CSE 642 Techniques in AI: Vision for HCI

SUNY Buffalo

Syllabus for Fall 2009

Instructor: Jason Corso (jcorso)

Course Webpage: http://www.cse.buffalo.edu/~jcorso/t/CSE642/.

Syllabus: http://www.cse.buffalo.edu/~jcorso/t/CSE642/syllabus.pdf.

Meeting Times: MW 3-4:20

Location: M Bell 340 and W Bell 224
We will meet directly in a computer lab for one of the two class meetings each week. The instructor will provide both a working software framework and a physical camera (which the student can take with them for the entire semester). The instructor will also provide a list of suitable projects at the beginning of the semester, but students are encouraged to design their own projects (subject to instructor approval). At the end of the semester, we will host an open-house demo of the projects.

Office Hours: Monday 1-3 or by appointment

Main Course Material

Course Overview: The promise of computer vision for enabling natural human-machine interfaces is great: vision-based interfaces would allow unencumbered, large-scale spatial motion; they could make use of hand gestures, movements, or other similar natural input means; and video itself is passive, cheap, and nearly ubiquitous. In the simplest case, tracked hand motion and gesture recognition could replace the mouse in traditional applications, but, computer vision offers the additional possibility of using both hands simultaneously, using the face, incorporating multiple users concurrently, etc. In this course, we will develop these ideas from both a theoretical and a practical perspective. From the theoretical side, we will cover ideas ranging from interaction paradigms suitable for vision-based interfaces to mathematical models for tracking (e.g., particle filtering), modeling high-dimensional articulated objects, and modeling a grammar of interaction, as well as algorithms for rapid and real-time inference suitable for interaction scenarios. From the practical side, we will each build (in pairs) an actual working vision-based interactive system. Each project must "close the loop" and be integrated directly into an interactive computer system (e.g., sort photos on the screen by grabbing them with each hand and moving them around). During the semester, very practical-minded topics such as interactive system design and architecture, debugging programs that process high-dimensional video data, and program optimization will be discussed alongside the underlying computer vision theory.

Course Project: Each student will be required to implement a course project that is either a direct implementation of a method discussed during the semester or new research in Bayesian vision. A paper describing the project is required at the end of the semester (8-10 pages two column IEEE format) and we will have an open-house poster session to present the projects. Working project demos are suggested but not required for the poster session.

Prerequisites: It is assumed that the students have taken introductory courses in machine learning/pattern recognition (CSE 555 or 574), and computer vision (CSE 573). Permission of the instructor is required if these prerequisites have not been met. Working knowledge of C/C++ is a must.

Course Goals: Students taking this course will learn the major ideas in the theory of computer vision for building natural human-machine interfaces, and they will gain practical experience in building large, real-time computer vision systems. These are challenging and complementary goals, unmatched in most vision curricula.

Textbooks: There is no textbook for the course. The instructor will prepare lectures and hand-out relevant academic articles. The OpenCV book (*Learning OpenCV* by Bradski and Kaehler) is suggested as OpenCV will serve as the core software library on which the projects will be built.

Grading: Grading will be discussed in the class. Basically, the grade will be a function of the students having achieved the course goals and it will ultimately be measured by the success of his or her project. Quantitatively speaking, the project will cover about 75% of the grade and the remaining 25% will be filled will class participation and the occasional minor homework/quiz.

Programming Language: C++ is the programming language.

Course Outline

The outline is presented in an idea-centric scheme rather than an application-centric one. This is an initial run; as the course evolves, this will be refined. During each lecture, we will ground the theoretical ideas with practical examples and real-world applications.

1. Computer Vision Background Material
   (a) Light-Models, Image Acquisition, and Backgrounds (Trucco & Verri, 1998; Forsyth & Ponce, 2003)
      - Adaptive Background Subtraction (Stauffer & Grimson, 1999)

   (b) Camera Geometry, Calibration, and Stereo (Trucco & Verri, 1998; Hartley & Zisserman, 2004; Zhang, 2000)
   (c) Interest Operators and Feature Extraction (There are many here, we survey some) (Mikolajczyk & Schmid, 2004; Harris & Stephens, 1988; Fraundorfer & Bischof, 2003; Lowe, 2004; Corso & Hager, 2005; Kadir & Brady, 2001; Matas *et al.*, 2002)
2. Tracking
   (a) Feature Tracking (Shi & Tomasi, 1994)
      - Selecting features (Collins & Liu, 2003)
      - Nose Tracking - Nouse (Gorodnichy *et al.*, 2002)

