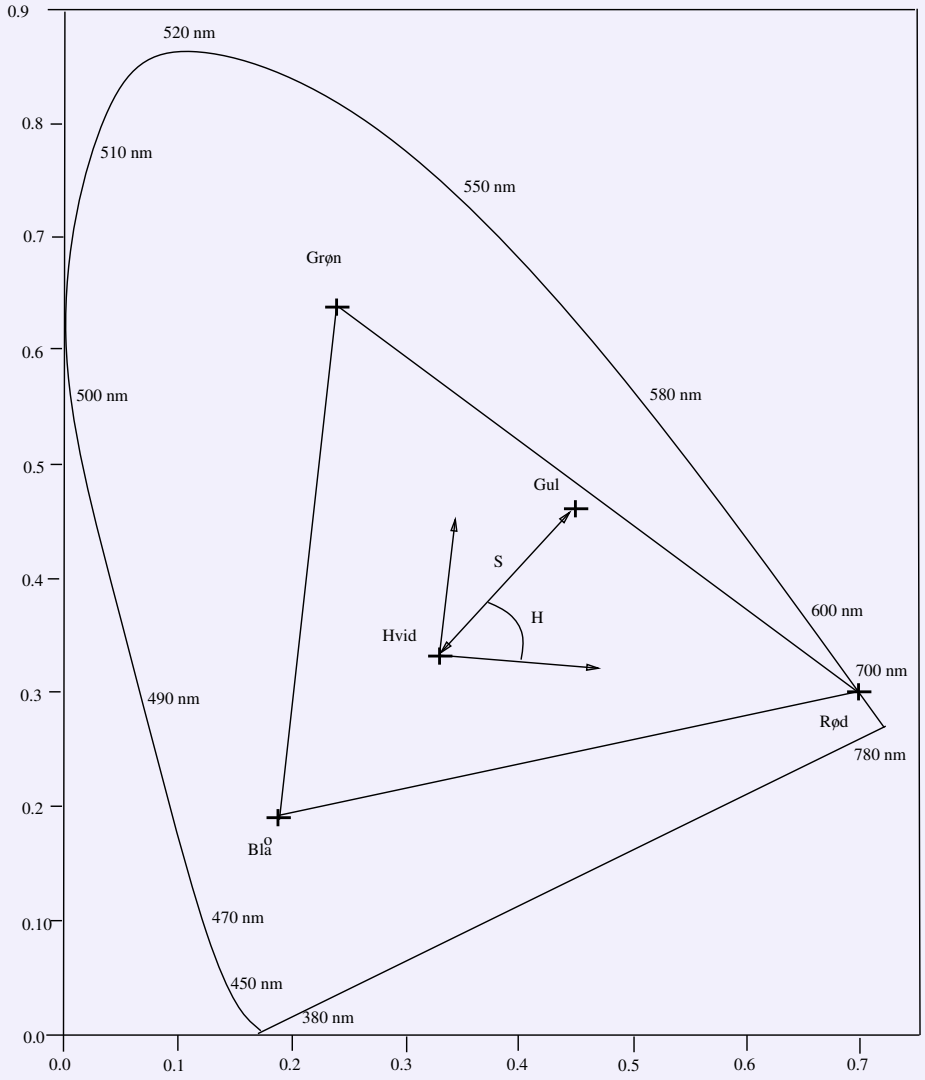
People generally agree that any (most) visible colors may be composed by a linear combination of three different reference/primary colors (read: R,G, and B). This is the principle of trichomacity.

However, no electronic wave of any wavelength λ may be constructed by linear combination of waves with other wavelengths.

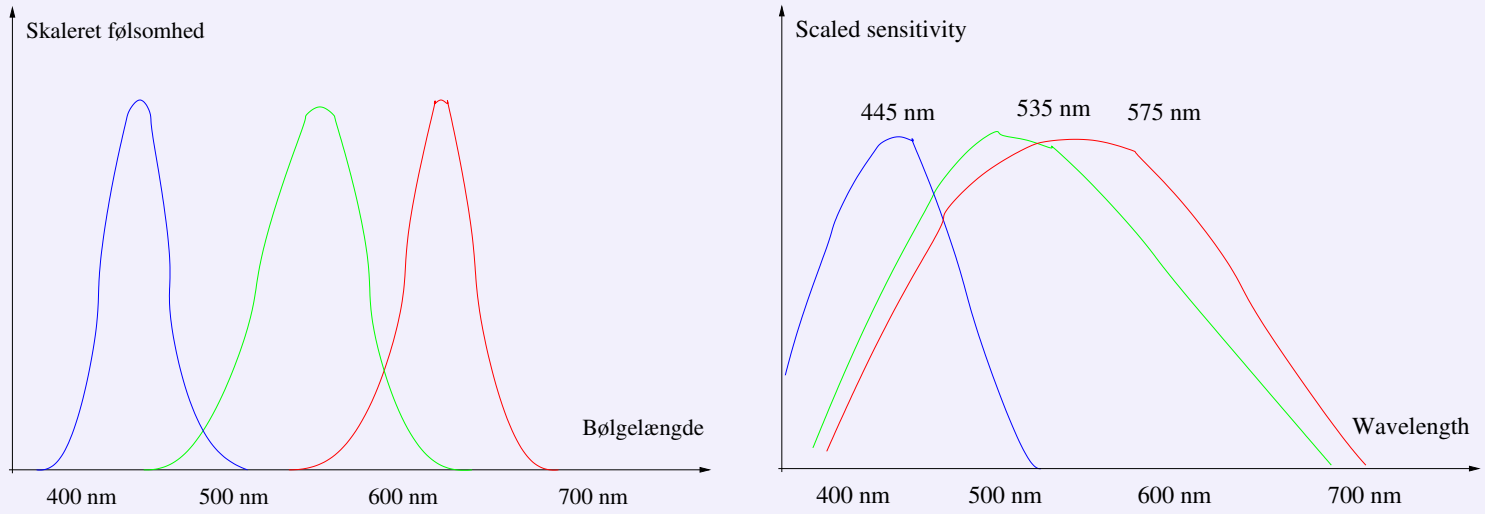
Thus, there are different wavelengths that people associate with the same color. Also, there are colors that cannot be represented uniquely in a linear vector space.

Visible light has a wavelength between 400 and 700 nm. The set of colors spanned by linear combination of *r*, *g*, og *b* is only a subset.

The triangle illustrates the gamut for a monitor. Colors outside the triangle cannot be show.



By using 3 sensors, sensitive to the red, the green and the blue part of the spectrum, it is possible to represent a subset of the range of possible colors by one RGB-value.

