Gesture recognition for robot control

Keywords: Gesture recognition, Active and Real-Time Vision

Abstract

This paper describes a system which enables a robot to recognize and respond to hand gestures from a human operator. The robot uses a color camera on a pan-tilt unit to track an operator as both the robot and the operator move, with the operator controlling the robot through visual gestures. The system uses a segmentation based on the hue of the image to find skin-colored areas. It then uses a convexity-based feature detector to find possible fingertips. The fingertips are filtered through geometry constraints to identify possible finger candidates. These are then matched against hand models to perform gesture recognition. This system is capable of recognizing a few simple gestures from a wide range of operators in an environment with complex backgrounds.

1 Introduction

It is increasingly important for robots to work with and around people in a cooperative and efficient manner. To do this, robots must be aware of where people are located, not only for safety precautions but also to accomplish tasks. Imagine a scenario where a robot is capable of moving heavy objects. An operator could specify which object the robot should pick up, tell the robot to follow him or her to another location, and then instruct it where to place the object. In order to operate effectively within this scenario, the robot must be able to track the operator in a complex and dynamic environment such as a factory or warehouse. Gesture-based recognition provides a useful and effective means of communicating with robots, under this and other circumstances, for the following reasons: gestures can be directed towards a particular robot, they are not confused with communication to people, they can be used in a noisy environment, and they do not require the operator to have a remote control device. Gesture-based recognition is a simple and intuitive way to provide accurate and high-level control of a robot.

In this paper, we describe a system that is capable of recognizing and responding to simple hand gestures from human operators in complex and changing environments. This system uses hue segmentation to locate hands, a convexity feature technique to identify fingertips, and a geometric constraint model to recognize hand gestures. The working system we describe is a useful contribution to understanding techniques of gesture recognition in robotics.

Figure 1: *OurRobotsNameHere* and operator

The work uses the Laboratory for Computational Intelligence's (*OurLab*) visually–guided mobile robot *OurRobotsNameHere* . *OurRobotsNameHere* is shown working with an operator in Figure 1.

2 Previous Work

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) [5], as a robot direction method [7], and even as a replacement for TV remote controls [4]. The problem is commonly investigated by addressing one or more of the following sub-problems:

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

Hand segmentation is generally handled in one of four methods: use of a controlled (uncluttered) background [7], use of a known background (i.e., background subtraction) [5], segmentation by motion [8], and color segmentation [3]. In our approach, we operate in a dynamic environment with both moving operator and robot. This makes use of controlled or known backgrounds infeasible. Motion cues are difficult to apply, due to apparent motion caused by a moving robot frame of reference. Consequently, we chose color segmentation as a fast and robust approach for hand segmentation, that has been shown to work well under varying lighting conditions and unknown backgrounds.

For a real-time, dynamic system, tracking the hand is a requirement in order to keep the hand in the field of view as the operator and robot move. Tracking has been facilitated with special markers [2], by use of correlation [4], or with a combination of color and shape constraints [8]. For our purposes, the latter approach was most appropriate. It has been shown that tracking colored regions can realistically be expected at frame rates [1].

The kinds of gestures used and the methods for recognizing them vary widely from researcher to researcher. Gesture are made up of 3 types of information. There is the actual hand shape, there is the spatial relation of the gesture to the world (e.g., a finger pointing at an object) and there is the hand motion (such as with beckoning). The Robogest system, by Katkere *et al.* [7], is an example of working gesture control for a robot using a constrained background and simple gestures. They use Zernike moments of the segmented hand to recognize gestures. An example of extensive shape recognition, by Uras and Verri [9], uses size functions to identify symbols from the American Sign Language.

The Perseus system [6] is an example of spatial gesture recognition. The robot is directed to objects by pointing at them. At this level, the gesture is really a body gesture (pointing with the arm) rather than a hand gesture. The work uses extensive scene knowledge (3D information, background, objects and people) from other vision operations to achieve this spatial understanding.

