Thesis Proposal: 3D Tracking & Recognition of Natural Objects

Cullen Jennings

This is a proposal for a thesis topic of robust tracking and recognition of finger gestures. It is anticipated that this work will also provide and demonstrate a design framework for combining multiple sources of information. These will including stereo range images, color segmentations, motion images, shape information, and various constraints about the information. This information will be used in robust model fitting techniques to track highly over-constrained models of deformable objects such as fingers. Other key contributions will be that this thesis will examine the utility of stereo range images for certain tasks and will provide an example of a robust vision system.

Introduction

APPLICATIONS

Visual communications can be significantly simpler than spoken language. Animals use visual signals; babies use them before they learn to talk; adults use them to supplement communications and when the environment does not allow for speech due to distance or noise or in other circumstances that prevent verbal communication, such as during underwater diving.

Natural hand gestures are fairly low bandwidth for most communications but are particularly suitable for indicating spatial relationships. A mouse - a 2 DOF pointer - has proven very useful for HCI; hand gestures could provide 5 DOF or more. Potential applications include robot and human collaboration, virtual reality interfaces, controlling scientific visualization, GIS, games, control of machines such as those used in forestry, and 3D CAD. Computers are becoming more than screens with keyboards attached, so gestures are likely to be an important part of user interfaces in the future.

RESEARCH CONTRIBUTIONS

Much of the previous work in tracking and recognition has concentrated on rigid models of non-natural objects with corners and edges. Visual recognition systems have often exploited these features and the assumption of rigidity. This research encounters problems in its applicability to many natural objects, such as hands, which have no inherent corners or edges and which are flexible, deformable objects whose shapes, sizes and colors differ from one person to the next. The research outlined in this thesis proposal will involve natural objects - fingers - which lack corners and are flexible and deformable.

As 3D vision systems acquire more practical usefulness, the need for robustness becomes increasingly apparent. In combining multiple input cues, including stereo images, and using these over-constrained models, this research will also provide an opportunity for an examination of the problems involved in building robust vision systems.

Stereo imaging should provide a considerable number of advantages over imaging using only one camera. Range images will simplify segmentation, which is one of the hardest problems for a robust system. Stereo imaging will also stabilize the depth component of estimating 3D positions and help to solve the scale problem where an object appears smaller in the image as it moves further away

The key results anticipated from the research described in this thesis proposal are that the system will be able to track deformable objects like fingers, it will be robust, and it will incorporate real-time stereo images. The system will be able to produce quick, accurate 3D positions and orientations of objects. The system is not intended to produce general object recognition.

It is anticipated that this work will also provide and demonstrate a design framework for combining multiple sources of information and constraints to do robust gesture tracking. The system will exploit the 3D range information contained in stereo images. It will use highly over-constrained models and robust methods for fitting data to the models.

Problem Statement

The goals of this work are to identify principles that increase the robustness of vision systems and to provide a framework for the design of gesture tracking systems. My hypothesis is that combining cues and constraints, particularly stereo images, can significantly increase the robustness of gesture tracking.

This work will describe the development and implementation of a system that provides robust 3D tracking of simple hand gestures to communicate spatial relations in a complex environment.

By simple hand gestures, I mean pointing and other simple gestures, such as “stop”. The goal is not sign language recognition. By a robust system I mean not just one that works under laboratory conditions but one that works for everyone, under various or changing lighting conditions, in a variety of distance ranges. I am concentrating on communicating spatial relations because humans often use their hands to communicate this sort of information to other humans, which suggests it is the natural and probably the fastest way for humans to share certain types of spatial information. By complex environment I mean one that has multiple people or objects in the background and one that works indoors (say in an office or factory) or outdoors.

The thesis work would implement a system that:

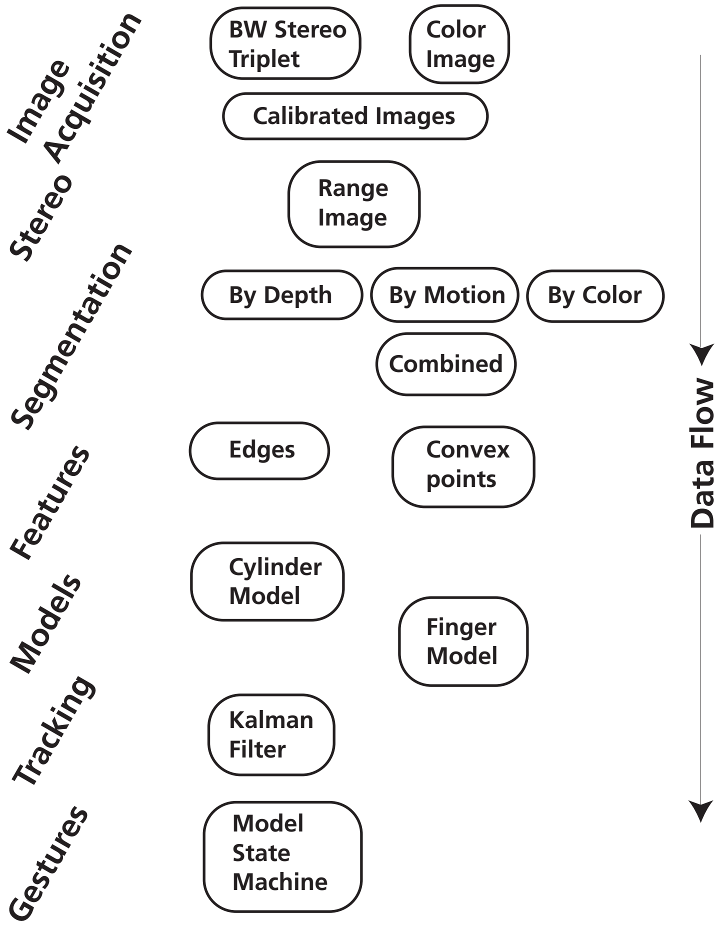
- allows a user to move and orient objects in a virtual environment using finger pointing gestures (basically a 5 DOF joystick using a finger);

- might track fingers on two hands to allow relative motion type gestures;
- works in a robust way in a wide variety of conditions; and
- allows for the use of accurate fine hand motion in 3D to precisely orient and move 3D virtual objects.

