Towards Automated Large Vocabulary Gesture Search

Alexandra Stefan, Haijing Wang, and Vassilis Athitsos
Computer Science and Engineering Department
University of Texas at Arlington, USA

ABSTRACT

This paper describes work towards designing a computer vision system for helping users look up the meaning of a sign. Sign lookup is treated as a video database retrieval problem. A video database is utilized that contains one or more video examples for each sign, for a large number of signs (close to 1000 in our current experiments). The emphasis of this paper is on evaluating the trade-offs between a non-automated approach, where the user manually specifies hand locations in the input video, and a fully automated approach, where hand locations are determined using a computer vision module, thus introducing inaccuracies into the sign retrieval process. We experimentally evaluate both approaches and present their respective advantages and disadvantages.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods; H.2.8 [**Database Applications**]: Data Mining; H.2.4 [**Systems**]: Multimedia Databases

Keywords

American Sign Language recognition, ASL recognition, gesture recognition, video indexing

1. INTRODUCTION

When we encounter an English word that we do not understand, we can look it up in a dictionary. However, when an American Sign Language (ASL) user encounters an unknown sign, looking up the meaning of that sign is not a straightforward process. In this paper we describe work towards designing a computer vision system for helping users look up the meaning of a sign.

Sign lookup is treated as a video database retrieval problem. A video database is utilized that contains one or more video examples for each sign, for a large number of signs (close to 1000 in our current experiments). When the user encounters an unknown sign, the user provides a video example of that sign as a query, so as to retrieve the most similar signs in the database. The query video can be either extracted from a pre-existing video sequence, or it can be recorded directly by the user, who can perform the sign of interest in front of a camera.

A crucial component of any similarity-based retrieval system is the choice of similarity measure for comparing the query with database objects. In this paper we evaluate two rather different approaches for similarity-based search in a database of signs. The first approach is not fully automatic: in that approach, it is assumed that the hand location for every frame of the query video is provided manually. In practice, this assumption can be enforced by requiring the user to review and correct, as needed, the results of an automatic hand detection module. This approach, while placing some burden on the user, allows us to use the popular Dynamic Time Warping (DTW) distance measure [9, 14, 15] for measuring similarity between signs.

The second approach we evaluate in this paper is a fully automated approach where the system simply uses the results of the hand detection module, without asking the user to correct those results. While this approach is less cumbersome for the user, it works rather poorly with DTW, because the correct hand location is not always identified unambiguously by the system. To handle ambiguities in hand detection we employ an extension of DTW called Dynamic Space-Time Warping (DSTW) [1]. The key difference between DSTW and DTW is that DTW takes as input a single (presumably correct) hand location per video frame, whereas DSTW takes as input a set of K candidate hand locations per frame, where K can range, for example, between 5 and 20.

In our experiments we have found that, while it is really hard to design a hand detection module that unambiguously identifies the hand location with high accuracy, it is rather easy, on the other hand, to design a hand detector that regularly includes the correct hand location in the top 10 candidates. In that sense, DSTW is a much more realistic choice for a fully automated system. At the same time, allowing multiple candidate hand locations makes it easier to obtain false matches for the query video, as the erroneous hand locations can match true hand locations in the database videos.

For both approaches we evaluate in this paper, we require that the hand locations for every frame of every database sign be known to the system. In our dataset, the hand locations in all database sequences are annotated manually. In our view, this requirement does not affect the user-friendliness of the system. Annotating the database is a task that is carried out by the designers of the system, and that is transparent to the user. Furthermore, database annotation is a one-time preprocessing cost.

We perform experiments using a video database containing 933 signs from 921 distinct sign classes, and a test set of 193 sign videos. The experiments are performed in a user-independent fashion: the signers performing the test signs are different from the

PETRA '09, June 09-13, 2009, Corfu, Greece.

signer performing the database signs. All signers are native ASL signers.

The results that we obtain illustrate the promise of the approach, but also the significant challenges that remain in order to produce a system ready to be deployed in the real world. As an example, using the second, fully-automatic approach, for 23% of the query signs the system ranks the correct class within the top 10 among all 921 database classes. At the same time, even when using the first, non-automated approach, for about 33% of the query signs the correct match is not included even in the top 100 matches, thus demonstrating the difficulty of obtaining accurate results with such a large gesture vocabulary.

2. RELATED WORK

A number of approaches have been proposed for sign language recognition (see [18] for a recent review). Many approaches are not vision-based, but instead use input from magnetic trackers and sensor gloves, e.g., [12, 16, 23, 29, 30, 32]. Such methods achieve good recognition results on continuous Chinese Sign Language with vocabularies of about 5,000 signs [12, 30, 32]. On the other hand, vision-based methods, e.g., [3, 7, 10, 11, 13, 25, 31] use smaller vocabularies (20-300 signs) and often rely on color markers, e.g., [3, 10]. One goal in our project is to make progress towards developing vision-based methods that operate on unadorned markerless images and can handle a more comprehensive vocabulary.