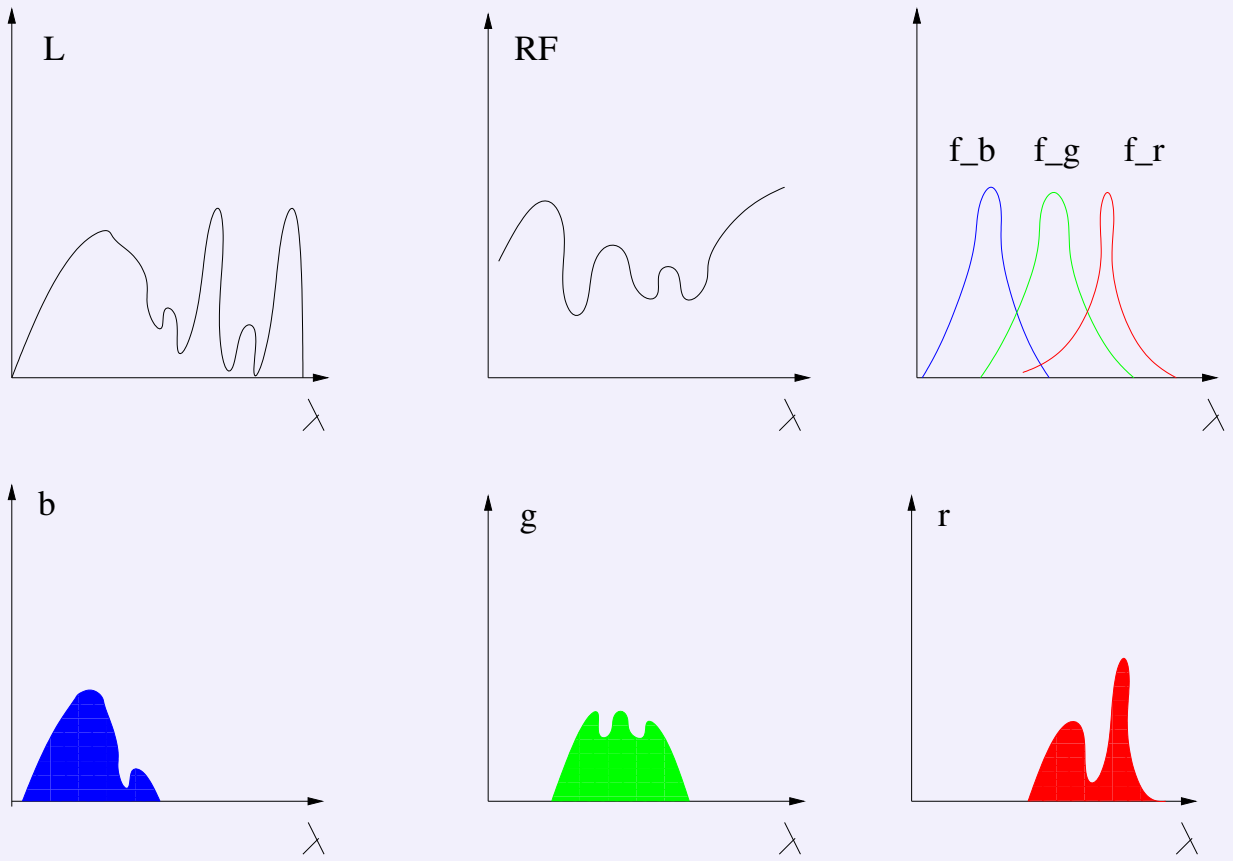


Within the human retina 3 types of cone-shaped cells exist. These contain pigment that makes them sensitive to different wavelengths.

- The spectral filters used by a specific camera, a color scanner etc. all are different.
- The registered value depend on the spectral distribution of the light source $L(\lambda)$, the reflection $RF(\lambda)$ from the illuminated object in view, and on the specific sensor sensitivity curves $f_{r/g/b}(\lambda)$.

$$
\begin{aligned}
R &= k_r \sum_{\lambda} L(\lambda) RF(\lambda) f_r(\lambda) \\
G &= k_g \sum_{\lambda} L(\lambda) RF(\lambda) f_g(\lambda) \\
B &= k_b \sum_{\lambda} L(\lambda) RF(\lambda) f_b(\lambda)
\end{aligned}
$$

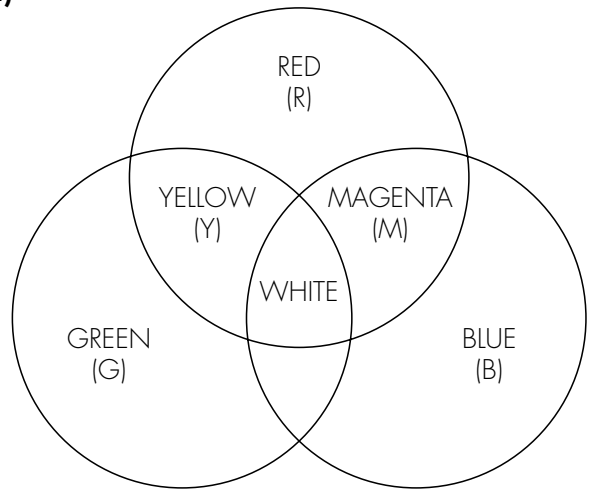
Example

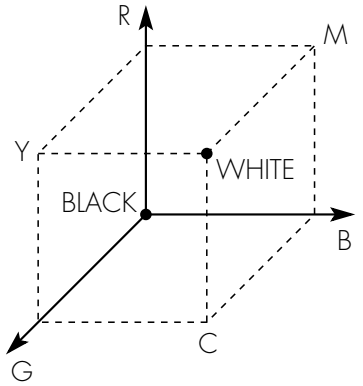


Metameterism

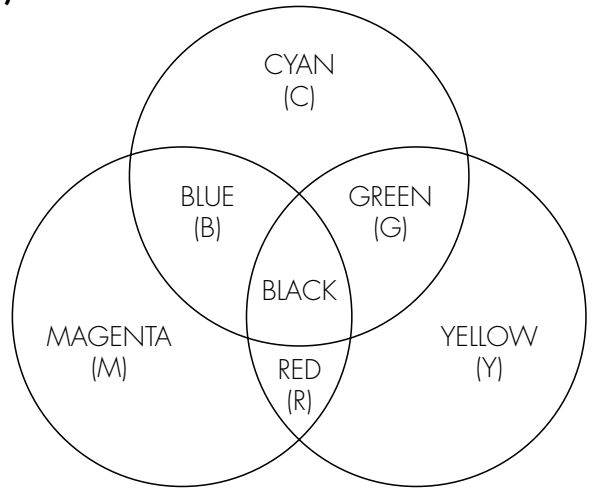
- Two spectral distributions may be registered as having the same RGB-value. There is many more different spectral distributions than RGB values.
- The set of spectral distributions having the same RGB-value is call a Metameric class.
- When a color is reproduced a representative for the metameric class must be selected. Different monitors and printers selects different representatives.

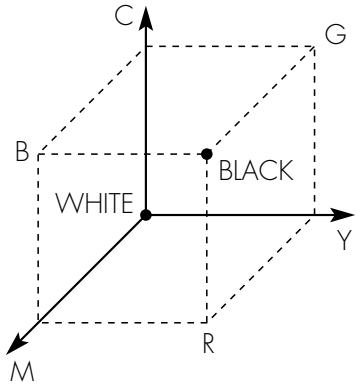
(a)


RED
(R)
YELLOW
(Y)
MAGENTA
(M)
WHITE
GREEN
(G)
BLUE
(B)


R
M
Y
WHITE
BLACK
B
C
G

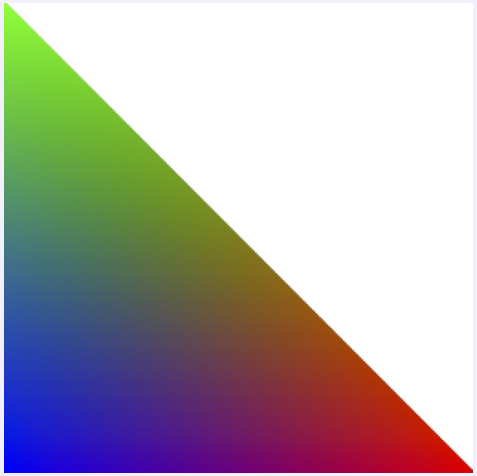
(b)


CYAN
(C)
BLUE
(B)
GREEN
(G)
BLACK
MAGENTA
(M)
RED
(R)
YELLOW
(Y)


C
G
B
BLACK
WHITE
Y
R
M

Farvesystemer

- R, G, og B are called primary colors.
- $(r, g, b) = (R/I, G/I, B/I)$ are called the trichromatic coefficients. $I = R + G + B$ is called the *Intensity*. Notice: $b = 1 - r - g$.
- A lot of other color coordinate systems exist: CMY = (1-r, 1-g, 1-b) are (together with black K) used in printers.
- For broadcast and video YIQ (in America) and YUV (in Europe) are used.
- In JPEG image coding YC_bC_r is dominating.
- Within graphic production the CIE Lab system is used.
- For Computer Vision the HSV-system often is appropriate.