From this work, I anticipate drawing important conclusions about what sources of information about a 3D object work well, how to combine sources and what level of robustness is attainable.

Approach

FIGURE 1. **Information flow in system.**



The overall technique is illustrated in Figure 1 and will consist of the following stages. First, images will be acquired from multiple calibrated cameras. From these images, stereo range images will be computed along with color and motion segmentation images. This information will be combined with information from previous frames to segment out, in a robust way, the finger candidates. A shapes-based feature detector will identify candidate finger tips and possible finger locations. The model fitter will then fit the finger model to the data and provide the 3D model parameters. These will be tracked and used to detect the gestures.

More specifically, the stages will be as follows:

IMAGE ACQUISITION

Images will be acquired from multiple calibrated cameras and corrected for lens distortion. Some of the important decisions at this stage are the number of cameras (probably 2 or 3), image resolution, frame rate and whether or not color is important. The current plan would be to use a system like the Triclops system from the TGAP project but to give it color cameras. It is hoped that it would run at about 10Hz with 320 by 240 pixel images. Calibration would be done using existing LCI software and the ACME facility.

STEREO

I believe stereo will help solve some of the key problems. It will, for example, simplify segmentation, which is one of the hardest problems for a robust system. It will stabilize the depth component of estimating 3D positions. It will also help solve the scale problem - that a hand appears smaller in the image as it moves further away.

The stereo imaging would be based on the Triclops/TGAP stereo system. This is a multi-baseline algorithm that uses a sum of squared differences correlation to compute depth images. It is a similar algorithm to the work at CMU described in [Kanade:94]. A significant amount of research has already been done on deciding what trade-offs result in highly reliable results without sacrificing too much performance. This research indicates strongly that accurate camera calibration is important for correlation-based stereo.

SEGMENTATION

The plan is to achieve robust segmentation by combining multiple cues. The basic signals will be depth images, color hue segmentation and image motion segmentation. Shape and size may also be used to further constrain the segmentation.

The depth segmentation gets a rough outline but does not define the edge of the object very well and often has spots of noise. The color segmentation can detect the skin reasonably well but often picks up other objects with similar color properties, such as wood furniture and doors. Once the hand region is found, the color can be tracked through lighting changes using the color consistency techniques described in [Funt:95].

Motion segmentation gives the locations where the hand used to be and where it has gone, so it is only good for estimating general locations. It is easily confused by shadows, changing light conditions and small motions or vibrations of the camera.

SEGMENTATION COMBINATION

The main issue is how to combine various signals and constraints to achieve a robust result. The general approach will be to try to find signals with significantly different failure modes and then, using information in any of the channels, to try to detect failure in a given channel. It would be necessary to find a way to give different weight to different channels according to their current failure detection results.

One approach would be simply to detect the channels that are failing and throw them out. Failure of a channel could be detected by the information in the channel violating some reasonability constraints, such as the size and number of regions or the general shape of the detect regions. The location of the hand in previous frames is a constraint that can be used for checking and filtering results. The remaining channels could be combined using a weighting scheme.

The characteristics of a given channel can also be exploited. For example, the stereo range images might provide a good segmentation but might bleed off the edge of the fingers, causing the finger to seem larger that they are. However, the color segmentation might provide good edge localization, so that it could be used to clean up the edges of the range segmentation. As the tracking progressed it could learn how to weigh and combine the channels optimally for the current situation.

These are only a few possible approaches to the combination problem; there are many others to explore.

FEATURES

The goal at the features stage is to combine the segmentation cues to form higher level cues that constitute strong hints of finger locations. Of assistance here is the fact that the human fingertip approximates a half sphere that looks the same from most orientations. This observation provides a strong geometric constraint that indicates that segmented shapes with a certain sort of convexity are likely to be fingertips. The scale of this feature is also constrained by using the stereo range information.

The goal of this stage is to find several candidate fingertip locations. These will greatly reduce the search space of the model fitting stage.

MODELS

The goal at the model fitting stage is to fit a highly over-constrained, parametrized model to the data. This will be done using a generalized least squares technique. The distance function would measure something like the distance from a given edge of the segmentation to the surface of the current finger model. A robust statistical approach - throwing out outliers and iterating the solution - would be used. The fingertip location data would provide an initial guess of the model location. The accuracy of measurement in various directions would have to be accounted for and used to compute the weighting for the data fitting, since the data would be much more accurate in the vertical plane than in the depth direction. The initial model would be a generalized cylinder. This could be extended to jointed cylinders. Constraints such as human hand geometry can be combined into the model fitting.

TRACKING

The goals of the tracking stage are to track the finger objects and, by combining multiple measurements in time, to increase the accuracy of the system. This stage will also propagate back a region of interest to reduce processing requirements and improve efficiency. Initially a Kalman filter will be used for each finger being tracked. The model fitting component will use some of the techniques described in [Lowe:92]. The goal of this stage is to output a high accuracy 3D position and orientation of a finger.

