Admistrivia

Still waiting for our mail list!

First assignment is ready.

We will have a group office hour on Monday as planned (11am, GS 919).

More discussion on ambiguity(*)

Example 1: Color you see on the screen in the water image is not that of the screen!

Example 2: This is related to color constancy.

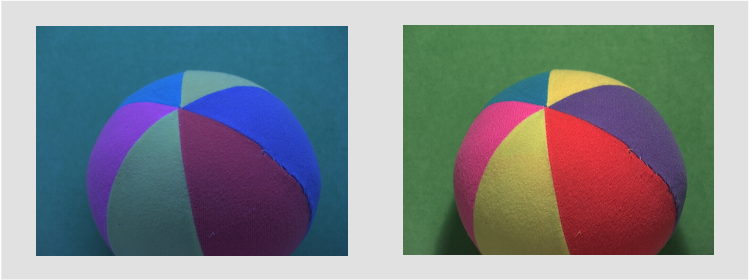
Main point: To solve the unsolvable, you have to risk being wrong while making intelligent guesses.

(Guessing wrong explains many illusions).

Example 3: The strong assumption of rectangular balconies lead to the wrong guess (they are parallelograms).

(*) Notes added after the class so that the images that follow make sense to those that were not there and/or who have forgotten the discussion.

The Computational Color Constancy Problem



(Same scene, but different illuminant)

Matlab Tricks

Matlab is good for quick experiments (use it!).

Good way to check C code.

Good way to check that the math does what you expect. Exploit “rand”.

Example

Recall that an eigenvector of a matrix Y satisfies.

$$Y\mathbf{v} = \lambda\mathbf{v}$$

Under “good” conditions, A has enough independent eigenvectors to form a basis.

We put those into the columns of a new matrix, A to get the eigenvalue decomposition.

We use unit length. (The scaling of v does not matter in the above equation).

We order the columns so that the eigenvectors corresponding to ascending/ descending eigenvalues.

Wisdom from tea dipper handle

Example

Recall that the transpose of a matrix, A, denoted in Matlab by A', is formed from A by making rows into columns (or vice versa).

```
X = rand(5,3)
Y= X'*X          % pos-def
[A,B]=eig(Y)     % positive e-vals
A'*A             % check orthogonal
A*B*A' - Y       % check decomp
```

A positive definite matrix, Y, is one where, for a vector, **v**, **v**'Y**v** is always positive or zero, and only zero where **v**=0. It is not too hard to show x'*x is positive definite. It is also not too hard to show that x'*x is also symmetric.

A symmetric matrix, A, is one where A'=A.

Symmetric matrices have real eigenvalues and the eigenvectors can be chosen to be orthonormal.

Positive definite matrices have positive eigenvalues.

The eigenvalue decomposition of the symmetric matrix, Y, gives Y=A*B*A' where B is diagonal, and A is orthogonal. Recall that orthogonal means A'*A = I. The eigenvectors of Y are the columns of A.

output

```
>> x = rand(5,3)

x =

    0.8147    0.0975    0.1576
    0.9058    0.2785    0.9706
    0.1270    0.5469    0.9572
    0.9134    0.9575    0.4854
    0.6324    0.9649    0.8003

>> y = x'*x

y =

    2.7345    1.8859    2.0785
    1.8859    2.2340    2.0461
    2.0785    2.0461    2.7591
```

```
>> [a,b]=eig(y)

a =

   -0.1065    0.8020    0.5877
    0.7923   -0.2886    0.5375
   -0.6007   -0.5229    0.6047

b =

    0.4291         0         0
         0    0.7006         0
         0         0    6.5979
```

```
>> a'*a

ans =

    1.0000   -0.0000    0.0000
   -0.0000    1.0000         0
    0.0000         0    1.0000

>> a*b*a'-y

ans =

  1.0e-14 *

   -0.1332   -0.0666   -0.1332
   -0.0666   -0.0444   -0.1332
   -0.1332   -0.0888   -0.0888
```

Image Formation

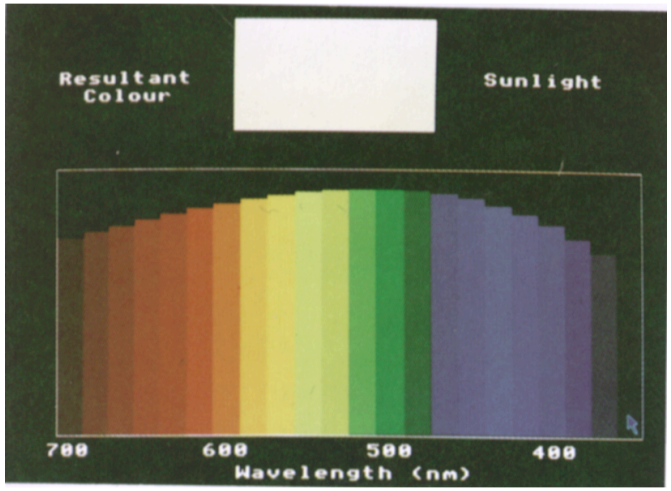
§1 (focus on §1.1.1, §1.4.2), highlights of §4

