12-1-2003

Interactive Visual System

Arthur Evans
Pace University

John Sikorski
Pace University

Patricia Thomas
Pace University

Jie Jou
Rensselaer Polytechnic Institute

George Nagy
Rensselaer Polytechnic Institute

See next page for additional authors

Authors
Arthur Evans, John Sikorski, Patricia Thomas, Jie Jou, George Nagy, Sung-Hyuk Cha, and Charles C. Tappert

SCHOOL OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

TECHNICAL REPORT

Number 196, December 2003

Interactive Visual System

Arthur Evans
John Sikorski
Patricia Thomas
Jie Zou
George Nagy
Sung-Hyuk Cha
Charles Tappert

PACE UNIVERSITY

Arthur Evans, ***John Sikorski***, and ***Patricia Thomas*** began work on the Interactive Visual System as a development team during the 2002-2003 academic year in the Software Engineering Seminar, taught by ***Professor Tappert***, in Pace University's Master of Computer Science program. Each is now holds an MS in CS from Pace.

Jie Zou is a doctoral student at the Rensselaer Polytechnic Institute pursuing research under the guidance of ***Professor Nagy***.

Sung-Hyuk Cha is Assistant Professor Computer Science at Pace University, based in Westchester. Dr. Cha holds baccalaureate and masters degrees from Rutgers University and a doctorate in computer science from the State University of New York at Buffalo. He joined the faculty at Pace University in September, 2001.

Dr. Cha's research interests lie in the areas of distance measure and pattern matching algorithms, pattern recognition, image analysis, and machine intelligence and data mining.

Charles C. Tappert holds a B.S. in Engineering Sciences from Swarthmore College, an M.S. and Ph.D. in Electrical Engineering from Cornell University, and was a Fulbright Scholar. He worked at IBM for 26 years, mostly at the T.J. Watson Research Center, on speech recognition and pen computing. He has over 100 publications: journal articles, book chapters, conference papers, patents, and technical disclosures. While at IBM, he adjuncted at Pace University, SUNY Purchase, and North Carolina State University. He taught full-time at the U.S. Military Academy at West Point for seven years before joining Pace University in 2000 as Professor of Computer Science.

At Pace Dr. Tappert has been involved primarily in the development of the Doctorate of Professional Studies in Computing, for which he is the Associate Program Chairperson and in leading the software engineering seminar, which is the two-semester capstone course in the Master of Computer Science program. His research interests include pattern recognition, pen computing and voice applications, graphics, algorithms, artificial intelligence, human-computer interaction, and e-commerce.

INTERACTIVE VISUAL SYSTEM

Arthur Evans[1], John Sikorski[1], Patricia Thomas[1], Jie Zou[2], George Nagy[2], Sung-Hyuk Cha[1], Charles Tappert[1]

[1]Pace University, White Plains, NY [2]Rensselaer Polytechnic Institute, Troy, NY

ABSTRACT

Mobile computing devices are being endowed with ever-increasing functionality. To demonstrate the augmentation of human cognition in an interactive visual recognition task, we reengineered a PC-based system called CAVIAR (Computer Assisted Visual Interactive System) for a handheld computer with camera attachment. The resulting Interactive Visual System (IVS) exploits the pattern recognition capabilities of humans and the computational power of a computer to identify flowers based on features that are interactively extracted from an image and submitted for comparison to a species database. While IVS has similar functionality to that of CAVIAR, because it runs on a handheld computer, it offers complete portability for use in the field. We find that the handheld IVS and PC-based CAVIAR systems outperform humans alone both on speed and accuracy and machines alone on accuracy.

Computer vision systems still have difficulty in differentiating flowers from rabbits [Hopgood 03]. Flower guides are cumbersome, and they cannot guarantee that a layperson unfamiliar with flowers will achieve accurate classification. We observed that it takes two minutes fifteen seconds on average identify a flower using a field guide or key. We describe a prototype interactive visual system (IVS) for identifying flowers, and show that this camera-handheld combination, which takes advantage of both human cognitive abilities and computational and storage technologies, is much faster and more accurate than unassisted humans, and far more accurate than a computer alone.