It remains to be seen whether this stage is necessary or adds to the robustness of the system.

GESTURES

To be really useful, the system needs a few gesture recognition elements, including some way to indicate the start and end of a gesture. The system needs about the same level of input that the buttons on a mouse provide. The current approach to this would be to watch the speed of motion to indicate the start of a gesture and to look for a quickly bent finger to indicate the end of the gesture.

One area to consider is that situated gestures provide considerably more constraints than do completely context-free gestures; it may be possible to use these constraints to increase robustness.

EXTENSIONS

This system could be extended to track multiple fingers, to use a more complex finger model that allows bent fingers, or to interpret more gestures.

State of the Art

Gesture recognition has been attracting interest from many researchers in recent years. It is being explored as a novel human-computer interface (a 3D mouse) [Ishibuchi:93], as a method for directing robots [Katkere:94], and even as a replacement for television remote controls [Freeman:95]. The problem is commonly investigated by addressing one or more of the following subproblems:

- how to segment the hand from the image;
- how to track the hand;
- what kinds of gestures are suitable; and
- how to recognize the gestures.

[Suetens:92] provides an overview of useful techniques for object recognition. [Pavlovic:97] is a survey paper more specifically addressing hand gestures.

CAMERA CALIBRATION

The system proposed in this research starts with the acquisition of calibrated images. This stage will combine the camera calibration techniques described in [Tsai:87] with the image centre corrections described in [Lenz:88].

STEREO

The stereo imaging builds on several important pieces of work. The multiple-baseline stereo described in [Okutomi:93] is an excellent technique for increasing the robustness of a vision system by combining multiple cues. Several cameras are used for stereo, but instead of computing pair-wise stereo correspondence and then combining the multiple pairs (as previous work did), the authors combine all the disparity errors and do one minimization for all the pairs. In essence, each pair contributes to the evidence for a given disparity.

Also important in correlation-based stereo is the work of [Fua:93]. Through careful analysis, the author comes up with optimal normalization of the correlation data, which leads to better stereo range images than does the naive correlation method. One important robustness enhancing clue from [Fua:93] is to do a “cross validation check”. This check makes sure that the best right image to left image stereo match is the same as the best left image to right image match.

The concept of using stereo for gesture recognition or even segmentation has not been widely pursued but has certainly been done in previous research such as [Kortenkamp:96].

SEGMENTATION

Hand segmentation has been handled in one of four ways in most of the literature: using a controlled (uncluttered) background [Katkere:94]; using a known background (i.e., background subtraction) [Ishibuchi:93]; using segmentation by motion [Ko:96]; or using color segmentation [Fleck:96].

For a dynamic environment, the use of controlled or known backgrounds has complications since the background can change over time. Motion cues can be difficult to apply due to the false impression of motion caused by changing light and small camera motions. Color segmentation, however, is a fast and fairly robust approach to hand segmentation that has been shown to work well under varying lighting conditions and against unknown backgrounds. Unfortunately it can easily be confused if the background behind the hand is close to skin color. All of these methods have particular failure modes when implemented in the simplest way, but for any particular failure mode, one can imagine a way to detect that the method has failed.

BLOBS

[Azarbayejani:96] describes work going on in the Pfinder system. This system uses “blobs” - oval regions of consistent color and texture - as features to track, typically attributing one blob to the head, one to the torso and one to each hand, arm, foot and leg. The Sfinder extension to this system takes tracked blobs from two cameras and gives 3D tracking data. The system seems to rely heavily on being able to use color to segment out the objects to track. It also has to “learn” the background image so that it can be ignored. Pfinder should therefore suffer from many of the problems that cause background subtraction and color tracing to fail. A more detailed description of this system is given in [Wren:97].

TRACKING

Tracking has been facilitated through the use of special markers [Davis:94], correlation [Freeman:95] and a combination of color and shape constraints [Ko:96]. It has been shown that tracking of colored regions can realistically be expected at frame rates [Barman:93].

One of the most encouraging systems for model tracking is described in [Lowe:92]. This system does edge detection on the image and then fits lines to the edges. These edges are matched to the model edges using a “best first” search approach. The model parameters are then solved for using a weighted least square approach. The system’s careful approach to errors in matching and measurement allows it to quickly find a good match and to weight the least squares so that it is not dominated by outliers and converges even with large motions. The system is robust and real-time, and it is easy to understand that it should work. A large part of the approach is readily applicable to tracking fingers. Fitting straight lines to edge images is probably not the best way to get stable features for flexible objects such as fingers, but it might work.

One technique that gets away from explicit models is the Eigen image idea. In [Black:96], Black and Jepson develop a system that tracks and recognizes simple hand gestures using what they call “EigenTracking”. This system builds a representation of the hand model from training images and uses this representation to compute an Eigen image set. It then finds and tracks these images in a video stream using a pyramid approach. The major advance of this work is it that it uses Eigen techniques to solve for the view transformation of the object at the same time that it finds the motion of the object in the scene.

Blake and Isard, in [Blake:94], describe a system whose approach to the tracking problem resembles that in “snakes” but which is based on Kalman filtering. The authors develop a finger and lip tracking system that uses a Kalman filter to estimate coefficients in a B spline. Measurements are made to find the minimum distance to move the spline so that it lies on a maximal gradient portion of the image. These measurements are used as the next input to the Kalman filter.

