An Integrated RGB-D System for Looking Up the Meaning of Signs

Christopher Conly, Zhong Zhang, and Vassilis Athitsos
Department of Computer Science and Engineering
University of Texas at Arlington
Arlington, Texas, USA
cconly@uta.edu, zhong.zhang@mavs.uta.edu, athitsos@uta.edu

ABSTRACT

Users of written languages have the ability to quickly and easily look up the meaning of an unknown word. Those who use sign languages, however, lack this advantage, and it can be a challenge to find the meaning of an unknown sign. While some sign-to-written language dictionaries do exist, they are cumbersome and slow to use. We present an improved American Sign Language video dictionary system that allows a user to perform an unknown sign in front of a sensor and quickly retrieve a ranked list of similar signs with a video example of each. Earlier variants of the system required the use of a separate piece of software to record the query sign, as well as user intervention to provide bounding boxes for the hands and face in the first frame of the sign. The system presented here integrates all functionality into one piece of software and automates head and hand detection with the use of an RGB-D sensor, eliminating some of the shortcomings of the previous system, while improving match accuracy and shortening the time required to perform a query.

Categories and Subject Descriptors

I.2.10 [**Vision and Scene Understanding**]: 3D/Stereo Scene Analysis, Motion, Video Analysis;
I.4.8 [**Scene Analysis**]: Depth Cues, Motion, Time Varying Imagery, Tracking

General Terms

Experimentation, Measurement

Keywords

gesture recognition, Kinect, hand location, tracking

1. INTRODUCTION

Video-based sign search systems offer users of sign languages some of the benefits that those of written languages enjoy. If we encounter a word in a written language that is unfamiliar to us, we can consult a dictionary or the internet and obtain a definition in short time; the ability to alphabetize letters affords us this opportunity and makes search simple. Sign languages, however, are composed of a series of motions and handshapes. It is more difficult to assign some type of ordering to a set of motions and shapes than it is to an alphabet, making sign-to-written language search awkward. Though such dictionaries do exist, they are not necessarily intuitive or easy to use and can be time consuming to find a specific sign. The American Sign Language Handshape Dictionary is one such dictionary that uses 40 handshapes to categorize the signs [20]. Assuming you have correctly chosen the handshape, with 1,900 signs in the dictionary, it can be tedious to find the desired sign in that category.

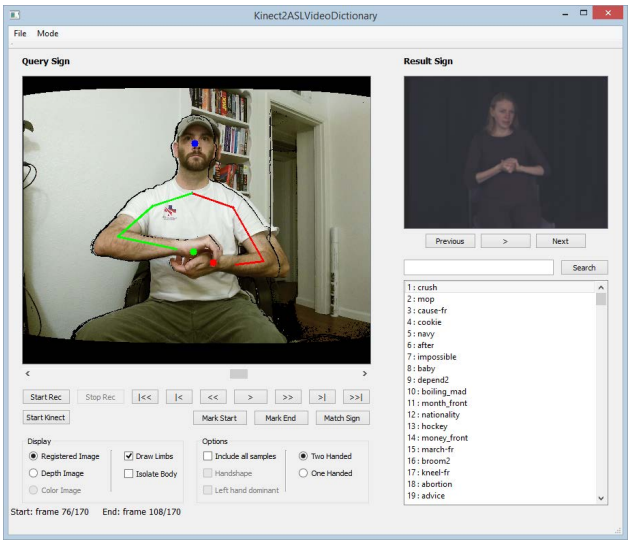


Figure 1: The ASL Video Dictionary System.

The American Sign Language (ASL) Video Dictionary System presented herein offers users the ability to quickly and easily search for the meaning of an unknown sign in a more integrated and automated manner. It eliminates inconsistencies in sign size normalization due to human factors and streamlines the dictionary search process.

There are several advantages over the previous system. The previous system required the use of two pieces of software. The user would first record a video of the sign using third-party web-cam software and would then import the recorded

PETRA '15, July 01 - 03 2015, Corfu, Greece
 ISBN 978-1-4503-3452-5/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2769493.2769534

video into the dictionary system for sign matching. The system described in this paper integrates the recording and matching into a single program, thus making sign search both easy and fast.