With respect to more general gesture recognition methods, in most dynamic gesture recognition systems (e.g., [8, 25]) information flows bottom-up: the video is input into the analysis module, which estimates the hand pose and shape model parameters, and these parameters are in turn fed into the recognition module, which classifies the gesture [19]. Skin detection, motion detection, edges, and background subtraction are commonly used for identifying hand location in each frame [5, 17].

A key drawback of bottom-up approaches is the assumption that we can have a preprocessing module that reliably detects hand locations in every frame. In many real-world settings, that assumption is simply unrealistic. For example, in Figure 2 skin detection yields multiple hand candidates, and the top candidate is often not correct. Other visual cues commonly used for hand detection, such as motion, edges, and background subtraction, would also fail to unambiguously locate the hand in the image. Motion-based detection and background subtraction may fail to uniquely identify the location of the hand when the face, non-gesturing hand or other scene objects (such as walking people in the background) are moving.

DTW, one of the two similarity measures used in this paper, was originally intended to recognize spoken words of small vocabulary [15, 21]. It was also applied successfully to recognize a small vocabulary of gestures [6, 9]. The DTW algorithm temporally aligns two sequences, a query sequence and a model sequence, and computes a matching score, which is used for classifying the query sequence. The time complexity of the basic DTW algorithm is quadratic in the sequence length, but more efficient variants have been proposed [24, 14].

In DTW, it is assumed that a feature vector can be reliably extracted from each query frame. Consequently, DTW-based gesture recognition falls under the category of bottom-up approaches, and, as discussed above, becomes problematic when the gesturing hand cannot be detected with absolute confidence. Dynamic Space-Time Warping (DSTW) [1], the second similarity measure that we are using in this paper, is an extension of Dynamic Time Warping (DTW), designed explicitly to work with multiple candidate hand locations per frame.

The CONDENSATION method [4] is, in principle, an approach that can be used for both tracking and recognition. In practice, however, the system described in [4] used CONDENSATION only for the recognition part, once the trajectory had been reliably estimated using a color marker. Even given the trajectory, system performance was reported to be significantly slower than real time, due to the large number of hypotheses that needed to be evaluated and propagated at each frame.

We should also note that, to use CONDENSATION, we need to know the observation density and propagation density for each state of each class model, whereas in our method no such knowledge is necessary. In our current training set, we have only one training example for the majority of sign classes. Clearly, no observation densities can be computed in this context, which renders CONDENSATION inapplicable in our setting. DSTW, being an exemplar-based method, can easily be applied in such single-example-per-class contexts.

3. THE ASL LEXICON DATASET

The long-term goal of our work on sign recognition is to design a system that makes it easy for users and learners of American Sign Language (ASL) to look up the meaning of an unknown sign. In such a sign lookup system, when the user encounters an unknown sign, the user submits to the system a video of that sign. The user can submit a pre-existing video, if such a video is available. Alternatively, the user can perform the sign in front of a camera, and generate a query video that way.

A key component of the sign lookup project is data collection. As described in [2], we are in the process of collecting a large video dataset containing examples of almost all of the 3,000 signs contained in the Gallaudet dictionary [26]. Each sign is performed by a native signer. The video sequences for this dataset are captured simultaneously from four different cameras, providing a side view, two frontal views, and a view zoomed in on the face of the signer. In both the side view and two frontal views the upper body occupies a relatively large part of the visible scene. In the face view, a frontal view of the face occupies a large part of the image. All sequences are in color.

For the side view, first frontal view, and face view, video is captured at 60 frames per second, non-interlaced, at a resolution of 640x480 pixels per frame. For the second frontal view, video is captured at 30 frames per second, noninterlaced, at a resolution of 1600x1200 pixels per frame. This high-resolution frontal view may facilitate the application of existing hand pose estimation and hand tracking systems on our dataset, by displaying the hand in significantly more detail than in the 640x480 views.

Due to the large number of signs, we can only collect a small number of exemplars for each sign. The lack of a large number of training examples per sign renders several model-based recognition methods inapplicable, e.g., Hidden Markov Models [20, 28]. At the same time, exemplar-based methods are readily applicable in cases with a small number of examples per class. In an exemplar-based method, processing a query involves identifying the most similar matches of the query in a database of training examples.

