Computer Vision

- Me. Robert Pless
  - pless@cse.wustl.edu
  - 518 Lopata Hall.
  - Office Hours: Thursday 7-8
  - Virtual Office Hours: Tuesday 3-4 (profpless on AIM, YahooMessenger)
- Class. CSE 519, Computer Vision
  - http://www.cse.wustl.edu/~pless/519
- TA office hours to be announced.
- Textbook: “Introductory Techniques for 3-D Computer Vision” by Trucco, Verry.

Expectations

- This is a graduate level computer science class.
- I expect you to be able to program, debug, etc.
- I will never look at your code, therefore, you can write your programs in whatever language you want: Java, C++, Matlab, Cobol, Lisp, Machine Language, perl, gawk (?), Pascal, Basic, ProLog (!), Scheme, VHDL, Javascript, Flash.
- Some of the above languages may be easier than others.

Grading

Class grading will be project based.

Limited project scope. Due approximately every two weeks.

Projects will get more complicated, but not intended to be longer (?!).

Projects will contain **both** programming and math/proof problem set type components.

Projects will be turned in as a web page.

Projects may have group options/ contests. I am open to suggestions on altering any project to be more relevant to your work/research area.

At the end of the class, each of you will have a web vision portfolio.

Grading

40% -- Initial grades for each project.

30-40% -- end of semester re-evaluation of portfolio (you can fix your project submissions after getting feedback)

20-30% -- final exam or project.

Our Goal

- To interpret images and video. What information can we extract from an image?

Our Goal

- To interpret images and video. What information can we extract from an image?
- What do we assume in order to extract that information?

- To interpret images and video. What information can we extract from an image?
- What properties of the world let us create 3d models from video?

- To interpret images and video. What information can we extract from an image?
- What properties of the world let us create 3d models from video?
- What are appropriate 3D models?

- To interpret images and video. What information can we extract from an image?
- What properties of the world let us create 3d models from video?
- How complicated can scenes be?

- To interpret images and video. What information can we extract from an image?
- What do we assume in order to extract that information?
- How complicated can the motions be?

- To interpret images and video. What information can we extract from an image?
- What do we assume in order to extract that information?

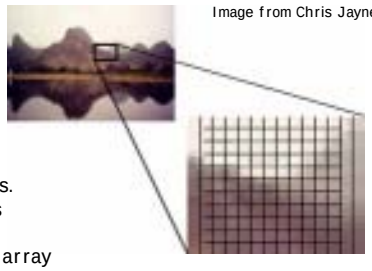
How to achieve this goal? (1) Cameras

History, progression and comparisons of different Cameras and optics.

Geometry, Linear Algebra

Images

Image from Chris Jaynes, U. Kentucky



Discrete vs. Continuous

I[x,y,t] ← array
I(x,y,t) ← function

Derivatives, Convolutions

Image transformations

Color/grayscale transformations

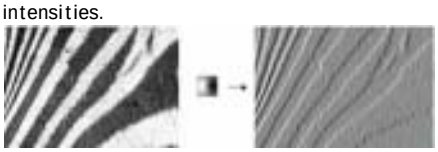
Geometric transformations.
Non-linear:
http://javaboutique.internet.com/AnStretch/

Linear: translations, rotations and uniform stretching and skewing of the image

Invertible vs. non-invertible transformations

Image Features.

- Drawing Features are points, lines, curves, shading etc.
- Image Features have to come from arrays of pixel intensities.



An image filter result

Fourier analysis and the frequency domain.

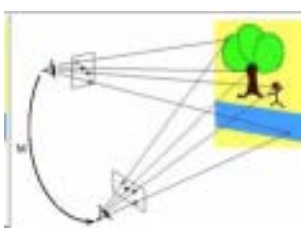
Image Features

http://bigwww.epfl.ch/demo/jedgedetector/start.php

Often, image features are defined relative to some scale. Multi-resolution features help to find features at many scales (useful because images of objects come at many scales).

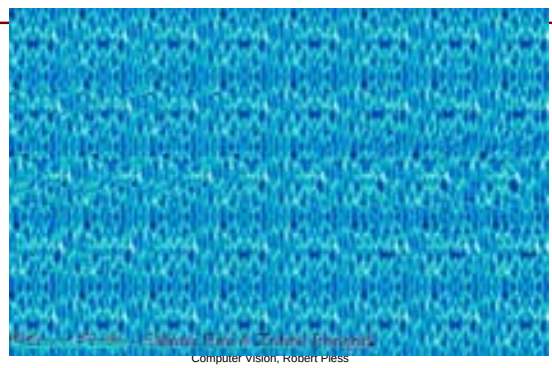
Stereo

1. What are the right measurements to make on the images?
2. What can you reconstruct from two images?



Linear Algebra, Geometry, Robust statistics?

