

Visual Interfaces to Computers: A Systems-Oriented First Course in Robust Control via Imagery

John R. Kender
Department of Computer Science
Columbia University
1214 Amsterdam, MC 0401
New York, NY 10027

Abstract

We present the rationale, description, and critique of a first course in image computing that is not a traditional computer vision principles-and-tools course. "Visual Interfaces to Computers" is instead complementary to standard Computer Vision, User Interface, and Graphics courses; in fact, VI:CV::UI:G. It is organized by case studies of robust working visual systems that use camera input for data or control information in service of higher user goals, such as GUI control, user identification, or automobile steering. Many CV scientific principles and engineering tools are therefore taught, as well as those of psychophysics, AI, and EE, but taught selectively and always within the context of total system design. Course content is derived from conference and journal articles and PhD theses, augmented with video tapes and real-time web site demos. Students do two homework assignments, one to design a "visual combination lock", and one to parse an image into English. They also do a final paper or project of their own choosing, often in teams of two, and often with surprisingly deep results. The course is assisted by a custom C-based tool kit, "XILite", a user-friendly (and bug-free) modification of Sun's X-windows Image Library for our lab's camera-equipped Sun workstations. The course has been offered twice to a wide audience to good reviews.

1 Introduction

We present the rationale, description, and critique of Columbia University Department of Computer Science course COMS W4735, Visual Interfaces to Computers, a three-point lecture and lab offering taken predominantly by computer science seniors as one of their "technical elective" courses to fulfill their requirements for the CS major. We first describe the motivation and pedagogical viewpoint behind its design, and show its relation to other courses. We then review and discuss its content and delivery. We describe the homework assignments and project required, noting the software that makes it possible, and indicate some of the notable systems that the students produced.

We conclude with a critical assessment of the course, from the standpoints of both the students and the instructor.

1.1 Rationale

As the processing, storage, and communication capacity of computers increase, it becomes possible to use imagery intelligently as a direct form of input. Images can be used for far more than simply transmitting pictures; they can be used to query, update, and govern systems. More importantly, they can now be used in these ways in both the classroom and in the undergraduate laboratory. This course explores the algorithms and systems, largely experimental, that understand and manipulate visual data directly, in ways that extend the power of keyboard, mouse, or even speech input. The emphasis is on full working systems in which visual input is a principal component, but not the ultimate system focus: all these systems do "real work" in service to some higher user goal. This visual modality of input is therefore best viewed as the complementary opposite of computer graphics. That is, it anticipates the near future in which the camera is as essential a part of a workstation environment as the display screen.

This course is therefore a complement to both CS W4731, Computer Vision, and CS W4170, User Interface Design. Our standard Computer Vision course, like most others, concentrates on the physical, psychophysical, geometrical, mathematical underpinnings of the science, but generally has little time for explorations of the design and performance of full systems. In this way, Computer Vision courses are not dissimilar in spirit to courses in Computer Graphics. On the other hand, User Interface courses tend to stress visual output to the near total exclusion of visual input; they tend toward mice input instead, with an occasional nod to speech input. The goals of this course is to do to Computer Vision what User Interfaces does to Graphics: it examines system level questions of user interaction and system performance, except that it does it all with respect to visual inputs rather than outputs. Like UI, this course can be designed and taught in parallel to its foun-

dation course; this course emphasizes breadth of coverage and significance of design decisions over depth of theory and exactness of mathematical detail.

However, of necessity, this course is research-oriented. There are currently relatively few existing commercial systems that use cameras for more than simple image capture and transmission. Trying to prize anything resembling a technical prospectus out of those few companies that make their income from visual input is virtually impossible, for obvious marketplace reasons. Nevertheless, the research-oriented flavor has been enthusiastically welcomed by the undergrads who have taken the course, since for many this has been their first exposure to experimental systems that haven't yet been codified in a textbook.

1.2 Relation to CS Curriculum

The course requires the same prerequisites as does the Computer Vision course, namely, CS W3139, Data Structures and Algorithms. For both courses this course and Computer Vision, the course CS W4701, Artificial Intelligence, is helpful but not required. Clearly, for this course, Computer Vision is also helpful, but it is not required; this course is self-contained.