The earlier version of the dictionary system also required the user to initialize the hand tracker and trajectory generation algorithm by drawing bounding boxes around the hands and face in the first frame of the sign. The tracker could then track the hands throughout the sign. We eliminate this requirement and use a *Microsoft Kinect for Windows* v2 [12] to automate the hand detection process.

As the recorded query signs are scaled based on the size of the user's face in the previous system, differences in the sizes of the bounding box users draw for the face can affect system performance. We seek to eliminate inconsistencies by using a proportion of the distance between two easily located joints in the user's skeleton as detected by the RGB-D skeleton detection algorithm. We learned this proportion through experimentation on a validation set containing none of the users that participated in this study.

We evaluate the system by performing a series of sign match accuracy and timing tests on a random set of signs from the 1,113 sign vocabulary, employing a user-independent experimental protocol. In order to recreate a realistic usage scenario in our tests, participants with little to no knowledge of ASL are used, and none of them are familiar with the signs they perform or the dictionary system itself. We demonstrate an improvement in the percentage of signs whose correct match is found in the top 20 results, from 46.7% using the old system to 62% using the new system. In informal timing studies performed with the two systems, we see a significant decrease in time required to perform a query, from an average of 106 seconds to an average of 22 seconds.

2. RELATED WORK

Computer-based sign language recognition systems can be broadly divided into three groups based on the the types of input: (1) RGB video input, (2) RGB-D (i.e. Kinect) input and (3) glove input.

Sign language recognition using RGB videos has been extensively studied in the computer vision community [2, 5, 21, 24, 4]. Most of these methods are model-based, using Hidden Markov Models [2, 5, 21], or alternative approaches such as recognizing motion patterns from hand trajectories using Time Delay Neural Networks [24] and classifying hand shapes using a recursive partition tree approximator [4]. All of these methods use a small vocabulary of signs (less than 100 signs) and have unknown potential for scalability.

The authors of [27] and [8] report promising recognition accuracies on vocabularies of more than 100 signs. However, these results are achieved under a user dependent experimental protocol. In [27], the recognition rate under a user-independent setting is about 44%, significantly lower that the 99.3% user-dependent accuracy. It is unrealistic to expect the users of a sign language video dictionary to prerecord and annotate all the signs in the system's vocabulary.

The vocabulary size has been further increased to almost 1,000 signs in the work [1, 19, 22]. Athitsos et al. [1] reported results on 992 signs using motion energy images. Stefan et al. [19] and Wang et al. [22] reported results on 921 and 1,113 signs, respectively, using dynamic time warping as a measure of sign similarity.

More recent work using RGB videos for sign language recognition is found in [3, 17, 15]. Bragg [3] proposed two systems: ASL-Search and ASL-Flash. When a user encounters an unfamiliar sign, ASL-Search requests the user to select sets of features including hand shapes, orientations, locations, and movements from the interface based on his/her observation, and recognition is based on these selected features. ASL-Flash is a sign learning tool which shows users online flashcards of signs; it also collects users' query signs. Roussos et al. [17] used handshape to classify signs. They propose a novel handshape model called dynamic affine-invariant shape-appearance model (Aff-SAM). Pfister [15] proposes a method to localize a gesture and classify it into one of multiple classes at the same time.

Gloves are another commonly used input source for sign language recognition [25, 10, 23]. Yao et al. [25] and Liang et. al [10] collected more than 1,000 signs and 250 signs by gloves, respectively. Both systems used HMMs to model the signs. Yao et al .[25] proposed a pre-processing method, called One-Pass pre-search, to speed up the recognition process. Wang et al. [23] studied how to track the movement of fingers through video capture of a glove with differently colored fingers and areas.

Depth sensing technology, like the Kinect, has been explored for sign language recognition recently [26, 14, 6]. Elliott et al. [6] proposed a Kinect camera based sign look-up tool which includes an interactive sign recognition system and a real-time sign synthesis system. Zafrulla et al. [26] evaluated the potential of Kinect depth camera for sign language recognition by comparing with the CopyCat system, which uses a colored gloves and embedded accelerometers to track hands. Both methods use hidden Markov models (HMMs) to model signs. Pavlakos et al. [14] combined visual cues (color and depth images) and audio under a HMMs framework, and the proposed method was evaluated on a public gesture dataset: the ChaLearn multi-modal gesture challenge dataset [7].