Combining shape and motion is common in real-time systems. Freeman and Weissman [4] present a cheap, fast hand tracking system based on template correlation for TV control. This may be limited in operating ranges and scales. Davis and Shah [2] track fingertips as they move with the aid of specially marked gloves.

The system described in this paper is based on color segemntation, feature dection, and recognition from a model built from the features. It works in an unconstrained environment with a clutter background. It does not require any training and operates in real time.

3 Approach

The goal of the system is to identify and respond to hand gestures from an operator who is standing between 8 and 24 feet from the robot. The resulting gesture recognition has the following features: it can be extended to numerous gestures, it works in a cluttered environment and under variable lighting conditions, and it is quickly computed.

Our approach to gesture recognition consists of four stages. First, areas in the image that are flesh colored (such as hands) are identified and segmented out. Second, certain features are detected by a scale space convexity feature method. Third, fingertip candidates are found by the clustering of convexity features and the geometry of the edges of the segmented regions. Finally, a hand model is formed by looking at the relative locations of candidate fingers. Each of these four stages is described below.

3.1 Segmentation

The goal at this initial stage is to find the outline of candidate hand objects. To do this, the images are segmented according to the hue of the pixels. The hue of human skin has been found to be fairly consistent under varying lighting conditions (i.e., whether a hand is in shadow or direct light) and across a range of people and races [3]. However, some undesirable materials, such as wood, are also included in the segmentation.

3.2 Convexity Features

An approximate model of a fingertip is a half-sphere attached to the end of a cylinder. Since a hemisphere has the same convexity from any viewing angle, this property is used as an invariant feature for detecting fingertips.

Convexity can be considered a scale space variant feature. The system must look for objects that match the convexity and scale of a fingertip, not that of a little bump on a finger or a large object like a head. This suggests the need to use a curvature measure of an edge to decide whether it is a convex segment.

Some previous approaches to convexity have measured the curvature of the edge. However, our edges contain few pixels and in some respects, measure of curvature is a second derivative. The small number of pixels available on an edge make second derivate measurements prone to noise. If they are smoothed, then it is not possible to detect convex segments as the system ends up working in too large a scale space.

Instead of measuring curvature, our convexity test is based on whether a tangent line can be drawn such that it touches the edge of the object without intersecting any other objects. If this tangent line is infinitely long, then we have a test for global convexity. As the line becomes very short, we get a convexity measure of local curvature. For example, if the line has a length of three pixels, convexity estimates are made using a three by three pixel window. By adjusting the length of this line, the scale space of the convexity feature can be adjusted.

The net result of the convexity feature detector is that many convex points cluster around the fingertips and are fairly sparse in the rest of the image. Since a hemispherical fingertip has many convex points in a small vicinity, this reduces the number of feature points in the image. Feature points that do not have several nearby supporting convexity features are unlikely to be fingertips and are removed from consideration. The features are clustered using a non-maximal suppression technique that suppresses all the features within a radius the approximate size of a fingertip.

3.3 Geometry Constraints

This stage uses the geometry of the segmented regions to validate that the fingertip candidates are at the end of an object that could be a finger.

First, a circle is fitted to the end of the finger by finding a circle that contains the convex feature point and maximizing the amount of the circle that lies on the edge of the segmented region. This effectively fits a circle to the end of the fingertip and localizes a center point for the end of the finger. Although the exact edges are not particularly stable from one image to the next (due to the instability of segmentation), the center of the fingertip is a very stable feature.

At this stage a series of rays are radiated out from the center of the fingertip in all directions until they meet the edge of the object. If the object is long and thin like a finger, most rays will only travel a short distance. However, in one narrow range of direction, the rays will travel for a fairly long distance. This is used to validate fingertips as fingers and to identify the fingers' approximate orientation.

3.4 Model Building & Gesture Recognition

This stage examines all the finger candidates and identifies groups of fingers that are likely to be a hand. The relative locations and direction of fingers are considered and tested against possible hand models for the gestures we wish to recognize.