IVS draws on innate perceptual ability to group "similar" regions, perceive approximate symmetries, outline objects, and recognize "significant" differences. It exploits computer capability of storing image-label pairs, quantifying features, and computing distances in an abstract feature space of shapes and colors. The IVS architecture was developed specifically for isolated object recognition in the field, where the time available for classifying each image is comparable to that of image acquisition. Whereas many of the more interesting mobile applications are communications based, IVS is autonomous.

(a) (b)

Figure 1. (a)The IVS in the field. (b) Close-up of IVS with attached camera.

The graphic user interface was designed, with the help of current user interface software tools [Myers01], for the small screen of the handheld. The interaction requires only pointing at designated parts of the image, and menu selection. The camera is controlled entirely by the pocket computer; the complete system is smaller and lighter than most flower guides, or most film cameras (Figures 1 (a) and (b)).

Additional samples of known species can be saved and eventually uploaded to a PC or the Web, and new species can be added to the database with little effort. As a result, the system *improves with use.*

OPERATION

User interaction is a defining feature of the IVS (Figure 2). Interaction with a handheld computer is similar to that with a desktop. The user points, and drags objects on the screen with a stylus, which is a better *direct action device* than a mouse [Schneiderman 83]. The display features are optimized to fit in a relatively small space (Figure 4). For detailed examination, thumbnail images can be zoomed to fill most of the screen.

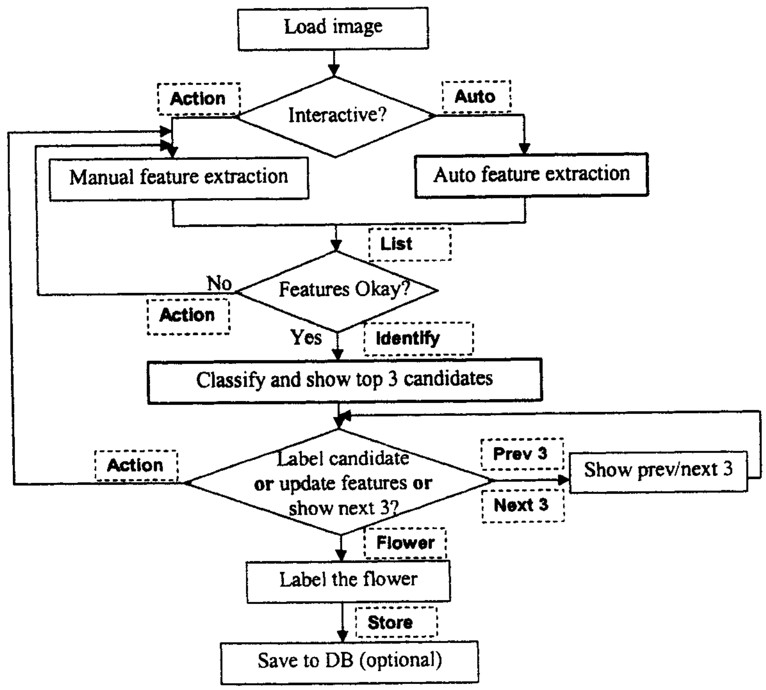
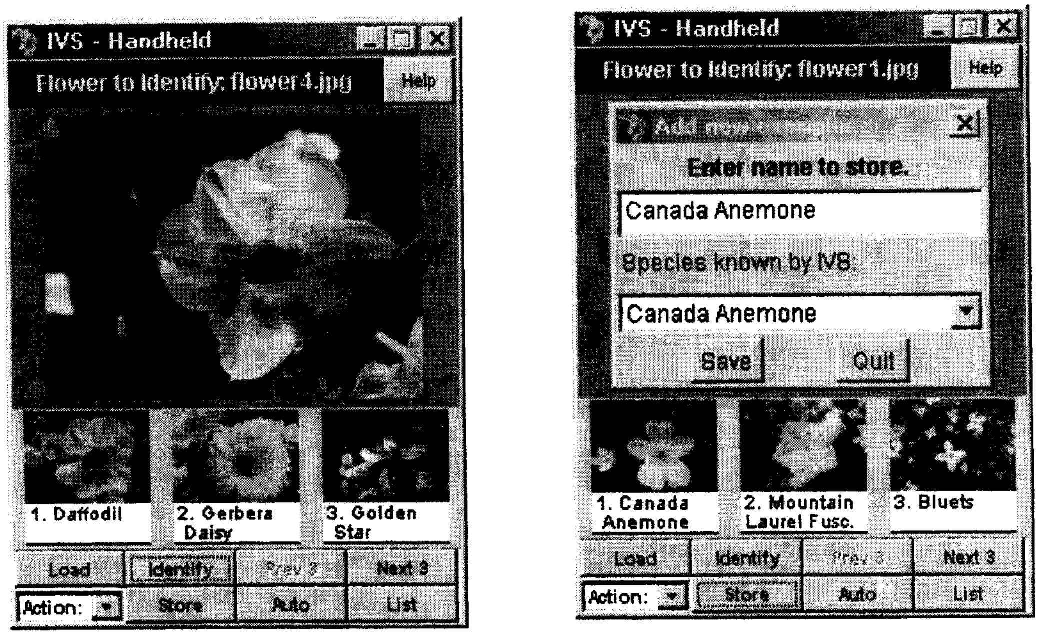