3. SYSTEM DESCRIPTION

The ASL Video Dictionary System is a combination of hardware and software: Microsoft's Kinect 2 RGB-D sensor and a custom Graphical User Interface (GUI). The system is written in C++, using the Qt 5.3 application and UI framework [16] for the GUI, OpenCV 2.4.9 [13] for image processing, and Microsoft's Kinect SDK v2 to access the sensors [11]. At present, the dictionary is trained with a vocabulary size of 1,113 signs using three examples of each sign class, obtained from the American Sign Language Lexicon Video Dataset [1]. A larger vocabulary is planned.

3.1 Hardware

The dictionary system uses a Microsoft Kinect v2 RGB-D sensor and the associated Kinect SDK v2 to provide several streams of data. Specifically, we utilize the depth, color, body, and body index streams. Whereas the color (RGB)

stream provides standard 1920×1080 pixel video frames, the depth (D) stream provides scene depth information with a resolution of 512×424 pixels. It is this scene depth information that enables the skeleton detector to work (see [18] for details). The output of the skeleton detector is provided by the body stream, which consists of the 3D coordinates (with the sensor as the frame of reference origin) of all 25 joint positions that the Kinect SDK v2 provides. The coordinates can be mapped to their corresponding 2D projected pixel positions in the depth image. The body index stream provides information about which depth frame pixels belong to different people found in the video.

3.2 Graphical User Interface

The system GUI is composed of two main sections: a query recording section and a results section. The stream recording section allows a user to select a video stream to display, record and review a video, ensure skeleton detection works properly, perform temporal segmentation of the sign, and initiate the matching process. When the user records a video, all data streams are recorded, though only one is displayed.

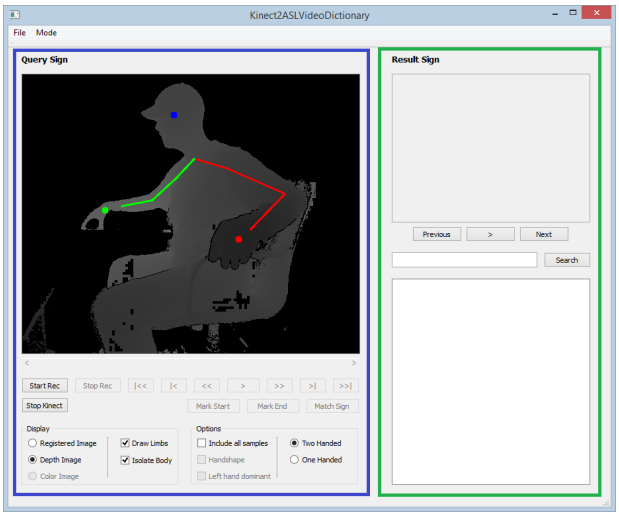


Figure 2: The system GUI: Highlighted in blue on the left is the query recording portion. On the right, highlighted in green, is the results section.

The results section presents a ranked list of sign matches, enables the user to view a video of each matched sign, and provides result search capabilities. Figure 2 shows the GUI before recording of a sign has begun. The depth stream has been selected, the user's body isolated, and the limbs are being drawn. Figure 1 shows the GUI after a search has been performed. In it, the registered image—in which each depth image pixel's corresponding color pixel is drawn—is displayed with the head and hands positions.

3.3 Feature and Trajectory Generation

The hand positions are expressed in a new coordinate system using the skeleton detector head position in the first frame of the sign as the origin. The sign size is normalized based on a proportion of the head-neck joint distance; this proportion was learned through experimentation. The new coordinate system and resizing ensures scale- and translation-invariance.

For each frame, we build a modified version of the feature vector presented in [22] to generate a trajectory representation of the sign. The feature vector includes the following information obtained by analyzing the 2D positional data of the hands in the depth image.

1. $L_d(X,t)$ and $L_{nd}(X,t)$: Dominant and non-dominant hands pixel positions in frame t of sign video X
2. $L_\delta(X,t) = L_d(X,t) - L_{nd}(X,t)$: Position of the dominant hand relative to the non-dominant hand.
3. $O_d(X,t)$ and $O_{nd}(X,t)$: Motion direction from frame $t-1$ to frame $t+1$ expressed as unit vectors for the dominant and non-dominant hands.
4. $O_\delta(X,t)$: Motion direction for L_δ from frame $t-1$ to frame $t+1$, expressed as a unit vector.