In [Rehg:93], Rehg and Kanade describe a system called DigitEyes that tracks a hand at 10 Hz, using a 27 DOF model based on cylinders. They describe the implementation of a 3D mouse with 3DOF application based on this sensor. One insight they provide is that

> [o]ther feature choices for hand tracking are possible, but the occlusion contours are the most powerful cue. Hand albedo tends to be uniform, making it difficult to use correlation features. Shading is potentially valuable, but complicated illuminance and self-shadowing of the hand make it difficult to use.[1]

My experience with stereo and hand tracking suggests that the contours of the hand provide the majority of the features that a correlation stereo algorithm will use for matching.

The Rehg and Kanade system fits a model to tracked features using a nonlinear least squares fit. The model is updated with a Kalman filter. The finger model is used to find the features in the next frame by searching orthogonally out from the centre of the finger and looking at the gradient of the greyscale image to find the edge of the finger. These contour edges of the fingers are used to establish the current feature location for the finger. The authors point out that difficult problems remain in tracking through occlusions and across complicated backgrounds.

The work of Huttenlocher and Rucklidge is interesting in that it provides tracking of flexible objects with no explicit model. This work is described in [Huttenlocher:92] and advanced in [Rucklidge:97]. First the system forms an edge image. It then matches this to previous edge images using a Hausdorff distance metric. The raw Hausdorff distance is modified to become a rank order distance to improve the robustness of the system. The model image is transformed by a discretization of the 2D projections of all the 3D transformations of the real object. The initial model is formed from the difference between the first two frames.

The important results seem to be:

- the system can compare feature images using Hausdorff rank distances;

[1] Page 6 of [Rehg:93].

- a search of the complete nearby “model space” is possible; and
- this search does not require knowledge of the model but only knowledge of what transforms can be applied to it.

One difficulty of a system with no explicit model is that to get the real 3D position of the finger, a model of some sort is still needed.

This work is significantly advanced in [Rucklidge:97], which presents several methods for quickly searching the transform space using a Hausdorff distance to match the image and model. Affine transforms are allowed. The system is certainly not real time, but it is feasible. The use of an nth percentile Hausdorff distance shows how robust statistics can be used to improve matching accuracy by eliminating outliers.

A similar concept is described in [Darrel:93]. This system uses view interpolation on training images to do the tracking and recognition. The images are dynamically warped to deal with nonrigid objects like hands. The computation is expensive, but the system achieves 10 Hz performance for small images. The matching of images is done using a correlation that is weighted by the variance of each pixel.

GESTURES

Gestures are made up of three types of information: the actual hand shape, the spatial relation of the gesture to the world (e.g., a finger pointing at an object) and the hand motion (e.g., with beckoning). The Robogest system [Katkere:94] is an example of working gesture control for a robot that uses a constrained background and simple gestures. The authors use Zernike moments of the segmented hand to recognize gestures. Providing an example of extensive shape recognition, Uras and Verri [Uras:94] use size functions to identify symbols from American Sign Language.

The Perseus system [Kahn:96] is an example of spatial gesture recognition. The robot is directed to objects by a user pointing at them. The gesture is really a body gesture (pointing with the arm), rather than a hand gesture. The system uses extensive scene knowledge (3D information, background, objects, and people) from other vision operations to achieve its spatial understanding.

CHI

[Hinckley:94] provides a review of literature on 3D input techniques from a CHI point of view. Given a 3D hand position sensing device, it discusses the advantages and disadvantages of different methods for using the raw 3D position information to solve some spatial input problem. It points out that the “clutching” problem - which the mouse solves by permitting a button to be held down to indicate that the user wants to clutch an object and move it - is quite hard to solve and can be the non-intuitive part of the interface.

One idea that the authors in [Hinckley:94] explore is the use of a two-handed interface, where one of the user’s hands “holds” the object and the other controls a tool that manipulates it. Since the hold hand provides the spatial reference for the other hand, this works quite well.

[Bolt:92] describes some actual experiments with using computers to position objects using speech and gestures. This work uses a data glove for gesture input. Similarly, the author of [Hauptmann:89] did experiments to determine the type of gestures people use to describe simple geometric manipulations of a cube. Most often the gestures were one fingered, although they were also commonly five or two fingered. In his conclusions, the author states that:

> Any system that restricts the user to a single hand or finger motion (like the “mouse”) will be inadequate for many of the manipulations analyzed in this experiment. We feel it is safe to further generalize to other manipulations in real world applications, which are likely to be even more complex.[2]

Initial Results

CAMERA CALIBRATION

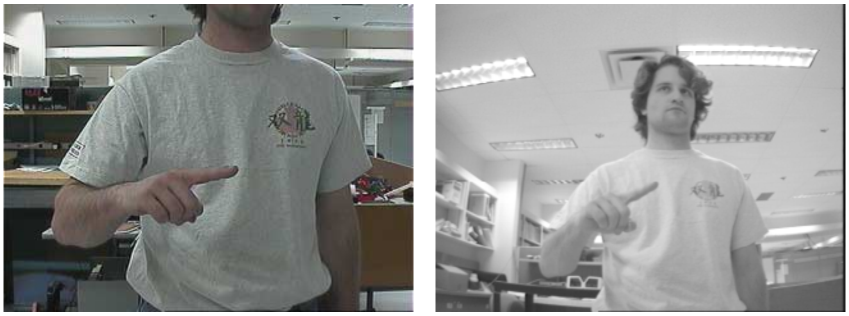
The camera calibration is currently done using the system developed by various LCI members including Rod Barman, Stewart Kingdon, Mike Sahota, Don Murray, and Cullen Jennings. The CIF is used to precisely move a plate with a grid of targets, thus allowing a cube of known reference points to be collected. The Tsai camera calibration method [Tsai:87],[Lenz:88] is used to fit the camera model parameters to this data. This is implemented in Matlab. No experiments have been done to check the accuracy of this calibration, but the results using the data seem quite good, suggesting that the calibration is correct.