Figure 2. Diagram of IVS operation. Automated functions are shown in thick boxes, buttons with dotted lines.

The Load button loads a new image. The IVS resizes and centers it in a 240x150 pixel area without changing its aspect ratio. The Auto operation directs IVS to attempt to outline the flower by edge detection [Davis75, Sobel] and filtering, and to determine the color(s) and shape of the petals and of the center. A nearest-neighbors classifier [DHS02] ranks the species and produces an ordered thumbnail display (80 x 50 pixels each) of the top three candidates. The user may then click on one of the candidates to confirm classification, or inspect other classes by means of the Next 3 button. It is also possible to enter or correct the features with the stylus.

As an alternative to the Auto operation the user can interactively assist IVS to extract features selected from the Action drop-down menu at the bottom of the screen. Selection prompts a toggled panel displayed at the top of the screen. Then, the user either enters the required information, such as the number of petals, or directs IVS to record the feature measurement, such as the petal or center color from the location on the image touched by the user. The OK button confirms the action and stores the feature.



(a) (b)

Fig. 3. (a) IVS graphical user interface. (b) Adding a new exemplar to IVS

Once some or all the features have been interactively extracted or corrected, the Identify button directs IVS to identify the flower by comparing the extracted feature measurements to those of species in the database. New species can be added to the database via the Store button (Figure 3 (b)). Context-sensitive Help provides a brief guide for every option.

IMPLEMENTATION

Our platform is the Sharp Zaurus SL-5500 with a 200MHz processor, 64 MB RAM, and Compact Flash and Serial Device ports [IVS03]. A Sharp CE-AG06 camera attachment is inserted into the Compact Flash port, allowing direct capture of images. Pictures from other cameras can also be uploaded through this port.

Our choice was based on the availability of a camera attachment and on the trade-off between price and performance. Performance includes Java capabilities as well as processor speed since Java is considered more portable than C++. We chose the Zaurus over the Ipaq 38xx series and the Sony Clie because it offered a high-end processor, a full-featured Linux OS with command line utilities, and Personal Java capabilities, in addition to the camera attachment. We favored Personal Java for its greater flexibility over MIDP (Mobile Information Device Profile) that runs on low-end PDAs and cell phones. As an indication of the rapid advance of the technology, the Zaurus SL-5500 has already been superseded by the SL-5600 and the Ipaq 38xx series by the 39xx series, both with 400MHz processors.

