Machine Vision and Gesture Recognition

6.866 Machine Vision Term Paper

Ying Yin (986278392)
yingyin@csail.mit.edu

December 9, 2008

1 Introduction

Gesture recognition allows the computer to understand human's body language, and therefore can be useful in many applications for natural human computer interaction. The use of gestures for interaction is especially useful in a virtual reality environment. Recent development in display technologies, like large-scale displays and table-top displays, also opens the realm for developing new gestural interaction techniques. Gesture recognition also has application in robotic control and teleconference.

Although there are special devices, such as Cybergloves, that can be worn to achieve the goal for gesture recognition, such devices are often expensive and unwieldy [4]. Cameras, especially web cameras, are getting cheaper and cheaper. As a result, since 1990s, there has been an increasing amount of research in computer vision to provide a cheaper and more convenient solution for gesture recognition.

Hand gesture is the most expressive part of the body language, and it also presents many challenging computer vision problems due to its complexity. Gestures can be classified into two categories: temporal gestures and static hand postures. This paper reviews some of the vision-based gesture recognition techniques for both categories. Both the merits and limitations of the techniques are discussed.

2 Temporal Gesture Recognition

Temporal gestures are characterized by the movement of body parts in the space. Hence, both spatial and temporal features are needed for recognition. Majority of the methods start by segmenting the images into blobs that model the human body, and then track the features of the blobs. Blob extraction can be based on motion, color, texture, position or model template. Both 2-D and 3-D tracking are possible, although 2-D tracking is generally view-dependent [8]. In the following sections, two papers on temporal gesture recognition with different methods are reviewed.

2.1 2-D Tracking of Motion Blobs

In their 1998 paper, Cutler and Turk [4] presented an optical-flow-based method for recognizing temporal gestures in real time. The motivation of the paper is to explore the use of hand and body gestures as an alternative method to the existing cumbersome input devices for human-computer interaction [4]. They also built a recognition system for children to interact with the computer in a game using hand and body gestures. In particular, they defined six actions they needed to recognize: waving, jumping, clapping, drumming, flapping and marching.

2.1.1 Method

Cutler and Turk used the optical flow method based on region-matching to extract motion blobs from video sequences captured by a single camera, and used features of the motion blobs to classify actions.

The basic idea of the region-based matching technique is to define velocity as a shift(dx,dy) that yields the best fit between image regions at different times [2]. For efficiency reasons, Cutler and Turk minimized sum of absolute distance (SAD) instead of sum-of-squared distance (SSD). After obtaining the flow field, they segment the image into motion blobs by clustering vectors with similar directions together with a region growing method. The motion blobs are then modeled as ellipses with major and minor axes determined.

The recognition of the gestures using the motion blobs are rule-based. For each gueature they defined a set of conditons for the blobs: the number of blobs, the absolute motion of the blobs, the relative motion, etc. The conditions must be satisfied over N consecutive frames in order for the gesture to be classifed as one the six possible actions.

2.1.2 Review

Cutler and Turk employed many simplifications in their method because they wanted to achieve real-time recognition. The simplifications are necessary due to the limitation on processor speed at that time. They used a dual-processor 300 MHz Pentium II PC. In the paper, they explicitly stated a list of requirements they wanted for they system. This provides a basis for the justification of the choices and the simplifications they made.

The assumptions they made on the environment are: (1) the background is relatively static, and (2) the illumination changes slowly [4]. A static background is necessary for extracting the relavant motion blobs successfully, however this assumption limits the generalizability of the method for more common conditions. In a usual interactive environment, even if it is indoor, we would expect that there are other people in the background moving around. The authors did mention that the system worked fine when there were other people in the scene if they did not move much. The system works because only the dominant motions with the largest motion blobs are used in recognition. The second assumption is reasonable because while we expect a robust system to work under different illumination conditions, the illumination usually changes slowly under normal conditions.

They recognize that there are several optical flow algorithms for estimating 2D motion field from image intensities. According to the requirements, they chose to use the region-based matching technique. The reason they gave, citing Barron et al. [2], is that region-based (i.e., correlation-based) matching techniques are more robust to noise than differential techniques. Although Barron et al. [2] mentioned about accurate numerical differentiation might be difficult due to noise, the evalutation of various optical flow methods reported in the paper [2] shows that the matching method did not perform as well as the differential method for both the synthetic and real image data. The citation used by Cutler and Turk here is not very accurate.