Currently this process is fairly cumbersome and involves several programs. I would like to simplify it, clean it up, perhaps integrate the methods in [Ayache:88], and then integrate it into the ACME system. I have also been doing work as part of the TGAP project on a simplified calibration process. This may get used, if it works sufficiently well.

IMAGE ACQUISITION

The system captures images from a color camera and a Triclops camera head with three black and white cameras. The color images are captured by a composite frame grabber card, while the three black and white images are captured in the channels of an RGB frame grabber. Even though the images are grabbed on separate cards, they are in the same computer and seem to be close to synchronized. They are captured at a resolution of 320 by 240. Example images are shown in Figure 2.

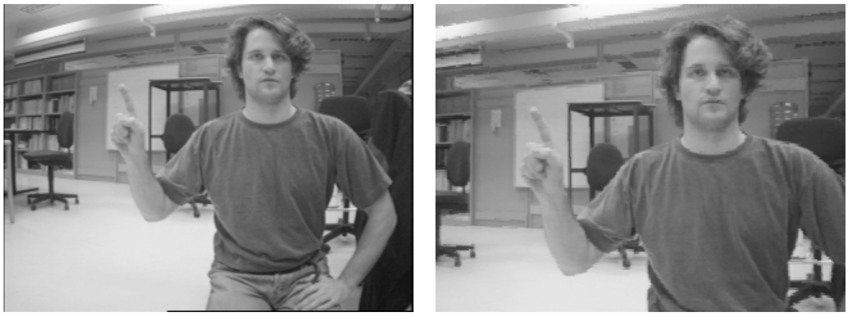
FIGURE 2. **Raw color and black and white images**



[2] Page 244 of [Hauptmann:89].

These images have considerable radial distortion due to the short focal length lenses. This distortion is corrected by using the camera calibration information to resample the image data into a “pin hole” camera model: first the image is smoothed slightly and then a non-interpolated lookup is used. The code to do this was implemented by Stewart Kingdon. Example images are shown in Figure 3. Note that the straight lines along the ceiling are bent in the uncorrected image.

FIGURE 3. **Raw image and image corrected for camera distortions.**



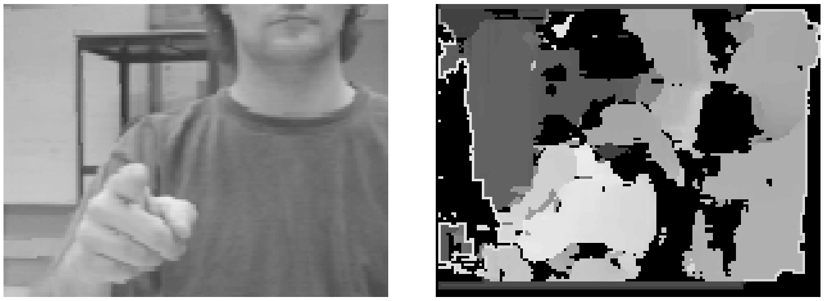
I am working on code to correct the color images as well as the black and white and to do proper pixel interpolation and filtering to reduce alias artifacts.

STEREO

I am currently experimenting with two stereo systems. The first is my implementation of the method described in [Fua:93]. This system uses two cameras and achieves reasonable results but is not particularly fast.

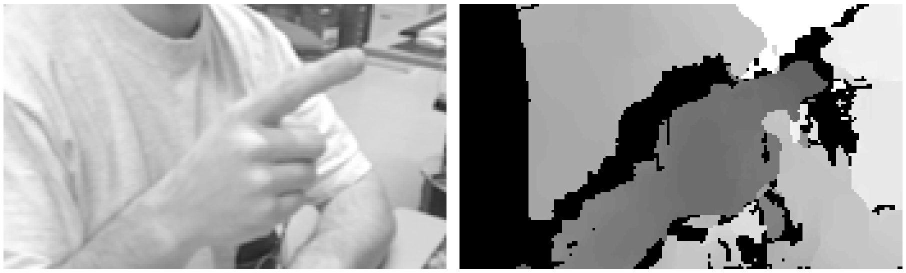
A sample range image from the first stereo system and the associated reference image are shown in Figure 4. The reference image is shown on the left, and the range image is on the right. Closer objects are lighter, and black areas indicate no match was found in the stereo algorithm.

FIGURE 4. **Stereo range image.**



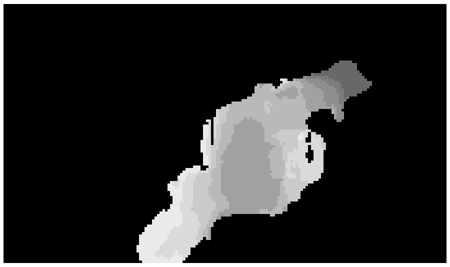
A second stereo image is shown in Figure 5. The reference image is on the left, and range values are on the right.

