To appear in *Proc. IEEE Workshop on Motion and Video Computing*, Jan. 2005.

Simultaneous Localization and Recognition of Dynamic Hand Gestures

Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff [*]
Computer Science Department
Boston University
Boston, MA 02215

Abstract

A method for the simultaneous localization and recognition of dynamic hand gestures is proposed. At the core of this method is a dynamic space-time warping (DSTW) algorithm, that aligns a pair of query and model gestures in both space and time. For every frame of the query sequence, feature detectors generate multiple hand region candidates. Dynamic programming is then used to compute both a global matching cost, which is used to recognize the query gesture, and a warping path, which aligns the query and model sequences in time, and also finds the best hand candidate region in every query frame. The proposed framework includes translation invariant recognition of gestures, a desirable property for many HCI systems. The performance of the approach is evaluated on a dataset of hand signed digits gestured by people wearing short sleeve shirts, in front of a background containing other non-hand skin-colored objects. The algorithm simultaneously localizes the gesturing hand and recognizes the hand-signed digit. Although DSTW is illustrated in a gesture recognition setting, the proposed algorithm is a general method for matching time series, that allows for multiple candidate feature vectors to be extracted at each time step.

1. Introduction

Hand gestures are an important modality for human computer interaction (HCI) [15]. Compared to many existing interfaces, hand gestures have the advantages of being easy to use, natural, and intuitive. Successful applications of hand gesture recognition include computer games control [7], human-robot interaction [22], and sign language recognition [21], to name a few. Vision-based recognition systems can give computers the capability of understanding and responding to hand gestures. The usability of such systems greatly depends on their ability to function reliably in common real-world environments, without requiring the user to wear special clothes or cumbersome devices such as colored markers or gloves [22].

Most hand gesture recognition systems assume that the gesturing hand can be reliably located in every frame of the input sequence. In many real life settings this assumption cannot be satisfied. For example, in Figure 1 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 [2, 13] 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 are moving. At the same time, such methods can be used to produce a relatively short list of candidate hand locations.

The proposed approach is a principled method for gesture recognition in domains where existing algorithms cannot reliably localize the gesturing hand. Instead of assuming perfect hand detection, we make the milder assumption that a list of candidate hand locations is available for each frame of the input sequence. At the core of our framework is a dynamic space-time warping (DSTW) algorithm, that aligns a pair of query and model gestures in time, while at the same time it identifies the best hand location out of the multiple hypotheses available at each query frame. The main advantages of our method are the following:

- Hand detection is *not* merely a bottom-up procedure. The gesture model is used to select hand locations in a way that the query-to-model matching cost is optimized.
- Recognition can be achieved even in the presence of multiple "distractors," like moving objects, or skin-colored objects (e.g., face, non-gesturing hand, background objects).
- Recognition is robust to overlaps between the gesturing hand and the face or the other hand.
- Recognition is translation-invariant; the gesture can occur in any part of the image.
- Unlike HMMs and CONDENSATION-based gesture recognition our method requires no knowledge of ob-

[*] This research was funded in part by NSF grants CNS-0202067, IIS-0308213, and IIS-0329009, and ONR grant N00014-03-1-0108.

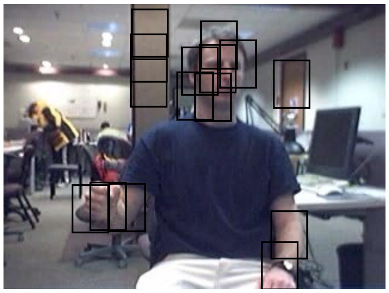

Figure 1: Detection of candidate hand regions based on skin color. Clearly, skin color is not sufficient to unambiguously detect the gesturing hand since the face, the non-gesturing hand, and other objects in the scene have similar color. On the other hand, for this particular scene, the gesturing hand is consistently among the top 15 candidates identified by skin detection.

servation and transition densities, and therefore can be applied even if we have a single example per class.

- Although this paper describes DSTW in the context of gesture localization and recognition, DSTW is a general method for matching time series, that can accommodate multiple candidate feature vectors at each time step.

Inspired by previous vision-based HCI systems (e.g., the virtual whiteboard by Black and Jepson [1], and the virtual drawing package by Isard [9], to name a few), we evaluate our framework on a vision-based character recognition task.