The authors also mentioned other reasons for choosing region-based matching method: (1) it works well even when the images are interlaced or decimated (which is one of their requirements); (2) citing Camus [3], correlation-based teniques are less sensitive to illumination changes between image frames than the differential methods. These reasons are reasonable because the the matching method finds the best matching pixel shift even though all matching strengths are low. Conversely, the differential method's constant brightness contraint would not apply since the image intensity does not remain constant when illumination changes.

The region-based matching technique can be computationally expensive. The authors introduced more simplifications so that computaton could be done in real time. As they assume that the background is relatively static, optical flow is only estimated for pixels where a motion is detected. The motion detection criterion is based on the temporal derivative of the image intensities at each pixel (x,y) which is thresholded to produce a binary mask M(x,y,t). The final motion mask M' is defined as:

$$M'(x, y, t) = M(x, y, t-1) \wedge M(x, y, t).$$

The flow at pixel (x,y) is calculated only if $M'(x, y, t) = 1$ which means pixel (x,y) is moving at time t (see Figure 1). This is a clever method because not only does it reduce the the number of pixels to compute optical flow, it also eliminates the problem of computing the flow for background pixels near the occlusion boundaries of the moving objects. The matching method usually requires confidence measure to evaluate the goodness of the match. Cutler and Turk did not use any confidence measure because the use of motion mask M' eliminated the two most common flow estimate errors, namely occlusion boundaries and low texture areas [4].

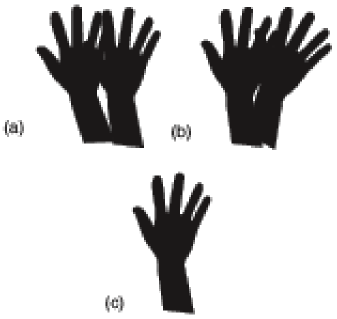

Figure 1: Mask applied to a hand moving left to right. Let A, B, and C be the hand at time t-1, t, and t+1. (a) $M(x, y, t-1) = A \vee B$; (b) $M(x, y, t) = B \vee C$; (c) $M'(x, y, t) = B$ [4]

Correlation matching can fail dramatically when used in arbitrary situations. It appears to work reasonably well if certain control strategy such as coarse-to-fine sequential processing of the images is employed [1]. Cutler and Turk did not use any control strategy and added many simplifications. In searching for the candidate matches, they simplified the search area to be along the X, Y, and diagonal directions. This is a very big simplification. However, as they only need to use the optical flow to combine similar motion vectors into blobs, they may not require an accurate motion field estimate. They claimed that it gave good results without giving any statistics, therefore it is not obvious how well their method worked.

A major drawback of the paper is that they did not report any systematic experiments being done or the accuracy of the recognition. They only reported that they tested the application with children and adults interacting using the predefined gestures, and the participants found it to be fun and intuitive. However, this does not say much for a scientific paper.

The idea of rule-based classification may work for a small of set of gestures, but it is not clear how well it can scale with more kinds of gestures. The conditions for some of the gestures can be very close (e.g., “drumming” and “jumping”). It is not shown how well these two gestures can be differentiated. In addition, the rules only work when the person is directly facing the camera. This view-denpency is also a shortfall of 2-D tracking.

Even the application did work well according to the simplified requirements, it is not obvious how the method can be generalized for a more complicated environment with more gestures to be recognized. The ultimate goal is gesture interaction with the computer, not just for a simplified application which may act as a testbed.

2.2 3-D Tracking of Skin Blobs

Different from Cutler and Turk, Shin et al. [7] extracted the hand from the background based on the skin color. 3-D position, instead of 2-D, of the hand is tracked by stereo cameras. Their main goal is to develop gesture-controlled visualization and manipulation techniques that can aid the exploration of complex scientific data on large-scale displays.

2.2.1 Method

Starting from simple commands for visualizing data, they wanted to recognize three basic gestures: zoom, rotation and translation. The hand blob is extracted based on skin statistics. The gesture on and off (start of manipulation) are determined according to the change of the area of the hand blob. Only the motion of the manipulating hand is tracked. The 3-D trajectory of the hand is computed and fitted into a Bezier curve. Classification of the gestures is based on the feature of the resulting curve.

2.2.2 Review

The authors’ assumptions about the environment are: (1) the background is not static, and (2) the interacting person (not only arms or hands) is moving. However, only the hand gestures need to be recognized. Hence, the use of skin color based segmentation for the hand blob is reasonable.

The authors presented detailed experiments and statistics for evaluating their methods. They broke down the evaluation into each stage of the recognition process: hand detection, manipulation detection, and gesture recognition. The overall performance is also reported. This allows readers to decern how well the methods work for each stage, and how well the methods can be generalized for other settings. The algorithm correctly recognized 93 out 100 gestures [7].