Plot of $(r,\ g,\ 1 - r - g)$:

YIQ, YUV and YCbCr

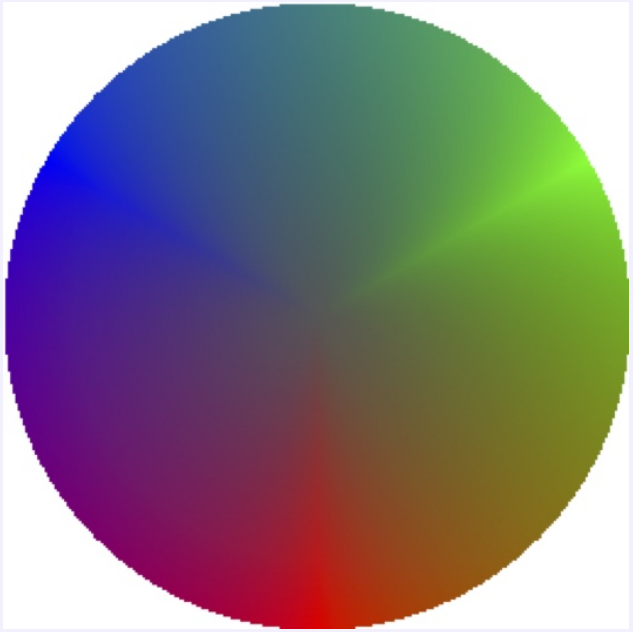
In all 3 transforms the luminance Y is defined by $Y = 0.299R + 0.587G + 0.114B$. All 3 transforms are linear:

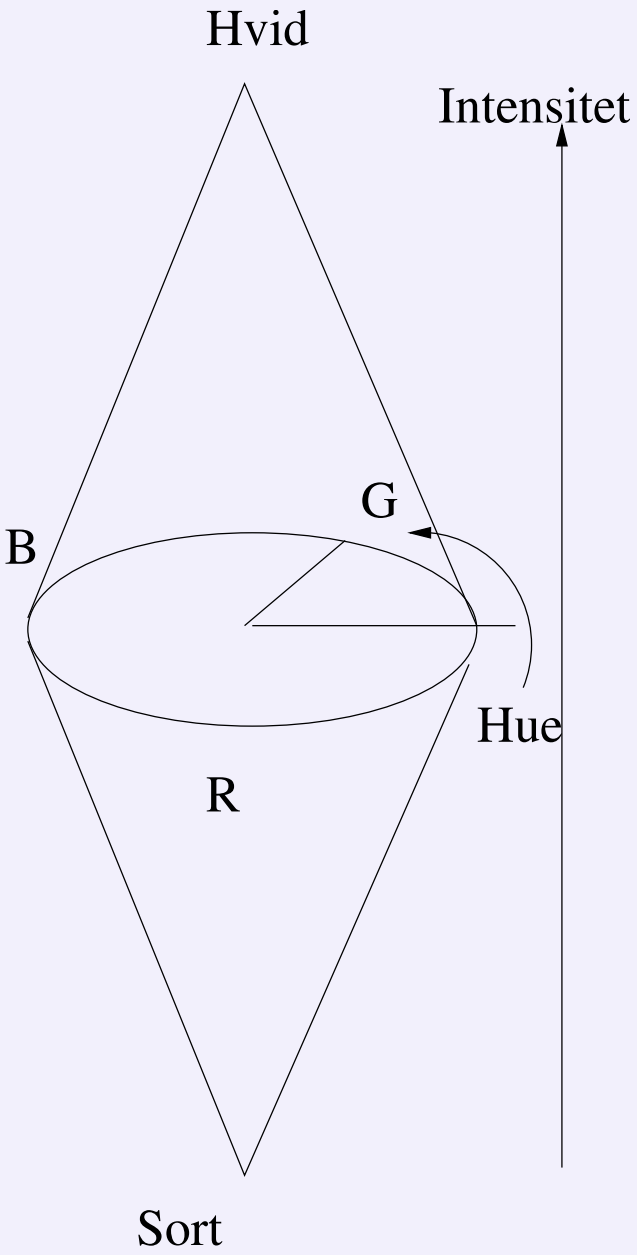
$$I = 0.74(R - Y) - 0.27(B - Y)$$
$$Q = 0.48(R - Y) + 0.41(B - Y)$$

$$U = 0.493(B - Y)$$
$$V = 0.877(R - Y)$$

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

HSV





RGB2HSV

Lad

$$\theta = \tan^{-1}\left\{\frac{(R-G)+(R-B)}{\sqrt{3}(G-B)}\right\}$$

Then Hue H is defined by:

$$H = \begin{cases} \theta & \text{if B} \leq \text{G} \\ 360 - \theta & \text{if B} > \text{G} \end{cases}$$

The Intensity is: $I = \frac{1}{3}(R+G+B)$ and the saturation S is defined by:

$$S = 1 - \frac{3}{R+G+B}\left[\min\{R,G,B\}\right] = 1 - \frac{\min\{R,G,B\}}{I}$$

Please notice that H is discontinuous at 0

Why HSV

Often we want to detect objects of specific colors in situations with varying illumination and viewing direction. When these factor change the RGB-values may change significantly. Often the hue is less sensitive.

In most other transform the color is represented in 2D. In computer Vision we are rarely interested in the saturation, so here the (used) color information essentially is 1D. This makes computation, e.g. thresholding, easier.

Example: Detection of colored object 1

If we want to detect a green object, first notice that the color geen has a hue value about 0.33 (on a scale from 0 to 1). Thus:

```
green = (abs(h - 0.33) < T)
```

where `T` is a user specified parameter, e.g.0.15, defining the width of acceptable (green) colors.

In some situations, depending on the white balancing of the camera, cloudy white may actually be greenish. To avoid this we may:

```
mygreen = (abs(h - 0.33) < T1) & (v < T2) & (s > T3)
```

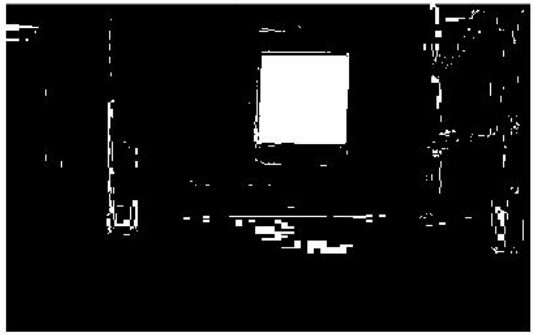
In this case `mygreen` will be a saturated green color.

Original

R thresholded

Windowed hue

Windowed and thresholded Hue



Simple thresholding of R does not work. Use of Hue is much better.

Example: White balancing

If something white is imaged as having another color, then the white balancing is wrong and may be corrected by scaling each of the color channels appropriately.