In our experiments, the database contains 933 examples of signs, corresponding to 921 unique sign classes. Experiments are performed in a user-independent manner, where the people performing signs in the query videos do not appear in the database videos. Out of the four camera views recorded, only the 60fps, 640x480 frontal view is used in our experiments. Figure 1 shows sample frames from four videos from this dataset.

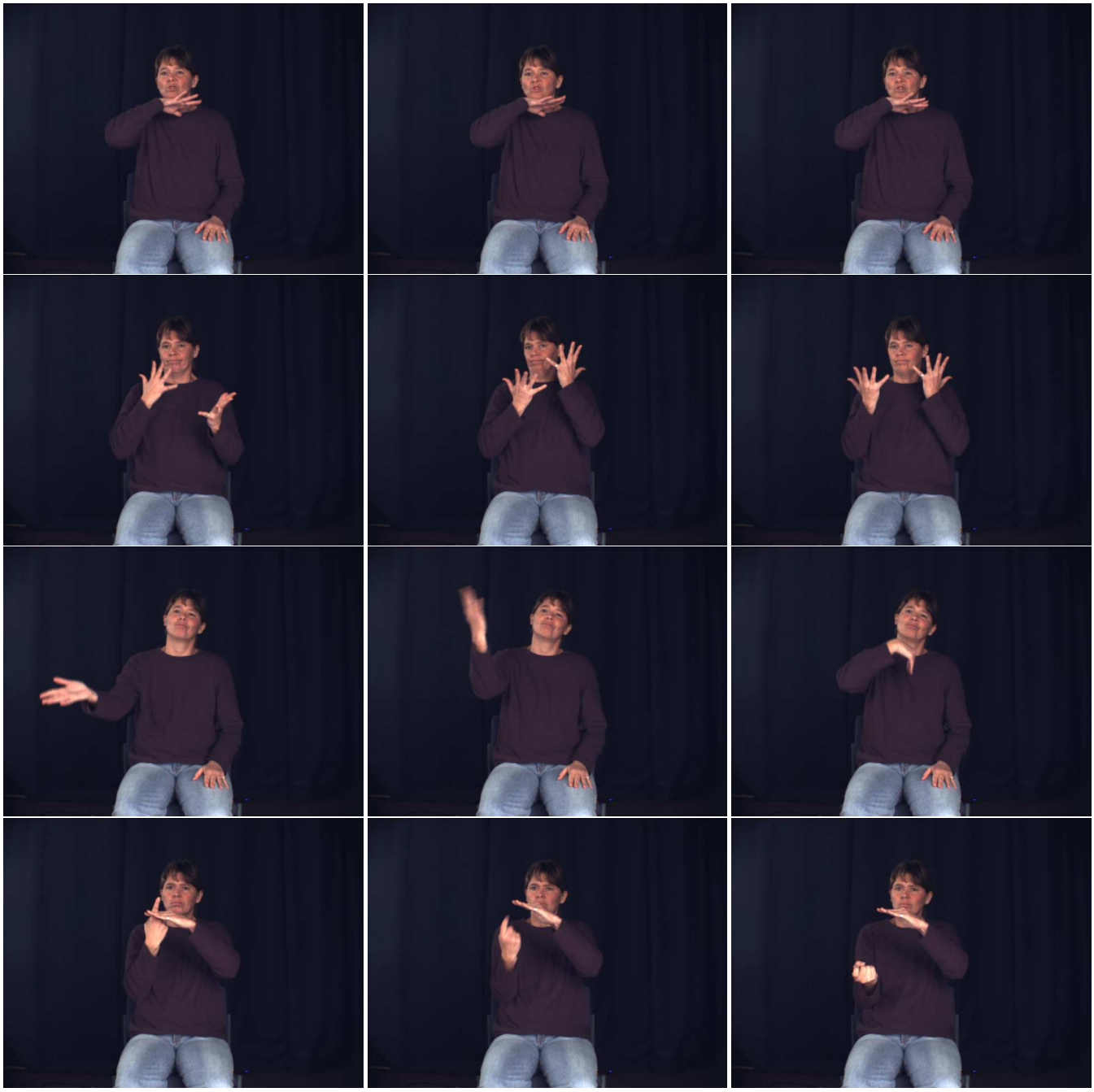

Figure 1: Examples of sign videos from the ASL lexicon video dataset [2]. For each sign, we show, from left to right, the first frame, a middle frame, and the last frame. First row: an example of the sign DIRTY. Second row: an example of the sign EMBARRASED. Third row: an example of the sign COME-ON. Fourth row: an example of the sign DISAPPEAR.