The experimental data was captured in a indoor setting with varying natural light and a complex background. Four different people with various skin tones participated in the experiemnts [7]. The variations of the lighting condition and the skin tone are important for the experiment setup, especially when the hand blob is extracted based on color. Depending on the color space used, the color intensities have different sensitivities to lighting effects. Shin et al. [7] used SCT (spherical coordinate transform) color space to reduce lighting artifacts.

Besides the skin color, the depth distance of the skin region is also used in identifying the manipulating hand because there are other skin regions in the image (e.g., face or other people's hands). They assume that the hand gesture occurs in front of the body, therefore the skin region closest to the camera is identified as the manipulating hand. While this assumption may be true for most of the time, it is not always valid. Their experiments also show cases where the face is mistaken for the hand because they are at the similar distance. This situation may not be just incidental. For example, the "zoom out" gesture involves moving one's hand towards the body which makes the hand close to the face. While it is possible to restrict the interacting person to make sure his/her hand is well in front of the body, this violates their initial intention of developing natural and intuitive gestures for manipulation. In addition to distance, other features can be used for differentiating skin regions, like area or motion.

Recognition of the "zoom" gesture requires the depth information of the hand because the hand is moving along the z-axis perpendicular to the large display. The 3-D position data is obtained from a Digiclops system based on a triangulation between three cameras [7]. Although the relative position of the cameras is fixed, the position of the stereo system in the environment could change. The authors did not state where they mounted the cameras. The video frames given in the paper show that the camera is directly in front of the person making the gesture (See Figure 2(a)). However, if the goal is for manipulating the data on a large display (see Figure 2(b)), we would imagine that the cameras are mounted above the screen and the person at an angle. In that case, additional calibration for the exterior orientation of the cameras is needed.

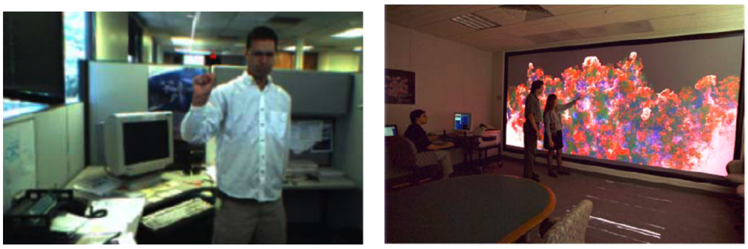

Figure 2: (a) Experiment condition from Shin et al. [7]. (b) A large display for interactive applications [7].

For a proof of concept, the simplification of the camera position is probably acceptable provided that the methods are generalizable. However, the manipulation detection is based on the area change of the hand. It is assumed that the area of the hand decreases when the hand closes signaling the grabbing of the object and the start of manipulation. This is true only when the plane of the palm directly faces the camera and the screen when the hand is opened. If the camera is above and at an angle to the person, it is not clear whether this area-change method can work.

2.3 Comparison

Accuracy is an important measure for the success of a recognition method. However, as the intention and the application of the above mentioned methods are different, it is difficult to compare in this dimension. In addition, Cutler and Turk did not report any statistics on their method based on motion blob extraction. The main focus, hence, is on the generalizability of the method, meaning how well the method can be applied in other common situations.

One way to assess generalizability is to evaluate the assumptions that are necessary for the method to work. Fewer assumptions implies that there are fewer constraints, and hence the method is likely to work under more general conditions. The validity of the assumptions is also important. Using motion blobs for tracking gestures requires the background to be static. However, using skin blobs does not require this assumption.

On the other hand, the idea of motion blob extraction allows recognition of body gestures (arms and legs) because it is not confined to detecting skin color of the hand. It would be difficult to extract blobs for body gestures based on color because of the large variety of the clothing.

It would be interesting to combine the two methods, extracting both the skin and motion blobs, for gesture recognition. In this ways, the benefits of the two methods can compensate their shortcomings. With the increasing computational speed, this can be possibly done in real-time.

Another difference is that Shin at el. used 3-D tracking while Cutler and Turk used 2-D. While the choice may be application dependent, 3-D tracking, though more complicated, should be applicable to more situations for spatial invariant recognition.

3 Hand Posture Recognition

While temporal gesture recognition focuses mainly on the high level hand (or other body parts) motion, hand posture recognition focuses on determining the hand orientation and finger shapes. It is at a finer level of granularity in terms of gesture recognition. Hand postures not only can express some concepts, but also act as some specific transitional states in the temporal gestures [8]. As a result, recognizing hand postures is also a main topic in gesture recognition [8].