FIGURE 5. **Stereo range image.**



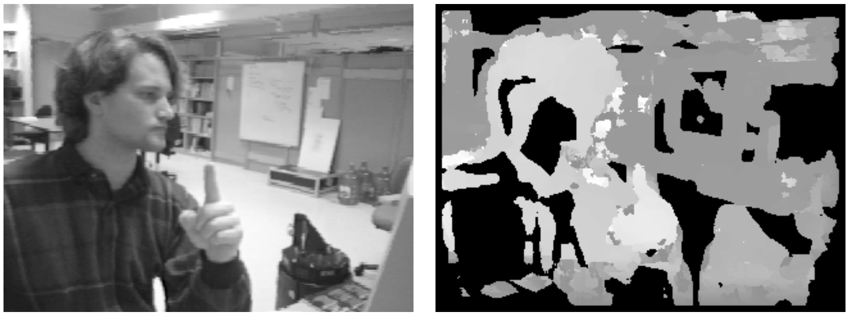
There are actually many distinct range values on the hand portion of the image. To make this more apparent, in Figure 6 I have increased the contrast of the display to make it easier to see the distinct range values in the hand area.

FIGURE 6. **Range values near hand**



The second stereo system I have been using is the Triclops system, which is much faster. Several people have worked on it. I was involved in algorithm design and implementation trade-off decisions. Stewart Kingdon implemented the bulk of the code and optimized it to use MMX. It is unclear whether the Triclops system gets better or worse results than my implementation of the method described in [Fua:93]. A sample image is shown in Figure 7.

FIGURE 7. **Triclops reference and stereo range image. (Black is no match.)**

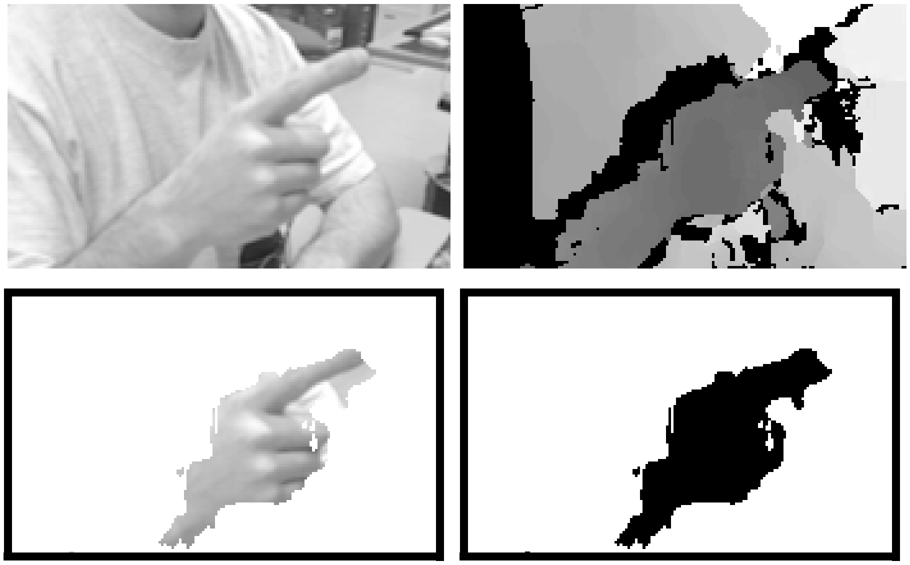


SEGMENTATION

The depth segmentation images are formed by finding things that are reasonably close to the camera. This is done by thresholding the range image in the top right quadrant of

Figure 8 to get the mask shown in lower right. This mask is used to select the pixels from the original reference image in the top left and to form the segmented image shown in the lower left.

FIGURE 8. **Reference image, depth image, depth segmentation, and mask.**



A second example uses the reference and depth images shown in Figure 9. The resulting mask and segmentation are shown in Figure 10.

FIGURE 9. **Second reference image and depth image.**

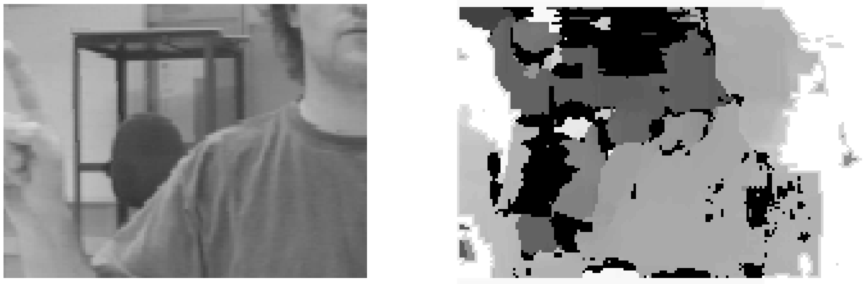
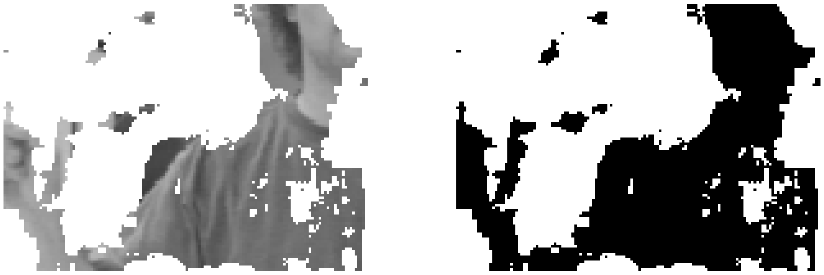


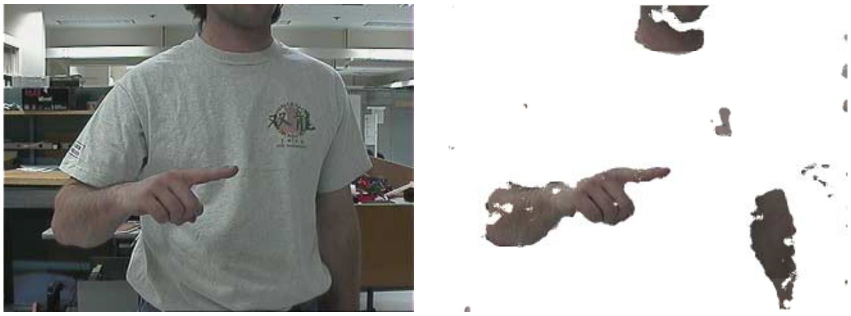
FIGURE 10. **Second depth segmentation and mask.**



Another approach may be to find the closest thing that moves, assume this is the hand to track, and then segment out the objects that are within about six inches from this object's depth plane.