The course is therefore potentially open to sophomores, but a review of the actual enrollments show that 70% of the students are undergraduates (or "older" students returning from the workplace to complete their undergraduate degree) and predominantly seniors; the remaining 30% are first year grad students. The course, offered twice so far (in Fall 1997 and Fall 1999), has drawn from 40 to 50 students each offering, about comparable to the course in Computer Vision. An additional 10 students in the distance-learning Master's program have taken it by tape in the Summers of 1998 and 1999, a relatively high number by the standards of the Columbia Video Network which offers it.

The lack of any subject-specific prerequisite implies that care has to be taken to explain from their beginnings all those system components and algorithms that are judged to be critical to system success. However, having a complete and final design structure of a working system in hand, this is eased substantially by being able to link such components and algorithms to each other, and to critically discuss their contributions to overall system performance. One drawback of this approach, other than the obvious one that many descriptions of such systems take a fair amount of basic principles and tools for granted, is that the coverage of the material of Computer Vision is both scatter-shot and somewhat somewhat repetitive. This is particularly true since working systems tend to place an emphasis on similar robust, often statistically-based, techniques, rather than on more varied abstract results, such as those of physics-based vision, or those of black box algorithms demonstrated on severely limited data. This does increase the burden on the

instructor; on the other hand, the instructor is free to select the systems and is free to emphasize those system components viewed to be the most critical.

2 Course Content and Delivery

The course is basically an introduction to the use of visual input as data and for control of computer systems. It is a survey and analysis of the architecture, algorithms, and underlying assumptions of commercial and research systems that: recognize and interpret human hand and body gestures, index into a database of pictures to retrieve related images, generate natural language descriptions of medical or map imagery, steer automobiles automatically, monitor large outdoor areas for types of activity, analyze imagery such as fingerprint or iris patterns for security data, and do other tasks. Along the way, the course explores, in varying depth, the foundations of these systems in human psychophysics, cognitive science, and artificial intelligence.

2.1 Scientific and Engineering Themes

Despite its case-study orientation, the course appears to be organized around, and continually repeats, five unifying principles about visual input systems.

- Visual data is coming to be an integral part of computing. The camera will become as essential as the keyboard, mouse, and microphone as in data and control device. Nor is visual data just something to compress and transmit; it can be used to make decisions through intelligent processing.
- Intelligent data reduction is absolutely necessary. Most visual input will be thrown away, and much of the rest is processed qualitatively, which often means statistically. Often, success is measured qualitatively too, against sometimes competing performance goals, such as accuracy versus precision.
- Deep spatial and temporal structures exists and can serve to guide data reductions. These structures are often based on psychophysics, and are sometimes explicitly formalized in user-defined or Hidden Markov Model-elicited grammars or rules of inference.
- Systems are based on scientific principles which unify phenomena into a small number of equivalent classes. Because of this, these systems can often predict future behavior for novel inputs, or under improved hardware or software. In short, these systems are extensible, predictable, measurable, and tunable.
- Systems are based on engineering design decisions which deliberately explore flexibility versus performance tradeoffs. Consequently, much emphasis is

placed on domain engineering, where the principles of physics and geometry do the work even before the computer gets the data, and on a study of human limits and preferences, which dictate the necessary precision and desired response times.

Because of the range of systems that can take visual input, the course eventually cover many of the tools and techniques of Computer Vision, but unequally and in varying depths. Among these are: visual sensor technology, projective geometry, stereo matching, color spaces and psychophysics, statistical thresholding, morphological operators, Hough transforms, edge following and fitting by snakes, geometric moments, shape approximations, neural nets, hidden markov models, eigenspaces, physical modeling for robotic control, affine transforms for spatial matching, dynamic programming and distance metrics for temporal matching, task grammars, spatial reasoning, applied graph search algorithms, iterative learning techniques, hardware assistance via DSPs, ROCs and other analytic measures of performance, and the ubiquitous concerns of domain engineering.

The emphasis in the course is neither on equations nor coding, and therefore tends towards comparative design overviews. In particular, several of the documents were deliberately chosen for the course because they provide complementary alternative designs. For example, Kjeldsen's gesture understanding system is in many ways a high-level and cognitive opposite to the low-level and sensor-based one of Segen and Kumar; the Navlab AI-based approach to vehicle guidance contrasts with the RALPH neural-net one; the paper of Wildes contains within it a comparison of his iris identification system with another by Daugman based on many different design choices; etc. Every system presents alternative ways in which massive amounts of data are reliably reduced, in spectrum, in space, in time, and ultimately, in meaning.