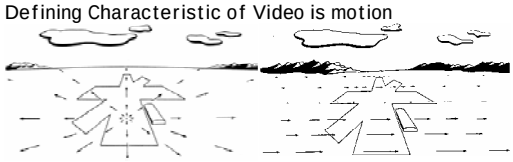
Autostereograms



What limits are there to the patterns?

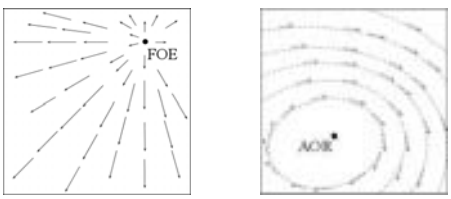
Video “Features”

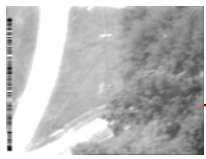
Defining Characteristic of Video is motion



Differential geometry

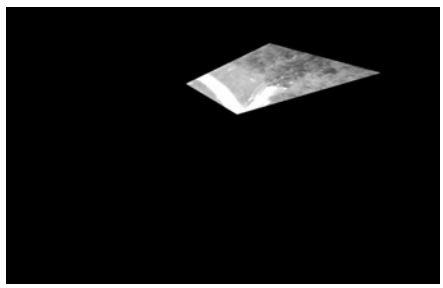
The motion field defines the 3D motion





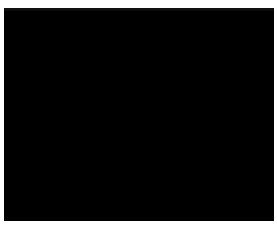
Special Topics: Surveillance

- Image transformations, function optimization, geometry



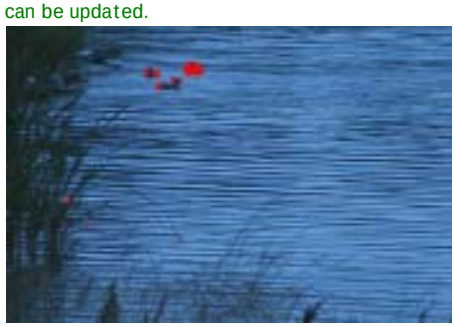
Special Topics: Surveillance

- Uses statistics, filters, tracking.



More surveillance.

- Not just any statistics (!) compact statistics that can be updated.



Human motion analysis

Model Based,

- From: **iris.usc.edu/~taozhao/papers/ACCV02/ACCV02.html**

Forward kinematics, function optimization, model fitting

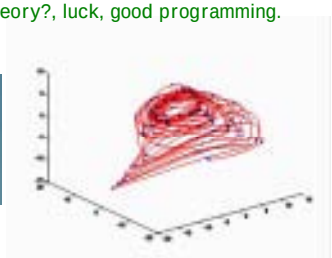
Motion analysis

Image Based, requires graph theory?, luck, good programming.

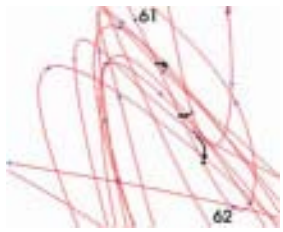
From Arno Schoedl.

Motion Video Analysis

Linear algebra graph theory?, luck, good programming.



Motion Analysis



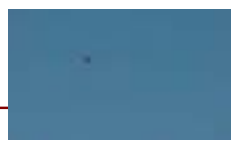
Temporal Super-Resolution

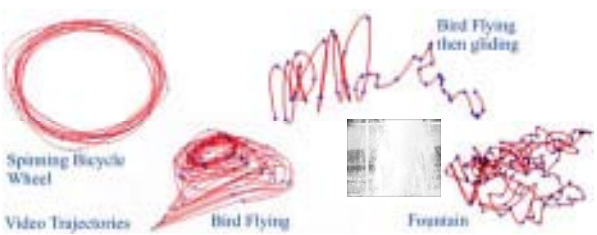
5 x Framerate

20 x Framerate
(different input)

Interpolation, splines

Complex Video Analysis

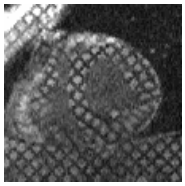




Motion Analysis

Medical Image Processing

Calculate strains in deforming tissue



Physics, strain theory, finite element analysis? Fourier transforms?