The dominant hand is the hand that will be moving in signs during which only one hand moves. All non-dominant hand information is set to zero for single-handed signs. The resulting trajectory representation is used by the sign matching algorithm.

As the skeleton detector does not provide bounding boxes for the hands, only a single point each, we do not include the hand appearance portion of the feature described in [22]. See section 5 for details about future work and the inclusion of handshape during sign matching.

3.4 Sign Matching

The dictionary system uses Dynamic Time Warping (DTW), a popular time series analysis method, as a similarity measure when comparing signs, thus accommodating variance in performance rate between the query and training signs [9]. There are three examples each of the 1,113 signs in the dictionary. For each of the three examples of each sign class that is the same type as the query (i.e. one-handed or two-handed), we use DTW to generated a lowest cost warping path, or alignment of query and example sign frame trajectory feature vectors using Euclidean distance as a metric. The total similarity score of a match given warping path W is calculated as the sum of the costs of aligning the frames feature vectors in W for query Q and example M:

$$C(W,Q,M) = \sum_{i=1}^{|W|} c(Q_{q_i}, M_{m_i}), \tag{1}$$

The DTW score D between query Q and example M is provided by the lowest cost of all warping paths:

$$D_{DTW}(Q,M) = \min_W C(W,Q,M). \tag{2}$$

Two sign rankings are generated. One contains all three examples of each sign of the same type, while the other contains just the best scoring of the three examples in each sign class of like type. Either set of rankings can be displayed.

3.5 Results Display

When sign comparison is complete and the results are ranked according to similarity, a list of the definitions of the results is generated and displayed. By clicking on any of the results the user can play a video of the sign to determine if it is correct match. Since there are three training examples of each sign and the two ranks lists are generated as described in section 3.4, the results display output mode can be selected to show either all matches or just the matches using the lowest of the three scores from the training examples. The results section of the GUI also provides search functionality, which can be useful in an experimental setting in which the definition of the sign is known.

3.6 System Usage

There are three phases to using the system. First a query sign is recorded. Second, temporal segmentation is performed to isolate the sign. Third, matching is performed and the results displayed.

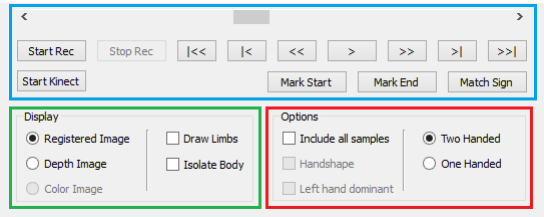


Figure 3: System Controls

When the ASL Video Dictionary System starts, the Kinect starts streaming and the live video is displayed along with the locations of the hands and head; the user has the option of adding limbs to the display. When ready, the user presses the record button and the system starts recording joint positions, joint states (i.e. tracked, inferred, or not tracked), the 16-bit depth video, the color video, and the registered video, in which each depth pixel is mapped to its corresponding color pixel. The left side of the blue outlined area in figure 3 contains recording and streaming controls that allow the user to start and stop the Kinect as well as the query recording. The green area of figure 3 contains the available data stream and display options. All streams, as well as timing and accuracy data, can be written to disk for later analysis and detection of bottlenecks.

The user then performs the sign and can use the skeletal overlay on the display to ensure the detector is working properly. When the sign has been performed to the user's liking, he can stop the recording, view what has been saved, and move on to temporal segmentation. The slider and buttons in the top and right sections of the blue area in figure 3 allow the user to set the start and end frames of the query and to play the full or cropped recording with the skeletal overlay.

During the temporal segmentation phase, the user marks what he or she feels are the start and end frames of the recorded sign. The user can again review the segmented sign and make any changes to the start and end frames that may be needed. After ensuring that the appropriate sign type (i.e. one or two handed sign) has been selected in the matching options section (red area of figure 3) and whether it is left hand dominant in the case of two-handed signs, the user can begin the match process.

(a) Color Frame

(b) Depth Frame

(c) Registered Frame

Figure 4: Example corresponding color, depth, and registered video frames.

When the *Match Sign* button is clicked, the joint position data is cropped and sent to the trajectory generation algorithm, which builds the feature vectors described in section 3.3. The output is then used for sign matching as described in section 3.4. After matching is complete and the results displayed, the user looks through the videos associated with the ranked list for the sign in question.