First find a robust estimate of the colors having largest and smallest intensities. Then compute a unit vector (a, b, c) pointing in this direction and scale the RGB-channels with $(\frac{1}{a}, \frac{1}{b}, \frac{1}{c})$ Finally, normalize to make alle values fit into $[0 : 1]$.

Example: Undoing automatic white balancing

Auto white balancing may create greyish images if the scene has no objects of e.g. the color blue. The camera incorrectly will crank up the sensitivity of the blue channel making the image greyish.

To undo, we do like in white balancing, but instead of adjusting to a the maximum vector $(1, 1, 1)$ we use the vector $(1, 1.0, 0.7)$ reflecting that the images show yellow and green colors and very little blue.

Example: Uneven illumination

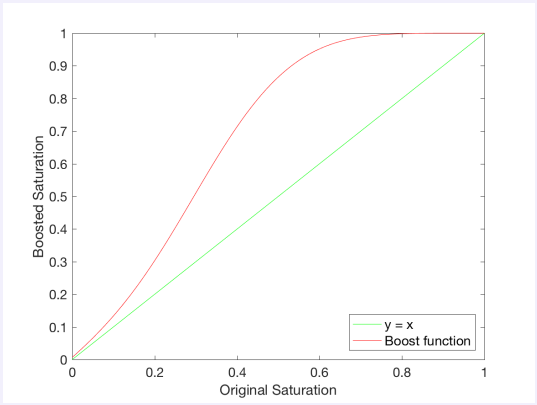
At low sun hight angle the illuminated straw is located oposite to the sun azimutal angle. Towards the sun, the unlit cereals is is sight. In a HSV-representation the effect is mostly visible in S and V.

Both components are compensated by fitting a plane to the local averages of saturation and value.

$$S(x,y) \longrightarrow \frac{\bar{S}}{a_S + b_S x + c_S y} S(x,y) \qquad \text{and} \qquad V(x,y) \longrightarrow \frac{\bar{V}}{a_V + b_V x + c_V y} V(x,y)$$

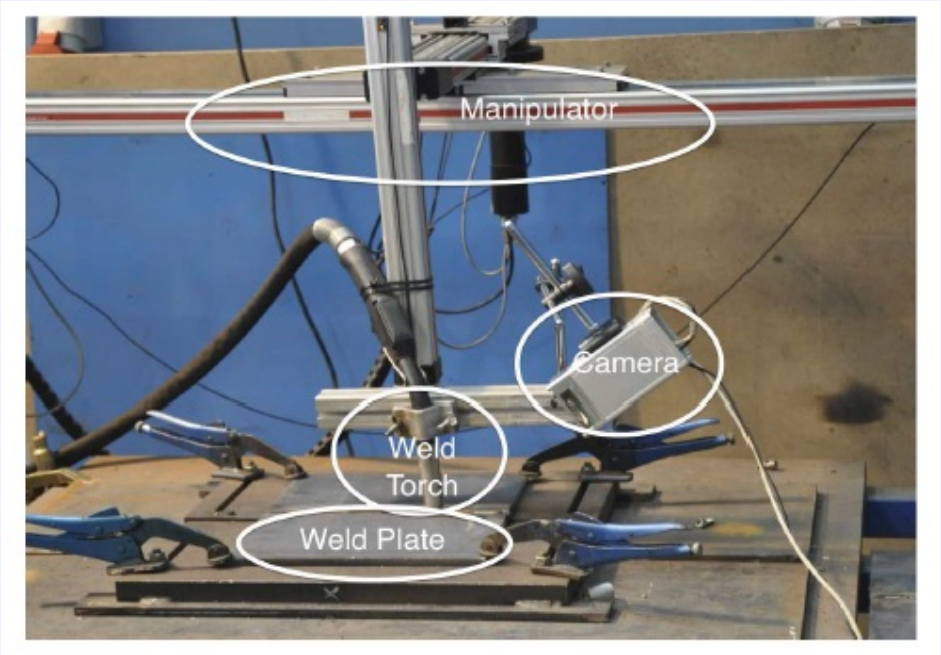
Example: Color boosting

When object detection is based on color information, Hue information alone may be used. If not, saturated colors are usually preferred. To increase saturation this may be nonlinearly increased:



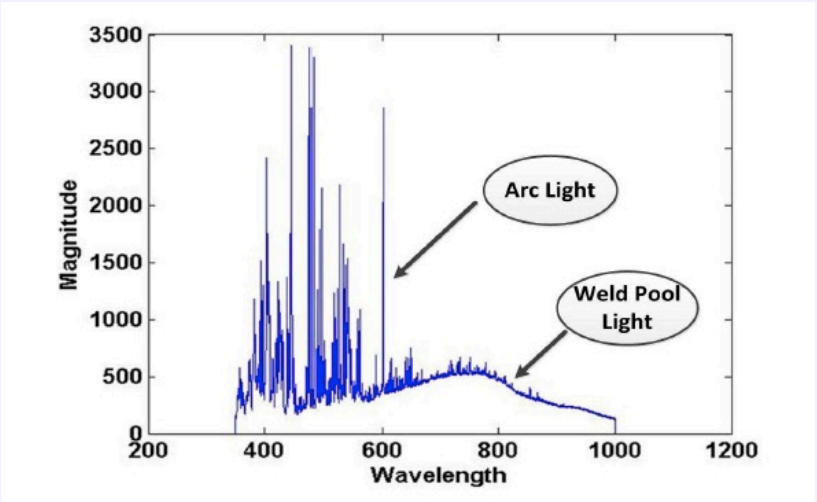
Example: Viewing the welding pool

The light from arch welding is extremely strong and strong shielding is needed in order to view the welding pool.



Work and images produces by former phD-studens Jinchao Liu

In automated welding, a model of the welding pool is essential. One approach is to select a tiny interval of the light spectrum using a special glass that blocks all other light.



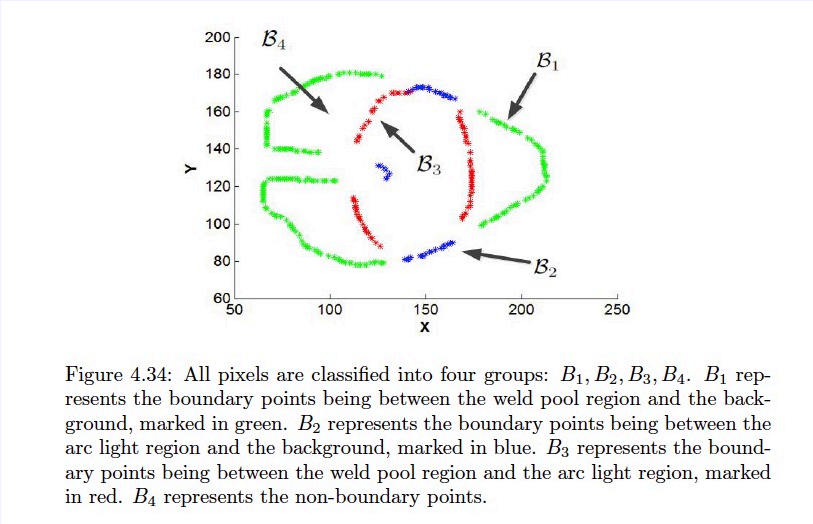
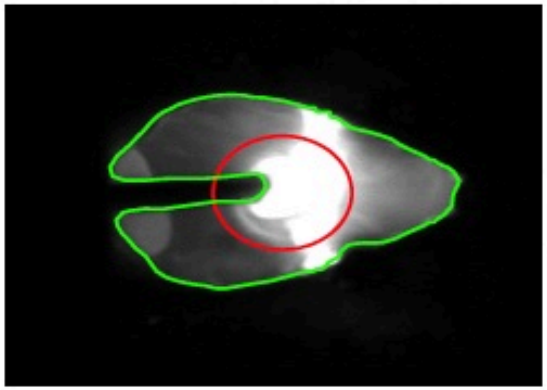
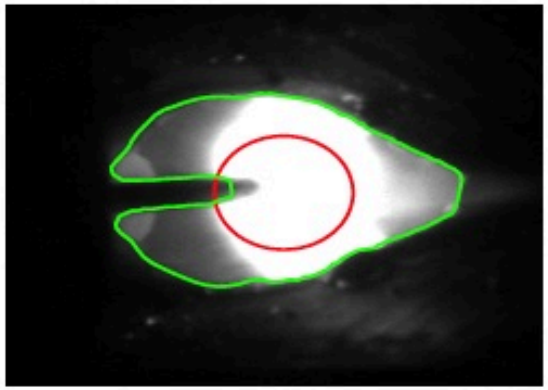