When a hand has been identified, a region of interest (ROI) is constructed near the hand in the image. The ROI from the previous image is used to verify subsequent images. The "stop" hand gesture is very unlikely to occur from noise, so it is always used to identify a hand even if it falls outside the ROI range.

3.5 Hand Localization

The location of the center of the hand is determined by a camera on a pan-tilt unit on the robot. This allows the robot to track the hand even when the person or robot is moving.

4 Implementation

4.1 System Architecture

This system was implemented on the *OurLab* mobile robot, *OurRobotsNameHere* , and a support Sun Ultra Sparc 1 workstation. The *OurRobotsNameHere* robot consists of a Real World Interface B-12 base with three black-and-white video cameras for trinocular stereo and a color camera mounted on a pan-tilt head. There are two TMS320C40 DSPs (digital signal processors) for image processing, two T805 transputers for control and a T225 transputer for communication. The robot communicates to the workstation over high-speed radio modems.

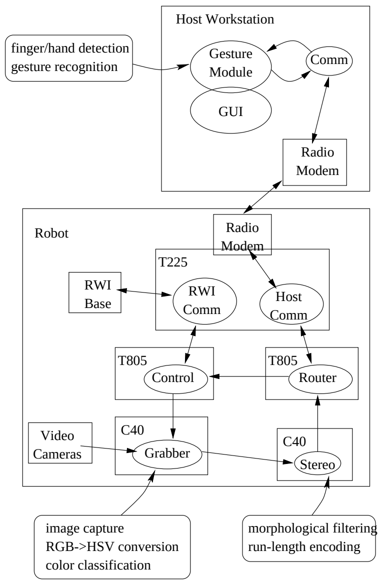


Figure 2: Architecture

Figure 2 shows the data flow through the subsystems and across the hardware. Image capture, color classification and morphological filtering operations are performed onboard *OurRobotsNameHere* via the C40 DSPs. The classified image is run-length encoded and sent to the workstation for gesture recognition. After gestures have been interpreted, commands are forwarded to the transputer controller via the radio modems. We intend to improve the performance of the system by using the SGS-Thomson A110 convolver hardware present onboard to speed up the erosion/dilation operations and the run-length encoding of the image.

To segment hands (and other skin regions) from the surrounding scene, we applied bandpass filters on the image HSV (hue, saturation and value) bands (as described by Fleck *et al.* [3]). Hue provides the basis of the segmentation, although saturation and intensity were used to reject extreme values where hue becomes unstable.

The image was captured as a 15-bit RGB (red-green-blue) image and converted to HSV before filtering. Initially, we attempted using 8-bit color images; however, these did not have the resolution required to successfully segment skin. To tune the system, we captured sample images under expected lighting conditions and determined the bandpass filter thresholds experimentally.

As Fleck *et al.* experienced, we had trouble with certain colors that strongly resemble skin in hue and saturation, most notably wood, cream-colored walls and brown cardboard. Since there is not a sharp cutoff in hue between skin and these similar materials, we could not always achieve a clean segmentation. We tuned the bandpass filter by adjusting the quantization of the RGB $\rightarrow$ HSV algorithm, which successfully removed all these colors except for some shades of wood. A few dots of wood regions passed the hue thresholds and similarly, a few pixels of skin failed. This resulted in a speckled segmentation that was difficult to analyze for shape information. To remove these artifacts, we employed morphological opening (erosion followed by dilation), which was quite effective in cleaning up the segmentation. The final values used were: hue $0.53 \rightarrow 0.67$, saturation $0.05 \rightarrow 0.5$ and value $0.25 \rightarrow 0.90$.

Since our approach is based on hue, it is fairly robust with respect to the intensity of lighting conditions. However, skin regions in dark shadows could not be successfully segmented, because as intensity drops, hue becomes less stable and an increasing portion of skin pixels fail the hue threshold. This led the algorithm to reject the region, and dark shadows remain a problem for the system.