2.2 Details of the Content

There is as yet no single text that gathers and discusses such visual interfaces. The course materials therefore consist mainly of reprints of research articles. Several examples of working systems are available for exploration on the Web, on the sites of the authors of the articles. The course is given in one of the university's "electronic classrooms", so that in every third class or so, demos available over the web are used to illustrate the concepts. In general, we have found that if a demo is worth displaying once, it is worth displaying three times: the first as overview, the second for talking-over and interruption to elaborate ideas, and the third as summary. (The students have not found this wasteful of class time.)

We detail below the papers selected, and comment on their major contributions to the themes of the course. Not

all of the papers are perfectly well-chosen, but we summarize our thinking in why they have been selected and what the lectures attempt to bring out of them.

Papers on Visual Gesture Interpretation

- Frederik Kjeldsen, "Visual Interpretation of Hand Gestures as a Practical Interface Modality", Ph.D. Thesis, Columbia University, 1997. This is the paradigmatic paper for the course, and it takes about 7 of 28 lectures. It describes a system for using the visual interpretation of gesture to control a graphical user interface via a gesture grammar. The system splits low level hand tracking, recognized by binary threshold vision, from hand pose recognition by neural nets. The paper provides a overview of the psychological theory of gesture, discusses domain engineering issues including calibration and digital signal processing assists, details skin detection via training and normalization and thresholding, discusses models of the hand as a physical mass with damping and spring forces, derives an affine mapping of image coordinates to world coordinates, elaborates on the construction and training of neural nets, derives and tunes task grammars, fits evaluation data to known models of human performance such as Fitts' law. The work is accompanied by a video tape of the system in operation.
- J. Segen, and S. Kumar, "Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999. This is a type of foil to Kjeldsen, substituting a reduction of time-dense data for the reduction of space-dense data, and introduces projective geometry, boundary tracking, and extrema detection. It provides evaluation statistics, with applications to several domains, complete with good downloadable videos of the system in operation.
- Axel Mulder, "Human Movement Tracking Technology", Technical Report 94-1, Simon Fraser University School of Kinesiology, July 1994. This is an early overview of visual methods and assist hardware for human limb detection. It provides a taxonomy of ways to monitor body or body part position, and a review of various sensor technology: all from a non-CS perspective (note the "Kinesiology").
- Robb Lovell and John Mitchell, "Using Human Movement to Control Activities in Theatrical Environments", Technical Report, Institute for Studies in the Arts, Arizona State University, 1995. This paper reviews Videoplace, one of the oldest (1976)

and non-CS ways of using visual input to control artistic creations in a theater environment. Includes very straightforward low-level vision re-invented by a non-expert. It is interesting that even the control panel which programmed the system to relate gestures to theatrical experiences was itself controlled by gestures! It also reviews the Virtual Stage, a second old interface for the arts, which included a specific grammar to map gestures to actions. Vision ideas re-discovered and clearly described include "hotspots" which detect the presence or absence of a person, a body part, a primitive motion, or stillness, in order to control actions such as MIDIs, or robots directing the stage lights.

- Mark Torrance, "Advances in Human-Computer Interaction: The Intelligent Room", CHI'95 Research Symposium, 1995. This paper is mostly useful as a forecast of visual interfaces, particularly for the conference room of the future (virtual Post-It notes, etc.). Supplementary web material describes particular successes with body understanding, specifically stereo tracking of people within cluttered environments, plus some discussions about domain engineering and human limits. It comes with an interesting related web site with videos of children's visual interactions in several environments, and videos of gestural interactions with a "personalized athletic trainer" demonstrating the concept of temporal templates.
- Andrew Wilson, Aaron Bobick, Justine Cassel, "Temporal Classification of Natural Gesture and Application to Video Coding", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997. This paper provides a detailed analysis of a hidden grammar behind "natural" gestures, combining psychology and computer vision. It introduces the ideas of semantic video compression, using imagery reduced in size, time, and resolution, and the idea of the eigen decomposition of a mainly static video sequence. Hidden Markov Model technology is described as a way to extract a simple gesture grammar, with commentary on temporal modeling. Accompanying web site video has video illustrations of the techniques.

