

CS377S

DESIGNING APPLICATIONS THAT SEE

Computer Vision for HCI



ASSIGNMENT #1: HUMAN VISION AND BASIC IMAGE PROCESSING

Due January 22, 2008 (in lecture)



Reflection



Ideation



Exercise



Bonus Challenge



The Man Who Mistook His Wife for a Hat (6 Points)

When Oliver Sacks has his initial encounter with “the man who mistook his wife for a hat,” he explains that Dr P. did not look at him in the normal way, but rather “made sudden strange fixations - on my nose, on my right ear, down to my chin, up to my right eye - as if noting (even studying) these individual features, but not seeing my whole face, its changing expressions, 'me' as a whole.”

Dr P. could experience the world only as small individual features. He was unable to group these low-level features into high-level constructs. Sacks writes that he “had no sense whatever of a landscape or a scene,” and when it came to recognizing people, “in the absence of obvious ‘markers,’ he was utterly lost.” In many ways, Dr P. functioned like a computer, construing the world “by means of key features and schematic relationships... without the reality being grasped at all.”

What tasks could Dr P. still accomplish by perceiving the world in this way? What tasks presented him with the most difficulty? What does this suggest about the capabilities of computer vision?



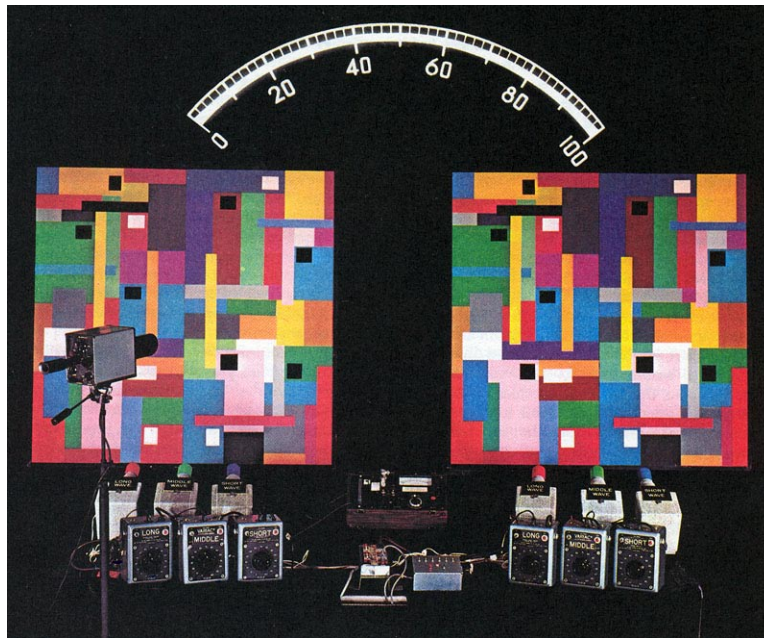
Vision by Man and Machine (6 Points)

Poggio describes the layout of ganglion cells in the retina as the biological equivalent of a center-surround filter, “approximating the Laplacian of a Gaussian.” From the readings, the lecture material, or from an outside source, cite two additional examples of components or processes in human vision that have equivalents in Computer Science or Electrical Engineering. These could be particular mathematical equations, algorithms, or electrical circuits, for example. How are the biological and electronic implementations of these techniques the same, and how do they differ? Is one implementation more flexible or robust than the other?

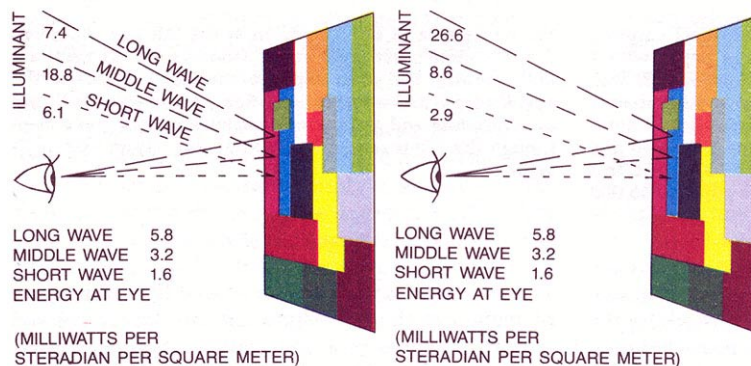


The Case of the Colorblind Painter (4 Points)

Land's "Mondrian" experiments, described in the reading, used two identical displays of sheets of colored paper mounted on boards. Each "Mondrian" was illuminated with its own set of three projectors. These projectors used bandpass filters and brightness controls to carefully adjust the intensities and wavelengths of light striking each board. A telescopic photometer could be pointed at any area to measure the flux, one wave band at a time.