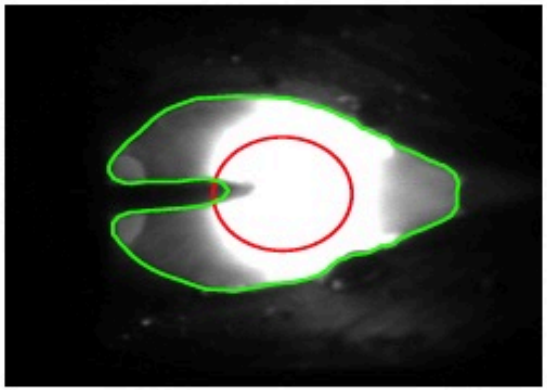
Figure 4.34: All pixels are classified into four groups: B_1, B_2, B_3, B_4. B_1 represents the boundary points being between the weld pool region and the background, marked in green. B_2 represents the boundary points being between the arc light region and the background, marked in blue. B_3 represents the boundary points being between the weld pool region and the arc light region, marked in red. B_4 represents the non-boundary points.



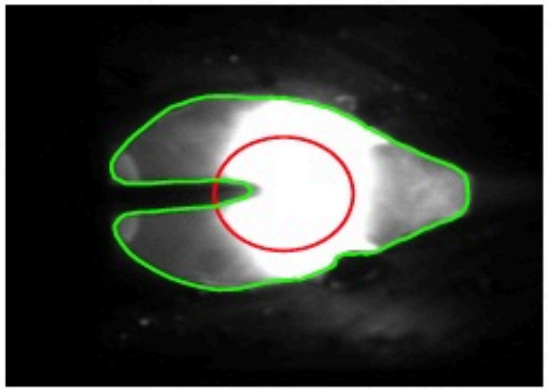
(a) Frame66



(b) Frame67



(c) Frame68



(d) Frame69

Example: Shadow edge removal

Both for viewing and for object recognition it would be beneficial to remove image edges related to shadows and leave only those related to the objects in the scene.

Method and images from *Finlayson, Drew, lu: Entropy Minimization for Shadow removal*, 2007.

Procedure for removal

1. Compute image gradients and detect edges in the image.
2. Classify the edges as shadow edges or not.
3. Remove the shadow edges by nulling the gradient information in a band along the shadow edges.
4. Reintegrate the gradient information and add mean color to get an images without shadows.

The first and third steps are easy. The last step requires advanced numerical integration routines, but is doable (we will not deal with this step). The challenging step is the classification.

The basic idea is to transform the image into another (grey) one where the intensity dependency of the illumination source is canceled. Thus, shadow edges will not exist.

Simplification

Remember relation between temperature of black body and emitted energy:

$$E(\lambda) = \frac{1}{\lambda^5} \frac{1}{e^{hc/k\lambda T} - 1}$$

You may look op the values for c, k and h and notice typical values of $\lambda \approx 500nm$ and $T \approx 10^3 - 10^4 K$. Then it is easy to see that $hc \gg k\lambda T$ and that:

$$E(\lambda) \approx C \frac{e^{-hc/k\lambda T}}{\lambda^5}$$

where C is some constant.

Remember that received energy r_k of color channel k is the product of the illuminant energy, the surface albedo, the reflection function and the sensor sensitivity profile:

$$r_k = c_k \int_\lambda E(\lambda)\rho(\lambda)RF(\lambda)f_k(\lambda)d\lambda$$

Assumptions

We have here assumed that the illuminant may be modeled as a black body. We will further assume that the scene surface is diffuse (no specularities), that the albedo $\rho(\lambda)$ is constant as a function of spatial coordinate and that our receptor sensitivity profiles f_k respond only to a single wavelength:

$$f_k(\lambda) \;=\; \delta(\lambda - \lambda_k) \;=\; \begin{cases} 1 & \text{if } \lambda = \lambda_k \\ 0 & \text{otherwise} \end{cases}$$

We will also assume that the reflection function RF is constant. Thus:

$$r_k \;=\; c_k \int_\lambda E(\lambda)\rho(\lambda)RF(\lambda)f_k(\lambda)d\lambda \;=\; K\rho(\lambda_k)RF(\lambda_k)\frac{e^{-hc/k\lambda_k T}}{\lambda_k^5}$$

where K is a new constant and where $k \in \{R, G, B\}$.

Now let $c_1 = \log(r_R/r_B)$ and $c_2 = \log(r_G/r_B)$.

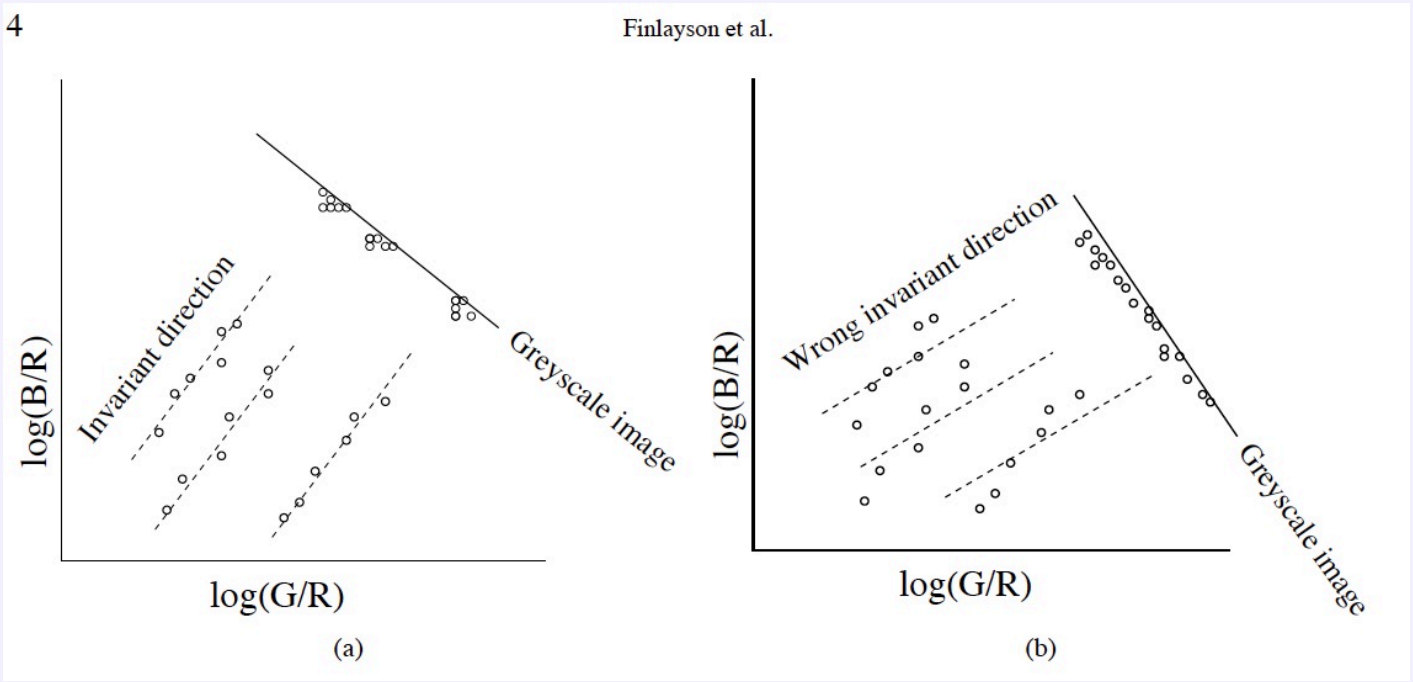
$$\begin{aligned} c_1 &= \log\left(\frac{K\rho(\lambda_R)RF(\lambda_R)\frac{e^{-hc/k\lambda_R T}}{\lambda_R^5}}{K\rho(\lambda_B)RF(\lambda_B)\frac{e^{-hc/k\lambda_B T}}{\lambda_B^5}}\right) \\ &= a_1 + \frac{1}{T}b_1 \end{aligned}$$

