COMS W4735y: Visual Interfaces to Computers
Spring, 2015
Assignment 1: TENTATIVELY due February 12, 2015

Course homework overview

This assignment is worth 20% of your course grade. It must be done individually and not in teams. A second and third assignment will also each be worth 20% of your course score. A project proposal will be worth 7% of your score. The final project with demonstration and 10 page write-up (or the final course research paper of 20 pages), will be worth 33% of your course score. There will be no exams.

Visual Combination Lock: Design Specifications

The goal of this assignment is to take a short sequence of visual images, and to determine from them if the user has placed some body part(s) in a predetermined sequence of locations and/or poses. For example, the program can require two images of the user's hand on a table, and the "combination" is if the hand is placed with a closed fist in the center of the first image, followed by a flat palm with fingers outwardly splayed (i.e., the gesture for "five!") in any of the four corners of the second. Many other variants are possible, and, in fact, part of the grade will depend on how creative the domain engineering and the grammar are demonstrated to be.

To do this assignment, you will need access to a digital picture-taking device: a webcam, a scanner, a laptop with built-in camera, a cellphone with camera, a digital still camera, etc. The camera need not work in real time--you can store the pictures for later analysis. It is not necessary to buy such a device for yourself. But, you must take your own pictures, and transfer them to your own computer. Part of this assignment is deliberately left underspecified: you will have to find out how to best capture an image. Size, color, resolution, etc., are also up to you, and lighting, positioning, and background in particular will necessarily be learning experiences. You can also use any language of your choice. (Please see Courseworks for URLs to the four most common image processing packages for C++, Java, Matlab, and Python.) All of these do-it-yourself steps will help set you up for the final project, if you choose to do one.

Because we do not have the manpower to allow live demos of this or the other two assignments, your documentation will be critical. Especially since your choice of hardware, operating system, language, and image package are entirely up to you, the graders will be relying on your writeup, and not on your code.

To help structure the assignment, it is broken down into four steps with approximately equal credit, with the full assignment worth the 20 points toward your final grade.

1. (For 5 points): Domain engineering step.

First, to get a feel for what the images and the domain looks like, do the following.

If you are working with your own camera, you will need to have the appropriate drivers that dump camera input into files. You will also need to understand the format of these files, particularly if what you get is a compressed image. Please note that on most Unix machines, the command "convert", written up in the "man" pages, can help move images to and from the various formats. Other systems have similar utilities.

Once you have the ability to access the data of an image internal to your code, you should make sure your data is good. Capture an image of a body part--it should still be attached to the body it belongs to, of course--against the background of your choice. To view it, you can use the program called "display" or others on Unix systems, or even just Media Player or Paint if using Windows.

In general, you can capture your imagery and store them all in a directory for processing later: the system does not need to run in real time, unless you want to do so in the creativity step below.

2. (For 5 points): Data reduction step.

Find a way to manipulate the images to get a good binary image of the body part, by defining some region of the color space as "skin". Then determine the (x,y) coordinates of the center of mass of this binary field of "isSkin()" bits. You will also need some way of testing the image to see if it looks more like a fist or more like a splayed palm. If you wish, you can clean up the image in various ways first, but good domain engineering should make much of that unnecessary.

To document this step, you need to each of the images you have taken in some way, indicating both the "what" and the "where". For example, "fist, upper right" or "splay, lower center", or "unknown, left", would be acceptable. A good way to do this is to superimpose a checklist on the image, with the attributes that your system found checked, since this would also make explicit the vocabulary your are using for the grammar.

3. (For 5 points): Parsing and performance step.

Define the grammar for handling the symbolic data derived from the imagery. This would require a definition of tolerances (what, exactly, does "center" or "corner" mean?), and a clear documentation of why these decisions were made. If the grammar is more complex, for example, if it includes symbols that indicate "reset", or "erase the previous symbol", or various forms of "it is the Nth image that matters", then that too must be documented. How you communicate the grammar, and how you demonstrate that a sequence has been successful, is up to you.