Human hand is especially rich in shape variation, and this adds to the complexity of estimating hand postures. Many methods recognize hand postures based on geometric features such as finger tips, finger direction and hand contours [8]. They differ, however, in the ways to extract and calculate these features from images. The following sections review two of such methods.

3.1 Shadow-Based Pose Estimation

Segen and Kumar [6] used a single camera and a point light source to track a user's hand posture in 3-D. The depth information is obtained from the images of the hand and its shadow. Their goal is to develop hand tracking system that can be used as an input interface for applications that require multidimensional control. This is because that, in addition to intuitive, hand gestures can offer higher dimensionality due to its high degrees of freedom.

3.1.1 Method

The system uses a point light source and a single camera, both mounted above a table. The camera is pre-calibrated with respect to the table. The system classifies gestures into four categories: point, reach, click and ground (which covers all other gestures). The algorithm starts with the 2-D analysis of the image. The boundaries of the hand and the shadow regions are obtained from background subtraction. The local curvature extrema of the boundaries are identified, and "peaks" and "valleys" are labeled on the boundaries. The peaks and valleys are also differentiated for the hand and the shadow based on the hue and the saturation values.

If a point gesture is detected, the system continues to find the 3-D position and the orientation of the tip of the pointing finger. The shadow image provides the additional information for calculating depth. If **P** is the point of the finger tip in the table coordinate frame and **S** is its shadow (Figure 3), the transformation M_t from **P** to **S** can be determined given the light source location. If **p** is the image of **P** as seen by the camera with transformation matrix M_c and **s** is the image of **S**, **s** can also be seen as the image of **P** as seen by a second camera with transformation matrix M_cM_t. Hence, the problem is translated to a standard stereo problem.

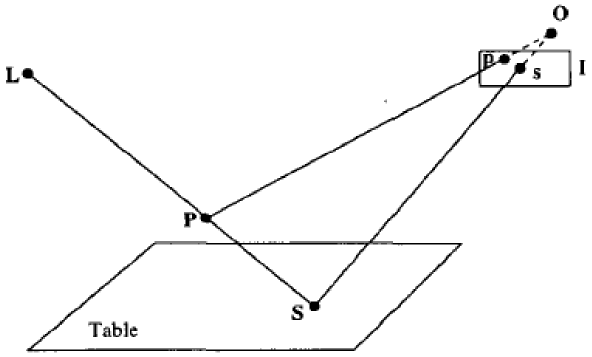

Figure 3: Imaging geometry from Segen and Kumar [6].

3.1.2 Review

The use of shadow for calculating depth information is interesting. Accurate calibration is important for a stereo system. The authors gave a description for how they calibrated the light source, but not for the calibration of camera's position and orientation relative to the table. They used a block of known height for calibrating the light source position with several measurements. An overdetermined system of equations are obtained. They did not mention whether they used a least-squares method to minimize error or not.

There are some drawbacks of this shadow approach, One limitation, also mentioned in the paper, is occlusion. When the hand is very close to the table, the shadow is hidden by the hand, and as a result, 3-D pose cannot be computed. The authors suggested to keep the hand at-least 15cm above the table. However, when the hand is too high above the table, the shadow can be less sharp, and its boundary may not be accurately defined. It is not clear also whether the system allows other lighting sources in the environment because they will affect the shadow too.

Another question the paper did not address is the reason for using the "shadow gesture". A pair of cameras can achieve the same function. Using shadow adds other limitations, and the benefit is not obvious. Calibration for the relative orientation of the cameras is avoided in this method, but there is an additional calibration for the light source.

3.2 Silhouette-Based Pose Estimation

Huang and Lin [5] presented algorithms for curve and peak detections from the silhouette pattern of a hand image. They did not mention any specific application or goal when developing the methods.

3.2.1 Method

The hand is segmented from the background using Support Vector Machines (SVMs) as the binary classifier. The feature vectors are the RGB values of the pixels. The silhouette is obtained using an erosion operator on the binary image of the hand.

The curve detection algorithm steps through the silhouette pattern block by block, calculating the frequencies of the occurrence of the 8 direction patterns (shown in Figure 4) in each block. If the silhouette pattern in a block is more straight, the frequencies of a certain direction mask and its neighboring directions will be high, while the other frequencies will be low. If the silhouette pattern in a block is more curved, the frequencies will be more uniform for more directions. The frequencies form a vector $\mathbf{v}$, and the norm of the vector $\|\mathbf{v}\|$ indicates the curvature of the silhouette pattern in a block. The peaks of the hand pattern is implied as they occur at the block locations where $\|\mathbf{v}\|$ is small. Statistical classification of gestures is done based on the feature vector which includes $\|\mathbf{v}_i\|$ for each block i.