The operating system is Embedix Linux with Personal Java support. All code was written to Personal Java specifications with code migration and extensibility in mind. The recognition engine is fully abstracted into generic and abstract classes, requiring only a few interfaces for data handling. Because generic classes exist to handle user-image interactions, GUI abstraction requires the implementation of only a few methods.

Based on Java 1.1.8, Personal Java predates current Java 2 Micro Edition (J2ME) specifications. J2ME provides Java functionality to a range of portable devices from smart cards to high-end handheld computers. J2ME was designed with the concepts of Configurations and Profiles. A Configuration targets a class of devices and takes into account their memory, display, connectivity and processing power. Profiles sit on top of Configurations, providing a set of libraries used by the programmer. The recently introduced Personal Profile, for high-end devices such as the Zaurus, provides the complete set of libraries available in the Java 2 Standard Edition. In time, Personal Profile is likely to replace Personal Java as the Java implementation of choice for this category of handhelds.

IVS extensions will likely be written to Personal Profile specification as it gains wider acceptance. The current code base will not be affected since all code written for Personal Java will run under Personal Profile. By designing to Java Standard Edition specifications, Personal Java and Personal Profile allow the same code to run equally well on handheld and desktop computers alike.

Because handhelds are only now becoming powerful enough to handle computationally intense applications like IVS, we had some implementation issues. With the Java Virtual Machine (JVM) starting with a default size of 3.5 MB we ran out of memory. Increasing the memory size to 7 MB not only cured the memory problems but also doubled the speed. Also, in contrast to CAVIAR, IVS avoids computer-intensive automatic segmentation and classification after each feature is entered, and leaves that decision instead to the discretion of the user.

DEVELOPMENT AND EVALUATION

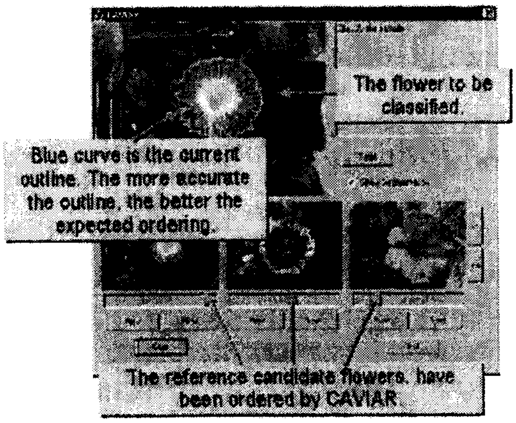


Fig. 4. CAVIAR graphical user interface.

Development and evaluation took place on a PC under the project name CAVIAR (Computer Assisted Visual InterActive Recognition) (Figure 4). The software was written in C++ with INTEL Open Source Computer Vision Library routines. It includes several options for experimenting with different families of objects (flowers, fruit, cell micrographs, and Chinese ideographs), automatic and inter-active segmentation methods (e.g., Electronic Scissors [MB98, Mortensen99]), some additional features, and a choice of experimental protocols. Most importantly, CAVIAR incorporates complete activity logging and statistical evaluation tools. Any testing session can be completely reconstructed and analyzed from the Excel worksheets created automatically from the log file.

For flower recognition, CAVIAR offers model-based automated and interactive segmentation, shape features based on moment invariants, hue and saturation histogram features, and the same Nearest Neighbor classifier as IVS [NZ02, ZN03]. Its browser can display not only alternative species, but also other instances of the same species. In anticipation of the port to a handheld, the CAVIAR window was restricted to 370 x 300 pixels. The simplicity of the interface draws on decades of HCI research by others [Myers99, DGHLRSSW02].