You must run the grammar on at least 11 different sequences. To stress that visual interfaces are a measurable science, at least seven of these sequences should run correctly, that is, the system must give the proper answer. But at least four of these

sequences must indicate a system failure of some sort--truly, this will not be hard to generate. For the failures, which should include at least two false positives and two false negatives, you must explain what failed and why, and how you determined what caused the failure. Part of your score will depend on your ability to create convincing representative sequences (as if you were making a pitch to some venture capitalists) and analyses (as if you were training a support staff).

To document this step, you need to use a program to create printable--and probably reduced--images of your inputs. Or, if you are doing this at home or at work, use some other way of capturing the image or the screen shot of it. Thus, what you turn in must be not only your code and your system's decisions, e.g. "yes, that is the combination", or "no, that is not", or, "oops, I am confused", but also some record of the images used in that particular sequence. Note that the graders will examine these images to ensure that they look like they were generated and analyzed by your code.

Note that you must determine, as part of this assignment, what additional intermediate data can be used to justify your claim to the grades (your "customers") that your system is operating as designed. For example, one usually good choice is to provide images showing the binarization of your input into black-and-white all-or-nothing images.

4. (For 5 points): Creativity step.

The default definition of this problem is the one given above: two images, one with a fist in the center and one with a "five!" in a corner. Doing and documenting this perfectly with the first three steps gets 15 points to your course score. However, to get full credit for the assignment, you have to do something else in addition, and it is up to you to define it, design it, verify it, and document it. If you are uncertain about your creation, please use Piazza or email to check with the instructor and/or TAs.

For example, you can use the user's head, or head and hand in combination; you can allow the domain and lighting to vary in some way, perhaps by using the body to shield the light dynamically; you can use relative positions rather than absolute ones (for example, consecutive poses can be "bigger" or positions can be "further from the center", etc.); you can use more poses (e.g., closed fist, karate chop, flat palm fingers together, digitus impudicus, the "OK" circle sign); you can use two hands; you can have the grammar allow a dynamic redefinition of a new password; you can make the system real-time, and if you do, you can impose certain timing delays as part of the combination; you can use motion rather than position (e.g., "move to right then move to top", regardless of size or position), etc.

Whatever variation is chosen should affect the grammar for parsing, and it should be documented in the code. A warning: it is still necessary for the system to work at least seven times, and fail four times, so don't try to be *too* creative. A good system is interesting but *doable*. Nevertheless, part of your score will depend on how interesting your variation is. It is up to you to find the "sweet spot" of design for something that satisfies both the sales staff and the support staff.

5. General rules

Please note that whatever you do in code or in write-up, style counts. It is your obligation to write it all up so that the instructor and/or the TA can understand it on the *very first* try. It is permitted to submit only one document, which is a code listing that incorporates within it as comments your design approach, you algorithms, your database of imagery, your testing, and your analysis. However, it is probably better for you to do a separate illustrated report followed by a separate code listing, since that is the style we will require for the final project.

6. Checklist of deliverables

1. Your assignment is to be submitted to Courseworks electronically as a single pdf.
2. Your assignment consists first of a writeup with examples, then a listing of all the code used. Any code that you did not write yourself has to be *documented* with a statement about its *source* and an explanation of why you have *permission* to use it.
3. For all steps, your writeup explains: what design choices you made and why, what algorithms you used, what you observed in the output, and how well you believe your design worked.
4. For step 1, you should describe the following. How you captured the imagery: camera, lighting, objects, background. Your environment: hardware, OS, language, packages. Your library: how many images, what they were *intended* to capture, the format they are stored in.
5. For step 2, you should describe the vocabulary you used to describe your symbols (particularly if you extended them) for "what" and "where", including any margins of safety, and show how each image your collected is labeled.
6. For step 3, you should describe your grammar (particularly if you extended it), the successes, the failures, and a convincing explanation of the reason for the failures. You might also want to describe to the TAs how you designed or discovered the successes and failures.
7. For step 4, you should describe your extensions. You might also want to describe to the TAs how you decided on what was your personal "sweet spot".
8. Please use Piazza, early and often. It is a good way to explore and learn, from and with the rest of the class. And, you can use it anonymously, so you don’t have to be afraid of looking stupid or unprepared. But, please start the assignment early, so that you can give yourself enough time to do so.

Table of Contents