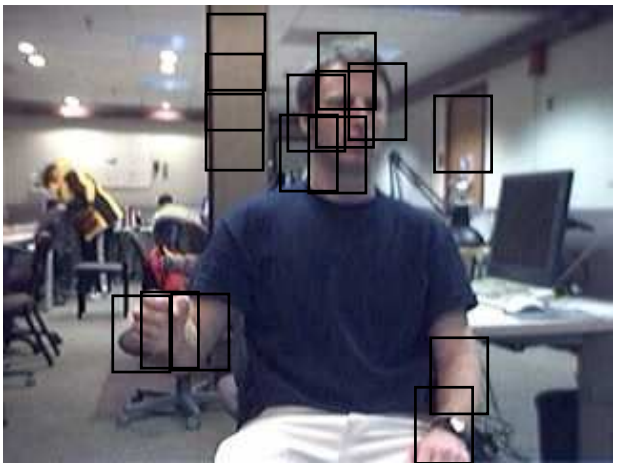

Figure 2: Detection of candidate hand regions based on skin color. Clearly, skin color is not sufficient to unambiguously detect the gesturing hand since the face, the non-gesturing hand, and other objects in the scene have similar color. On the other hand, for this particular scene, the gesturing hand is consistently among the top 15 candidates identified by skin detection.

4. DETECTION AND FEATURE EXTRACTION

The fully-automated version of the system has been designed to accommodate multiple hypotheses for the hand location in each frame. Therefore, we can afford to use a relatively simple and efficient hand detection scheme. In our implementation we combine two visual cues, i.e., color and motion; both requiring only a few operations per pixel. Skin color detection is computationally efficient, since it involves only a histogram lookup per pixel. Similarly, motion detection, which is based on frame differencing, involves a small number of operations per pixel.

The skin detector computes for every image pixel a skin likelihood term, given the skin color model that was built based on the results of face detection. The motion detector computes a mask by thresholding the result of frame differencing (frame differencing is the operation of computing, for every pixel, the absolute value of the difference in intensity between the current frame and the previous frame). If there is significant motion between the previous and current frame the motion mask is applied to the skin likelihood image to obtain the hand likelihood image. Using the integral image [27] of the hand likelihood image, we efficiently compute for every subwindow of some predetermined size the sum of pixel likelihoods in that subwindow. Then we extract the K subwindows with the highest sum, such that none of the K subwindows may include the center of another of the K subwindows. If there is no significant motion between the previous and current frame, then the previous K subwindows are copied over to the current frame.

A distinguishing feature of our hand detection algorithm compared to most existing methods [5] is that we do not use connected component analysis to find the largest component (discounting the face), and associate it with the gesturing hand. The connected component algorithm may group the hand with the arm (if the user is wearing a shirt with short sleeves), or with the face, or with any other skin-colored objects with which the hand may overlap. As a result the hand location, which is typically represented by the largest component's centroid, will be incorrectly estimated. In contrast, our hand detection algorithm maintains for every frame of the sequence multiple subwindows, some of which may occupy different parts of the same connected component. The gesturing hand is typically covered by one or more of these subwindows (See Figure 2).

As described above, for every frame j of the query sequence, the hand detector identifies K candidate hand regions. For every candidate k in frame j a 2D feature vector $Q_{jk} = (x_{jk}, y_{jk})$ is extracted. The 2D position (x, y) is the region centroid.

4.1 Tolerating Differences in Translation and Scale

Since the only information we use in measuring sign similarity is hand position, and hand position is *not* translation invariant or scale invariant, we need to take additional steps to ensure that the matching algorithm tolerates differences in translation and scale between two examples of the same sign.

We address differences in translation by normalizing all hand position coordinates based on the location of the face in each frame. Face detection is a relatively easy task in our setting, since we can assume that the signer's face is oriented upright and towards the camera. Mature, publicly-available real-time face detection systems have been available for several years [22, 27], that work well in detecting upright, frontal views of faces. In our experiments, the face location in database sequences is manually annotated, whereas for query sequences we use the publicly available face detector developed by Rowley, et al. at CMU [22].

Differences in scale can also cause problems, as a small difference in scale can lead to large differences in hand positions, and consequently to large DTW distances or DSTW distances. Our approach for tolerating differences in scale is to artificially enlarge the database, by creating for each database sign multiple copies, each copy corresponding to different scaling parameters. In particular, for each time series corresponding to a database sign video, we generate 361 scaled copies. Each scaled copy is produced by choosing two scaling parameters S_x and S_y, that determine respectively how to scale along the x axis and the y axis. Each S_x and S_y can take 19 different values, spaced uniformly between 0.92 and 1.1, thus leading to a total of $19^2 = 361$ possible value for each (S_x, S_y) pair.