Medical Image Processing

Registration/ self organization/ motion analysis of MRI images

MRI imaging theory (Fourier Analysis again!), good programming, abstract algebra.

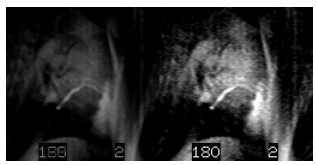
Demo of which I am irrationally proud.

http://www.cse.wustl.edu/~pless/wumapDemo

Html web programming(!), differential/manifold geometry.

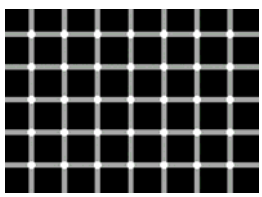
Noise reduction

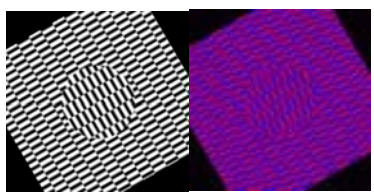
http://www.cse.wustl.edu/~pless/wumapDemo



Statitistics, cool new(ish) tricks like thin-plate-splines.

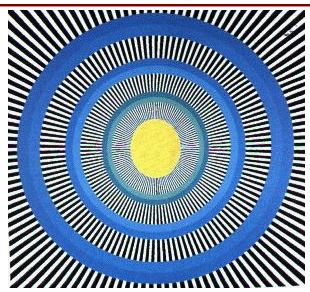
Human visual system and illusions



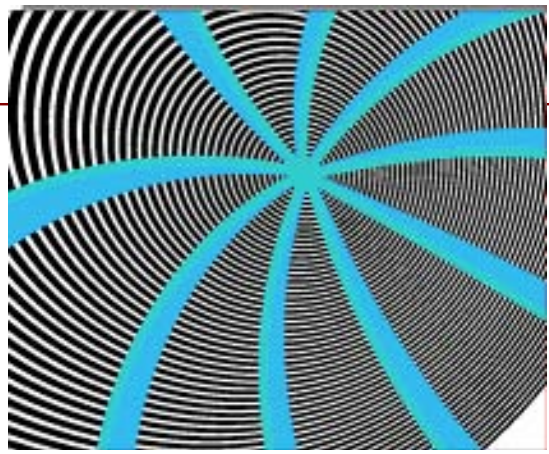


Estimation theory, statistical bias

Computation & Perception



Feedback loops, control theory?



Momentary Pause with Gratuitous Image

Photo by: Alexey Ratinoff

First Day, First Project

Hello world, in image terms, is to apply a filter to an image.

Read image into your program.
(open the file, read in the data, put it into an array of integers, floats, whatever).

If you have a color image, then perhaps you have a red array, green array, and a blue array, or maybe you have a 3d which is "width x height x 3".

Project 1

1)

10	29	11	9	18	,,,
5	11	23	99	98	…
17	28	39	58	17	…
89	86	88	88	82	…
17	81	71	72	72	…
…	…	…	…	…	…

2) Define a "filter"

-1	-2	-1
-2	12	-2
-1	-2	-1

Project 1

3) Apply

filter:

-1	-2	-1
-2	1	-2
-1	-2	-1

-1 x 10 + -2 x 29 + -1 x 11 + -2 x 5 + 12 x 11 + -2 x 23 + -1 x 17 + -2 x 28 + -1 x 39 = -141

Orig image

			9	18	,,,
			99	98	…
			58	17	…
89	86	88	88	82	…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141				…
					…
					…
					…
…	…	…	…	…	…

Project 1

filter:

3) Apply

-1	-2	-1
-2	1	-2
-1	-2 -2	-1

-1 x 29 + -2 x 11 + -1 x 9 + -2 x 11 + 12 x 23 + -2 x 99 + -1 x 28 + -2 x 39 + -1 x 58 = ??

Orig image

10				18	,,,
5				98	…
17				17	…
89	86	88	88	82	…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141	??			…
					…
					…
					…
…	…	…	…	…	…

Project 1

filter:

3) Apply

-1	-2	-1
-2	1	-2
-1	-2 -2	-1

-1 x 29 + -2 x 11 + -1 x 9 + -2 x 11 + 12 x 23 + -2 x 99 + -1 x 28 + -2 x 39 + -1 x 58 = ??

