Administrivia

Assignment one is now posted:
http://www.cs.arizona.edu/classes/cs477/spring08/ua_cs_only/assignments
Slides now being posted:
http://www.cs.arizona.edu/classes/cs477/spring08/ua_cs_only/lectures

Need to connect via a UA machine or use id (“me”; pw=“vision4fun”).

Lectures and assignments will require either connecting from a UA machine, OR
login id (“me”) and password (“vision4fun”).

Office hours (**tentative**) on T/R 5-5:30 and Friday 11:30-12 (not every week) by electronic signup.
login id (“public”) and password (“meetkobus”).
Also, if you coming from off campus, you will need
login id (“me”) and password (“pw4cal”) to start.

Eight machines in 9th floor lab (gr01-gr08) will be available for this course (only).

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”.

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

```
A = 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, A, is one where, for a vector, **v**, **v**’A**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
```

Very Brief Course Intro

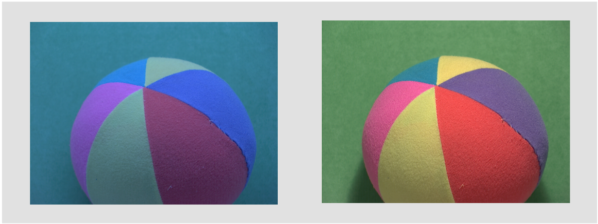
(Subject to change)

Part* I: How images are formed

- Image abstraction level is a single pixel
- Cameras
  - What a camera does
- Light
  - How to measure light
  - What light does at surfaces
  - How the brightness values we see in cameras are determined
- Color

* Here "part" refers to the book section. We cannot do them all in detail.

The Computational Colour Constancy Problem



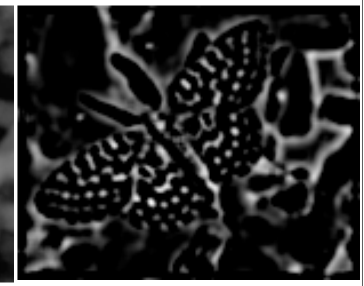
(Same scene, but different illuminant)

Part II: Early Vision in One Image

- Representing small patches of image
  - Often want to match points in different images, so we need to describe the neighborhood of the points (e.g for stereo)
  - Sharp changes are important in practice --- known as "edges"
  - Representing texture by statistics of local structure.
    - Zebras have lots of bars, few spots, leopards are the other way

Filters as templates

Filters respond to structures that "look" like the filter

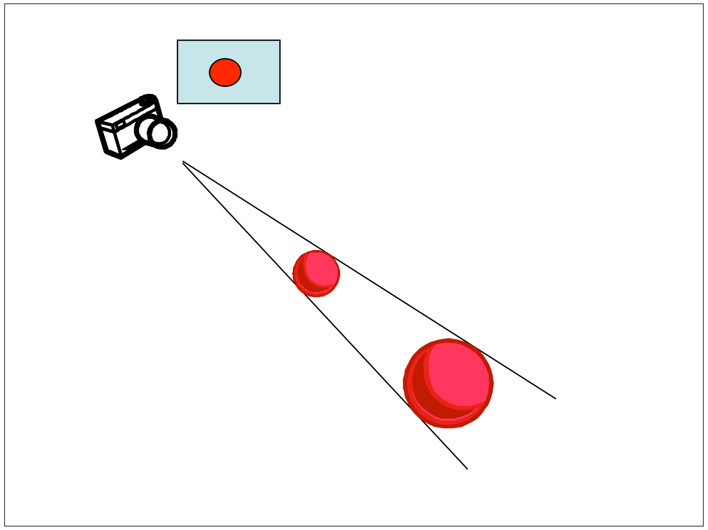


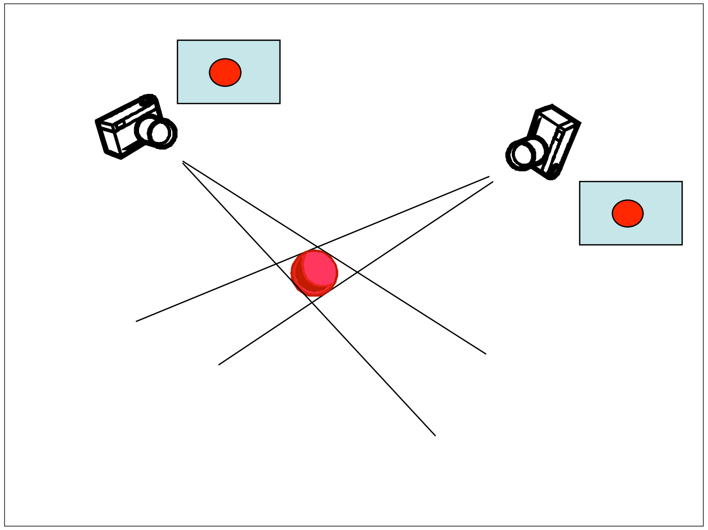
Texture

- Many objects are distinguished by their texture
  - Tigers, cheetahs, grass, trees
- We represent texture with statistics of filter outputs
  - For tigers, bar filters at a coarse scale respond strongly
  - For cheetahs, spots at the same scale
  - For grass, long narrow bars
  - For the leaves of trees, extended spots
- Objects with different textures can be segmented
- The variation in textures is a cue to shape