4. EXPERIMENTS

We ran an assortment of experiments to evaluate the performance of the system with respect to sign match accuracy and the amount of time it takes to use the system.

4.1 Description

To simulate a typical usage scenario, we chose experiment participants with little to no experience with sign languages. This enables us to asses the performance of the system with a typical user, instead of a regular user of ASL who knows how to perform the signs properly. Furthermore, they had never used the system before and had not yet developed the ability to very quickly perform a sign search.

For the study, five participants, designated **P01**–**P05**, were given a brief introduction to the system so they could observe how to use it and were presented with video examples of 30 signs chosen randomly from the 1,113 signs in the dictionary system's vocabulary. A separate set of 30 random signs was generated for each participant. After viewing video of the sign to be performed, the participant used

the system to search for the meaning of the sign without intervention from the experiment coordinators.

For each sign that the participants performed, the color, depth, and registered videos were written to disk, as well as position information for all 25 joints output by the skeleton detector, the ranked results lists, and timing information, including the time from the start of the query recording to results display, the time required for the entire matching algorithm (trajectory generation, results ranking and display), and the time required by DTW. Example corresponding color, depth, and registered video frames can be seen in figure 4.

If the participant made a mistake, for example forgot to mark a sign as one-handed, and needed to rerun the matching algorithm on one-handed signs, the entire time from their first attempt until they received appropriate results was logged. As real users of the system are expected to make mistakes, especially when learning the system, this provides more realistic usage timing data.

In section 4.2, we provide a comparison of sign recognition results from the old and new systems on the same videos. To generate accuracy results on the old system, we imported the color videos recorded on the new system and used the same start and end frames as determined by the participant. As the old system is not intuitive to use, an individual experienced with the system performed all experiments. Since the old system offers the ability to incorporate handshape into the matching algorithm, we ran the signs twice, both with and without handshape, and recorded each sign's best rank between the two.

We also recorded timing data with the old system. Since the video had already been recorded on the new system, however, we did not include the recording time for these signs. Instead, we record the time from the beginning of sign video importation to the display of results. Timing data from the two systems is compared in section 4.3.

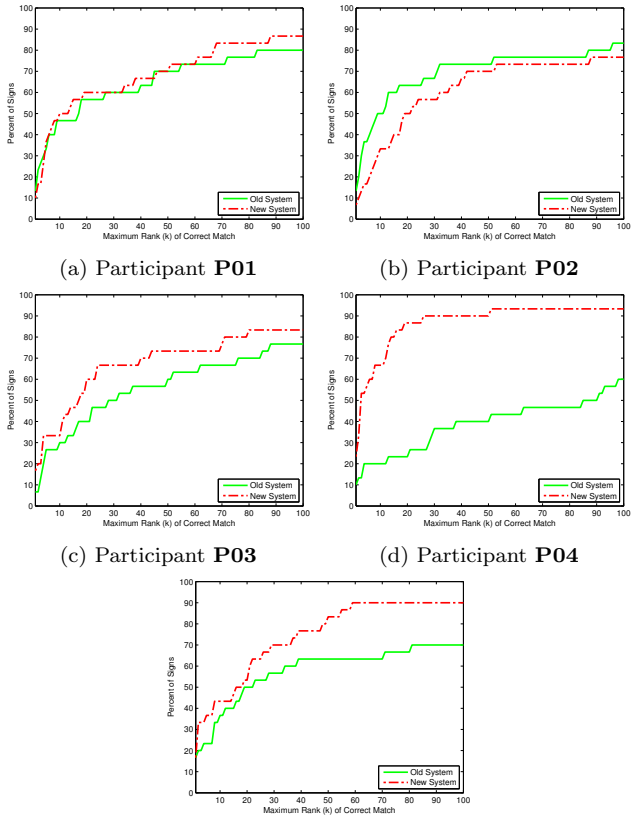
4.2 Accuracy Results

System accuracy is computed as the percentage of signs whose correct match is found in the top k results returned by the system. Figures 5a–5e show system accuracy for the individual participants. It can be seen that in all but one case, the new system outperforms the old system to varying degrees. For example, for participant **P04**, the old system returned the correct sign in the top 10 matches for 20% of the query signs versus 66.7% with the new system.