2. Related Work

In most dynamic gesture recognition systems (e.g., [4, 21]) 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 [15]. In a bottom-up framework, tracking and recognition typically fail in the absence of perfect hand segmentation.

The method proposed in this paper is an extension of Dynamic Time Warping (DTW). DTW was originally intended to recognize spoken words of small vocabulary [12, 16]. It was also applied successfully to recognize a small vocabulary of gestures [3, 5]. 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 [19, 11]). 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. A framework that allows for multiple detections of candidate hand regions, or more generally multiple observations, is therefore required.

In multiple hypothesis tracking (e.g., [17]) multiple hypotheses are associated with multiple observations. Each observation corresponds to a different object with a different model. In contrast, in the proposed framework a single consistent hypothesis is selected among multiple distinct observations (detections), only one of which is correct. The CONDENSATION-based framework can also be applied to gesture recognition [1]. Although in principle CONDENSATION can be used for both tracking and recognition, in [1] CONDENSATION was only used 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. Also, 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.

The work by Sato and Kobayashi [20] is the most related to our work. In the Hidden Markov Model (HMM) framework, Sato and Kobayashi extended the Viterbi algorithm so that multiple candidate observations can be accommodated at each query frame; the optimal state sequence is constrained to pass through the most likely candidate at every time step. HMMs have found wider application for problems with large vocabulary (of words or gestures) primarily due to their ability to probabilistically encode the variability of the training data. However, DTW can still be appropriate for smaller problems because it is simpler to implement: there is no need to worry about the HMM structure, and no training is required. Furthermore, our approach differs from [20] in that it incorporates translation invariance, and is evaluated in a more challenging setting (users are wearing short sleeve shirts and the hand is not an isolated skin-colored blob).

3. Detection and Feature Extraction

The overall algorithm consists of three major components: detection of multiple candidate hand regions, feature extraction, and hand gesture recognition.

3.1. Detection

The proposed method 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.

The skin detector first computes for every image pixel a skin likelihood term. For the first frames of the sequence, where a face has still not been detected, we use a generic skin color histogram [10] to compute the skin likelihood image. Once a face has been detected [18], we use the mean and covariance of the face skin pixels in normalized *rg* space to compute the skin likelihood image.

The motion detector computes a mask by thresholding the result of frame differencing. 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 [23] 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 [2] 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 1).

3.2. Feature Extraction

For every frame j of the query sequence, K candidate hand regions are found as described in the previous section. For every candidate k in frame j a 4D feature vector $Q_{jk} = (x_{jk}, y_{jk}, u_{jk}, v_{jk})$ is extracted. The 2D position (x, y) is the region centroid, and the 2D velocity (u, v) is the optical flow averaged over that region. Optical flow is computed using a block-based matching method [24].

In our current implementation, when we collect the model sequences, a colored glove is used to reliably detect the gesturing hand. Using such additional constraints, like colored markers, is often desirable for the offline model-building phase, because it simplifies the construction of accurate class models. It is important to stress that such markers are not used in the query sequences, and therefore they do not affect the naturalness and comfort of the user interface, as perceived by the end user of the system.

Based on the features extracted from all database sequences, we compute parameters that translate and scale those features, so that they lie inside the unit hypercube. We transform the feature vectors of each query frame using the same parameters.

4 Dynamic Space-Time Warping

One of several publications that describe the DTW algorithm is [11]. In this section we will describe dynamic space time warping, which is an extension of DTW that can handle multiple candidate detections in each frame of the query.

Let $M = (M_1, \dots, M_m)$ be a model sequence in which each M_i is a feature vector. Let $Q = (Q_1, \dots, Q_n)$ 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 and velocity 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 our algorithm, Q_j is a *set* of feature vectors: $Q_j = \{Q_{j1}, \dots, Q_{jK}\}$, where each Q_{jk}, for $k \in \{1, \dots, K\}$, 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 defines an alignment between M and Q. Formally, $W = w_1, \dots, w_T$, where $\max(m, n) \leq T \leq m + n - 1$. Each $w_t = (i, j, k)$ is a triple, which specifies that feature vector M_i of the model is matched with feature vector Q_{jk}. We say that w_t has two *temporal* dimensions (denoted by i and j) and one *spatial* dimension (denoted by k). The warping path is typically subject to several constraints (adapted from [11] to fit the DSTW framework):

- **Boundary conditions:** $w_1 = (1, 1, k)$ and $w_T = (m, n, k')$. This requires the warping path to start by matching the first frame of the model with the first frame of the query, and end by matching the last frame of the model with the last frame of the query. No restrictions are placed on k and k', which can take any value from 1 to K.

