A Framework for Recognizing Hand Gestures

David Knight, CS229 Student
Matthew Tang, CS229 Student
Hendrik Dahlkamp, Mentor
Christian Plagemann, Mentor

December 10, 2010

1 Introduction

Traditional methods for user input, consisting of fixed key input and point-click devices, are slowly being supplemented and enhanced by touch technologies and their associated touch gestures. While these methods have proven effective, they are limited in scope in that they require a physical object or surface with which a user must interact.

The use of hand gestures in free space is often seen as an intuitive next step in the progression of user input technologies. A system that takes cues from such gestures would unshackle a user from physical devices. Using such systems, a user only needs their body motion to signal specific actions that can be picked up by a camera and interpreted by appropriate computer vision algorithms.

In this paper, we explore the use of machine learning to process and interpret image sequences in order to correctly recognize and classify a couple simple hand gestures. Our work is part of a larger body of research by computer science Ph.D. student Hendrik Dahlkamp and computer science Postdoctoral researcher Christian Plagemann to create a gesture-based UI that allows users to manipulate virtual objects in 3D space. An active infrared camera is used to capture both infrared intensity data as well as depth information. Unlike a passive RGB camera, the depth information captured by the infrared camera greatly simplifies segmentation of foreground objects. Existing components of Dahlkamp's vision system can track the movement of a hand once it has been recognized as such, crop out the region around the hand, and perform the necessary background subtraction to produce the training sets we use. Recognizing a hand gesture, then, requires that we be able to 1) differentiate a "hand image" from a "nonhand image", and 2) correlate a sequence of hand images with a specific gesture. This paper comprises the development of these two functional blocks using machine learning techniques.

2 Hand Classification

2.1 Training Data

Given an image, we need to be able to classify it as a "hand image" or "nonhand image". To do this using a machine learning algorithm, we need an appropriate set of training data. This data is obtained by capturing typical scenes of people giving hand gestures in a room with an active infrared camera. The pre-existing hand tracker provided by Hendrik Dahlkamp then crops out 32x32 pixel, 8-bit grayscale images from these scenes and performs the appropriate background subtraction on them. Some of these images contain hands, while others contain various other objects in the scene.

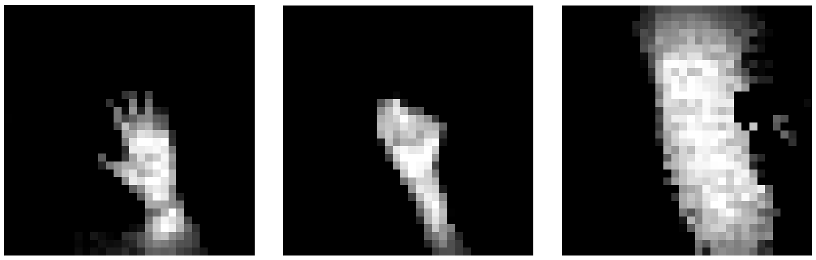
(a) Hand Image (b) Hand Image (c) Nonhand Image

Figure 1: Sample “hand” and “nonhand” images.

Labeling these images is a matter of separating the “hand” images (e.g. Figs. 1(a), 1(b)) from the “nonhand” images (e.g. Fig. 1(c)).

2.2 Feature Selection

The main features for characterizing our various images are normalized bin values from a Pyramid of Histogram of Oriented Gradients (PHOG). Obtaining a Histogram of Oriented Gradients (HOG) involves calculating a gradient direction and magnitude for every pixel in the image and binning these gradients by their direction with a weight based on their magnitude [1]. PHOG extends this method by calculating a histogram for a region, subdividing the region into sub-regions, calculating histograms for each sub-region, and repeating this process. We used an implementation of PHOG created by the Visual Geometry Group at the University of Oxford [2], allowing us to configure subdivision level depth, bin resolution, and histogram range.

Our use of HOG is motivated by the similarity of our problem of recognizing hands to the objective of detecting humans in the original HOG paper [1]. Additionally, intuition suggests that gradient direction information should be able to highlight differences between a hand versus other blob shapes. PHOG gives us the added flexibility of specifying sublevels on which to take HOG measurements, in addition to other HOG related parameters. The exact parameters we used were chosen by performing 70-30 cross validation on our test data.

2.3 Training and Testing

Recall that our goal is to label and track hand positions. We can therefore afford to occasionally classify a “hand” as a “nonhand” since we would, over a large number of frames, overwhelmingly label the image content as a “hand”. Once a “hand” has been confirmed over a few frames, the object tracker will start providing a steady stream of known “hand” images. However, we can ill afford to classify a “nonhand” object as a “hand” on more than a few occasions, since the hand tracker would start tracking this “nonhand” object. The precision of our algorithm, then, is more important than its recall.