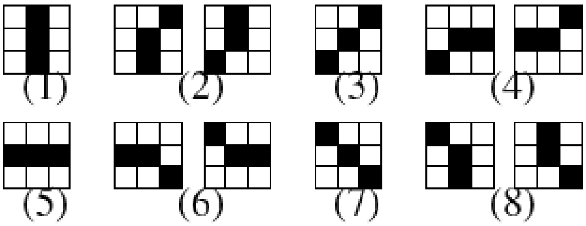

Figure 4: 8 direction masks [5].

3.2.2 Review

In this paper, the explanation of the method is not clear, and sometimes conflicting. It is worsened by many spelling and grammar mistakes. The use of direction masks to evaluate curvature is reasonable though.

The extraction of hand from the background is based on a classifier trained with one sample image. Variations in the lighting condition and the skin tone are not considered.

The authors did not state any specific goal and what kind of hand postures they wanted to recognize at the beginning. Although they showed good recognition rates from the experiment, the result is not convincing. The recognition is based on three gestures, but the choice of the gestures seems arbitrary (Figure 5). The good recognition rates may simply due to the fact that the three gestures are easily discernible. It would be more useful to test similar postures, like postures with same number of fingers extended, and more complex postures with self-occlusion.

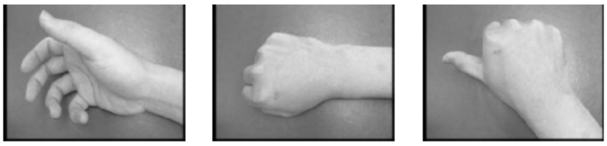

Figure 5: Three examples of the gestures classified by Huang and Lin [5].

3.3 Comparison

Both methods use table tops as background for the obvious simplicity. The background is also assumed to be static and simple under constant lighting condition. For obtaining the hand contour, both methods use simple boundary extractions. Segen and Kumar used background subtraction, while Huang and Lin used maximum-margin binary classification. The simple background subtraction is sensitive to noise, and Segen and Kumar [6] did suggest the use of heuristic screening to discard unrelated regions. No edge detection method is used or considered which may actually generalize better and make less stringent assumption on the background.

Both methods evaluate the curvature of the hand contour. Segen and Humar used a k-curvature measure to find the angle between two vectors connecting three points on the contour to locate local curvature extrema. This method seems to be more direct and efficient compared to the direction mask method used by Huang and Lin, although no specific results were reported by Segen and Kumar.

Based on the description, both methods seem to work only when the plane of a fully-opened hand is parallel to the table. If not, self-occlusion will occur, and the evaluation of hand contour and the detection of finger tips will fail. However, the parallel assumption is not stated in either paper.

The success of the two methods in estimating hand postures is limited. This is not surprising as hand posture recognition is a hard problem to solve.

3.4 Conclusion

Gesture recognition is a multidisciplinary area involving computer vision, machine learning and psycholinguistics [8]. Feature selection and data collection are important and non-trivial tasks in visual gesture learning, and these are the areas where machine vision has the direct application. Four papers related to using machine vision techniques to solve gesture recognition problems are reviewed. Many methods in machine vision are shown to be useful in gesture recognition. These include optical flow, feature extraction from binary and color images, image segmentation, stereo imaging, and many others. Development in the machine vision field would help the advancement of gesture recognition. However, more importantly, the choice of the right technique for the right application is also crucial for the successful implementation of a recognition system.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. Technical report, Amherst, MA, USA, 1987.

[2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow tech-

niques. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 236–242, 1992.

[3] T. Camus. Real-time quantized optical flow. *Proceedings of IEEE Conference on Computer Architectures for Machine Perception*, 3:71–86, 1997.

[4] R. Cutler and M. Turk. View-based interpretation of real-time optical flow for gesture recognition. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:416, 1998.

[5] C.-H. Huang and D.-T. Lin. Fast silhouette-based hand gesture feature extraction algorithm. *Multimedia and Expo, IEEE International Conference on*, 0, 2001.

[6] J. Segen and S. Kumar. Shadow gestures: 3D hand pose estimation using a single camera. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1:–485 Vol. 1, 1999.

[7] M. C. Shin, L. V. Tsap, and D. B. Goldgof. Gesture recognition using bezier curves for visualization navigation from registered 3-d data. *Pattern Recognition*, 37(5):1011 – 1024, 2004.

[8] Y. Wu, T. S. Huang, and N. Mathews. Vision-based gesture recognition: A review. In *Lecture Notes in Computer Science*, pages 103–115. Springer, 1999.