- **Temporal continuity:** Given $w_t = (a, b, k)$ then $w_{t-1} = (a', b', k')$, where $a - a' \leq 1$ and $b - b' \leq 1$. This restricts the allowable steps in the warping path to adjacent cells along the two temporal dimensions.

- **Temporal monotonicity:** Given $w_t = (a, b, k)$ then $w_{t-1} = (a', b', k')$ where $a - a' \geq 0$ and $b - b' \geq 0$. This forces the warping path sequence to increase monotonically in the two temporal dimensions.

```
input  : A sequence of model feature vectors $M_i, 1 \leq i \leq m$, and a sequence of sets of query feature
         vectors $Q_j = \{Q_{j1}, \ldots, Q_{jK}\}, 1 \leq j \leq n$.
output : A global matching cost $D^*$, and an optimal warping path $W^* = (w_1^*, \ldots, w_T^*)$.
// Initialization
$j = 0$
for $i = 0 : m$ do
    for $k = 1 : K$ do
        $D(i, j, k) = \infty$
    end
end
$D(0, 0, 1) = 0$
// Iteration
for $j = 1 : n$ do
    for $i = 0 : m$ do
        for $k = 1 : K$ do
            if $i = 0$ then
                $D(i, j, k) = \infty$
            end
            else
                $w = (i, j, k)$
                for $w' \in N(w)$ do
                    $C(w', w) = \tau(w', w) + D(w'),$
                end
                $D(w) = d(w) + \min_{w' \in N(w)} C(w', w)$
                $b(w) = \text{argmin}_{w' \in N(w)} C(w', w)$
            end
        end
    end
end
// Termination
$k^* = \text{argmin}_k \{D(m, n, k)\}$
$D^* = D(m, n, k^*)$
$w_T^* = (m, n, k^*)$
// Backtrack
$w_{t-1}^* = b(w_t^*)$
```

Algorithm 1: The DSTW algorithm

Note that continuity and monotonicity are required only in the temporal dimensions. No such restrictions are needed for the spatial dimension; the warping path can "jump" from any spatial candidate k to any other spatial candidate k'.

Given warping path element $w_t = (i, j, k)$, we define the set $N(i, j, k)$ to be the set of all possible values of w_{t-1} that satisfy the warping path constraints (in particular continuity and monotonicity):

$$N(i, j, k) = \{(i-1, j), (i, j-1), (i-1, j-1)\} \times \{1, \ldots, K\} \quad (1)$$

We assume that we have a cost measure $d(i, j, k) \equiv d(M_i, Q_{jk})$ between two feature vectors M_i and Q_{jk}. We also assume that we have a transition cost $\tau(w_{t-1}, w_t)$ between two successive warping path elements. DSTW finds the optimal path W^* and the global matching score D^* as described in Algorithm 1.

For this algorithm to function correctly it is required that $\tau(w', w) = 0$ when $w' = (0, j, k)$ or $w' = (i, 0, k)$. For all other values of w', τ must be defined appropriately in a domain-specific way. The function τ plays a similar role in DSTW as state transition probabilities play in the HMM framework.

4.1. Translation Invariance

In recognizing hand gestures, a commonly used feature is position of the hand. Using positions as features is appealing because they are simple to extract, and are highly informative about the gesture content. However, they are not invariant to translation. In simple DTW, there is a single candidate per frame, and therefore the trajectory of the hand is known. In that case we can achieve translation invariance (i.e., invariance with respect to global translation of the entire gesture) by subtracting from the entire trajectory the position of the hand in the first frame.

In DSTW, we can apply this strategy to the model, where the trajectory is known (recall that a colored glove is used in the model sequences). However, for the test sequence, there are multiple candidates at each frame and therefore the position of the hand in the first frame is not known. When position is used as a feature, we can achieve translation invariance as follows: given the K candidate regions detected at the first frame, we start K separate DSTW processes, running in parallel. Each such process P_k corresponds to a candidate k among the K regions detected in the first frame. The process P_k makes the assumption that k was the correct candidate in the first frame, and normalizes all position features in subsequent frames by subtracting from them the position of the k'th candidate in the first frame. Note that this normalization is only applied to the position component of the feature vector. The velocity features used in the experiments are translation-invariant by definition. When all frames have been processed, to find the best match of the observation sequence with the model, we need to find which of the K parallel DSTW processes gave the lowest matching cost D^*.