- Images
  - spatial sampling
    - typically of light on a plane
    - typically encoded as an array of values
  - carry information about the world when light has interacted with it
- Image formation is essentially at the level of a "pixel"
- We will briefly study in turn
  - light and its interaction with the world (with more to come later)
  - camera spectral response
  - camera geometry

Light

- Geometrically approximate light by rays (vectors)
- Typical scene has light going in a multitude of directions
- A bit of light has additional characteristics (energy/wavelength, polarization)
- The light in a certain direction is a mix, so we get a “spectrum” over wavelengths
- Spectrum records how much power is at each wavelength
- Visible portion is about 400 to 700 nm (Certain applications may require modeling some IR and/or UV also).

Example---sunlight spectra



Two disparate source spectra

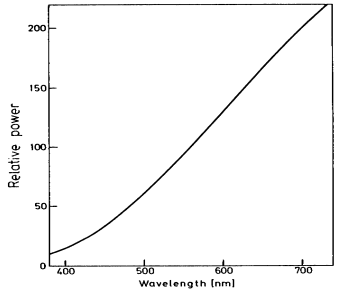


Fig. 4.1. Wavelength composition of light from a tungsten-filament lamp [typified by CIE ILL A (Sect. 4.6)]. Relative spectral power distribution curve. Color temperature: 2856 K

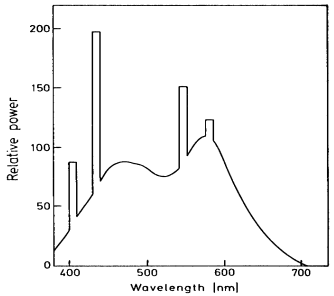
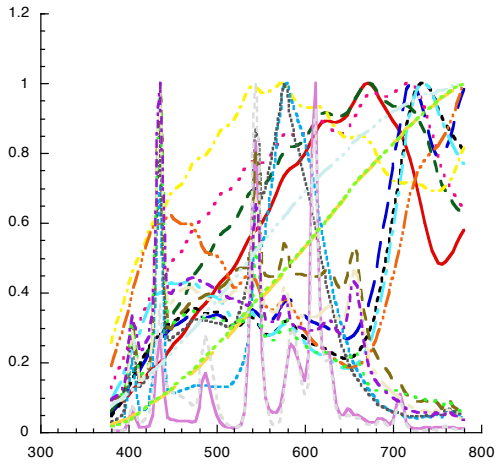
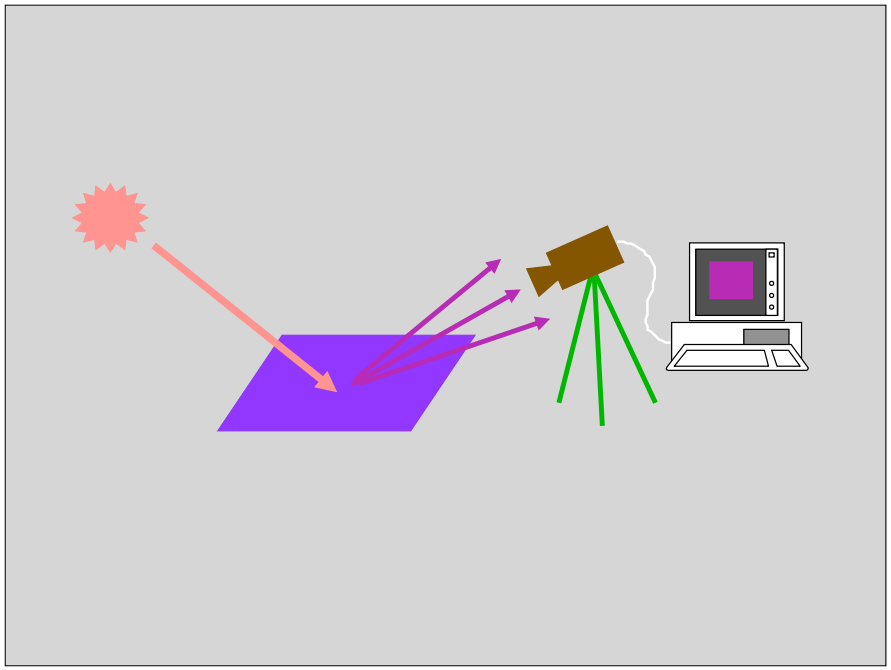


Fig. 4.2. Wavelength composition of light from a daylight fluorescent lamp. Typical relative spectral power distribution curve. Correlated color temperature: 6000 K. (Based on data of Jerome reported in [Ref. 3.14, p. 37])

Spectra of many sources

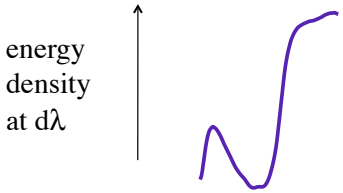




Light (summary)

Light energy reaching a camera sensor has a distribution over wavelength, λ.