5. COMPARING GESTURES VIA DYNAMIC PROGRAMMING

In order for the system to identify the most similar database matches to a query video, we need to define a distance measure between sign videos. In this section we describe Dynamic Time Warping (DTW), which requires the hand location in each frame to be known, and Dynamic Space Time Warping (DSTW), which can tolerate a significantly larger amount of ambiguity in the output of hand detection.

5.1 Dynamic Time Warping

Given the position of the dominant hand in each frame, each sign video is naturally represented as a 2D time series $((x_1, y_1), \ldots, (x_n, y_n))$, where n is the number of frames in the video, and each (x_i, y_i) represents the pixel coordinates of the centroid of the hand in the i-th frame. Consequently, comparing sign videos to each other becomes a time series matching problem.

For the purpose of measuring distance between these time-series representations of signs, we use the dynamic time warping (DTW) distance measure [9, 14, 15]. DTW is a popular method for matching time series, and satisfies a key requirement for a time series distance measure: the ability to tolerate temporal misalignments,

so as to allow for time warps, such as stretching or shrinking a portion of a sequence along the time axis, and differences in length between time series. We now proceed to briefly describe DTW.

Let Q be the time series representation of a query video with $|Q|$ frames, and let X be the time series representation of a database video with $|X|$ frames. A warping path $W = ((w_{1,1}, w_{1,2}), \ldots, (w_{|W|,1}, w_{|W|,2}))$ defines an alignment between two time series Q and X, and $|W|$ denotes the length of W. The i-th element of W is a pair $(w_{i,1}, w_{i,2})$ that specifies a correspondence between element $Q_{w_{i,1}}$ of Q and element $X_{w_{i,2}}$ of X. The cost $C(W)$ of warping path W that we use is the sum of the Euclidean distances between corresponding elements $Q_{w_{i,1}}$ and $X_{w_{i,2}}$:

$$C(W) = \sum_{i=1}^{|W|} \|Q_{w_{i,1}} - X_{w_{i,2}}\| \quad (1)$$

As a reminder, in our setting, $Q_{w_{i,1}}$ and $X_{w_{i,2}}$ denote respectively the center of the dominant hand in frame $w_{i,1}$ of the query video and frame $w_{i,2}$ of the database video.

For W to be a legal warping path, W must satisfy the following constraints:

- **Boundary conditions:** $w_{1,1} = w_{1,2} = 1, w_{|W|,1} = |Q|$ and $w_{|W|,2} = |X|$. This requires the warping path to start by matching the first element of the query with the first element of X, and end by matching the last element of the query with the last element of X.
- **Monotonicity:** $w_{i+1,1} - w_{i,1} \geq 0, w_{i+1,2} - w_{i,2} \geq 0$. This forces the warping path indices $w_{i,1}$ and $w_{i,2}$ to increase monotonically with i.
- **Continuity:** $w_{i+1,1} - w_{i,1} \leq 1, w_{i+1,2} - w_{i,2} \leq 1$. This restricts the warping path indices $w_{i,1}$ and $w_{i,2}$ to never increase by more than 1, so that the warping path does not skip any elements of Q, and also does not skip any elements of X between positions $X_{w_{1,2}}$ and $X_{w_{|W|,2}}$.

The optimal warping path between Q and X is the warping path with the smallest possible cost. The DTW distance between Q and X is the cost of the optimal warping path between Q and X. Given Q and X, the DTW distance between Q and X and the corresponding optimal warping path can be easily computed using dynamic programming [14].

Computing the DTW distance takes time $O(|Q||X|)$, i.e., time proportional to the product of the lengths of the two time series. If Q and X have comparable lengths, computing the DTW distance takes time quadratic to the length of the Q.

5.2 Dynamic Space-Time Warping

Here we describe dynamic space time warping [1], which is an extension of DTW that can handle multiple candidate detections in each frame of the query.

Let $M = (M_1, \ldots, M_m)$ be a model sequence in which each M_i is a feature vector. Let $Q = (Q_1, \ldots, Q_n)$ be a query sequence. In the regular DTW framework, each Q_j would be a feature vector, of the same form as each M_i. However, in dynamic space-time warping (DSTW), we want to model the fact that we have multiple candidate feature vectors in each frame of the query. For example, if the feature vector consists of the position of the hand in each frame, and we have multiple hypotheses for hand location, each of those hypotheses defines a different feature vector. Therefore, in DSTW, Q_j is a *set* of feature vectors: $Q_j = \{Q_{j1}, \ldots, Q_{jK}\}$, where each Q_{jk}, for $k \in \{1, \ldots, K\}$, is a candidate feature vector. K is the number of feature vectors extracted from each query frame. In our algorithm we assume K is fixed, but in principle K may vary from frame to frame.

