A Similarity Measure for Vision-Based Sign Recognition

Haijing Wang, Alexandra Stefan, and Vassilis Athitsos

Computer Science and Engineering Department, University of Texas at Arlington
Arlington, Texas, USA

Abstract. 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. It has been recently proposed that this problem can be addressed using a computer vision system that helps users look up the meaning of a sign. In that approach, sign lookup can be treated as a video database retrieval problem. 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. A necessary component of such a sign lookup system is a similarity measure for comparing sign videos. Given a query video of a specific sign, the similarity measure should assign high similarity values to videos from the same sign, and low similarity values to videos from other signs. This paper evaluates a state-of-the-art video-based similarity measure called Dynamic Space-Time Warping (DSTW) for the purposes of sign retrieval. The paper also discusses how to specifically adapt DSTW so as to tolerate differences in translation and scale.

Keywords: Gesture recognition, sign language recognition, American Sign Language, Dynamic Space-Time Warping, video databases, similarity-based retrieval.

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. A recent approach for facilitating sign lookup is to develop a computer vision system that, given a sign as a query, computes the similarity between the query sign and every sign in a large database, and outputs the most similar matches to the query [2].

In this paper, as in [2], 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 necessary component of such a sign lookup system is a similarity measure for comparing sign videos. Given a query video of a specific sign, the similarity measure should assign high similarity values to videos from the same sign, and low similarity values to videos from other signs. Also, the similarity measure should tolerate differences in translation and scale, and should also tolerate motion and clutter in the background. In this paper we evaluate a state-of-the-art video-based similarity measure called Dynamic Space-Time Warping (DSTW) [1] for the purposes of sign recognition. We also discuss how to specifically adapt DSTW so as to tolerate differences in translation and scale.

DSTW is an extension of the popular Dynamic Time Warping (DTW) similarity measure for time series [7], [11]. A key limitation of DTW when applied to gesture recognition is that DTW requires knowledge of the location of the gesturing hands in both the database videos and the query video. In contrast, DSTW only requires known hand locations for the database videos. Since the database videos are known a priori, hand locations in those videos can be specified manually, incurring a one-time preprocessing cost that does not affect the user experience. Given a query video, instead of requiring the hand locations to be specified (as DTW does), DSTW makes the much milder assumption that a hand detection module has produced a relatively short list of candidate hand locations. The ability to handle multiple candidate hand locations allows DSTW to be seamlessly integrated with existing imperfect hand detectors. As a result, DSTW-based systems can be readily deployed for gesture recognition in challenging real-world environments.

While DSTW is explicitly designed to tolerate imperfect hand detection and cluttered backgrounds, DSTW is not inherently invariant to translation and scale. In this paper we discuss specific strategies for tolerating significant differences in translation and scale. To tolerate translation differences, the location of the face is used to map each video's image coordinates to a canonical coordinate system. To tolerate differences in scale, multiple scaling factors are applied to the query video, so as to identify the scaling parameters that optimize the similarity score with each database video.

We perform experiments using a video database containing 933 sign videos 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 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, for 33% of the query signs, the system ranks the correct class within the top 1% of all database classes. For a dictionary containing 3000 signs (such as the Gallaudet Dictionary of American Sign Language), 1% of all signs would correspond to 30 signs. In such a dictionary, if the correct sign is ranked in the top 1%, the user would have the browse at most 30 signs in order to identify the sign of interest.

2. Related Work

In most dynamic gesture recognition systems 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. In a bottom-up framework, tracking and recognition typically fail in the absence of perfect hand segmentation. DSTW does not suffer from the bottom-up approach drawbacks, as it does not place unrealistic requirements upon the low-level task of hand detection: creating a relatively short list of candidate hand locations is sufficient, as long as the correct location is included in that list. Hand detection does not have to precisely identify the left and right hand at each frame.

DSTW is an extension of Dynamic Time Warping (DTW). DTW was originally intended to recognize spoken words of small vocabulary [11]. It was also applied successfully to recognize a small vocabulary of gestures [5], [7]. 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. In DTW, it is assumed that a feature vector can be reliably extracted from each query frame. However, this assumption is often hard to satisfy in vision-based systems, where the gesturing hand cannot be located with absolute confidence. In contrast, DSTW can take as input a list of several candidate hand regions per frame, a requirement that is easier to satisfy in complex real-world scenes.

With respect to recognition of signs and sign languages, a number of approaches have been proposed in the literature (see [13] for a recent review). Many approaches are not vision-based, but instead use input from magnetic trackers and sensor gloves, e.g., [9], [12], [16], [20], [21], [23]. Such methods achieve good recognition results on continuous Chinese Sign Language with vocabularies of about 5,000 signs [9], [21], [23]. On the other hand, vision-based methods, e.g., [3], [6], [8], [10], [17], [22] use smaller vocabularies (20-300 signs) and often rely on color markers, e.g., [3]. The approach described in this paper is a step towards developing vision-based methods that can handle a more comprehensive vocabulary.

