Interpreting American Sign Language with Kinect

Frank Huang and Sandy Huang

December 16, 2011

1 Introduction

Accurate and real-time machine translation of sign language has the potential to significantly improve communication between hearing impaired people and those who do not understand sign language. Previous research studies on computer recognition of sign language have taken input from technology including motion-capturing gloves and computer vision combined with colored gloves. These projects commonly apply Hidden Markov Models or neural network systems [1].

We created a system that recognizes American Sign Language (ASL) using the Microsoft Kinect sensor, in particular its skeletal tracker. Such a system has the additional benefit of extending the Kinect to be a platform for people to learn and practice sign language. The Kinect sensor is an exciting device to use for human gestural applications because of its ease of use: it does not require gloves or special markers for tracking, it does not require a background image or room calibration, it works well even in low lighting, and it is also quite inexpensive.

We were able to train a single-sign recognizer to recognize ten ASL signs with a cross-validation accuracy of 97%, using LIBSVM [2]. We also developed a basic way to segment a sequence of signs, in order to translate an entire sign sequence.

2 Methodology

2.1 Training data

We recorded training data for 10 different ASL signs with the Kinect sensor, using the Microsoft skeletal tracking solution. Training data is in the form of pre-segmented single signs. We have about 100 training examples – 50 each from two people – for each of the ten signs: “baby,” “day,” “drive/car,” “I/me,” “past,” “now,” “sleep,” “tonight,” “wash,”

and "with." The signs we chose involve arm movements, rather than individual finger movements, since the Kinect sensor does not resolve the hands or fingers clearly at typical skeletal tracking distances (6 ft) [3].

For each sign in our training set, a sequence of skeleton frames (composed of 20 joint positions per frame, defined by x, y, and depth coordinates) is saved. Each sign's frame sequence contains on average 20 frames, making for around 20,000 frames in total. Joints are normalized relative to the HipCenter joint: their positions are relative to that joint, while the HipCenter joint's position is absolute. Data is captured at 30 FPS, although frame skipping sometimes occurs due to processing load.

2.2 Single-sign classification

Our single-sign classifier takes as its input a sequence of frames that compose a single sign already segmented out of a full sequence, and predicts which sign it is. To achieve this, we first train an SVM (using LIBSVM) on the individual frames of the training data (e.g. a "drive" sequence becomes around 20 frames which are all labeled "drive"), using a set of frame-dependent features determined by feature selection. When the classifier is asked to predict on a test sequence, it classifies each frame separately. The result is then selected by simple majority – the classification with the most frames assigned to it.

We pre-processed the individual input frames into feature vectors suitable for entry into LIBSVM, creating a library of almost 100 features that includes:

- {x, y, depth} of {right, left} {hand, wrist, elbow, shoulder} position
- {x, y, depth} of separation between right and left {hand, wrist, ...}
- {x, y, depth} of {right, left} {hand, wrist, ...} velocity (computed from neighboring frames)

For example, one specific feature was the x coordinate of the right hand position. For improved accuracy, we selected a subset of features from this set by running an automated feature selection process, using filter feature selection [4]. Since our feature values were continuous, we elected to use the correlation between feature values and class labels for feature selection. In order to fit this multi-class case, for each feature we computed 10 correlation values: its correlation with each of the 10 class labels. We then computed the variance of this set of values and picked the K features with the highest "correlation variance," the intuition being that these features capture the differences between the labels.

To select K for the above process, we used model selection by running 10-fold cross-validation using the top K features, varying K. Ultimately we found that K = 70 gave the best classification performance (Figure 1).

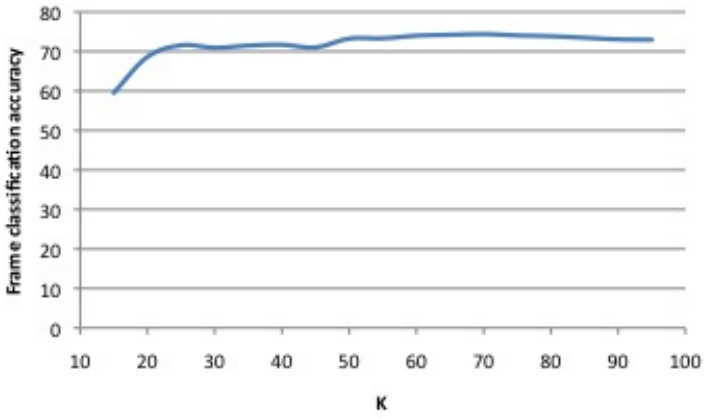

Figure 1: Frame classification accuracy for varying values of K.

2.3 Segmenter

We also worked on a system to solve the second problem in real-time recognition: segmentation. After some experimentation, we decided to use the following approach. The segmenter keeps track of a sliding window of K frames (through experimentation, we chose $K = 10$). As frames are read from the input stream, the segmenter runs the SVM classifier on each frame to produce a frame label. The sliding window contains the last K frames seen, and the simple majority of these K labels is taken. If the majority label changes, the segmenter reports that a new sign has begun.