where $b_1 = \frac{hc}{k}(\frac{1}{\lambda_B} - \frac{1}{\lambda_R})$

is a constant only depending on the wavelengths and where a_1 is a constant depending on λ_R and λ_B and on the ρ- and RF values of these.

Similarly $c_2 = a_2 + \frac{1}{T}b_2$. Thus, when the light source radiation, i.e. T is changing, then (c_1, c_2) is moving on a straight line. Notice that b_1 and b_2 do not depend on the scene, but only the sensors. Thus, **all such lines are parallel**. The orientation is called **color temperature direction**.

The color temperature direction



When projecting (c_1, c_2) on a line perpendicular to the color temperature direction we will get a grey image called The invariant image. This image do not contain any illumination effects such as shadows.

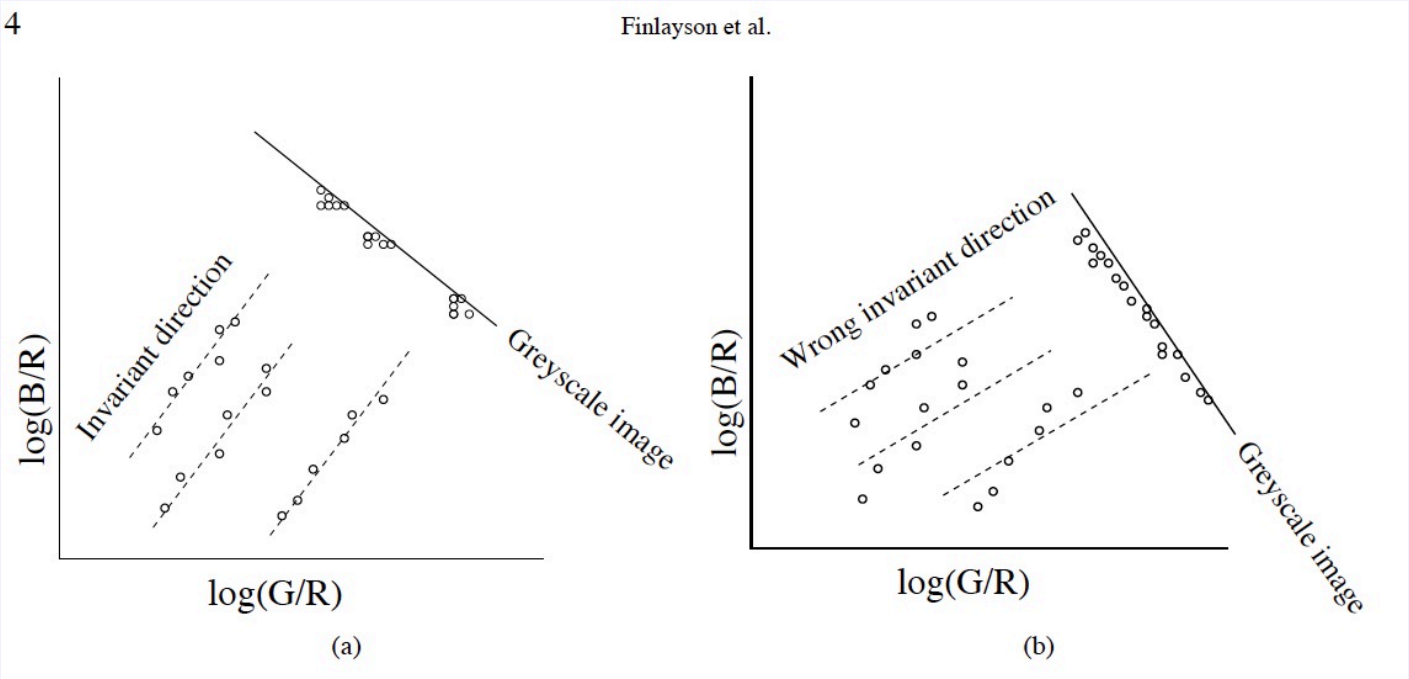
The Invariant Image

We may compute the images $c_1 = \log(R/B)$ and $c_2 = \log(G/B)$. If we somehow know the color temperature direction we may project the image on a unit vector orthogonal to this.

The edges visible in the original image but not in the invariant images may be classified as shadow edges.

How to find the projection direction

If many shadow edges are visible, then a lot of pixels will be arranged along the color temperature lines in the (c_1, c_2)-space. Thus, when projected on the orthogonal direction they will create a peak. If projected on a line with any other direction they will spread more out.



Entropy based estimation

The basic idea (only having access to a single image) is to project on all possible angles ([0:180] degrees) e.g. using a quantisation of 1 degree. For each projection, i.e. a grey-valued image, we compute how peaked the projection is, and select the one maximising this.

Entropy, is a statistical measure of the spread of a probability density distribution. Entropy is used intensively in Computer Vision and Image Analysis to characterise distributions