In a typical experiment, the illuminators were adjusted so that an area of the Mondrian at the left and some differently-colored area of the Mondrian at the right were both sending the same triplet of radiant energies to the eye. In the example below, the eye perceives the exact same triplet of long-, middle-, and short-wave energies from the red area (left) and the blue area (right).



In this experiment, what color sensations would a normal subject consciously perceive? What does this say about the relationship between reflectance and illumination of the objects in our world (the energy reaching our eyes) and the sensation of color?



Digital Video Capture (8 Points)

Begin this exercise by capturing a short segment of digital video (2-5 minutes in length) of an interesting environment or sequence of events in the real world. This could be the view out your apartment window, what you see when riding your bicycle between classes, footage of a sporting event, video of your pet or your friends... the possibilities are unlimited. Just make sure that there is something interesting going on in the video (no videos of blank walls). Try to come up with an example where you might appreciate using a computer as an “extra set of eyes” to watch things for you – for example, warning you when a pot is boiling over in the kitchen, or when someone has removed a book from your bookshelf. Here are some examples of the type of video you might create:

<http://cs377s.stanford.edu/assignments/sushi.avi>

<http://cs377s.stanford.edu/assignments/skateboard.avi>

<http://cs377s.stanford.edu/assignments/crosswalk.avi>

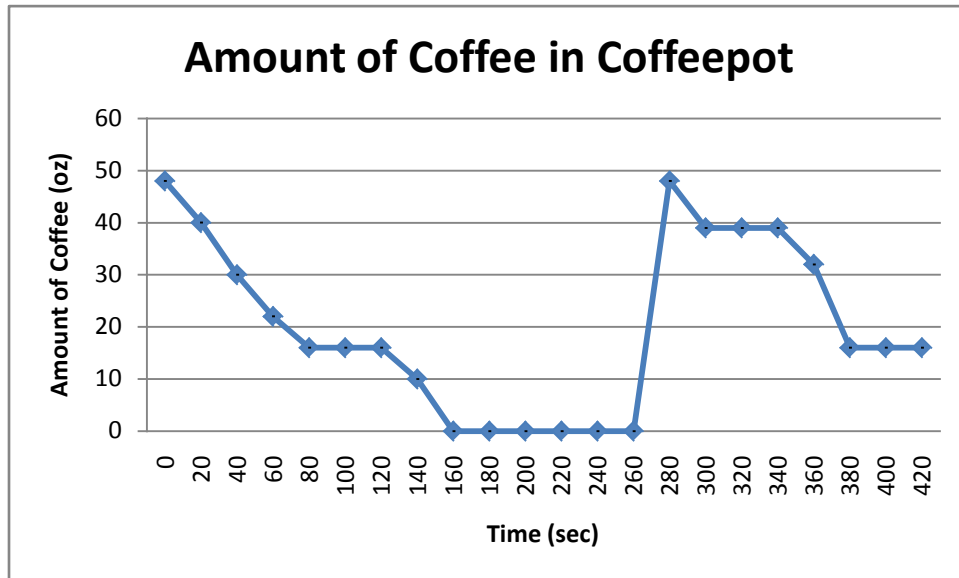
To play these videos, you may need to install the DivX codec, available from <http://divx.com/>.

You can capture your video in any of the following ways:

1. Many handheld point-and-shoot digital cameras are capable of capturing short, low-resolution video clips directly in AVI, MPG, or MOV format. If you or one of your friends has a digital camera with this capability, you can use it to capture a video.
 2. Most USB webcams will allow you to capture digital video, using a program like Windows Movie Maker or VB VidCap. If you have a USB webcam, you can capture your video this way (though it will restrict you to capturing your video in the same location as your computer).
 3. If you have a handheld DV camera with a Firewire connection, you can use it to record video, and copy the video to your computer using a program like Adobe Premiere, Windows Movie Maker, or iMovie. If you have access to a video camera, but no way to get the video onto a computer, you can bring your camera to office hours to have your video captured to computer.
- a) Once you have captured your video, watch through it a few times and mark the frames that contain the events or information that you are interested in. For example, you might create a list of events like this:

Timecode	Event
0:35	A Train arrives
0:44	A Train departs
2:23	C Train arrives
2:56	C Train departs
3:33	A Train arrives

If you are interested in a continuous parameter or value rather than discrete events, you can create a graph instead, like this:



- b) From what you know so far about human visual processing, how is your visual system picking out the event, value, or object of interest in your video?
- c) From what little you know about computer vision, how might the same event, value, or object be extracted by a computer algorithm? If you're not sure, just hazard your best guess.
- d) Post your video online, preferably by uploading the original video file to your personal Stanford web space. If you have difficulty doing this, you can use a video sharing site like YouTube instead, but the original source video is preferred. Include the URL of the video in your assignment hand-in.