We calculated an average accuracy for both systems, as well as and expected accuracy for a random system. Since there are fewer one-handed signs in the system dictionary than two-handed signs, the maximum possible rank m is the number of two-handed signs, and the expected accuracy $f(r)$ at a rank level $r \in [1..m]$ is calculated:

$$f(r) = \begin{cases} \frac{2r}{N} & : r <= n \\ \frac{r+n}{N} & : r > n \end{cases} \quad (3)$$

for number of one-handed signs n in a dictionary of size N,



(a) Participant **P01** (b) Participant **P02**

(c) Participant **P03** (d) Participant **P04**

(e) Participant **P05**

Figure 5: System sign recognition accuracy. The green represents old system accuracy, while the red represents new system accuracy. The system whose curve is higher than the other is the better performing system.

alternatively expressed:

$$f(r) = \frac{r + \min(r, n)}{N}. \quad (4)$$

Figure 6a shows the accuracy for all rank levels and table 1 for a small subset. It can be seen that while both systems far outperform a system that randomly ranks the result signs, the new system shows a significant performance increase over the old. Figure 6b is a closer view of the accuracy in the maximum rank 1–100 levels. In the new system, for 62% of the query signs, the correct match is returned in the top 20 results, whereas this percentage drops to 46.7% in the old system. It is apparent that skeleton detection using scene depth information outperforms the skin color and motion-based hand tracker in the previous generation software. The same results can be seen in tabular format in table 1.

4.3 Query Time Results

The informal timing experiments show a significant performance increase in the new system. Table 2 shows the average and median times required by each user to perform a query, as well as the standard deviations. The *Average*

Table 1: Accuracy of both systems.

Max Rank	Old System	New System
1	12.0%	14.7%
2	16.7%	22.7%
3	20.7%	27.3%
4	26.0%	32.7%
5	28.0%	36.0%
10	36.7%	45.3%
15	40.7%	54.0%
20	46.7%	62.0%
30	54.0%	68.7%
50	61.3%	77.3%

row contains the averages of participants **P01**–**P05**, while the *System* row includes timing data from all participants in the calculations.

Table 2: Timing data in seconds for study participants.

Participant	Mean	Median	Std. Dev.
P01	13.1	11.0	6.18
P02	25.1	21.9	9.33
P03	15.1	14.8	3.19
P04	27.4	24.7	13.0
P05	29.2	27.5	9.93
Average	22.0	20.0	8.33
System	22.0	19.4	11.1

An experienced user performed the matching on the old system with the same videos. The timing data obtained here is informal, as it was obtained from a stopwatch. Furthermore, it did not include the time to record the videos, since they were not recorded with this system. The timed portion consisted of importing the video into the system, marking the start and end frames as the participant marked them in the new system, initialization of the hand tracker, tracking, the matching algorithm, and results display. Once the first set of results was displayed, the timer was stopped. Whereas the new system showed an average query search time of 22.0 seconds, the old system had a mean time of 106.2 seconds per query. See table 3 for details.