percentile of rank of correct class	percentage of queries		
	DTW	DSTW	Auto-DTW
0.5	21.8	14.5	15.0
1.0	31.6	21.8	19.7
2.0	42.0	31.6	27.5
3.0	51.8	35.8	32.1
4.0	54.9	38.9	34.7
5.0	57.5	43.5	36.8
10.0	66.3	59.1	52.3
20.0	80.8	73.6	68.4
30.0	86.5	82.9	75.7
40.0	90.7	89.6	83.4

Table 1: P-percentile accuracy statistics for non-automated DTW, automated DSTW, and the automated version of DTW (marked "Auto-DTW"). The first column specifies values of P. For each such value of P, for each of the three methods, we show the percentage of test signs for which the correct sign class was ranked in the highest P-percentile among all 921 classes. For example, using DSTW, for 21.8% of the queries the correct class was ranked in the top 1.0% of all classes, i.e., in the top 9 out of all 921 classes.

As in DTW, a warping path W in DSTW defines an alignment between M and Q. However, in contrast to DTW, where each element of W is a pair, in DSTW each element of W is a triple: $W = ((w_{1,1}, w_{1,2}, w_{1,3}), \ldots, (w_{|W|,1}, w_{|W|,2}), w_{|W|,3}))$ which specifies a correspondence between element $Q_{w_{i,1}}$ of Q and element $X_{w_{i,2}}$ of X (as in DTW), but also specifies that, out of the multiple candidate hand locations in frame $Q_{w_{i,1}}$, the location indexed by $w_{i,3}$ is the one that optimizes the similarity score between query and model sequence.

In matching Q with M, and allowing K candidate hand locations per frame, the number of possible warping paths for DSTW is $O(K^{|Q|+|M|})$ larger than the number of possible warping paths for DTW. Despite the exponential number of possible warping paths for DSTW, it is shown [1] that the optimal warping path can still be found efficiently, in polynomial time, using dynamic programming.

6. EXPERIMENTS

The query and database videos for these experiments have been obtained from the ASL Lexicon Video Dataset [2]. Our test set consists of 193 sign videos, with all signs performed by two native ASL signers. The video database contains 933 sign videos, corresponding to 921 unique sign classes (we had two videos for a few of the sign classes). The database signs were performed also by a native ASL signer, who was different from the signers performing in the test videos. From the original database of 933 time series we created an extended database of 336,813 time series, by creating multiple scaled copies of each original time series, as described in Section 4.1. While somewhat more than half the signs in our dataset are two-handed, we only use the right (dominant) hand locations for the purposes of matching signs. All signers in the dataset we used are right-handed.

Performance is evaluated using P-percentile accuracy, which is defined as the fraction of test queries for which the correct class is among the top P-percentile of classes, as ranked by the retrieval system. Parameter P can vary depending on the experiment. In