4.2 Color Classification

The coarse shape of skin regions was stable and consistent from image to image. However, details in the shape's contours were unstable and fluctuated from image to image, even when no movement was present. This created noisy edges from which we expected to extract shape information based on convexity. Since convexity was locally unstable, but quite reliable at a coarser resolution, we used a coarse-scale convexity approach to successfully extract hands and fingers (as described in Section 3.2). Figure 3(a) shows a sample image. Figures 3(b) and (c) show image hue and skin segmentation respectively. One can see that the skin region contours, though locally noisy, accurately convey the overall shape.

4.3 Finger detection

After segmentation, the next step in the algorithm was to detect convex points and verify or reject them as belonging to fingertips. As described above, while the segmented image had consistent and stable shapes, the local contours of the shapes were prone to noise. This resulted in the occurrence of numerous locally convex points at random locations along the region contours and the number of potential fingertips was large.

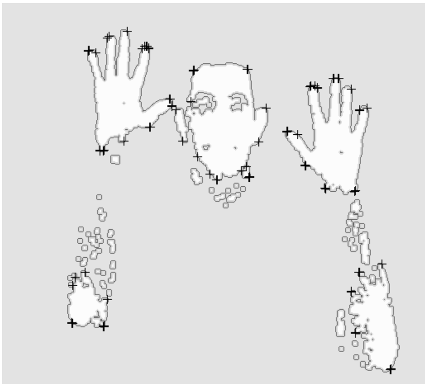
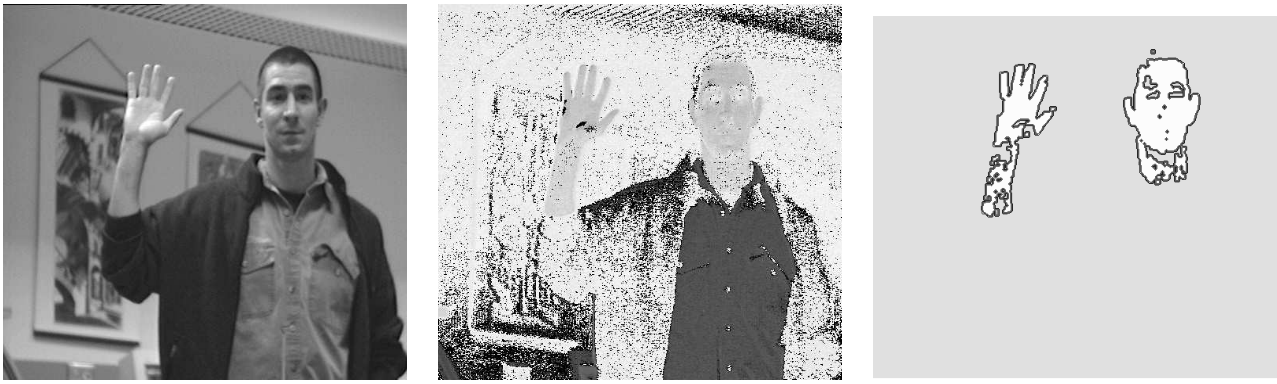


Figure 4: Convex points

As described in Section 3, we adopted an approach that detects convex points at a sufficiently coarse scale to successfully label candidate fingertips. This approach reduced the total number of contour points to be considered by two orders of magnitude, greatly reducing the computational cost of fitting finger models to the contour since most contour points could be rejected. The result of this approach was to distill the total number of edge points to a small set that could be considered. Figure 4 shows an example of convex point detection. Convex points are marked in an already segmented image.

After candidate fingertips were identified, we fit circles to the contour at the convex point. This allowed us to combine multiple convex points on a fingertip



(a) (b) (c)

Figure 3: Input, hue, and segmented images

into a single representation, since overlapping circles were detected and combined. In order to achieve this for the correct scale of the finger, it was necessary to dilate the region contours to smooth local noise. Only circles between a set range of scales that matched fingertip size at expected ranges were considered. The fitted circles gave better localization than convex points.