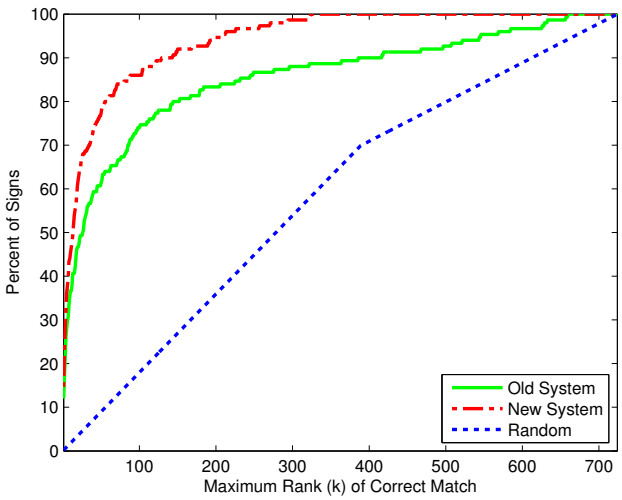
Table 3: Query Time Comparison

System	Mean	Median	Std. Dev.
Old	106.2	106.4	9.847
New	22.00	19.40	11.06

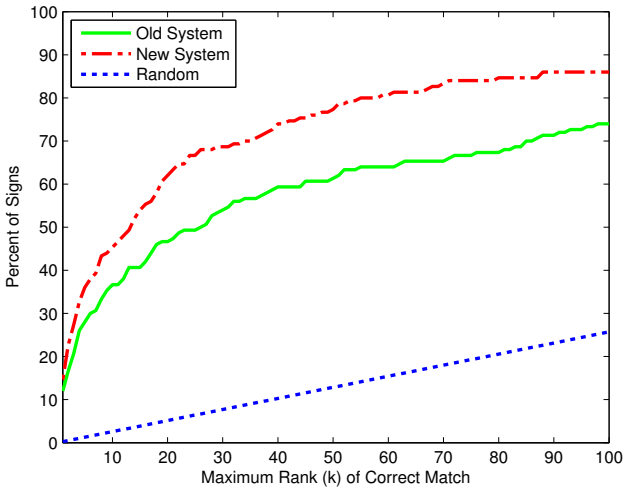
5. DISCUSSION AND CONCLUSIONS

We have demonstrated that the ASL Video Dictionary System presented in this paper outperforms the existing system in a two main respects: speed and accuracy.

The system has proven to be significantly faster to use. Even with an experienced user performing the experiments on the old system and inexperienced users on the new, the average time per query is roughly 5 times slower than the new system using the same query videos and frame information. Two



(a) All ranks.



(b) Closeup of maximum ranks 1–100.

Figure 6: Average System Accuracy Comparison

things account for this. The integration of recording functionality into the new system eliminates the time required to write video to disk with one program and then load it into the dictionary system. With the new system, nothing is written to disk until the results have been displayed. Until then, everything is recorded to memory.

Automatic real-time skeleton detection also provides a large speed performance increase. The previous system required the time-comsuming process or drawing bounding boxes on the hands and face in the first query frame and then using a tracking algorithm on those bounding boxes. The new system avoids this by using skeleton detection performed as each new frame arrives from the Kinect.

The new system has also outperformed the old in query retrieval accuracy. Since the general sign match method is the same across both systems (DTW), there are two potential sources of the increase. First, the skeleton detection algorithm used in the new system seems to generally work

better than the tracker of the old system. The skeleton detector has the ability to recover after losing track. The *tracking* performed in the new system is actually frame-by-frame skeleton detection. Whereas the old system's tracker could begin following the wrong object as a hand and not have to potential to recover, the per frame detection allows the new system to recover if it did detect the skeleton wrong in some frames. There are instances, however, when the old tracker outperforms the new. Signs during which the arms are crossed or the hands are close to the face seem to track better using the old system. In these cases, the Kinect skeleton detector destabilizes, and the joint positions can become unpredictable and incorrect.

Another potential source of the improved accuracy may be the automatic resizing of the signs. The old system resizes the sign based on the the face size in the first frame of the query. This size comes from the user and depends on how they draw the bounding box. Some users tightly crop the face, whereas other draw a loose box around the entire head. The new system introduces some consistency in this respect and can be tuned to improve performance. By using a proportion of the distance between easily detected joints, signs are resized in a much more reproducible manner. Further study is required to determine the exact effect of bounding box size on accuracy.

There are several areas we are examining for improvement and possible future work. In order to make the system more user friendly, a few aspects of additional automation need to be introduced. Ideally, the user would not need to mark start and end frames. Algorithms exist to spot a sign in a query video, but informal experimentation has not produced an acceptable level of accuracy.

Furthermore, the user should not be required to indicate whether the sign is one- or two-handed. This was often the reason for some of the longer times involved in new system usage. Participants would occasionally forget to change the setting and have to rerun the query. There are characteristics that two-handed signs tend to share that one-handed signs don't. We may be able to exploit this knowledge to automatically detect sign handedness. Related to this concept, is the idea of automatic two-handed sign sub-classification into one of four categories to reduce the search space and maximize accuracy.

Another area to examine is handshape. In using the old system, we found that handshape can significantly improve the rank of the correct sign, sometimes from a ranking of 57 to 4. Unfortunately, it can also do the opposite. Some study is warranted into better handshape representations; the one used by the old system is somewhat naive and simple.

If we are to incorporate handshape into the matching process, however, we must develop a method to cluster the hand pixels in the depth image or point cloud produced by the Kinect. As the detector gives us a single point for the location of the joints, we must expand that point to include hand pixels but exclude others.