   (b) Direct Method (Image-Based Tracking) (Lucas & Kanade, 1981; Baker & Matthews, 2004)
      - Head Tracking (Hager & Belhumeur, 1998)
      - Face feature tracking (Black & Yacoob, 1997)

   (c) Model-based Tracking
      - Hand tracking (Rehg & Kanade, 1994)
      - Body tracking (Gavrila & Davis, 1995)
      - Fingertip Tracking (Hardenberg & Berard, 2001)

   (d) Exemplar-based Tracking
      - Toyama and Blake method (Toyama & Blake, 2002)

   (e) Kalman Filtering (Kalman, 1960; Welch & Bishop, 95)
      - People tracking (Wren *et al.*, 1997)
      - Multiple hand tracking (Utsumi & Ohya, 1999)

   (f) Particle Filtering (Arulampalam *et al.*, 2002; Isard & Blake, 1998)
      - Hand tracking (Isard & Blake, 1998)

   (g) Mean shift / Kernel-based tracking (Collins, 2003; Comaniciu *et al.*, 2003; Comaniciu & Meer, 2002; Cheng, 1995; Hager *et al.*, 2004)
3. Recognition
   (a) Space-time Methods
      i. Discriminant Analysis
         - ASL Recognition (Cui & Weng, 2000)

      ii. State-based methods (Bobick & Wilson, 1997)
      iii. Templates/Exemplar Methods
         - Space-Time Gestures (Darrell & Pentland, 1993; Darrell *et al.*, 1996)
         - Body Motion Recognition (Bobick & Davis, 2001)

      iv. Bayesian Network Classifiers
         - Expression Recognition (Cohen *et al.*, 2003)

   (b) Temporal Methods

   i. Model-Based Methods
      - Markov-Dynamic Networks (Pentland & Liu, 1999)
   ii. Hidden Markov Models (Rabiner, 1989)
      - Parametric HMMs for Gesture Recognition (Wilson & Bobick, 1999)
      - Learning Variable-Length Markov Models (Galata *et al.*, 2001)
   iii. Dynamic Bayesian Networks (Murphy, 2002; Ghahramani, 1998)
      - Facial Activity Recognition (Tong *et al.*, 2007)
      - Hand Gesture Sequence Recognition (Corso *et al.*, 2005)
   iv. Stochastic Parsing (Ivanov & Bobick, 2000)

4. Localized Interaction
   (a) VICs framework (Ye *et al.*, 2004; Corso *et al.*, 2008)
   (b) Dynamically Reconfigurable UIs (Kjeldsen *et al.*, 2003)
   (c) Tracking the scene
      - Planar structures (Simon *et al.*, 2000; Zhang *et al.*, 2001)
      - Planar structures from stereo (Corso *et al.*, 2003a)
      - Deformable surfaces from stereo (Corso *et al.*, 2003b)

Project

The goal of the project is threefold: first, we want to put the theory of Vision for HCI to work; second, we want to build substantial student experience in programming and debugging a large system with images, video, and interactive/near-real-time constraints (on a Unix system); third, we want to give the students a chance to explore the fun and exciting area of building next generation and useful user interfaces.

The projects will be tackled in pairs (if there is an odd-student in the course then we can have a student work alone or have one group of three). At the beginning of the semester, the instructor will hand out commodity webcams to each student (on loan from his lab). If the students need additional hardware to complete their project, they should discuss it with the instructor as early as possible. (Erector sets will be available in the instructor's lab for some simple construction tasks.)