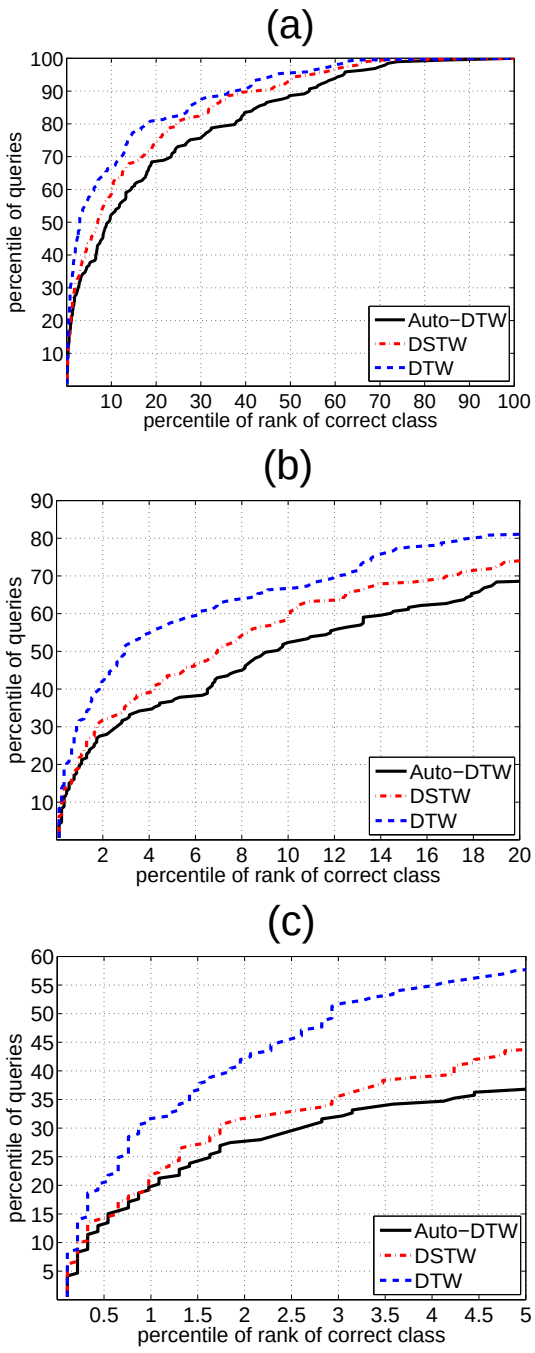

Figure 3: P-percentile accuracy plot for the ASL sign dataset, for non-automated DTW, automated DSTW, and the automated version of DTW (marked "Auto-DTW"). The x-axis corresponds to values of P. For each such value of P, we show the percentage of test signs for which the correct sign class was ranked in the highest P-percentile among all 921 classes. For example, using DSTW, for 22.8% of the queries the correct class was ranked in the top 1.1% of all classes, i.e., in the top 10 out of all 921 classes. Plots (b) and (c) focus on P-percentile ranges from 0 to 20 and 0 to 5, respectively, as results for low P-percentile values are of far more interest to system users.

order to compute P-percentile accuracy, we look at the similarity scores produced by comparing the query to every video in the database, and we choose for each class its highest-ranking exemplar. We then rank classes according to the score of the highest-ranking exemplar for each class. For example, suppose that the top three database matches come from class A, the fourth and fifth match come from class B, the sixth match comes from class C, and the seventh match comes from class A again. Then, A is the highest-ranking class, B is the second highest-ranking class, and C is the third highest-ranking class.

Figure 3 and Table 1 illustrate the results obtained on our dataset using our two approaches, which are based respectively on DTW and DSTW. For DSTW, parameter K was set to 11, i.e., 11 candidate hand locations were identified for each frame in the query videos. For comparison purposes, we have also included results with an automated DTW-based approach (denoted as "Auto-DTW"), where the hand location used in each frame is simply the most likely location identified by the hand detector, with no manual corrections.

The results demonstrate that, as expected, the fully automated DSTW-based approach is less accurate than the DTW-based approach. At the same time, we note that the results of Auto-DTW are mostly inferior to the results of DSTW, with the exception of the results for 0.5-percentile accuracy, where the percentage of query signs attaining that accuracy is marginally higher for Auto-DTW than for DSTW. These results demonstrate the advantages of DSTW with respect to DTW when the goal is to have a fully automated gesture recognition system, and the ability of DSTW to better tolerate ambiguities in the output of the hand detection module.

The results show that we have still quite some work to do, in order to obtain retrieval accuracy that would be sufficiently high for real-world deployment. We note that, even with the non-automated DTW-based approach, for about 33% of the query signs the correct class was not included even in the top 100 matches. At the same time, we need to take into account that these results were obtained using only hand location features. Incorporating hand appearance as well as additional body part detection modules (such as a forearm detector) can bring significant improvements to retrieval accuracy, and these topics are the focus of our current work.

At the same time, even with these rather minimal hand location features, using the fully automated DSTW-based approach, for about 23% of the queries we get the correct sign ranked in the top 10, out of 921 sign classes. We believe that visually inspecting 10 signs can be an acceptable load for users of the system, especially given the current lack of alternative efficient methods for looking up the meaning of a sign. We hope that including more informative features will help increase the percentage of queries for which the system attains a satisfactory level of retrieval accuracy.

7. DISCUSSION

This paper has described ongoing work towards a system for automatically looking up the meaning of ASL signs. Our focus has been on comparing a non-automated DTW-based approach, where hand locations are assumed to be known to the system, and a fully automated DSTW-based approach where hand locations are obtained, with a certain amount of ambiguity, using a computer vision-based hand detector. As expected, the fully automated approach resulted in lower accuracy. At the same time, the results obtained using DSTW indicate the advantages of DSTW over DTW when the goal is a fully automated system.

Several challenging research problems are posed by our dataset, that remain topics for future investigation. For example, our current approach is purely exemplar-based, and no learning is performed

by the system. An interesting question is whether it is possible to develop learning methods that can be applied in this context, with a large number of classes and only a single example (as of now) or a couple (as we proceed with data collection) of examples per sign class available for training. Another interesting topic is investigating novel similarity measures, that may overcome some limitations of both DTW and DSTW that stem from the underlying dynamic programming formulation. One significant limitation is that both DTW and DSTW cannot explicitly model dependencies between non-consecutive frames; the warping path cost only depends on matches between individual frames and transitions between consecutive frames.