Papers on Visual Information Retrieval

- Amarnath Gupta and Ramesh Jain, "Visual Information Retrieval", Communications of the ACM, Vol. 40, No. 5, May 1997. This paper is a fine overview introducing the issues of encoding of image and video information for databases in terms of spectral, textu-

ral, spatial, and temporal features, and their combinations (for example, mixed shape graphs), and the issues of user interface, such as interactions through user sketching or shaping of regions and motions. It also details current deficiencies in visual interfaces to such systems, and the relative absence of agreement in how to measure the effectiveness of such systems.

- Myron Flickner, Harpreet Sawhney, Wayne Biblack, Jonathan Ashley, Quan Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, Peter Yanker, "Query by Image and Video Content: The QBIC System", IEEE Computer, September 1995. This is a fine overview of one of the first working visual interfaces to visual databases, and is complemented by one of the best web demos illustrating its concept of iterative and interactive user query refinement. It addresses standard low-level vision algorithms for color, texture models, and shape, including region-filling and snakes. It introduces several concepts in motion analysis, particularly with respect to video shot-detection.
- John Smith, Shih-Fu Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System", ACM Multimedia 96. This highly compact paper presents several engineering-based optimizations for visual retrieval, derived from statistical observations of color and shape distributions of imagery in visual databases. The interesting ideas of color set and 2-D shape strings are introduced as types of coarse-fine approximations to search; they also elaborate on color distance metrics. The paper concludes with a strong example of analytic performance evaluation via several interesting well-controlled simulations, as well as actual queries. Several web demos allow experimentation (but are starting to disappear; the student has graduated to industry).

Papers on Natural Language Interpretation

- Alicia Abella, "From Imagery to Saliency: Locative Expressions in Context", Ph.D. Thesis, Columbia University, 1995. This paper surveys the psychological, linguistic, and artificial intelligence literature on the deep semantic encodings of position, orientation, and spatial relationships of objects, then shows how objects in these circumstances are best to be approximated by best-fit ellipses. Prepositions are encoded as geometric relationships of these approximations, and are filtered by common-sense rules of spatial inference which reflect either a novice or expert level of user familiarity with the environment. These latter

map into two kinds of symbolic optimization: Quine-McClusky and Dijkstra algorithms, respectively. The paper concludes with empirical tests of the effectiveness of the generated descriptions, in the context of path descriptions in complex environments. A video tape illustrates the system at work.

Papers on Visual Guidance of Vehicles

- Charles Thorpe, Martial Hebert, Takeo Kanade, Steven Shafer, "Vision and Navigation for the Carnegie-Mellon Navlab", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 10, No. 3, May, 1988. An aging but useful model of the AI approach to visual interfaces. It gives a detailed system design with several interacting components, the lowest level of which models roads as statistical collections of colors on a plane that differ from off-road colors. The method uses Hough transform to detect road centerline, and a learning technique to update road color estimates. It has an discussion of coordination of modules, including those sensing the world via range sensors, but is mainly interesting in the anthropomorphic flavor of its design.
- Dean Pomerleau, "RALPH: Rapidly Adapting Lateral Position Handler", Technical Report, Carnegie-Mellon University, 1995. This paper serves as a wonder foil to Navlab, cutting out much of its system in order to optimize performance based on a few assumptions of the visual input. Highly redundant temporal data replaces the careful analyses of spatial data; the system clearly shows the impact of 10 years of hardware development. Nevertheless, retains a learning component to gracefully adjust to changing visual input, very similar in spirit to its predecessor.

Papers on Visual Surveillance

- Chris Stauffer, and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-time Tracking", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1999. Most notable for its excellent interactive on-line demo, this paper is an exercise in robust statistical modeling of absolutely uncontrolled visual input (i.e. continuous day and night imagery taken of the outdoors from an office). It is useful as a reprise of the concepts of statistical color learning methods.

Papers on Visual Methods for Biometrics

- Anil Jain, Lin Hong, Sharath Pankanti, and Ruud Bolle, "An Identity-Authentication System Using

Fingerprints", Proceedings of the IEEE, Vol. 85, No. 9, September 1997. This overview of visual input for identification via fingerprints also surveys (visual) biometrics as a field. Most of the paper is a detailed description of design choices made to optimize capture and processing, with heavy emphasis on data reduction ultimately leading to coarse-fine approach to visual matching of derived data, not unlike that use in the coarse-fine methods of Smith's visual database retrieval system. Empirical performance data is clearly described and evaluated, and contrasted to face recognition performance.