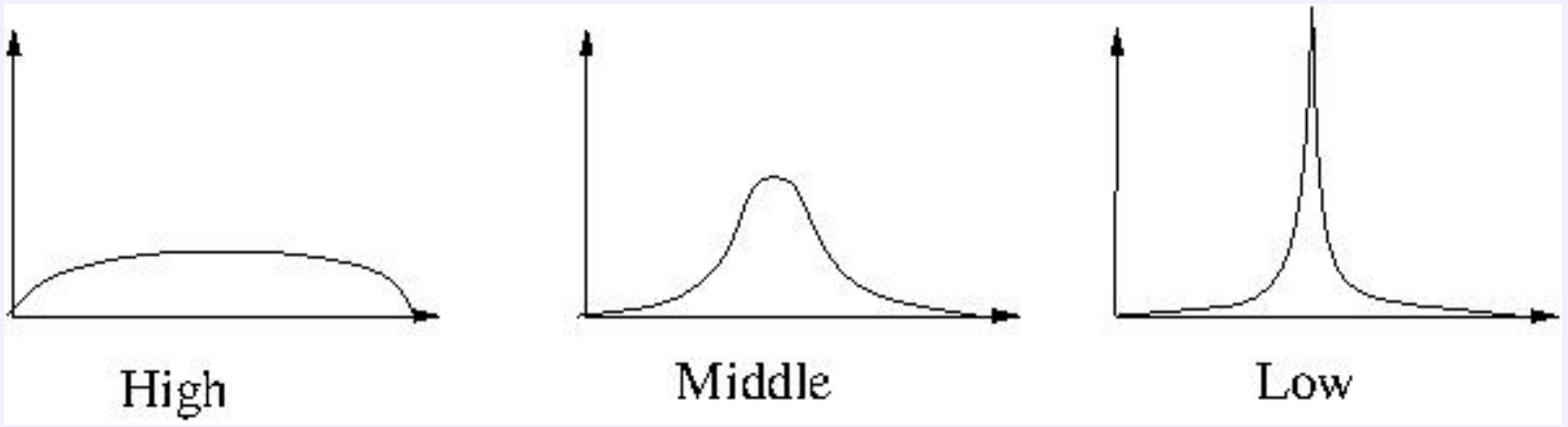
$$H = -\sum_i p_i \log p_i$$

In our case p_i is the empirical probability that a pixel takes the value i.

In practice Finlayson et.al. do not use entropy directly, but a more robust measure called *Information potential*, but this is of minor importance.

Entropy

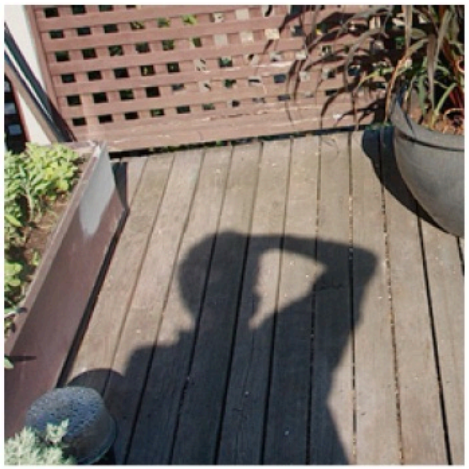
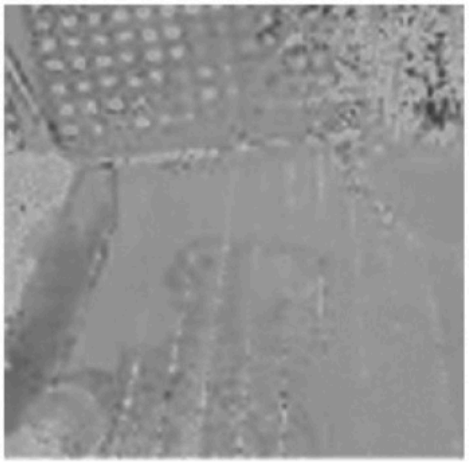
Please notice that if p is uniformly distributed on say 64 bins, then $H = -64\frac{1}{64}\log_2(\frac{1}{64}) = \log_2(64) = 6$. If p is ultimately peaked (i.e. a delta-function) $H = 0$. In all other cases H is in between the two values.



Please notice that the projection $(c_1, c_2) \cdot (\cos v, \sin v)^\top$ gives an image of real values and need to be quantized when computing a histogram of the values. Finlayson use 64 bins. Also note that the histogram needs to be normalized to have unit sum before entropy computation.

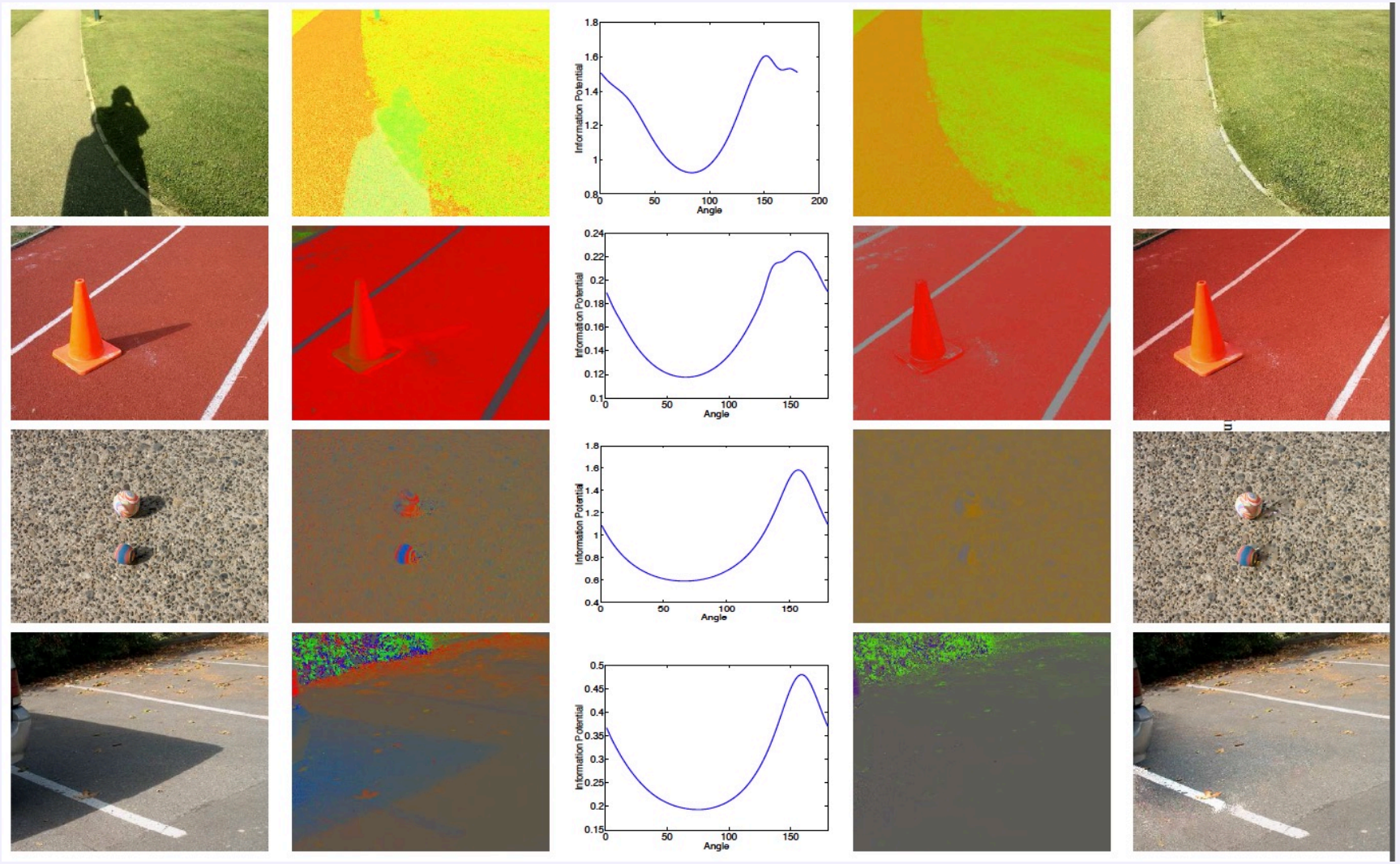
Shadow edges

Edges visible in the original image, but not in the invariant images are deleted by setting the partial derivatives to zero in all such edge pixels and in the neighboring pixels up to a distance of 2 pixels.