The usability aspects of such a system also need to be better studied. While expecting the user to manually mark hand locations in every frame of the query video is not realistic, it is not clear at this point what amount of manual intervention would be acceptable to most users, and it is also not clear what kind of trade-offs can be achieved by allowing users to provide some limited help to the system. For example, an interesting experiment will be to allow users to mark hand locations for only the first frame of the video, and let a hand tracker track the hands in the remainder of the video. Marking hand locations for only a single frame does not seem to be an excessive burden, and it will be interesting to evaluate how much such minimal manual annotation will impact retrieval accuracy.

Acknowledgements

This work has been supported by NSF grants IIS-0705749 and IIS-0812601, as well as by a UTA startup grant to Professor Athitsos, and UTA STARS awards to Professors Chris Ding and Fillia Makedon. We also acknowledge and thank our collaborators at Boston University, including Carol Neidle, Stan Sclaroff, Joan Nash, Ashwin Thangali, and Quan Yuan, for their contributions in collecting and annotating the American Sign Language Lexicon Video Dataset.

8. REFERENCES

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. Simultaneous localization and recognition of dynamic hand gestures. In *IEEE Motion Workshop*, pages 254–260, 2005.

[2] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, Q. Yuan, and A. Thangali. The American Sign Language lexicon video dataset. In *IEEE Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB)*, 2008.

[3] B. Bauer and K. Kraiss. Towards an automatic sign language recognition system using subunits. In *Gesture Workshop*, pages 64–75, 2001.

[4] M. Black and A. Jepson. Recognizing temporal trajectories using the condensation algorithm. In *Face and Gesture Recognition*, pages 16–21, 1998.

[5] F. Chen, C. Fu, and C. Huang. Hand gesture recognition using a real-time tracking method and Hidden Markov Models. *Image and Video Computing*, 21(8):745–758, August 2003.

[6] A. Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems (RATFG-RTS)*, pages 82–89, 2001.

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

[8] R. Cutler and M. Turk. View-based interpretation of real-time optical flow for gesture recognition. In *Face and Gesture Recognition*, pages 416–421, 1998.

[9] T. Darrell, I. Essa, and A. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(12):1236–1242, 1996.

[10] J. Deng and H.-T. Tsui. A PCA/MDA scheme for hand posture recognition. In *Automatic Face and Gesture Recognition*, pages 294–299, 2002.

[11] K. Fujimura and X. Liu. Sign recognition using depth image streams. In *Automatic Face and Gesture Recognition*, pages 381–386, 2006.

[12] W. Gao, G. Fang, D. Zhao, and Y. Chen. Transition movement models for large vocabulary continuous sign language recognition. In *Automatic Face and Gesture Recognition*, pages 553–558, 2004.

[13] T. Kadir, R. Bowden, E. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *British Machine Vision Conference (BMVC)*, volume 2, pages 939–948, 2004.

[14] E. Keogh. Exact indexing of dynamic time warping. In *International Conference on Very Large Data Bases*, pages 406–417, 2002.

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

[16] J. Ma, W. Gao, J. Wu, and C. Wang. A continuous Chinese Sign Language recognition system. In *Automatic Face and Gesture Recognition*, pages 428–433, 2000.

[17] J. Martin, V. Devin, and J. Crowley. Active hand tracking. In *Face and Gesture Recognition*, pages 573–578, 1998.

[18] S. C. W. Ong and S. Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):873–891, 2005.

[19] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7), 1997.

[20] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77:2, 1989.

[21] L. Rabiner and B. Juang. *Fundamentals of speech recognition*. Prentice Hall, 1993.

[22] H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, pages 38–44, 1998.

[23] H. Sagawa and M. Takeuchi. A method for recognizing a sequence of sign language words represented in a Japanese Sign Language sentence. In *Automatic Face and Gesture Recognition*, pages 434–439, 2000.

[24] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 34(1), pages 43–49, 1978.

[25] T. Starner and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, 1998.

[26] C. Valli, editor. *The Gallaudet Dictionary of American Sign*

Language. Gallaudet U. Press, Washington, DC, 2006.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518, 2001.

[28] C. Vogler and D. N. Metaxas. Parallel hidden markov models for american sign language recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 116–122, 1999.

[29] C. Vogler and D. N. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. In *Gesture Workshop*, pages 247–258, 2003.

[30] C. Wang, S. Shan, and W. Gao. An approach based on phonemes to large vocabulary Chinese Sign Language recognition. In *Automatic Face and Gesture Recognition*, pages 411–416, 2002.

[31] M. Yang and N. Ahuja. Recognizing hand gesture using motion trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 466–472, 1999.

[32] G. Yao, H. Yao, X. Liu, and F. Jiang. Real time large vocabulary continuous sign language recognition based on OP/Viterbi algorithm. In *International Conference on Pattern Recognition*, volume 3, pages 312–315, 2006.