Project Software, Platform, and Requirements

All projects will be performed primarily in the Bell 340 lab working environment, which is a Linux environment. Students will be able to gain important working knowledge of Linux systems from this experience. However, the project will be built on the following open-source cross-platform libraries and the students should hence pursue cross-platform project code:

1. OpenCV (`http://opencv.willowgarage.com/`) is a widely used open-source computer vision toolkit. It contains the necessary low-level classes like images, matrices, and capture devices; algorithms like linear system solving, and frequency transformations; as well as many advanced computer vision algorithms like face detection with boosting. **Constraint:** the temptation will be to design a project and build it by stitching together bits and pieces of OpenCV; this is not permitted. OpenCV is there to provide you with a substantial working base to build real software. Each project must include substantial new algorithm development on top of the existing methods in OpenCV; this will be a challenge.
2. wxWidgets (`http://www.wxwidgets.org/`) is a cross-platform GUI library. If a project demands advanced interface tools that HighGUI does not provide (such as buttons!), then the students must use wxWidgets. **Exception:** if the project involves extending an existing open-source piece of software to add new computer vision-based functionality, such as Firefox, then wxWidgets need not be used. Such exceptions must first be discussed with the instructor at the beginning of the semester.

Both of these libraries and additional ones have been installed by the CSE IT staff and are available on the CSE and Bell 340 networks. Although the students may use an IDE of their choosing if they so wish, the projects must provide a standard Makefile solution for compiling on the instructor's machine (paths may be manually configured). The projects must use a revision control system of the students choosing (either CVS or SVN).

List of Possible Projects

No two teams may tackle the same project. First come are first served. See project schedule below. Teams and project selection are due in class by Sept. 21 at which time project work will begin in the lab.

For reasons that may or may not be obvious, the actual project list will be distributed via hard-copy on the first day of class and not posted to the website.

Project Schedule

The project schedule is largely student organized within the following date framework:

Sept. 21 — Teams are formed and projects are selected.

Sept. 28 — Requirements document must be submitted to the instructor in class. We will discuss what this means in class. Basically, this is a less than three page document that describes the project goals, anticipated methods to be used, clearly states what the new research developments will be (e.g., implementation beyond those in OpenCV, or altogether new ideas), and defines a schedule that includes two milestone deliverables and the ultimate end-game deliverables.

Oct. 26 — Milestone 1 Due. Content is specified by the students via their requirements document.

Nov. 23 — Milestone 2 Due. Content is specified by the students via their requirements document.

Dec. 14+ — Demo day. Final deliverables and project report are due. (Specific date of demo-day is yet to be determined, but it will be on or after the 14th.)

Project Write-Up

The project write-up is a standard two-column IEEE conference format at maximum of 8 pages. The students can use the CVPR style files. It should be approached as a standard paper containing introduction and related work, methodology, results, and discussion.

Project Evaluation and End-Goals

The projects can have multiple non-exclusive end-points, and it is the instructor's goal that all projects achieve all of these end-points. At the very least, a project must achieve at least one of these end-points to be considered a success. The three end-points are

1. Project yields a working interactive demo at demo-day that can be used by all or most of the demo visitors. Demo visitors will be asked to to informally evaluate the projects and rank them.
2. Project yields a nice demo video that is uploaded and well-received to YouTube.
3. Project yields a research manuscript that is submitted to a good conference in vision or interfaces such as CVPR, or CHI. Since paper reviewing will not happen until after the end of the semester, the instructor has the ultimate say in judging the quality of the manuscript.

In addition to these possible end-states, the final project deliverable will include the project report, all of the project code, and Doxygen generated documentation. These can be emailed to the instructor (if the size is not prohibitive) or submitted via CD/key.

Here is how the project scores will be computed:

How to get an A: Achieve two or three of the end-points mentioned above. Coherent code and report.

How to get a B: Achieve one of the three end-points mentioned above. Coherent code and report.

How to get a C: Clear effort has been made but non of the three end-points mentioned above have been achieved; project does some thing but is incomplete. Coherent code and report.

How to get a D: Unclear how much effort was put in; project does not do anything.

How to get an F: Students have not done anything. There is no project to speak of.

In most cases, both partners of a project team will receive the same grade, but if there are cases of clear irresponsibility of a project team member, these will be addressed on an individual basis with the instructor and the project team.