DSTW takes $O(Kmn)$ time; the translation invariant version takes $O(K^2mn)$. Overall, adding translation invariance increases both the space and the time complexity of the algorithm by a factor of K.

5. Experiments

To test the DSTW algorithm we implemented a hand-signed digit recognition system in Matlab. For the experiments we have collected video clips of three users gesturing the ten digits in the style of Palm's Grafitti Alphabet [14] (Figure 2). The video clips were captured with a Logitech 3000 Pro camera using an image resolution of 240×320. A total of 270 digit exemplars were extracted from three different types of video clips depending on what the user wore:

- Colored Gloves: 30 digit exemplars per user were stored in the database (See Figure 3).
- Long Sleeves: 30 digit exemplars per user were used as queries.
- Short Sleeves: 30 digit exemplars per user were used as queries.

Given a query frame, K candidate hand regions of size 40×30 were detected as described in Section 3.1. For every candidate hand region in every query frame, a feature vector was extracted and normalized as described in Section 3.2. The query digit was then matched with the model exemplars in the database: for the user-dependent experiments, 30 query digits of one user were matched with 30 database digits of the same user; for the user-independent experiments, 30 query digits of one user were matched with all 60 database digits of the two other users. The class of the query was estimated using the one nearest neighbor (1-NN) rule, and classification accuracy rates were averaged over the three users. Examples of a correct match and a false match are shown in Figures 6 and 7 respectively.

5.1. Experiment 1: DSTW vs. DTW

The purpose of the first experiment is to demonstrate that the DSTW algorithm outperforms the simple DTW algorithm when using a hand detection method based on color and motion [2]. The classification rates depicted in Table 1 show a significant $(11.1\% - 21.1\%)$ increase in classification accuracy between the simple DTW algorithm, which can only handle a single (best) candidate, and the proposed DSTW algorithm, which can handle multiple candidates. In addition, the graphs in Figure 4 show the initial decreasing trend of the classification error rate as K increases. At some point the error rate stops decreasing since additional candidates cause more false matches. The optimal value for K can be estimated using cross validation.

The results in Table 1 also show that the classification accuracy rates for the short sleeves sequences are slightly worse than the classification accuracy for the long sleeves sequences. This is to be expected, because the gesturing hand is more accurately localized when the user wears a long sleeved shirt. However, it is important to note that the classification accuracy for the short sleeves sequences would be much worse without handling multiple candidate observations, unless much more sophisticated hand segmentation and detection algorithms were employed.

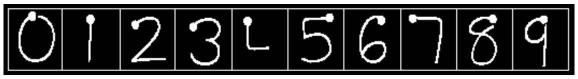

Figure 2: Palm's Graffiti digits.