3 Results

Our single-sign classifier achieves sign classification accuracy of 97% on pre-segmented signs when running 10-fold cross-validation on our training data. As mentioned, sign classification involves running the classifier on each individual frame of the sign sequence, and then taking the simple majority of the class labels to determine the overall label for the sign. The accuracy of our model in classifying individual frames was about 80%.

Some of the top features selected by feature selection included:

- R wrist–L elbow depth difference (+6 other similar differences)
- R wrist depth position (+1)
- R wrist–L shoulder depth difference (+1)

- R wrist–L hand Y difference (+1)
- L hand X position

We chose the following sign sequences to test our segmenter: "I wash baby tonight," "I wash car past," "Today I drive with baby," and "I sleep now." (Note: "today" in ASL is signed by "now" followed by "day.") These sequences cover all ten signs. Our approach to segmentation produced recognizable but noisy results (i.e. the classifier reported incorrect signs between the "ground truth" signs). Even after some filtering in the segmenter – ignoring the first few frames in the input, not allowing for segmented sequences that are too short, and so forth – there was still a considerable amount of noise in the segmenter output. We conclude that a more sophisticated segmentation approach would be necessary to achieve realistic accuracy for a real-time sign language translator.

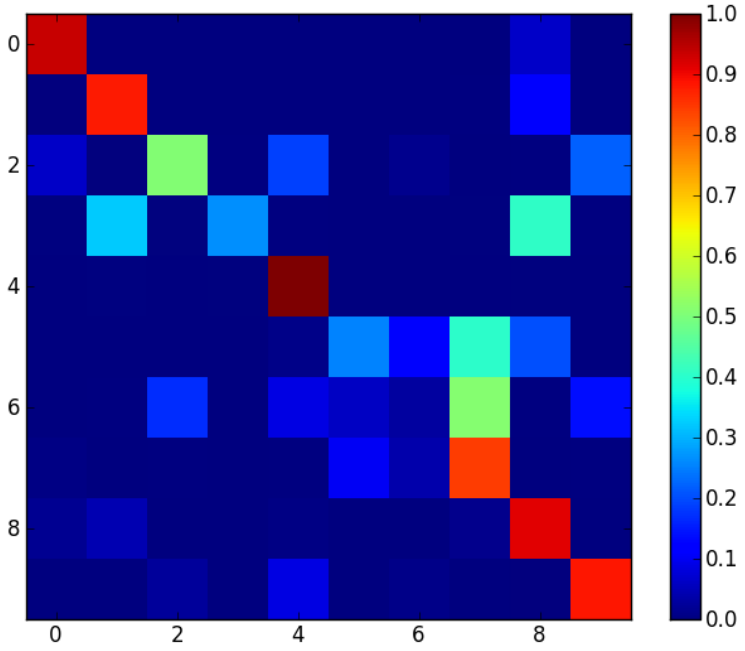

Figure 2: Confusion matrix generated from cross-validation of frame classification with SVM. The left axis corresponds to indices of correct labels, and the right axis to indices of predicted labels. The mapping from label to index is as as follows: "past": 0, "wash": 1, "tonight": 2, "sleep": 3, "baby": 4, "now": 5, "with": 6, "drive": 7, "me": 8, "day": 9

Based on the confusion matrix generated from cross-validation of the SVM classifier (Figure

2), we found the signs "now" and "with" were frequently mis-classified as "drive". In the future, features describing the correlation of joint velocities (e.g. whether two joints are both traveling in the same direction) will likely reduce this mislabeling.

4 Conclusion

The high accuracy of our single-sign classifier shows it is possible to classify ASL signs with only joint position data. Many ASL signs involve gestures using the arms, and the Kinect is designed to capture large body movements such as these. However, using skeletal data from the Kinect to recognize signs has limitations. In particular, the inability of the Kinect to detect hand shapes and finger movements makes it difficult to interpret certain ASL signs, especially letters, which involve small and subtle finger motions.

Our greatest challenge was solving the segmentation problem. Similarly to speech, ASL is not signed with obvious pauses in-between signs, nor do fluent signers return to an easily-recognizable"neutral pose" in-between signs. Consequently, to people unfamiliar with ASL, it can be difficult to recognize where one sign ends and another begins. In order for our ASL recognizer to be effective in real-time, it will need to be able to segment a series of signs.

The advantage of using the Kinect rather than other technology to capture signs is that such a system has the additional benefit of extending the Kinect to be a platform for people to learn and practice sign language.

5 References

[1] Parton, B. S. (2006). Sign language recognition and translation: A multidisciplined approach from the field of artificial intelligence. *Journal of Deaf Studies and Deaf Education, 11*(1): 94-101.

[2] Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*(3): 27:1-27:27.

[3] Yin-Poole, W. (2010). Source: MS quadrupling Kinect accuracy. *Eurogamer*. Retrieved from http://www.eurogamer.net/articles/2010-12-17-source-ms-quadrupling-kinect-accuracy

[4] Ng, A. (2011). CS 229 lecture notes: Regularization and model selection. *CS 229 Machine Learning*. Retrieved from http://cs229.stanford.edu/notes/cs229-notes5.pdf