- Richard Wildes, "Iris Recognition: An Emerging Biometric Technology", Proceedings of the IEEE, Vol. 85, No. 9, September 1997. This prize-winning paper has a complete detailed description not only of its own design for this form of visual input, but a running comparison of these design choices with a competitor's. It discusses the necessary domain engineering required to obtain good iris imagery, and the Hough transforms necessary to model them. The system is notable in that different spatial resolutions of iris patterns have image features arranged in different spatial patterns (flecks, rings, radial lines); the paper discusses how these differing perceptions can be weighed together in a single evaluation using probabilistic measures of closeness. Performance is carefully analyzed.

3 Course Deliverables

Based on the experiences of the first offering of the course and student feedback, the second offering of the course was tuned to require the following workload, which was judged by the second offering's students to be approximately appropriate for the amount of course credit.

There are two homeworks worth 20% each, which are designed to be completed in the undergraduate workstation laboratory using (for Homework 1) custom camera-oriented software and (for Homework 2) custom data sets and modified version of the program, "xv". However, much of this is portable and it was permitted, and not unusual, for students to do the work on their own PCs. A deliberate feature of the two assignments was that 1/4 of the grade depended on the student's design of a creative extension of the basic requirements of the assignment.

Based on long experience, a course project proposal was required in order to ensure reasonable expectations by both instructor and students. To make it a serious attempt at design, it was graded, discussed privately, and worth 10%. Students had the option for the final project to do either a paper or a project, either singly or in teams of two. In the first offering, 1/3 of the class did a project, usually on

their home PCs, usually in teams of two. But by the second offering of the course, the number of projects rose to 2/3 of the class, again in teams, but mostly in the undergrad lab, in large part because of the availability of the custom software. Details of the course deliverables follow.

Homework 1 (worth 20% of the final grade) The following is an abridged description of this assignment, which comes early in the course, after the material on human gestural interfaces. It basically calls for the repeated calculation and verification of the center of mass of a binary thresholded image taken in a well-designed environment.

This homework is a theoretic and programming exploration of spatial description minimization and inference, given restricted visual data. 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. For example, the program can ask for two images of the user's hand on the table, and decide if the hand is in the upper left in the first image, then in the lower right in the second. Many other variants are possible, and, in fact, part of the grade depends on how creative the domain engineering and the grammar has been.

The assignment is done (preferentially) in the undergraduate workstation lab, in C or C++, on one of the dozen or so SUN workstations there that have a camera, using custom modified SunVideo image-handling software available on the class web page. The assignment has four steps of equal value:

- Domain engineering step. This requires selecting a body part and a background with good contrast, and capturing a color JPEG image. The imagery is viewed with the program, "xv", and converted to PPM format with the program, "convert".
- Data reduction step. Using the custom software with header files and method files that handle PPM format, find the center of mass of the body part.
- Parsing and performance step. Define a grammar for handling the symbolic data derived from the imagery, including various tolerances for its symbols. Run the grammar on at least 10 different sequences, at least seven of which succeed and at least three of which fail; explain the failures.
- Creativity step. To get full credit for the assignment, you have to do something of your own design beyond the upper left to lower right hand placement stated as default. For example, you can use the user's head, or head and hand in combination; you can allow the domain to vary in some way; the combination lock

could include a reset signal, or some decoy positions, or ways to signal which part of the sequence "really" should be processed; you can use relative positions rather than absolute ones; you can use poses that vary in area (e.g. closed fists, karate chops, flat palms), etc. Whatever variation is chosen should affect the grammar for parsing, and it should be documented in the code.

Homework 2 (worth 20% of the final grade) The following is an abridged description of this assignment, which comes mid-way in the course, after the material on natural language description of imagery. It basically calls for geometric calculations made on previously segmented and labeled binary information taken from a widely available map of the Columbia University main campus.

The goal of this assignment is to write a program that describes the location of a "visitor" to the "Columbia campus". More accurately, you are to modify the standard image-viewing program "xv" so that it displays a binary image of the main campus as seen from above, and which responds to any mouse click within it with an English description of where the visitor would be. For example, it could print out "Inside Low" or "Between Schapiro and Uris" or "South of Dodge and North of Journalism".

Much of the code will be found on the web page, as are the required images. Your main job is to encode the buildings as shapes, to determine their spatial relationships with the (x,y) coordinate that the mouse click will give you, and to filter out any relationships that are unnecessary because they can be easily inferred. For example, if the visitor is "South of Uris", it is not necessary to also say "South of Schapiro", as this can be inferred from the relation that "Uris is South of Schapiro".