Finally, we are reexamining the cost function used by the new system during sign trajectory alignment. A new cost function, or at the least, the addition of a good transition cost may improve accuracy. We are currently performing work in this area.

6. ACKNOWLEDGEMENTS

This work was partially supported by National Science Foundation grants IIS-0812601, IIS-1055062, CNS-1059235, CNS-1035913, and CNS-1338118.

7. REFERENCES

[1] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali. The american sign language lexicon video dataset. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008.

[2] B. Bauer, H. Hienz, and K.-F. Kraiss. Video-based continuous sign language recognition using statistical methods. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 2, pages 463–466. IEEE, 2000.

[3] D. Bragg, K. Rector, and R. E. Ladner. A user-powered american sign language dictionary. 2015.

[4] Y. Cui and J. Weng. Appearance-based hand sign recognition from intensity image sequences. *Computer Vision and Image Understanding*, 78(2):157–176, 2000.

[5] P. Dreuw, T. Deselaers, D. Keysers, and H. Ney. Modeling image variability in appearance-based gesture recognition. In *ECCV Workshop on Statistical Methods in Multi-Image and Video Processing*, pages 7–18, 2006.

[6] R. Elliott, H. Cooper, E.-J. Ong, J. Glauert, R. Bowden, and F. Lefebvre-Albaret. Search-by-example in multilingual sign language databases. In *Proc. Sign Language Translation and Avatar Technologies Workshops*, 2011.

[7] S. Escalera, J. Gonzàlez, X. Baró, M. Reyes, O. Lopes, I. Guyon, V. Athitsos, and H. Escalante. Multi-modal gesture recognition challenge 2013: Dataset and results. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 445–452. ACM, 2013.

[8] T. Kadir, R. Bowden, E.-J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *BMVC*, pages 1–10, 2004.

[9] J. B. Kruskal and M. Liberman. The symmetric time warping algorithm: From continuous to discrete. In *Time Warps*. Addison-Wesley, 1983.

[10] R.-H. Liang and M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 558–567. IEEE, 1998.

[11] Microsoft.com. Developing with kinect, 2015.

[12] Microsoft.com. Kinect for windows, 2015.

[13] Opencv.org. Opencv, 2015.

[14] G. Pavlakos, S. Theodorakis, V. Pitsikalis, S. Katsamanis, and P. Maragos. Kinect-based multimodal gesture recognition using a two-pass fusion scheme. In *Proc. IntâĂŹl Conf. on Image Processing*, 2014.

[15] T. Pfister, J. Charles, and A. Zisserman. Domain-adaptive discriminative one-shot learning of gestures. In *Computer Vision–ECCV 2014*, pages 814–829. Springer, 2014.

[16] Qt-project.org. Qt project, 2015.

[17] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Dynamic affine-invariant shape-appearance handshape features and classification in sign language videos. *The Journal of Machine Learning Research*, 14(1):1627–1663, 2013.

[18] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2821–2840, 2013.

[19] A. Stefan, H. Wang, and V. Athitsos. Towards automated large vocabulary gesture search. In *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, pages 16:1–16:8. ACM, 2009.

[20] R. Tennant. *American Sign Language handshape dictionary*. Gallaudet University Press, Washington, D.C, 2010.

[21] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 116–122. IEEE, 1999.

[22] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle, and F. Kamangar. A system for large vocabulary sign search. In *Proceedings of the 11th European conference on Trends and Topics in Computer Vision - Volume Part I*, ECCV'10, pages 342–353, Berlin, Heidelberg, 2012. Springer-Verlag.

[23] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics (TOG)*, 28(3):63:1–63:8, 2009.

[24] M.-H. Yang and N. Ahuja. Recognizing hand gestures using motion trajectories. In *Face Detection and Gesture Recognition for Human-Computer Interaction*, pages 53–81. Springer, 2001.

[25] G. Yao, H. Yao, X. Liu, and F. Jiang. Real time large vocabulary continuous sign language recognition based on op/viterbi algorithm. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 312–315. IEEE, 2006.

[26] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, and P. Presti. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, pages 279–286. ACM, 2011.

[27] J. Zieren and K.-F. Kraiss. Robust person-independent visual sign language recognition. In *Pattern recognition and image analysis*, pages 520–528. Springer, 2005.