Using a simple Support Vector Machine (SVM) with a linear classifier, and using 70-30 cross validation to train and test, we find that we are able to quickly achieve a precision of 99.79% and a recall of 86.61% on a training set of size 27,000. These results were obtained by using PHOG with 8 bins/level, and 2 sublevels, resulting in a feature vector size of $8(1 + 2^2 + 4^2) = 168$.

We next add to our feature vector a histogram of intensities, binning intensity values along with our original oriented gradient values. This quick addition, effectively doubling the length of our

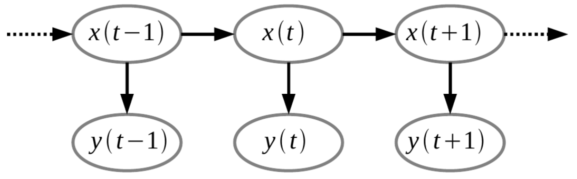

Figure 2: Hidden Markov Model.

feature vector to 336 dimensions, manages to improve our precision to 99.90% and our recall to 90.96%, which is sufficient for our purposes.

3 Gesture Classification

3.1 Hand Position Discrimination

In the previous section, we were able to differentiate between "hand" and "nonhand" images with high accuracy. That classifier, used in conjunction with the hand tracker, will now give us a sequence of hand images from which we can infer gestures. To do so, we first characterize a gesture with a few defining positions. For example, a grasping motion might be characterized by an open palm followed by a closed fist (or any number of intermediate positions; for the sake of simplicity we will consider only a two state model for our gesture). Thus, one way to recognize a gesture from our set of images is to label any sequence consisting of an "open hand" followed quickly by a "closed hand" as a "grasp" gesture.

To classify hand positions, we take our image sequence and manually label the images preceding a grasping motion as "open" and label the images following a grasping motion as "closed". We then attempt to train a machine learning algorithm on this set of data in order to give us an initial classification of hand states. We find that with PHOG and binned intensity values, we are unable to achieve an accuracy greater than 85%. Various attempts to improve accuracy using these features (e.g. using a Mixture of Gaussians model) proved ineffective. Our improved approach is to use the images' pixel intensity values as a feature vector coupled with a SVM with a second order polynomial kernel. This method yields a classifier with 95% accuracy (90-10 cross validation on a training set of 4,500) and is used in the final recognition system.

3.2 Modeling Image Sequences with a Hidden Markov Model

So far, we have attempted to label every image in our sequence independently, ignoring the correlation between temporally adjacent images. We can exploit this temporal correlation by modeling the sequence with a Hidden Markov Model. Suppose that at time t the hand we are tracking has a true state ("open" or "closed") given by x_t. Given an image of the hand y_t, we can calculate $p(y_t|x_t)$ by fitting a logistic function to the margin output of the SVM classifier. Furthermore, we can estimate $p(x_{t+1}|x_t)$ by counting transitions in our training sequences. That is, given a typical sequence of $x_0, x_1, \ldots, x_N$ ordered in time, we estimate $p(x_{t+1} = a|x_t = b) = \frac{1}{N}\sum_{n=0}^{N-1} 1\{x_{n+1} = a\}1\{x_n = b\}$.

Using forward recursion [3] we can calculate, on the fly, $p(x_t|y^t)$ where $y^t = \{y_0, y_1, \ldots, y_t\}$. This allows us to take into account all previous observations rather than just the current observation. To improve the accuracy of our estimate x_t, we can delay estimation until time $t+d$, allowing us to use forward-backward recursion to calculate $p(x_t|y^{t+d})$. This will introduce a delay d in our system, but

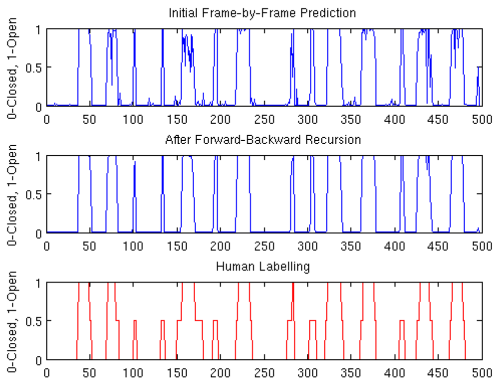

Figure 3: Predictions on a captured 30 frames/s image sequence (shown as Gesture Sequence 1 in Figure 4). Images that did not resemble either "open" or "closed" hands were labelled 0.5. Introduced a 3 frame delay for the forward-backward recursion. All parameters were obtained by training on 9 other gesture sequences of similar length (90-10 cross validation).

allow us to consider future observations in our estimation of x_t. We note here that the larger the delay d, the larger the number of running estimates we need to store, since $p(x_t|y^{t+d})$ is calculated from $p(x_t|y^t), p(x_t)|y^{t+1}), \ldots, p(x_t|y^{t+d-1})$; however, this extra storage can be considered negligible because the delay d should only be on the order of a few frames. The fundamental tradeoff in choosing d is therefore accuracy vs. system response time.