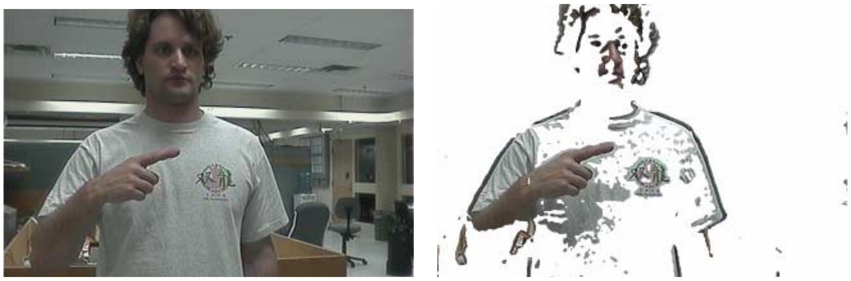
The color segmentation images are built by looking at the hue of the skin and also considering the color intensity. This gives a very noisy segmentation, but it can be cleaned up with erosions and dilations to the image. Once the initial image is found, it can be used to refine the color search class for subsequent images. A final segmentation is shown in Figure 11. My experiments have indicated that color segmentation can be done fairly well with 16 bit color images but does not work with 8 bit color images.

FIGURE 11. **Color image and cleaned up segmentation**



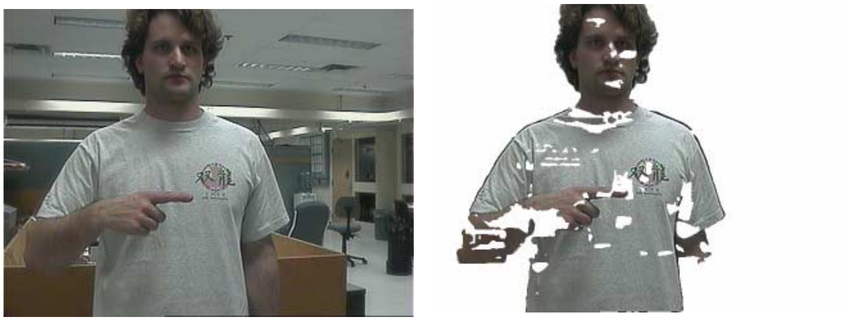
The difference between two consecutive frames can be thresholded to form an outline of the moving objects in the image. An example is shown in Figure 12. One problem here is that it shows where the object was as well as where it has gone. This is not a significant problem if the number of images per second is large enough that the motion between each image pair is small.

FIGURE 12. **Motion segmentation**



A technique that is often used but seldom works robustly is background subtraction. The problem is that it is very susceptible to changes in lighting conditions and camera vibration. It should be possible to detect these sorts of failure modes and perhaps compensate for them. An example of thresholding is shown in Figure 13.

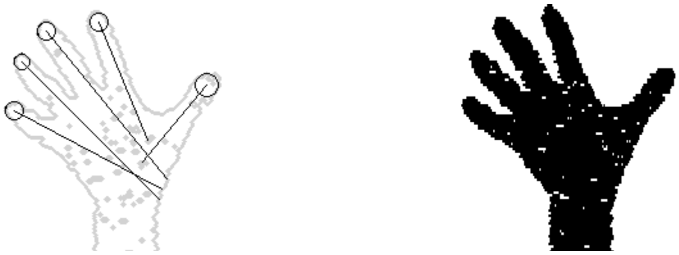
FIGURE 13. **Background subtraction**



FEATURES

I have put considerable effort into producing a fast feature detector for fingertips given a reasonable segmentation of a hand. My feature detector currently runs at 5Hz on a Sun Sparc 20. The resulting feature locations are shown in Figure 14. The concept is to find regions of the image that have the right convexity to be fingertips.

FIGURE 14. **Convexity features from segmentation**



By combining stereo information with the input segmentation, the detector should be able to compute the correct scale space for a given location in the image. I think this will improve the detection and make it more robust.

Experiments

Experiments will evaluate the performance of this system. Items to be considered include: how often it tracks the correct finger; its accuracy and precision; what factors affect its performance; and over what spatial range it works.

References

AYACHE:88 N. Ayache and C. Hansen. “Rectification of images for binocular and trinocular stereo vision”, ICPR’88, October 9, 1988, Bejing, China, pp. 11-16.

AZARBAYEJANI:96 A. Azarbayejani, C. Wren, and A. Pentland. “Real-Time 3-D Tracking of the Human Body”, M.I.T Media Lab. Tech. Report #374. Appears in Proc. IMACE’COM 96, Bordeaux France, May 1996.

BARMAN:93 R. Barman, S. Kingdon, J.J. Little, A.K. Mackworth, D.K. Pai , M. Sahota, H. Wilkinson, and Y. Zhang.“DYNAMO: real-time experiments with multiple mobile robots”, Intelligent Vehicles Symposium, July 1993, Tokyo.

BLACK:96 M. Black and A. Jepson. “EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation”, ECCV’96.

BLAKE:94 A. Blake and M. Isard. “3D position, attitude and shape input using video tracking of hands and lips”, SIGRAPH’94.