References

Arulampalam, S., Maskell, S., Gordon, N., & Clapp, T. 2002. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, **50**, 174–188. 2

Baker, S., & Matthews, I. 2004. Lucas-Kanade 20 Years On: A Unifying Framework. *International Journal of Computer Vision*, **56**(3), 221–255. 2

Black, M. J., & Yacoob, Y. 1997. Tracking and Recognizing Rigid and Non-Rigid Facial Motions using Local Parametric Models of Image Motion. *International Journal of Computer Vision*, **25**(1), 23–48. 2

Bobick, A., & Davis, J. 2001. The Recognition of Human Movement Using Temporal Templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(3), 257–267. 2

Bobick, A. F., & Wilson, A. 1997. A State-based Approach to the Representation and Recognition of Gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(12), 1325–1337. 2

Cheng, Y. 1995. Mean Shift, Mode Seeking and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(8), 790–799. 2

Cohen, I., Sebe, N., Cozman, F. G., Cirelo, M. C., & Huang, T. S. 2003. Learning Bayesian Network Classifiers for Facial Expression Recognition using both Labeled and Unlabeled Data. *In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2

Collins, R. 2003. Mean-Shift Blob Tracking Through Scale Space. *In: IEEE Conference on Computer Vision and Pattern Recognition*. 2

Collins, R., & Liu, Y. 2003. On-Line Selection of Discriminative Tracking Features. *Pages 346–352 of: International Conference on Computer Vision*, vol. 1. 2

Comaniciu, D., & Meer, P. 2002. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(5), 603–619. 2

Comaniciu, D., Ramesh, V., & Meer, P. 2003. Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(5), 564–577. 2

Corso, J. J., & Hager, G. D. 2005. Coherent Regions for Concise and Stable Image Description. *In: IEEE Conference on Computer Vision and Pattern Recognition*. 2

Corso, J. J., Burschka, D., & Hager, G. D. 2003a. Direct Plane Tracking in Stereo Image for Mobile Navigation. *In: Proceedings of International Conference on Robotics and Automation*. 3

Corso, J. J., Ramey, N., & Hager, G. D. 2003b. *Stereo-Based Direct Surface Tracking with Deformable Parametric Models*. Tech. rept. The Johns Hopkins University. 3

Corso, J. J., Ye, G., & Hager, G. D. 2005. Analysis of Composite Gestures with a Coherent Probabilistic Graphical Model. *Virtual Reality*, **8**(4), 242–252. 3

Corso, J. J., Ye, G., Burschka, D., & Hager, G. D. 2008. A Practical Paradigm and Platform for Video-Based Human-Computer Interaction. *IEEE Computer*, **42**(5), 48–55. 3

Cui, Y., & Weng, J. 2000. Appearance-Based Hand Sign Recognition from Intensity Image Sequences. *Computer Vision and Image Understanding*, **78**, 157–176. 2

Darrell, T., & Pentland, A. 1993. Space-Time Gestures. *Pages 335–340 of: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 2

Darrell, T. J., Essa, I. A., & Pentland, A. P. 1996. Task-specific Gesture Analysis in Real-Time using Interpolated Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**(1236–1242). 2

Forsyth, D. A., & Ponce, J. 2003. *Computer Vision*. Prentice Hall. 2

Fraundorfer, F., & Bischof, H. 2003. Detecting Distinguished Regions By Saliency. *Pages 208–215 of: Scandinavian Conference on Image Analysis*. 2

Galata, A., Johnson, N., & Hogg, D. 2001. Learning Variable-Length Markov Models of Behavior. *Computer Vision and Image Understanding*, **83**(1), 398–413. 3

Gavrila, D. M., & Davis, L. S. 1995. Towards 3-D Model-Based Tracking and Recognition of Human Movement: A Multi-View Approach. *International Conference on Automatic Face and Gesture Recognition*. 2

Ghahramani, Z. 1998. Learning Dynamic Bayesian Networks. *Pages 168–197 of:* Giles, C. L., & Gori, M. (eds), *Adaptive Processing of Sequences and Data Structures*, vol. Lecture Notes in Artificial Intelligence. 3

Gorodnichy, D. O., Malik, S., & Roth, G. 2002. Nouse Use Your Nose as a Mouse - A New Technology for Hands-free Games and Interfaces. *Pages 354–361 of: International Conference on Vision Interface*. 2

Hager, G. D., & Belhumeur, P. N. 1998. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(10), 1025–1039. 2

Hager, G. D., Dewan, M., & Stewart, C. V. 2004. Multiple Kernel Tracking with SSD. *IEEE Conference on Computer Vision and Pattern Recognition*. 2