In order to compare human, machine, and interactive human-machine performance, we photographed several samples of each of 29 species at nature gardens and flower shows (Figure 5). Some of the species are quite similar (e.g. second row, 4 & 5; third row, 4 & 5), while different exemplars of the same species may exhibit marked differences in color and shape. This is by no means an easy recognition task for either laypersons or pattern recognition systems.

The experimental design devised for testing required each subject to recognize 174 new flowers. Five other instances of each species were used for estimating shape and color features for classification and some other parameters required for model-based segmentation. Test sessions can be configured to distinguish the effects of computer assistance.

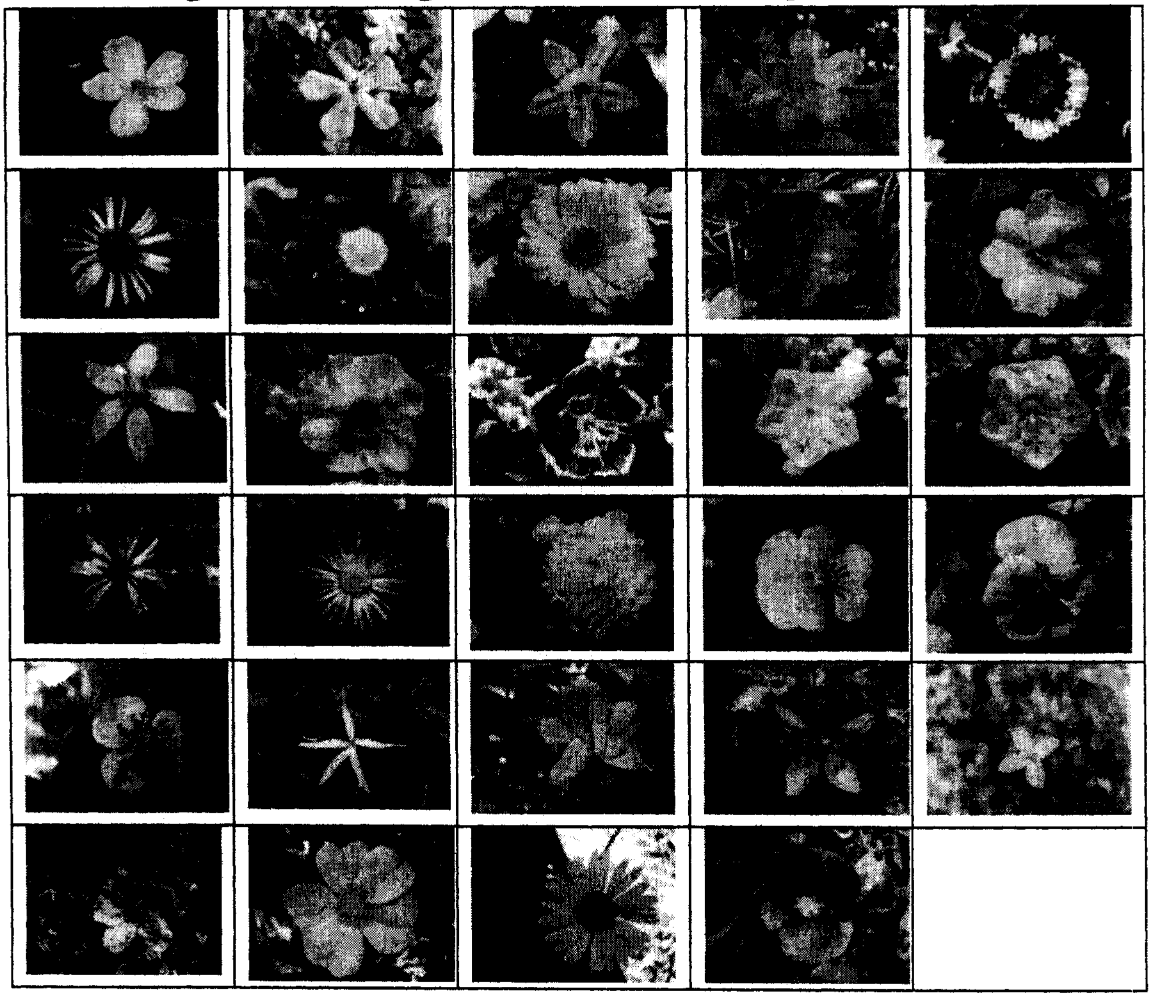


Figure 5. Exemplars of various species from our database.