BOLT:92 R. Bolt and E. Herranz. “Two-Handed Gesture in Multi-Modal Natural Dialog”, UIST’92, Monterey, California, November 15-18, 1992, pp. 7-14.

DARREL:93 T. Darrell and A. Pentland. “Space-Time Gestures”, CVPR’93, pp. 335-340.

DAVIS:94 J. Davis and M. Shah. “Visual Gesture Recognition”, IEEE Proc. Vision Image Signal Processing 141(2), April 1994, pp. 101-106.

FLECK:96 M. Fleck, D, Forsyth, and C. Breggler. “Finding Naked People”, ECCV’96, April 15-18, 1996, pp. 593-602.

FREEMAN:95 W. T. Freeman and C. D. Weissman. “Television control by hand gestures”, IEEE International Workshop on Automatic Face and Gesture Recognition, June 1995.

FUA:93 P. Fua. “A parallel stereo algorithm that produces dense depth maps and preserves image features”, Machine Vision and Applications, 1993, vol. 6, pp. 35-49.

FUKUMOTO:92 M. Fukumoto, K. Mase, and Y. Suenaga. “Real-time detection of pointing actions for a glove-free interface” In IAPR Workshop on Machine Vision Applications, December 7-9, 1992, pp. 473-476.

FUNT:95 B. Funt and G. Finlayson. “Color constant color indexing”, PAMI 17(5), May 1995, pp. 522-529.

HAUPTMANN:89 A. Hauptmann. “Speech and Gestures for Graphic Image Manipulation”, In Proc. CHI’89, April 30-May 4, Austin, Texas, pp. 241-245.

HUTTENLOCHER:92 D. Huttenlocher, J. Noa, and W. Rucklidge. “Tracking Non-Rigid Objects in Complex Scenes”, Cornell University Computer Science Dept. Tech Report, CUCS TR-92-1320.

HINCKLEY:94 K. Hinckley, R Pausch, J. Goble, and N. Kassell. “A Survey of Design Issues in Spatial Input”, UIST’94, November 2-4, 1992, Marina Del Rey, California, pp. 213-222.

ISHIBUCHI:93 K. Ishibuchi, H. Takemura, and F. Kishino. “Real Time Hand Gesture Recognition using 3D Prediction Model”, International Conference on Systems, Man, and Cybernetics, Volume 5, Le Touquet, France, October, 1993, pp. 324 - 328.

KAHN:96 R. Kahn, M. Swain, P. Prokopowicz, and R. Firby. “Gesture Recognition Using the Perseus Architecture”, CVPR’96, June 18-20, 1996, San Francisco, California, pp. 734 - 741.

KANADE:94 T. Kanade. “Development of a Video Rate Stereo Machine”, Proc. 1994 ARPA Image Understanding Workshop, November 14-16, 1994, Monterey, California, pp. 549-558.

KASS:87 M. Kass, A. Witkin, and D. Terzopoulos. “Snakes: Active Contour Models”, IJCV 1(4), 1987, pp. 321-331.

KATKERE:94 A. Katkere, E. Hunter, D. Kuramura, J. Schlenzig, S. Moezzi, and R. Jain. “ROBOGEST: Telepresence using Hand Gestures”, Technical report VCL-94-104, Visual Computing Laboratory, University of California, San Diego, December 1994.

KORTENKAMP:96 D. Kortenkamp, E. Huber, and R. Bonasso. “Recognizing and interpreting gestures on a mobile robot”, AAAI’96, August 4-8, 1996, Portland, Oregon, pp. 915-921.

KO:96 I. J. Ko and H. I. Choi. “Extracting the hand region with the aid of a tracking facility”, Electronic Letters 32(17), August 1996, pp. 1561- 1563.

LENZ:88 R. Lenz and R. Tsai. “Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology”, PAMI 10(5), September 1988, pp. 713-720.

LOWE:92 D. Lowe. “Robust Model-based Motion Tracking Through the Integration of Search and Estimation”, IJCV 8(2), 1992, pp. 113-122.

OKUTOMI:93 M. Okutomi and T. Kanade. “A Multiple-Baseline Stereo”, PAMI 15(4), April 1993, pp. 355-363.

PAVLOVIC:97 V. Pavlovic, R. Sharma, and T. Huang. “Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review”, PAMI 19(7), July 1997, pp. 677-695.

REHG:93 J. Rehg and T. Kanade. “DigitEyes: Vision-Based Human Hand Tracking”, Technical report CMU-CS-93-220, School of Computer Science, Carnegie Mellon University, December 1993. Also see ECCV’94.

RUCKLIDGE:97 W. Rucklidge, “Efficiently Locating Objects Using the Hausdorff Distance”, IJCV 24(3), September/October 1997, pp. 251-270.

SUETENS:92 P. Suetens, P. Fua, and A. Hanson. “Computational Strategies for Object Recognition”, ACM Computing Surveys 24(1), March 1992, pp. 7-61.

TSAI:87 R. Tsai. “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, IEEE Journal of Robotics and Automation RA-3(4), August 1987, pp. 221-244. Also see CVPR’86.

URAS:94 C. Uras and A. Verri. “On the Recognition of the Alphabet of the Sign Language through Size Functions”, International Conference on Pattern Recognition, Jerusalem, Israel, Session B18.4, October 9-13, 1994.

WREN:97 C. Wren, A. Azarbayejani, T Darrell, and A. Pentland. “Pfinder: Real-Time Tracking of the Human Body”, PAMI 19(7), July 1997, pp. 780-785.