The assignment has four steps of equal value:

- Basic infrastructure. Uncompress and compile the software package consisting of a modified version of the code for "xv" which you also will be modifying, a Makefile to generate your own custom version of "xv", some utility files, and three data files. Left clicking on the campus map displayed by this program should print to the standard output the x and y coordinates of the mouse location in the image. Change the code so that it prints out for each non-zero code integer that is clicked on the following building shape features: the (x,y) of the center of mass, the area, and the upper left and lower right coordinates of its minimum bounding rectangle (MBR), and the English string of its name.
- Creating simple spatial relations. Design and code the boolean-valued functions for In(x,y,N), North(x,y,N),

South(x,y,N), East(x,y,N), and West(x,y,N). Each function takes the coordinates of the mouse click (x,y), and the building code integer N, and returns true if the "visitor" is in the given relationship to the "building".

- Filter simple spatial relationships using transitivity. Use a graph representation of the relationships.
- Creativity step. You must add at least one more filterable trinary spatial relationship to the mix, and filter it. For example, you can do Between (which is static, involving one visitor position and two buildings), or do MovingAlongSideOf (which is dynamic, involving the visitor's present location, the visitor's prior location, and one building), or something else.

Paper/project proposal (worth 10% of the final grade)

The following is required approximately at midterm, and it is graded and discussed with the student in order to forestall three software engineering-related phenomena observed in prior project-oriented courses. First, most students attempt projects that are an order of magnitude larger in scope than is appropriate for a three-credit course: they basically try to do MS theses. Demanding some early design work and a forecast helps both instructor and student narrow the topic to one that is reasonable. Second, most students start too late, and most often to their own regret, as they generally enjoy doing a self-defined project. The proposal gets them started earlier. Third, teams of more than two simply don't work: they end up as pitched battles between the "doers" and the "slackers", who are variously defined by the participants. The proposal limits teams to two and demands twice as much from them.

Basically, the proposal asks for the specifications of the proposed project or paper; the anticipated methods, data, and results; and a sizable bibliography. The following is an abridged version of the call for proposals.

For a paper, of about 20 pages: Investigate and report on the state of the art in human and/or machine visual and/or spatial perception. You are not required to make an original contribution, although a well-written paper might see publication. You must show some evidence of an attempt at a synthesis, thus, the last four or five pages should serve future readers in the area as a personal guide. Prior course papers have included: analyses of current commercial systems (such as iris verification, fingerprint analysis, blood cell analysis, robot spacecraft), analyses of current research systems (gesture or sign language analysis, face recognition, gestural control of robots), or designs for new visual interface application areas (psychology of visual interfaces, musical instrument simulation, robotic delivery vehicles).

For a project, with a demo and writeup of about 10 pages: Program up a small version of some visual input processor of human data. Two dimensional data are acceptable. You can use a real-time camera, or just a real camera, or input that has been digitized off-line. The emphasis should be on the assertion of a small symbol: a binary decision like an authorization, or a token from a small set like a gesture class, perhaps with modifying scalar data, like certainty factors, positions, speeds, etc. Prior course projects have included: a visual burglar alarm (demonstrated in my own office), a system which detected class changing times from a video of campus pedestrian traffic, and two separate systems using "visual passwords" for computer logins.

For both: The proposal should describe the phenomena or program to be investigated, the limits you have placed on the investigation, at least five major references, a two-page sketch of the anticipated methods and results, and a description of any special help that may be required.

Project or paper ideas can be taken from, among others, the medical domain, the security and surveillance domain, the sporting world, the art world. Systems can include visual login control by face, motion, or pose; simple activity sensors; user reinforcement of visual database retrieval. [In the actual document, this list was considerably elaborated.]

Paper/project (worth 50% of the final grade) Due at final exam time (there was no course examination), was either the 20-page research paper surveying some aspect of visual interfaces, or the demonstrable working project documented with a 10-page write-up.

The papers tended to be straightforward, with two notable social phenomena taking place. The first was that few papers were written in teams, but those that were turned out to be inferior, even given their greater length, to those written singly. The second was that over the two course offerings and two summers, three students attempted to submit as their own work someone's Master's thesis; the average time to locate the original document on the Internet was 15 minutes, by using proper names from the bibliographies.