1. *Computer-assisted (interactive human-machine) recognition.* As soon as a new flower is loaded, CAVIAR segments it as best it can, extracts the shape and color features, classifies the flower, and displays the top three candidates. The user has the option of confirming one of these candidates, looking at other exemplars or other species, or correcting the segmentation. After every correction click, CAVIAR reclassifies the flowers and offers the same options as before.

2. *Unassisted (human) recognition.* Here interactive segmentation, feature extraction and classification are disabled. Exemplars of each species are presented in the bottom panel in a fixed order, with similar species in adjacent panels. On the average, the user must inspect one half of the species before encountering the correct class.

The experiment was introduced by a short PowerPoint presentation that explained the objective of the experiment and the usage of the various buttons: no other training was conducted. The results of tests based on 18 naïve subjects (faculty, students, secretaries, significant others) are shown in Table I. The *completely automatic (machine only) recognition* scenario simply returns the top candidate that the machine initially obtains in the computer-assisted recognition scenario.

Table I – Performance of various recognition scenarios

Recognition method	Median error rate %	Median time per flower seconds
Computer-assisted recognition (12 subjects)	1.4	3.9
Unassisted recognition (6 subjects)	4.0	6.8
Completely automatic recognition	51.1	0.0

The times shown are human time only (on a PC the computer time required for each function is less than 0.5 seconds). These times are the medians over all 174 flowers of the test suite (which are presented in random order) and over all subjects. There is little variation from subject to subject, but the errors are concentrated on the similar classes. Not shown here but noteworthy is that the time-per-flower, for assisted recognition, decreased by 50% during the second half of the test. The slower initial pace is due to the time required by the subjects to reach a satisfactory balance between segmentation and browsing.

The automatic classifier is very effective in eliminating unlikely candidates from the top few even when it is wrong on the top candidate. Therefore a few segmentation or browsing clicks are almost always sufficient to bring the correct candidate to the display.

Experimentation with CAVIAR, which produced several counterintuitive results, helped to avoid some false starts in the implementation of IVS. One of our goals was to port CAVIAR's most effective features to the handheld while considering potential architectural limitations. IVS is a computationally intensive application, pushing the handheld system to its limits. The same code running on a desktop computer is more than an order of magnitude faster than on the handheld. Since handheld processing power is significantly less than that of a desktop, we recorded results with IVS on a desktop as a system comparison. The test system was a Dell Optiplex GX240 with a 2000 MHz Pentium IV processor. Having a faster processor and possibly a more optimized JVM, the desktop times were significantly less than on the handheld. As expected, accuracies for both systems are similar.

The user has the option to extract color features automatically and verify their accuracy or skip auto extraction entirely. The user initiates classification when he or she believes that enough features have been identified. Subjects quickly learn to identify which features will more likely lead to accurate identification. Because of the efficient Java implementation and simplified segmentation, the overall time for computer-assisted recognition on the handheld is only about three times slower than on CAVIAR. The machine-time component should drop below human reaction time in 2-3 years.

EXTENSIONS

In any given geographic region, season, and elevation, it is uncommon to encounter more than thirty different species of wildflowers. Nevertheless, we are expanding our database to many more species. We expect to add further features, such as sepal length, leaf shape and size, height of the plant, and bloom density. We intend to derive scale information from the recorded camera focus, and to implement some type of normalization for illumination.

Mobile, interactive visual recognition is likely to prove applicable to other families of "natural" objects, such as fruit, trees, beetles, and traces of disease or stress on specific plants. It may also be useful in the final stages of some biometric applications, especially face recognition.