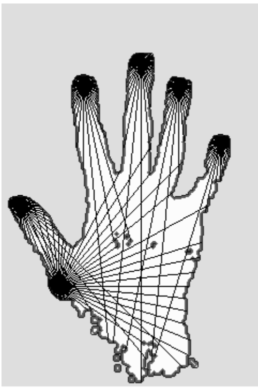


Figure 5: Radiating lines from fingertips

We then accepted or rejected the finger based on the skin region that was visible to the fingertip center in a direct "line-of-sight". Figure 5 shows an example of radiated lines from circles fitted to candidate fingertips that define these "insight" regions. Regions that had multiple lobes had multiple "insight" lobes, or allowed too obtuse an angle between viewable directions were rejected.

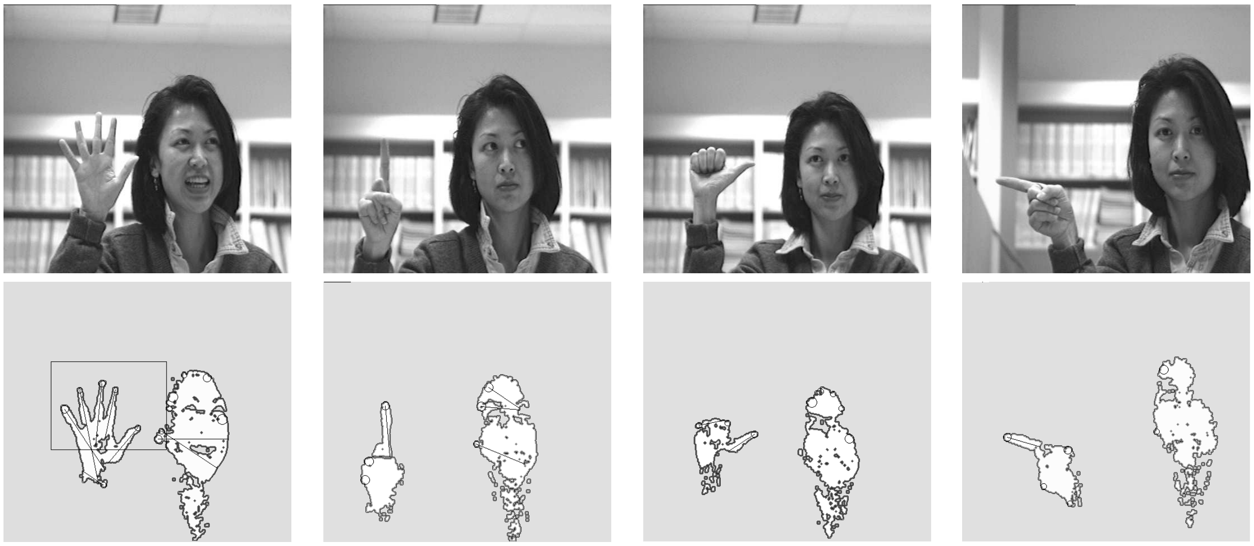
Unfortunately, even after dilating edge contours, the fingertip circle was sometimes smaller than the actual fingertip. While localization remained good, this prevented us from using the circle size as a scale indication for finger size. Noisy edges gave rise to multiple incorrect fingertips. False finger also occurred because of segmentation holes. Overall, the approach rejected many false fingertips. However, in typical noisy images, there were often as many false fingers identified as correct ones. This degree of uncertainty in finger detection led us to implement a higher level verification of fingers as described in the next section.

4.4 Hand localization and gesture recognition

Individual fingers could not be used directly for gesture recognition, as the number of spurious fingers was too great. Since false fingers appeared at random, the chance of false positives was fairly high. We therefore constrained the problem by first identifying the hand, and then looking for gestures only in the local region of the hand in the image.