The projects in general were a great success on both the instructor and student sides, although many of them were so dependent on careful tuning to their environment that we made the rule that the systems must be developed in the same place they were to be demonstrated. Nonetheless, it was common to offer a badly failing project the option of a second chance later in exam week. The net result was that for the second offering of the course, virtually all projects were demonstrated in the undergraduate workstation laboratory, a semi-public place. Thus, many demos had an small informal audience of their peers; the added pressure, which was midway between a solo demonstration in the instructor's office and a full-blown show-and-tell in front of

the full class, seemed to be just the right amount for optimal focus and performance of the presenter(s).

Some of the best projects are already described in the call for proposal document above. But perhaps the most notable project was one in the second offering, which used a combination of background subtraction and skin color matching to derive the approximate outline of the user's head and (separately) the user's face. Then, based on the observation that most user's faces were rimmed with hair on the top and sides, it calculated from these separate images the relative displacements of their visual centers. By modeling the head as a sphere, this led to a simple calculation of head tilt and pan, roughly based on the projection of the vector from the center of the the head to the tip of the nose. Thus, in real time, the system was able to superimpose on the image of the user's head a smiley-faced sphere whose position, size, and two angles of orientation exactly mirrored the motions of the user in X, Y, Z, tilt, and pan. The project, impressive as it was, was done under pure "vanilla" circumstances: an otherwise academically undistinguished undergraduate, using only the hardware, software, and theory provided by the course.

4 Course Tools

The ability to offer the course depended on the availability of a large enough number of mutually interchangeable workstations with cameras. Our undergraduate workstation laboratory has about a dozen Sun stations equipped with cameras, and a (clumsy and buggy) software package from Sun for accessing data, called the XIL (X-windows Image Library). We were able to rework that library, and blend it with some additional image-referencing code from the Computer Vision course, so that a new package, called XILite, is now available. It handles image capture, window management, and mouse query within a window, obviating the need for special versions of "xv". It is also nicely compatible with C or C++, and is virtually bug-free. It was written by the TAs for the second offering of the class, two talented seniors, and is available on request.

In lieu of the lab hardware and software, we note that cheap cameras for use with laptops are surprisingly common, and that with Linux, most of the code written for the course is portable to them. Several of the class projects even in the second offering of the course were done on laptops, although only one project use as completely different image infrastructure.

5 Critique

The course has been offered twice. The first offering, offered in trial as a "Topics in CS" course, required only a course project, and was (surprisingly?) down-rated by the students for the lack of work during the semester. Nevertheless, the class was rated 4.3 out of 5.0 in quality. The

second offering addressed the workload issue, and provided custom software so that the majority of the students were able to do two lab assignments and a team project. This offering was rated 4.6 out of 5.0. Comments indicated that the course was "very interesting" and "enjoyable", and that students thought it should "become a mainstay in the curriculum."

This instructor feels that future offerings could include an exam, although it is not easy to write quantitative questions in a course that stresses qualitative design analyses and comparisons. Such an exam would have to be somewhat essay-like, except perhaps for some questions explicitly referring to the lab exercises.

It remains an open question how idiosyncratic the course is. The basic outlines are general enough that other instructors can tailor the systems, papers, algorithms, and principles that are covered to their own expertise and preferences. If the analogy to User Interfaces holds, there would be a lot of ways to teach visual systems without necessarily requiring a deep understanding of Computer Vision. This is both a strength and a weakness; probably any such course will necessarily be heavily influenced by the viewpoint of the instructor. The present course does lean heavily on two recent PhD theses advised by this instructor; however, any Computer Vision researcher with an interest and familiarity in psychophysics, psychology, artificial intelligence, and software engineering could probably replace these two theses with comparable investigations.

In general, this instructor is very pleased with about 2/3 of the references used, and will probably replace the other 1/3 in the third offering of the course. The students don't seem to mind being asked to pay a duplication fee for them in lieu of the cost of a textbook, and express interest and satisfaction at the experience of being lead through state-of-the-art research documents. More problematic is the associated web content, some of which have proven to be ephemeral, even over the course of only two years. It is imperative to teach the course in an electronic media-enhanced classroom, even if the students can play with the URLs privately, as many of the best points by talking over the demos.

The success of both offerings of the course has inspired the creation of a trial Visual Databases course next, based on the content of the middle lectures of this course, first offered in Spring 2000.