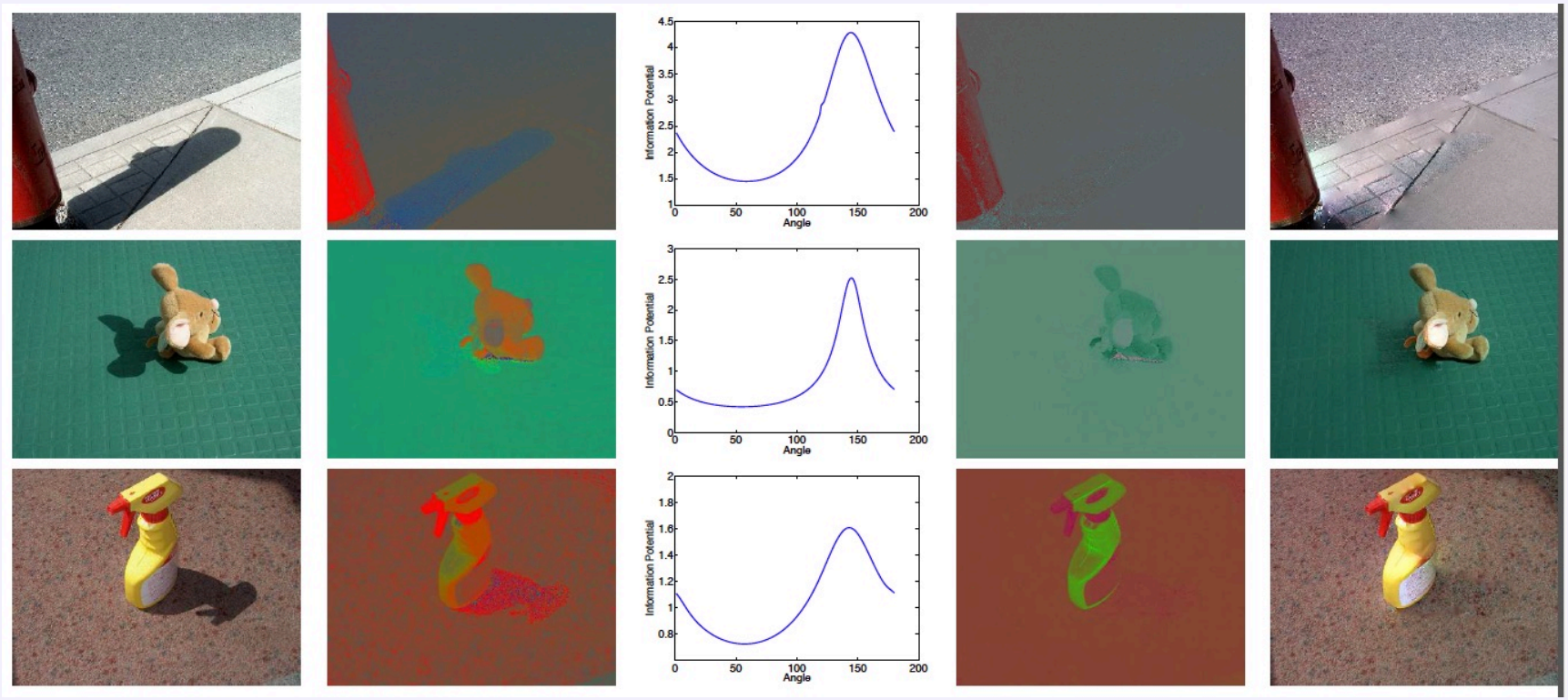




Before zeroing however, all detected shadow edges need to be closed in order for the following reintegration procedure to work. The image border may participate in this closing.

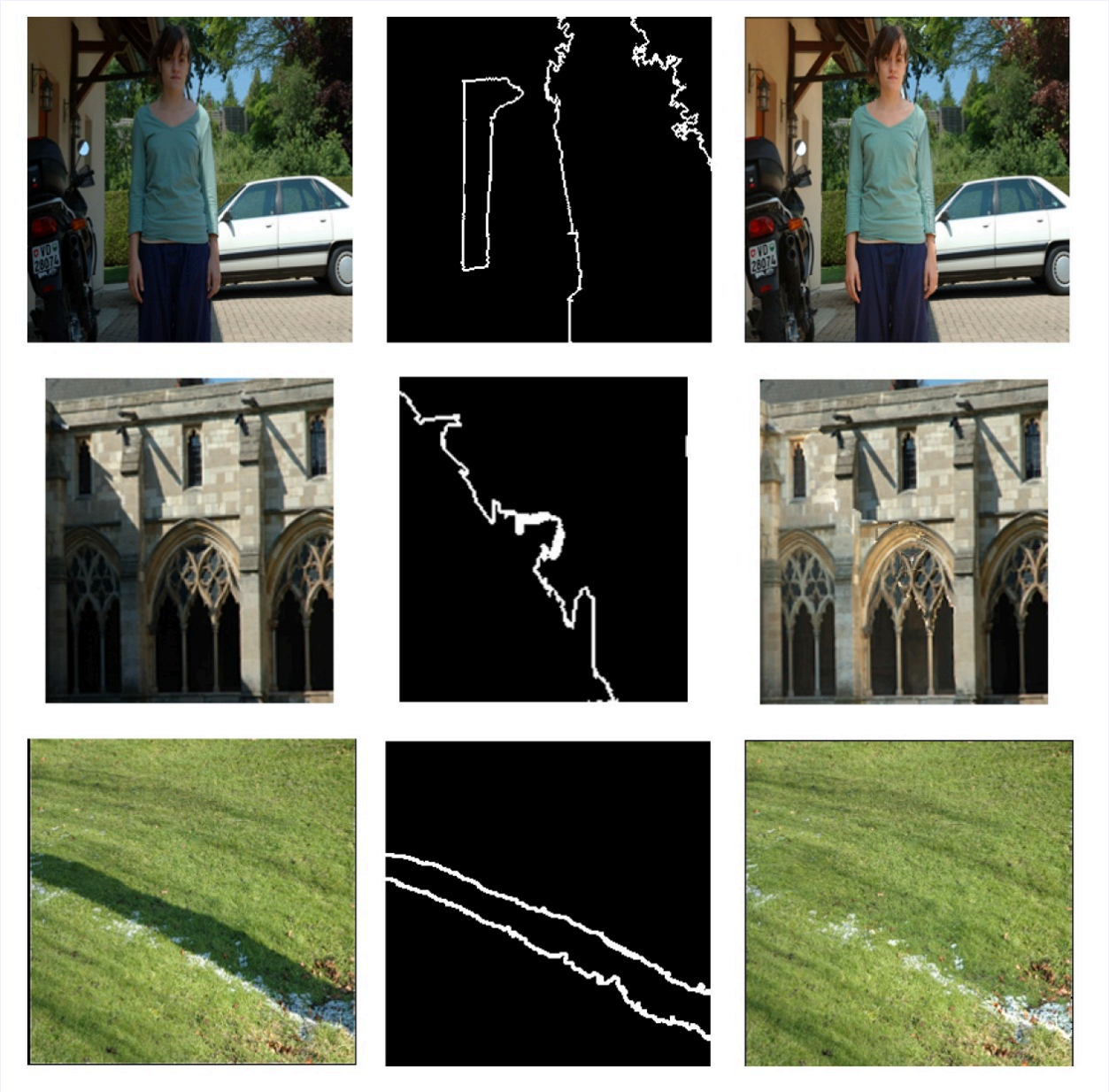
Examples from Finlayson et.al



More examples from Finlayson et.al.



Examples from Fredembach et.al.



Questions ?

Next time

- A bit on Computer Vision history
- Neural Net basics
- Modern Convolutional Neural Nets (CNN's)
- Image classification, Object detction, and image segmentation