3. The Dataset

The dataset used in this paper is the ASL Lexicon Video Dataset [2]. Here we briefly summarize the main features of that dataset. The goal in the ASL Lexicon Video Dataset is to eventually contain examples of almost all of the 3,000 signs contained in the Gallaudet dictionary [18]. 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. Out of the four camera views recorded, only a single 60fps, 640x480 frontal view is used in our experiments.

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 [14], [19]. At the same time, exemplar-based methods are readily applicable in cases

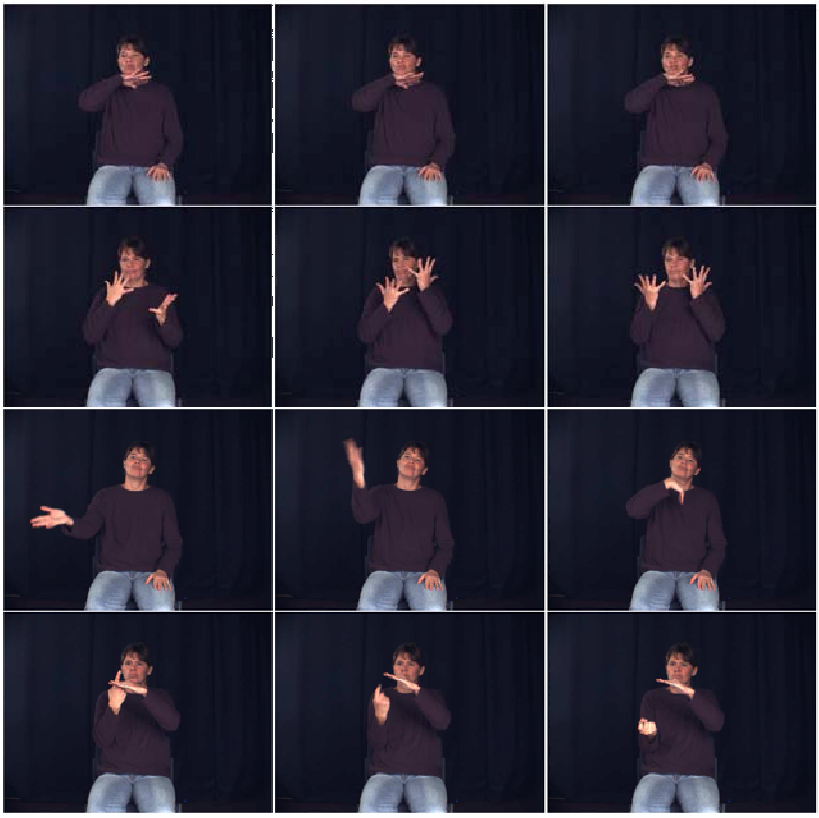

Fig. 1. Examples of sign videos from the ASL lexicon video dataset [2]. For each sign, we show, from left to right, the first frame, a middle frame, and the last frame. First row: an example of the sign DIRTY. Second row: an example of the sign EMBARRASED. Third row: an example of the sign COME-ON. Fourth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

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.

The dataset includes manually annotated hand locations for all frames of the dataset. In our experiments, we use these manual annotations so that the system knows the hand location in every frame of every database video. However, these manual annota-

tions (although available) are not used for the queries. Hand detection and feature extraction is performed on the queries as described in Section 4.

Figure [1] shows sample frames from four videos from this dataset.

4. Feature Extraction

DSTW 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. We 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 [4] 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.

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.

5. Dynamic Space Time Warping

In this section we briefly summarize the Dynamic Space Time Warping (DSTW) method described in [1].

For the database video sequence, since we are 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. We use notation $M = (M_1, \ldots, M_m)$ for the model sequence, where each M_i is a feature vector (x_i, y_i).

Let $Q = (Q_1, \ldots, Q_n)$ be a query sequence. In DSTW, Q_j is a *set* of feature vectors: $Q_j = \{Q_{j1}, \ldots, Q_{jK}\}$, where each Q_{jk}, for $k = \{1, \ldots, K\}$, 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.

A warping path W in DSTW defines an alignment between M and Q. Each element of W is a triple: $W=((w_{1,1}, w_{1,2}, w_{1,3}), \ldots, (w_{|W|,1}, w_{|W|,2}, w_{|W|,3}))$. Triple $(w_{i,1}, w_{i,2}, w_{i,3})$ specifies a correspondence between frame $w_{i,1}$ of Q and frame $w_{i,2}$ of X, but also specifies that, out of the multiple candidate hand locations in the $w_{i,1}$-th frame of Q, the candidate hand location indexed by $w_{i,3}$ is the one that optimizes the similarity score between query and model sequence.