Hardenberg, C. von, & Berard, F. 2001. Bare-Hand Human-Computer Interaction. *Pages 113–120 of: Perceptual User Interfaces*. 2

Harris, C., & Stephens, M. J. 1988. A Combined Corner and Edge Detector. *Pages 147–151 of: ALVEY Vision Conference*. 2

Hartley, R. I., & Zisserman, A. 2004. *Multiple View Geometry in Computer Vision*. Second edn. Cambridge University Press, ISBN: 0521540518. 2

Isard, M., & Blake, A. 1998. CONDENSATION – conditional density propagation for visual tracking. *International Journal of Computer Vision*, **29**(1), 5–28. 2

Ivanov, Y. A., & Bobick, A. F. 2000. Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(8), 852–872. 3

Kadir, T., & Brady, M. 2001. Saliency, Scale and Image Description. *International Journal of Computer Vision*, **43**(2), 83–105. 2

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASMA; Journal of Basic Engineering*, **82**, 35–45. 2

Kjeldsen, R., Levas, A., & Pinhanez, C. 2003. Dynamically Reconfigurable Vision-Based User Interfaces. *Pages 323–332 of: International Conference on Computer Vision Systems*. 3

Lowe, D. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, **60**(2), 91–110. 2

Lucas, B.D., & Kanade, Takeo. 1981. An iterative Image Registration Technique with an Application to Stereo Vision. *Pages 674–679 of: International Joint Conference on Artificial Intelligence*. 2

Matas, J., Chum, O., Urban, M., & Pajdla, T. 2002. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *In: British Machine Vision Conference*. 2

Mikolajczyk, K., & Schmid, C. 2004. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, **60**(1), 63–86. 2

Murphy, K. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California at Berkeley, Computer Science Division. 3

Pentland, A., & Liu, A. 1999. Modeling and Prediciton of Human Behavior. *Neural Computation*, **11**(1), 229–242. 3

Rabiner, L. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, **77**(2), 257–286. 3

Rehg, J. M., & Kanade, Takeo. 1994. Visual Tracking of High DOF Articulated Structures: An Application to Human Hand Tracking. *Pages 35–46 of: European Conference on Computer Vision*, vol. 2. 2

Shi, Jianbo, & Tomasi, Carlo. 1994. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, **0**(0). 2

Simon, G., Fitzgibbon, A. W., & Zisserman, A. 2000. Markerless Tracking Using Planar Structures in the Scene. *Pages 137–146 of: International Symposium on Augmented Reality*. 3

Stauffer, C., & Grimson, W.E.L. 1999. Adaptive Background Mixture Modeling for Real-Time Tracking. *Pages 246–252 of: IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2. 2

Tong, Y., Liao, W., Xue, Z., & Ji, Q. 2007. A Unified Probabilistic Framework for Facial Activity Modeling and Understanding. *In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 3

Toyama, K., & Blake, A. 2002. Probabilistic Tracking with Exemplars in a Metric Space. *International Journal of Computer Vision*, **48**(1), 9–19. 2

Trucco, E., & Verri, A. 1998. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall. 2

Utsumi, A., & Ohya, J. 1999. Multiple-Hand-Gesture Tracking Using Multiple Cameras. *In: IEEE Conference on Computer Vision and Pattern Recognition*. 2

Welch, G., & Bishop, G. 95. *An Introduction to the Kalman Filter*. Tech. rept. University of North Carolina at Chapel Hill. 2

Wilson, A., & Bobick, A. 1999. Parametric Hidden Markov Models for Gesture Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**(9), 884–900. 3

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. 1997. Pfinder: Real-Time Tracking of the Human Body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 780–785. 2

Ye, G., Corso, J. J., Burschka, D., & Hager, G. D. 2004. VICs: A Modular HCI Framework Using Spatio-Temporal Dynamics. *Machine Vision and Applications*, **16**(1), 13–20. 3

Zhang, Z. 2000. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(11), 1330–1334. 2

Zhang, Z., Wu, Y., Shan, Y., & Shafer, S. 2001. Visual Panel: Virtual mouse, keyboard, and 3D Controller With an Ordinary Piece of Paper. *In: Perceptual User Interfaces*. 3