(*Recall from physics that wavelength is inversely related to photon energy)



Typical light energy spectrum reaching a camera sensor.

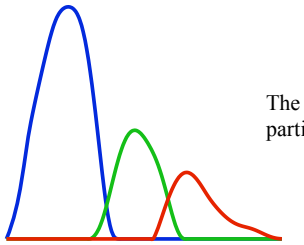
<--- wavelength --->

* Things marked with red stars are optional comments in the current context. If cover them in more detail later on, then they might not be optional anymore.

Sensors

Sensors (including those in your eyes) have a varied sensitivity over wavelength

Different variations lead to different kinds of sensor responses ("colors" in a naïve sense)

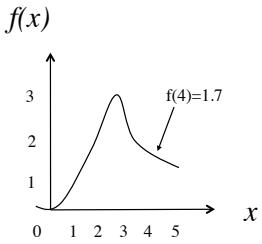


The sensor sensitivity spectra for a particular color camera (they vary).

<--- wavelength --->

Math warm-up --- functions and vectors

Function



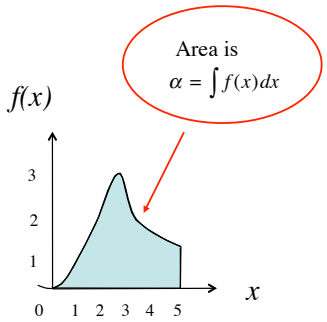
Vector Representation

$\mathbf{f} = (0, 1, 2, 3, 1.7, 1.4)$

$\mathbf{f}_i = f(i)$

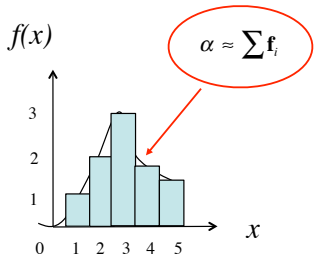
Function

Vector Representation



Bar *i* has height, $\mathbf{f}_i$

Area is approximately the sum over vector elements



Two simple linear operators

Discrete: $h(\mathbf{f}) = \mathbf{f} \cdot \mathbf{k}$

Maps the vector **f** to a number. **k** is also a vector.

Continuous: $h(f) = \int f(\phi)\ k(\phi)\ d\phi$

Note that h() is the area under the curve of the product of two functions---example to come soon.

Linear sensor response

For each wavelength, the sensor responds in proportion to the signal, AND the sensor sensitivity (i.e., the response is to the product of L(λ)*R(λ)).

The response to the entire signal is the total of the above for all wavelengths, λ.

Calculus way

Take the integral over wavelength of L(λ)*R(λ)

Linear algebra way

Add up discrete samples of L_i*R_i

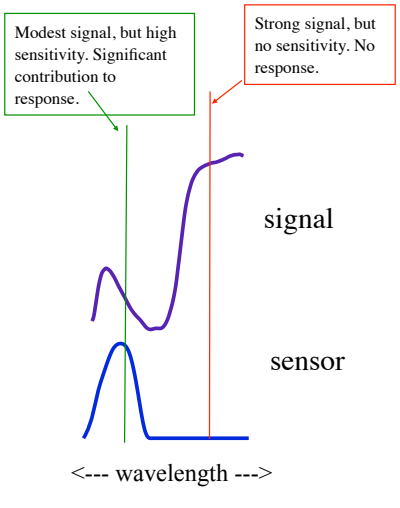
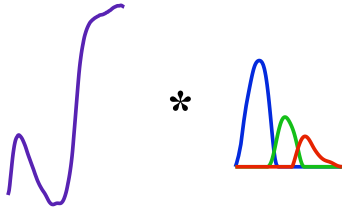


Image Formation (Spectral)

$$(R,G,B) = \int \quad * \quad d\lambda$$



On the next page, (R,G,B) is the row vector, **ρ**, with elements $\rho^{(k)}$. So, $R=\rho^{(1)}$, $G=\rho^{(2)}$, $B=\rho^{(3)}$.