We don't consider IVS as just a temporary solution until computer vision gets to be "good enough." Its potentially most significant applications may well be in education and training. After sufficient experience with an IVS, students could learn to classify objects of a given family as accurately and as fast as a domain expert and, like the expert, rarely require computer help. The IVS is an example of human cognitive augmentation.

ACKNOWLEDGMENTS

We are grateful for expert advice to City Naturalist Rebecca Seth (Lincoln, NE), New York State Botanist Dr. Richard Mitchell (Albany, NY), Prof. Angelo Marcelli (University of Salerno), and Prof. Robert Ingalls (Rensselaer Polytechnic Institute). We thank Adam Callahan for photographing many flowers and fruit.

REFERENCES

[IVS03] Evans, Sikorski, and Thomas, Interactive Visual System, *CSIS Technical Report,* Pace University, White Plains, May 2003.

[Davis75] L.S. Davis, A survey of edge detection techniques, *Computer Graphics and Image Processing,* Vol 4, pp. 248-270, 1975.

[DGHLRSSW02] Z. Duric, W.D. Gray, R. Heishman, F. Li, A. Rosenfeld, M.J. Schoelles, C. Chunn, and H. Wechsler, Integrating Perceptual and Cognitive Modeling for Adaptive and Human-Computer Interaction, *Proceedings of the IEEE,* Vol. 90, no. 7, pp. 1272-1289, (Special issue on technology and tools for visual perception, C. Guerra and V. Cantoni, editors), July 2002.

[DHS02] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Wiley, 2000.

[Hopgood93] Adrian A. Hopgood, Artificial Intelligence: Hype or Reality? IEEE *Computer* Vol. 36, no. 5, pp. 24-28, May 2003.

[MB98] E.N. Mortensen and W.A. Barrett, Interactive Segmentation with Intelligent Scissors, *Graphical Models and Image Processing,* vol. 60, No. 5, pp. 349-384, 1998.

[Myers 99] B. A. Myers, A Brief History of Human Computer Interaction Technology, *ACM interactions*, Vol. 5, no. 2, March, pp. 44-54, 1998.

[Myers 01] B. Myers, Scott E. Hudson, and Randy Pausch, Past, Present and Future of User Interface Software Tools, in John M. Carroll, ed. *HCI In the New Millennium,* New York: ACM Press, Addison-Wesley, pp. 213-233, 2001.

[NZ02] G. Nagy, J. Zou, Interactive Visual Pattern Recognition, *Proc. Int. Conf. Pattern Recognition XIV,* vol. 2, pp. 478-481, Quebec City, 2002.

[Schneiderman 83] B. Schneiderman, Direct manipulation: A step beyond programming languages, IEEE *Computer*, vol. 16, no. 8, pp. 57-69, 1983.

[Sobel] http://www.cee.hw.ac.uk/hipr/html/sobel.html accessed December 2002.

[ZN03] J. Zou, G. Nagy, A Trainable Object Segmentation Algorithm, *Procs. ICCV*-03, Beijing, if accepted, RPI tech report otherwise.

School of Computer Science and Information Systems
Pace University
Technical Report Series

The School of Computer Science and Information Systems, through the Technical Report Series, provides members of the community an opportunity to disseminate the results of their research by publishing monographs, working papers, and tutorials. *Technical Reports* is a place where scholarly striving is respected.

All preprints and recent reprints are requested and accepted. New manuscripts are read by two members of the editorial board; the editor decides upon publication. Authors, please note that production is Xerographic from the pages you have submitted. Statements of policy and mission may be found in issues #29 (April 1990) and #34 (September 1990).

Please direct submissions as well as requests for single copies to:

Allen Stix
School of CS & IS - Suite 412 Graduate Center
Pace University
1 Martine Avenue
White Plains, NY 10606-1932