We can now define "grasp" and "release" gestures as points in the frame classification output where confidence values transition from <0.5 to >0.5 and >0.5 to <0.5, respectively. Without using forward-backward recursion, the raw frame-by-frame output of classification is fairly noisy as is shown in Figure 3. Results in Figure 4 show that trying to identify gestures from the raw output results in an overall high number of false identifications. In particular, some gesture sequences show more false identifications than true identifications when using the raw output. Applying forward-backward recursion significantly reduces the number of false identifications, retains most of the true identifications, and slightly increases the frame classification accuracy.

Finally, we note that there is a large amount of freedom in choosing criteria for a grasping event. For example, we can declare a gesture occurrence when our estimated state is "open" for more than some fixed number of frames, followed by "closed" for some fixed number of frames. Such a criterion would filter out overly quick open-close-open events, requiring the user to hold his/her hand closed for a fixed duration before a grasping event is recognized. Additionally, the derivative of the classification output at a transition point can provide information about the speed of a gesture. Different gestures can be defined for the same basic motion that occurs at different speeds.

4 Conclusion

While using forward-backward recursion greatly reduces the number of false gesture identifications, a few false identifications remain in many gesture sequence tests. Most of the anomalies come from interpreting quick bursts of classification uncertainty as a full transition and can likely be addressed by imposing more advanced criteria for registering a gesture as was explained in the

Gesture Sequence	[FBR / No FBR] True "Grasp"	[FBR / No FBR] True "Release"	[FBR / No FBR] False "Grasp"	[FBR / No FBR] False "Release"	"Grasp" Ground Truth	"Release" Ground Truth	[FBR/ No FBR] Frame Accuracy
1	9 / 9	9 / 9	0 / 5	0 / 5	9	9	99.8% / 98.3%
2	5 / 5	6 / 6	2 / 4	1 / 3	5	6	94.9% / 91.8%
3	11 / 11	11 / 11	1 / 4	1 / 4	11	11	96.0% / 95.4 %
4	8 / 9	8 / 9	1 / 5	1 / 5	9	9	97.1% / 96.6%
5	6 / 7	7 / 8	3 / 17	3 / 17	8	9	72.6 % / 74.2 %
6	5 / 5	4 / 5	4 / 11	5 / 12	5	5	91.4% / 87.0%
7	10 / 10	9 / 9	3 / 13	3 / 13	10	9	94.7% / 92.6%
8	7 / 7	7 / 7	1 / 3	2 / 4	7	7	94.5% / 92.1%
9	10 / 10	10 / 10	0 / 2	0 / 2	10	10	97.6% / 97.2%
10	8 / 8	9 / 9	1 / 3	1 / 3	10	10	96.7% / 95.8%

Figure 4: Gesture recognition results both with and without forward-backward recursion (FBR). Gestures were identified by state transition without consideration for gesture speed. Results for each sequence were obtained by training on the 9 other gesture sequences of similar length (90-10 cross validation).

previous section. Also, forward-backward recursion slightly decreased the number of true gesture identifications in a few of the sequence tests. The exact cause of this effect needs to be further investigated, but it might actually be desirable behavior if the gestures that are being skipped over are due to uncertain user input. The next step in this project is to implement the gesture recognition system in C so that it can be integrated with the rest of the vision system developed by Dahlkamp and Plagemann. This integration will allow the vision system to be tested as a whole.

Overall, we have provided a basic framework with which to classify hand gestures. Given appropriate methods for distinguishing between "hand" images and "nonhand" images, and for tracking a hand once it has been identified, we can produce a sequence of hand images that can be further analyzed for the purposes of gesture recognition. Given a specific gesture we want to detect, we identify key positions of that gesture and develop classifiers that are able to correctly bin the captured images into one of the identified key positions. We can then estimate transition probabilities between these positions by counting and averaging transition occurrences in a typical sequence of gestures, allowing us to use forward-backward recursion to obtain accurate estimates of the underlying state at any given time. Various criteria can then be used to recognize a gesture (e.g. when these key positions have been realized in a certain order), providing us with a robust hand gesture detector.

References

[1] N. Dalal, B. Triggs. "Histograms of Oriented Gradients for Human Detection". *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, 2005: 886-893.

[2] A. Bosch, A. Zisserman, "Pyramid Histogram of Oriented Gradients (PHOG)". *University of Oxford Visual Geometry Group*, http://www.robots.ox.ac.uk/ vgg/research/caltech/phog.html.

[3] L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes". *Inequalities*, vol. 3, 1972: 1-8.