More formally,

The response of an image capture system to a light signal $L(\lambda)$ associated with a given pixels is modeled by

$$\upsilon^{(k)} = F^{(k)}(\rho^{(k)}) = F^{(k)}\Big(\underbrace{\int L(\lambda)\ R^{(k)}(\lambda)\ d\lambda}_{\text{from previous slide}} \Big)$$

where $R^{(k)}(\lambda)$ is the sensor response function for the k[th] channel and $\upsilon^{(k)}$ is the k[th] channel result.

$R^{(k)}(\lambda)$ includes the contributions due to the aperture, focal length, sensor position in the focal plane.

$F^{(k)}$ absorbs typical non-linearities such as gamma.

Important

Discrete Version

Often we represent functions by vectors. For example, a spectra might be represented by 101 samples in the range of 380 to 780 nm in steps of 4nm.

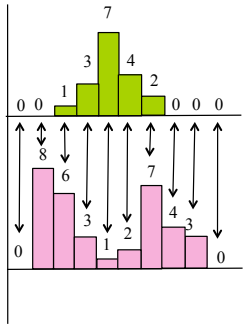
Then $L(\lambda)$ becomes the vector **L**, $R^{(k)}(\lambda)$ becomes the vector $\mathbf{R^k}$, and the response (ignoring non-linearity issues) is given by a dot product:

$$\rho^{(k)} = \mathbf{L} \bullet \mathbf{R^{(k)}}$$

Important

Sensor/light interaction example

R=(0,0,1,3,7,4,2,0,0,0)



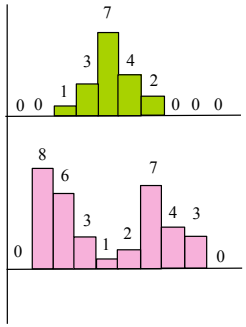
L=(0,8,6,3,1,2,7,4,3,0)

Multiply lined up pairs of numbers and then sum up

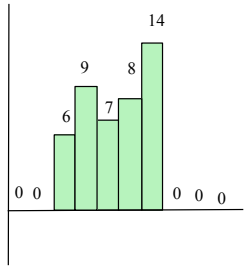
Important

Sensor/light interaction example

R=(0,0,1,3,7,4,2,0,0,0)



L=(0,8,6,3,1,2,7,4,3,0)



L .* R=
(0*0, 0*8, 1*6, 3*3, 7*1, 4*2, 2*7, 0*4, 0*3, 0*0)
=(0,0,6,9,7,8,14,0,0,0)

L • R = 0 + 0 + 6 + 9 + 7 + 8 + 14 + 0 + 0 + 0
= 44

Image Formation (Spectral)

- Note that (ignoring $F^{(k)}$) image formation is linear.
- Formally this means **if**:

$$L_1(\lambda) -> \rho_1^{(k)} \text{ and } L_2(\lambda) -> \rho_2^{(k)}$$

- **Then**:

$$aL_1(\lambda) + bL_2(\lambda) \; -> \; a\rho_1^{(k)} + b\rho_2^{(k)}$$

Image Formation (Spectral)

- Note that image formation loses spectral information
- Technically, the process is a projection
- This means that two **very** different spectra can map into the same color
- This is the key to color reproduction

Supplemental material

Image Formation (non-linear transform)

$F^{(k)}$ is often ignored (assumed to be the identity), but this is not a safe assumption, especially when color or radiometric measurements matter.

Commonly images are "gamma" corrected by raising the RGB values (normalized to [0,1]) to the power 1/(2.2).

Note that in such an image, a number twice as large does not mean that the light had twice the power!

To linearize RGB's from such a signal we compute:

$p=F^{-1}(v)=255*(v/255)^{2.2}$

Supplemental material

Image Formation (non-linear transform)

The non-linear transformation is added by captured devices **after** the raw capture (which is typically linear).

Because it is a single function applied to responses, it is easy to measure and compensate for.

Supplemental material

Image Formation (non-linear transform)

Why are images typically encoded in this way?

Historically, images have been gamma corrected on the assumption that their values drive a CRT (cathode ray tube) monitor which are non-linear devices. Their theoretical response to a voltage is energy output proportional to that voltage raised to the (5/2) power. Appropriately gamma corrected images display as linear on such devices.

Supplemental material

Image Formation (non-linear transform)

Coincidentally, this typically gamma correction is a sensible way to encode image data into a limited number of values (e.g. 256) due to the noise sensitivity of the human vision system.

Hence, while CRT displays are now obsolete, images are still typically non-linear, and the signal to modern displays (which are linear) are typically adjusted assuming typical incoming non-linear in images.

Supplemental material

Image Formation (non-linear transform)

If you have access to a Mac, then you can play with this under System Preferences --> Displays --> Color --> Calibrate (may need to select “expert”)

Demo!