Orig image

10	29				,,,
5	11				…
17	28				…
89	86	88	88	82	…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141	??	??		…
					…
					…
					…
…	…	…	…	…	…

Project 1

filter:

3) Apply

-1	-2	-1
-2	1	-2
-1	-2 -2	-1

-1 x 29 + -2 x 11 + -1 x 9 + -2 x 11 + 12 x 23 + -2 x 99 + -1 x 28 + -2 x 39 + -1 x 58 = ??

Orig image

10	29	11	9	18	,,,
			99	98	…
			58	17	…
			88	82	…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141	??	??		…
	??				…
					…
					…
…	…	…	…	…	…

Project 1

filter:

3) Apply

-1	-2	-1
-2	1	-2
-1	-2 -2	-1

-1 x 29 + -2 x 11 + -1 x 9 + -2 x 11 + 12 x 23 + -2 x 99 + -1 x 28 + -2 x 39 + -1 x 58 = ??

Orig image

10	29	11	9	18	,,,
5				98	…
17				17	…
89				82	…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141	??	??		…
	??	??			…
					…
					…
…	…	…	…	…	…

Project 1

filter:

3) Apply

-1	-2	-1
-2	1	-2
-1	-2 -2	-1

-1 x 29 + -2 x 11 + -1 x 9 + -2 x 11 + 12 x 23 + -2 x 99 + -1 x 28 + -2 x 39 + -1 x 58 = ??

Orig image

10	29	11	9	18	,,,
5	11				…
17	28				…
89	86				…
17	81	71	72	72	…
…	…	…	…	…	…

output:

					,,,
	-141	??	??		…
	??	??	??		…
					…
					…
…	…	…	…	…	…

At the core this program is a nested for loop.

```
For x = 1 : x_size
    For y = 1: y_size
        filterResponse = 0
        For i = 1: x_size_filter
            For j = 1: y_size_filter
              filterResponse += image(x+i,y+j) x filter(i,j)
            End
        End
        OutputImage(x,y)= filterResponse.
    End
end
```

0) Read chapters 3,4
For sample image A of your choice:

1) Pick a smoothing filter F. The book defines "smoothing filters" on page 56, 62, and mean filter at bottom of 62.

Apply F to image A to get A1
Apply F to image A1 to get A2
Apply F to image An to get image An+1

The book defines "edge detection" algorithms on page 80. These edge detection algorithms work as follows:

2) Apply a pair of filters to get image B1, B2
Output(x,y) = sqrt(B1(x,y)^2 + B2(x,y)^2)

Run edge detection on A, A1, ... An

3) Make up new filter (can be larger, much larger, than 3x3), Can be linear or non-linear (like mean filter).

Contest: Most interesting output filtered image gets bonus point.

4) For an (a x b) pixel image, and a (m x n) pixel filter, how many computations are necessary to compute the filtered image?

5) How many computations are necessary to compute first, the filtered image for an (a x b) pixel image, then a (1 x n) filter, then filter the result with a (m x 1) filter.

6) Give an example of a 3x3 filter than can be separated into two components as suggested in question 5.

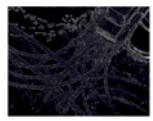
7) Give an example of a 3x3 filter that cannot be separated into two components as suggested in question 5.

8) You may work by yourself OR in pairs. This option will be available for many of the projects throughout the semester. However.
(1) You may only work with the same partner once this semester.
(2) Pairs must do parts 1-2 using both a linear smoothing filter, AND the median-filter. Results for both must be presented and a short explanation of the differences should be presented.

Web page to turn in:

Your name(s):

Smoothing filter:

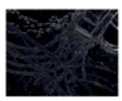


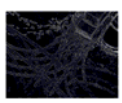
Orig **filter**

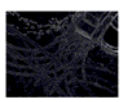
More

And

More Blurred







(if group project, you need three columns here, plus Explanation paragraph.).

3) Your new filter. If linear, show filter

If non linear, give description like
MED_filter at bottom of page 62.

4,5) Answer in form of function of a,b,m,n, such as:
$178\ a + 15\ b + m^n - 14 \sin(m)$

6,7) Give example 3 x 3 filter

8) Does NOT in any way affect your grade, but may affect future projects

Estimate how much time you spent:

0) On this project, but independent of any problem
1) On problem 1
2) On problem 2
.
.
7) On problem 7
8) Extra time for making webpage.

A Final Note.

The programming required for this project is a fairly minimal level considering that this is a graduate computer science class. If you have trouble (perhaps with the help of GOOGLING "how do I read in an image") how to successfully read an image in, loop through an array and write an image out, you are likely to have difficulty in many parts of the class.

But, for those of you who stay, this class will be a tour of several different fields of computer vision at the forefront of today's research.