To localize the hand, we used the distinctive gesture with all fingers splayed out in the traditional "stop" symbol. This gesture has the maximum number of fingers detectable and provides the best opportunity for minimizing false matches. Since the chance of randomly finding four fingers oriented upright and one to the side at a reasonable scale is very unlikely, we had an extremely low false positive rate for hands. Figure 6(a) shows an example of successfully detecting and localizing the "stop" gesture. Once a hand was detected and localized, an ROI was constructed in the vicinity of the hand in the image. Simple one finger gestures could now be recognized by the orientation of a single finger within the ROI.

While the false positives were rare, often individual fingers were missed and prevented the system from locating the hand. However, over a sequence of images, hands were correctly recognized in four out of five frames. Similarly, for single-finger gestures, given suf-



(a) Stop/hand localize (b) Move forward (c) Turn right (d) Turn left

Figure 6: Results of gesture recognition (bottom images must be originals)

ficient frame rates, the correct (stable) gesture could be identified over four to five frames. Figure 6 shows the results of recognizing four gestures used to direct the robot.

A drawback to this approach is the necessary use of the "stop" command to localize the hand. It would be useful to track the hand and adjust the ROI dynamically without requiring the "stop" gesture for each hand movement. As well, by increasing the number of fingers per gesture, we could constrain the problem so as to yield more stable gesture recognition without hand localization.

5 Conclusion

The system can recognize simple gestures in a typical lab environment. It is not unique or trained for a single person and it works from a wide range of distances. Further, the system is quick and can be implemented on current robotic systems. The hue segmentation is able to extract hands from the scene. Finger candidates are easily identified by the convexity feature technique. The geometry and model constraints ensure that there are virtually no false positives and a relatively low rate of false negatives. These techniques represent an important working step in the development of a system that is capable of recognizing a wide range of hand-based gestures.

Future Work

In the future, we would like to achieve hand tracking using an initial hand localization followed by color/optical flow tracking. This would improve the real-time performance of the system. Use of real-time convolver hardware for morphology and encoding would speed the system up to 5 frames per second, enabling temporal features to be included into gestures. Finally, a more complex hand model would allow more sophisticated gestures to be recognized.

References

[1] 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. In *Intelligent Vehicles Symposium*, Tokyo, July 1993.

[2] J. Davis and M. Shah. Visual gesture recognition. In *IEE Proc. Vision Image Signal Processing*, volume 141, April 1994.

[3] Margret M. Fleck, David A. Forsyth, and Chris Breggler. Finding naked people. In *Comp. Vision – ECCV '96, 4^{th} European Conf. on Comp. Vision*, pages 593–602, Cambridge, UK, April 1996.

[4] William T. Freeman and Craig D. Weissman. Television control by hand gestures. In *IEEE Intl. Wkshp. on Automatic Face and Gesture Recogniion*, Zurich, June 1995.

[5] K. Ishibuchi, H. Takemura, and F. Kishino. Real time hand gesture recognition using 3d prediction model. In *International Conference on Systems, Man, and Cybernetics*, volume 5, pages 324 – 328, Le Touquet, France, October 1993.

[6] R. Kahn, M. Swain, P. Prokopowcz, and R. Firby. Gesture recognition using the Perseus architecture. In *Proc. of Computer Vision and Pattern Recognition*, pages 734 – 741, San Francisco, California, June 1996.

[7] Arun Katkere, Edward Hunter, Don Kuramura, Jennifer Schlenzig, Saied Moezzi, and Ramesh Jain. Robogest: Telepresence using hand gestures. Technical Report VCL-94-104, Visual Computing Laboratory, University of California, San Diego, December 1994.

[8] I. J. Ko and H. I. Choi. Extracting the hand region with the aid of a tracking facility. *Electronic Letters*, 32(17):1561 – 1563, August 1996.

[9] C. Uras and A. Verri. On the recognition of the alphabet of the sign language through size functions. In *Internation Conference on Pattern Recognition*, Jerusalem, Israel, October 1994.