Part III: Early Vision in Multiple Images

- (May not have time in 2008)
- The geometry of multiple views
  - Two views of the same patch reveal geometry (if you can **match** them---"correspondence")
- Stereo vision, structure from motion



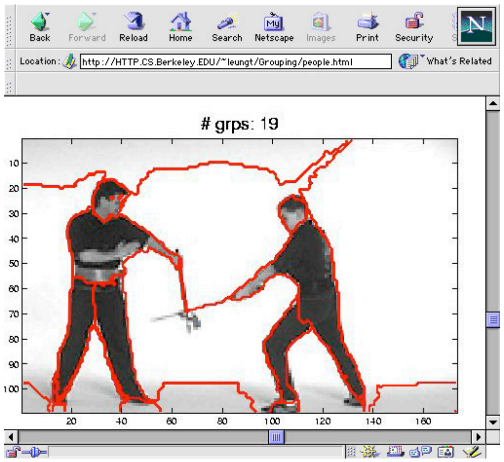


Part IV: Mid-Level Vision

- Finding coherent structure so as to break the image or movie into big units
  - Segmentation:
    - Breaking images and videos into useful pieces
      - finding video sequences that correspond to one shot
      - finding image components that are coherent in internal appearance
  - Tracking:
    - Keeping track of a moving object through a sequence of views
    - Use a model to predict next position and refine using next image

Grouping

- Which image components “belong together”?
- Belong together == lie on the same object
- Cues
  - similar colour
  - similar texture
  - not separated by contour
  - form a suggestive shape when assembled
  - move together



High Level Vision

- Visual information --> semantics
- Object recognition
  - Specific object (my car)
  - Object category (a car)
- Object / scene understanding
  - What are the pose (orientation) and location of scene objects
  - What about the object structure (very leafy plant?)
  - What might an object be good for (a step?)
- Activity recognition

High Level Vision

- Many applications
  - Digital libraries
    - Find me the picture of JFK and Marilyn Monroe embracing
  - Surveillance
    - Warn me if there is a mugging in the grove
  - HCI
    - Do what I show you
  - Military
    - Shoot this, not that

What are the problems in recognition?

- Variable appearance
  - Objects appear different due to pose, illumination, occlusion, etc.
- Which bits of image should be recognized together?
  - Segmentation (issue--segment first or recognize first)
- How can objects be recognized without focusing on detail?
  - Levels of abstraction (object categories vs individuals)

Part V: High Level Vision (Geometry)

- We will not do much of this part in 2008
- The relations between object geometry and image geometry
- One difficulty in recognition is pose change
  - One approach is model based vision
    - find the position and orientation of known objects
  - Another approach is to use descriptors which are invariant to (small) pose changes

Part VI: High Level Vision (Probabilistic)

- Using probabilistic models and classifiers to recognize objects
  - Probabilistic methodology
    - Bayes rule: p(object | data) ~ p(data | object) p(object)
  - Templates and classifiers
    - Find objects from a canonical view (e.g. frontal face) by matching a template
      - best when combined with a probabilistic classifier
      - transformed views require looking with more templates (e.g. rotated, scaled, faces)
  - Relations
    - find the parts with a classifier, and then reason about the relationships between the parts to find the object.

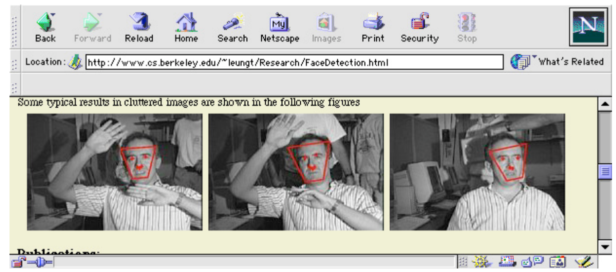
Matching templates

- Some objects are 2D patterns
  - e.g. faces
- Build an explicit pattern matcher
  - need to discount changes in illumination
  - changes in background are hard
  - changes in pose are hard

http://www.ri.cmu.edu/projects/project_271.html

Relations between templates

- e.g. find faces by
  - finding eyes, nose, mouth
  - finding an assembly of the three that has the “right” relations