MATLAB Introduction (6 Points)

This exercise is a very basic introduction to image processing in MATLAB that will prepare you for a more detailed walkthrough in lecture.

- a) Your first task is to learn how to run MATLAB, preferably on your own computer. You can run MATLAB in one of four ways:
 - 1. MATLAB is already installed on the machines in the Myth cluster (Gates B08). You can complete this exercise on one of the Myth machines, but you will not be able to follow along during the in-class tutorial. Once you log in, type `matlab` at any prompt to begin.
 - 2. You can run MATLAB remotely, but display it on your machine, using the computers in Stanford's Remote Computing facility. Information on the facility is available here: <http://www.stanford.edu/services/unixcomputing/environments.html#remote>

Instructions for remotely running X-Windows programs such as MATLAB can be found here:

<http://www.stanford.edu/services/unix/moreX.html>

3. You can run an online trial of MATLAB in your web browser at the MathWorks website:
http://www.mathworks.com/programs/trials/online_trials/index.html

However, your trial is limited to two hours in length, and you will not be able to upload, save, or print your work, so this method is not recommended.

4. If you are a member of the Stanford Graphics Lab, you can install MATLAB on your personal machine and use the Graphics Lab license server. The installation files are shared as:

<\\blur\Matlab installers>

License files and installation instructions are available here:

<http://graphics.stanford.edu/lab/soft/matlab/>

- b) Watch the six-minute demo video entitled "Introduction to the Image Processing Toolbox," available on the MathWorks website:

<http://www.mathworks.com/products/image/demos.html>

This video gives you an introduction to some of the image processing capabilities in Matlab.

- c) In this exercise we will write a MATLAB routine to detect round objects in an image. Begin by downloading the following image:

<http://cs377s.stanford.edu/assignments/tennis.jpg>

Load the image into Matlab with the following commands:

```
RGB = imread('tennis.jpg');  
imshow(RGB);
```

- d) Now follow the step-by-step instructions in the MATLAB demo entitled "Identifying Round Objects," available here:

<http://www.mathworks.com/products/demos/shipping/images/ipexroundness.html>

Since you have already loaded your image, you can begin from Step 2 of the instructions.

- e) How well does this algorithm perform? Where does it break down?
- f) See if you can adjust any of the parameters in the sequence of commands to improve the output. For example, the threshold is chosen automatically using the commands

```
threshold = graythresh(I);  
bw = im2bw(I, threshold);
```

But you could also choose a threshold manually, like this:

```
bw = im2bw(I, 0.5);
```

- g) Save your output image as a JPEG file. Include a printout of your output image with your assignment hand-in, and explain any changes you made to the code in order to produce it.



Where's Waldo? (6 Points)



This is an open ended exercise based on the “Where’s Waldo?” book series. You are asked to create an automated “Waldo Detector” using image processing techniques in MATLAB.

We provide the following series of images for you to analyze:

<http://cs377s.stanford.edu/assignments/waldo/wheresWaldo1.jpg>

<http://cs377s.stanford.edu/assignments/waldo/wheresWaldo2.jpg>

<http://cs377s.stanford.edu/assignments/waldo/wheresWaldo3.jpg>

<http://cs377s.stanford.edu/assignments/waldo/wheresWaldo4.jpg>

In each image, Waldo is hiding in the midst of a busy crowd. Waldo always wears the same red and white striped sweater and hat. However, he may be carrying a stack of books that vary from scene to scene. Here are a few sample images of Waldo that you may find helpful in solving this problem:

<http://cs377s.stanford.edu/assignments/waldo/waldo0.jpg>

<http://cs377s.stanford.edu/assignments/waldo/waldo1.jpg>

<http://cs377s.stanford.edu/assignments/waldo/waldo2.jpg>

<http://cs377s.stanford.edu/assignments/waldo/waldo3.jpg>

<http://cs377s.stanford.edu/assignments/waldo/waldo4.jpg>

<http://cs377s.stanford.edu/assignments/waldo/waldo5.jpg>

Unfortunately, there are also several Waldo lookalikes in each scene. Try not to be fooled by these impostors! Here are some fake Waldos to look out for:

<http://cs377s.stanford.edu/assignments/waldo/waldoLookAlikes.jpg>

Write a MATLAB function
`FindWaldo(filename)`

That loads the image specified by `filename`, and displays the image with the detected location of Waldo marked by a rectangle. You may implement any of the techniques discussed in class or in this assignment, or invent your own approach. For example, you might try template matching, examining color distributions, or looking for colored stripes or circles. You may assume that each input image contains only one valid Waldo.

In your assignment hand-in, include the code listing for your `FindWaldo` function and a printout of its output on one of the input images.