The cost C(W) of warping path W that we use is the sum of the Euclidean distances between the |W| pairs of corresponding feature vectors defined by W. In matching Q with M, the number of possible warping paths for DSTW is exponential to |Q| and to |M|. Despite the exponential number of possible warping paths for DSTW, it is shown in [1] that the optimal warping path can still be found efficiently, in polynomial time, using dynamic programming. The DSTW distance between Q and M is defined as the cost of the optimal (min-cost) warping path between Q and M.

6. Multiscale Search

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. 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 [15].

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 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 441 scaled copies. Each scaled copy is produced by choosing two scaling parameters Sx and Sy, that determine respectively how to scale along the x axis and the

y axis. Each Sx and Sy can take 21 different values, spaced uniformly between 0.9 and 1.1, thus leading to a total of $21^2 = 441$ possible values for each (Sx, Sy) pair.

7. Experiments

The query and database videos for these experiments have been obtained from the ASL Lexicon Video Dataset [2]. The 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 411,453 time series, by creating multiple scaled copies of each original time series, as described in Section 6.

In the database, 612 of the 921 signs are two-handed, meaning that they are performed with both hands, and 309 signs are one-handed. Among the queries, 109 are two-handed and 84 are one-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. We assume that the system knows if a query sign is one-handed or two-handed, since such information can easily be input by the user. Database signs that do not match the number of hands used in the query sign are not considered during search.

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 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.

Table 1 shows the results obtained on our dataset using DSTW. For DSTW, parameter K was set to 7, i.e., 7 candidate hand locations were identified for each frame in the query videos. For comparison purposes, we have also included results obtained using the classical DTW algorithm. We should note that, to use DTW, we simply use as hand location the top-ranked candidate hand location at each query frame. That location is oftentimes wrong, and that causes DTW to yield less accurate results than DSTW. At the same time, the background in our query videos is relatively simple, with no clutter or skin-colored objects, and that allows our simple hand detection module to work relatively well: in the majority of the video frames, the highest ranked candidate hand location is the correct hand location. We expect the difference in accuracy between DSTW and DTW to be more pronounced in videos with more complicated background, especially in scenes including clutter and other moving humans in addition to the person performing the sign.

Table 1. P-percentile accuracy statistics for DSTW and DTW. The first column specifies values of P. For each such value of P, for each method, 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 32.6% of the queries the correct class was ranked in the top 1% of all classes, i.e., in the top 9 out of all 921 classes..

Percentile of rank of correct class	Percentage of queries	
	DSTW	DTW
0.5	21.2	18.1
1.0	32.6	26.4
2.0	37.3	31.1
3.0	45.6	34.7
4.0	51.3	40.9
5.0	54.4	47.15
10.0	65.3	60.1
20.0	76.2	73.6
30.0	88.1	87.6
40.0	93.3	92.2
50.0	99.0	96.7
60.0	100.0	100.0

8. Discussion

Our experimental results, while demonstrating that DSTW is more accurate than DTW, also show that it remains a challenge to obtain retrieval accuracy that would be sufficiently high for real-world deployment. We note that for about 30% of the query signs the correct class was not included even in the top 100 matches.

At the same time, using the DSTW approach, for about 33% of the queries we get the correct sign ranked in the top 9, out of 921 sign classes. We believe that visually inspecting 9 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. Furthermore, 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.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.

Another interesting topic for future exploration is improving retrieval time, by eliminating the need for brute-force search of all database videos in order to find the best match for the query. Brute-force search was used in the current experiments,

leading to a retrieval time of about 4 minutes per query. We are currently investigating methods for drastically reducing retrieval time, so as to make it acceptable for interactive applications.

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.

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. 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 (2003)
5. 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)
6. Y. Cui and J. Weng. Appearance-based hand sign recognition from intensity image sequences. *Computer Vision and Image Understanding*, 78(2):157—176 (2000)
7. 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)
8. K. Fujimura and X. Liu. Sign recognition using depth image streams. In *Automatic Face and Gesture Recognition*, pages 381–386 (2006)
9. 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)
10. T. Kadir, R. Bowden, E. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *British Machine Vision Conference (BMVC)*, volume I
11. J. B. Kruskall and M. Liberman. The symmetric time warping algorithm: From continuous to discrete. In *Time Warps*. Addison-Wesley (1983)
12. 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)
13. 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)

14.L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77:2 (1989)
15. H. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 38--44 (1998)
16.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)
17. 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 (PAMI)*, 20(12):1371—1375 (1998).
18. C. Valli, editor. *The Gallaudet Dictionary of American Sign Language*. Gallaudet U. Press, Washington, DC (2006)
19.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)
20. C. Vogler and D. N. Metaxas. Handshapes and movements: Multiple-channel american sign language recognition. In *Gesture Workshop*, pages 247--258 (2003)
21.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)
22. M. Yang and N. Ahuja. Recognizing hand gesture using motion trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 466--472 (1999)
23.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 (ICPR)*, volume 3, pages 312–315, 2006.