Official Start of Course

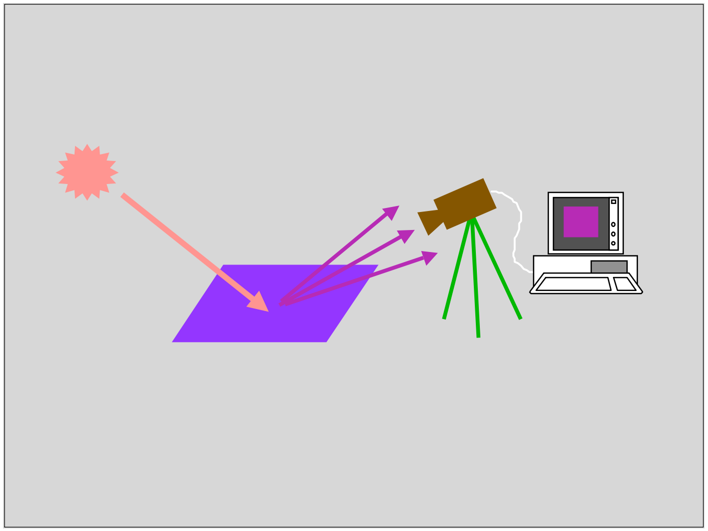
Image Formation

§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”

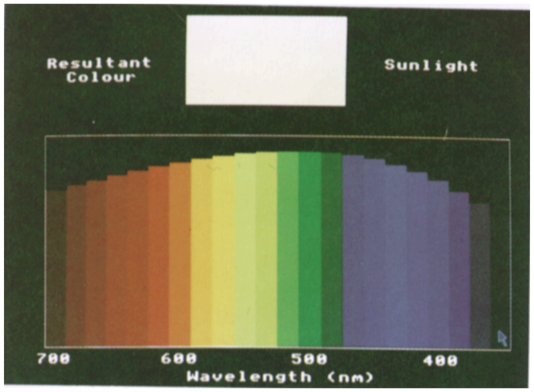
- We will briefly study in turn
  - light and its interaction with the world (more in a few lectures)
  - camera spectral response
  - camera geometry



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

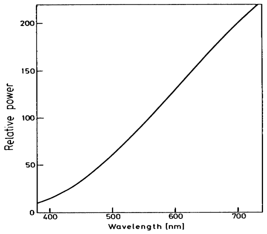


Fig. 4.1. Wavelength composition of light from a tungsten-filament lamp [typified by CIE ILL A (Sect. 4.6)]. Relative spectral power distribution curve. Color temperature: 2856 K

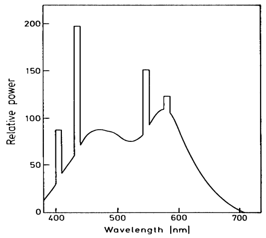
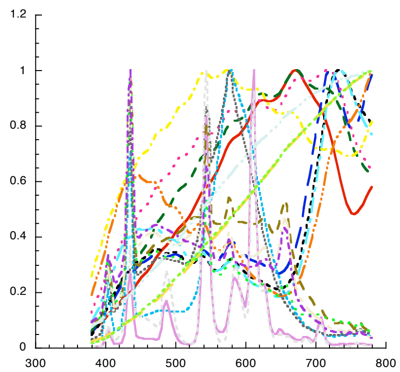


Fig. 4.2. Wavelength composition of light from a daylight fluorescent lamp. Typical relative spectral power distribution curve. Correlated color temperature: 6000 K. (Based on data of Jerome reported in [Ref. 3.14, p. 37])

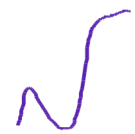
Spectra of many sources



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)



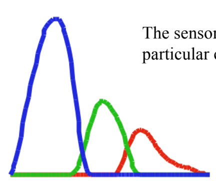
<--- wavelength --->

* 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 a lot).

<--- wavelength --->

Math warm-up --- linear operators

h(x) is a linear operator means that ?

Math warm-up --- linear operators

h(x) is a linear operator means that

$$h(a{*}x+b{*}y) = a{*}h(x) + b{*}h(Y)$$

Examples?

Math warm-up --- linear operators

Examples of linear operators

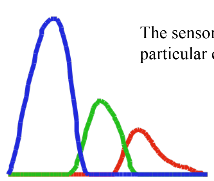
Discrete: $h(\mathbf{x})=\mathbf{k}\cdot\mathbf{x}$
($\mathbf{x}$ and $\mathbf{k}$ are vectors)

Continuous: $h(f)=\int k(x)f(x)dx$

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)



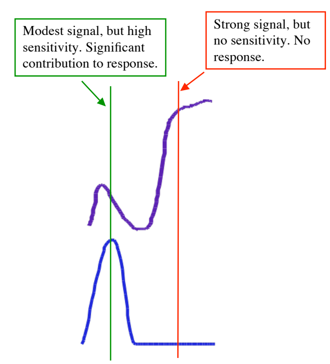
The sensor sensitivity spectra for a particular color camera (they vary a lot).

<--- wavelength --->

Linear sensor response

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

The response to the signal is the total for all wavelengths, λ (i.e. the integral over wavelength of L(λ)*R(λ)).



<--- wavelength --->