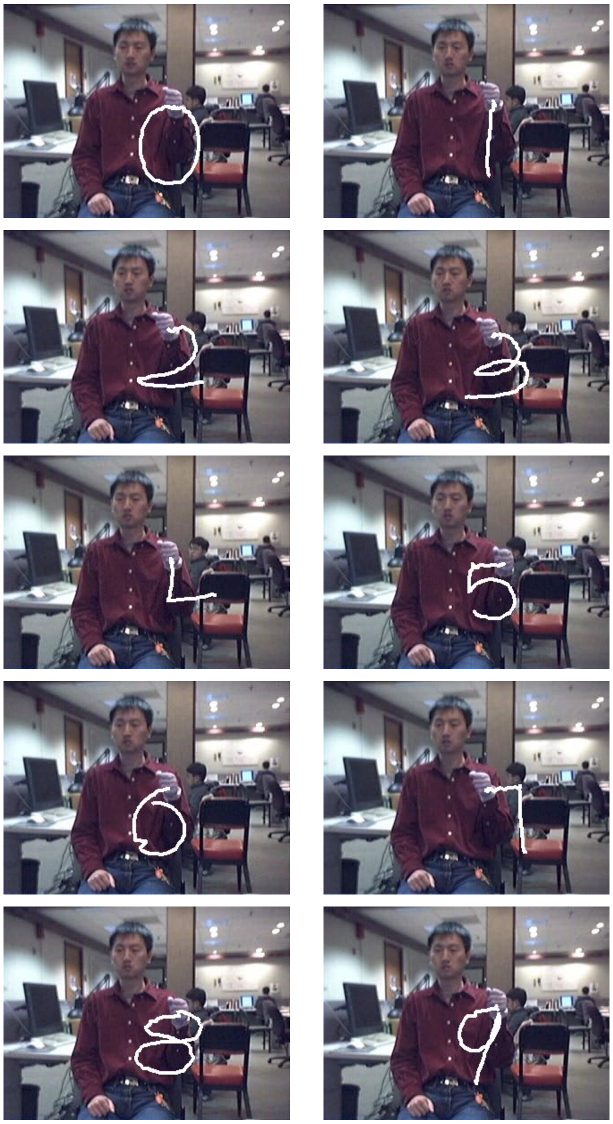

Figure 3: Example model digits extracted using a colored glove.

Experiment	User-dep.		User-indep.	
Method	DTW	DSTW	DTW	DSTW
Long Sleeves	81.1	96.7	76.7	91.1
Short Sleeves	82.2	93.3	70.0	91.1

Table 1: Classification accuracy results. The results for DSTW are for $K = 8$.

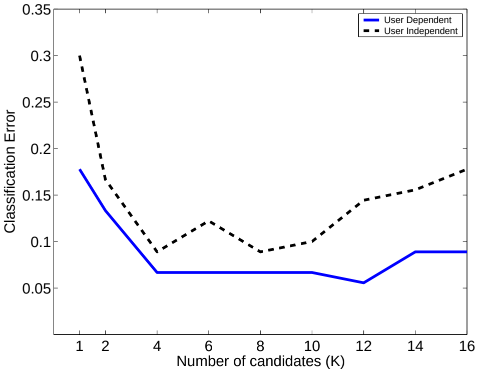

Figure 4: Classification error as a function of the number of candidates for the experiment with short sleeves sequences. The experiment with long sleeves sequences showed similar behavior.

5.2. Experiment 2: Translation Invariance

The purpose of the second experiment is to demonstrate the additional benefit of incorporating translation invariance in the DSTW framework as proposed in Section 4.1. In principle, translation invariance could be obtained using translation invariant features such as relative position with respect to the hand position in the first frame, or velocity. However, in practice the hand position in the first frame is not known, and using only velocity as a feature causes dramatic drops in classification accuracy. For example, accuracy drops from 85.6% to 22.2% in the user-independent experiment with short sleeves sequences using $K = 12$.

On the other hand, if both absolute position and velocity are included in the feature vector and translation invariance is not handled, then if there is a shift of the gesture in the image plane, the classification error rate will increase. The graph in Figure 5 shows the increase of the error rate as a function of (a synthetic) increase in translation, for the user-independent experiment with short sleeves sequences. Clearly, for the translation invariant formulation, the error rate does not change as translation increases, as indicated by the horizontal line in the graph. We also note that the reason that the error rates when using absolute position and velocity are relatively low for small translation is that all gestures were performed approximately at the same location in the image plane.

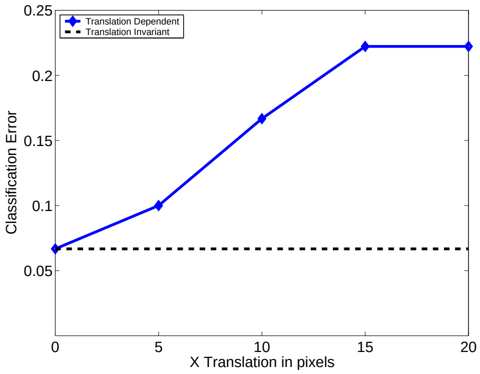

Figure 5: Classification error as a function of translation for the user-independent experiment with the short sleeves sequences and $K = 8$. The experiment with long sleeves sequences showed similar behavior.

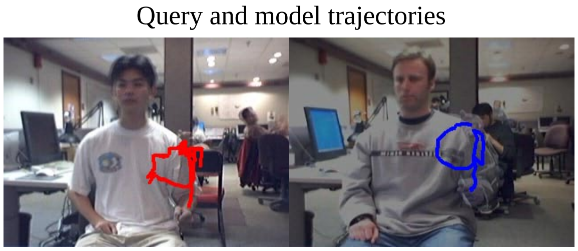

Figure 6: Example query trajectory (left) and corresponding model trajectory (right) for a correct match between two users signing the digit 9.

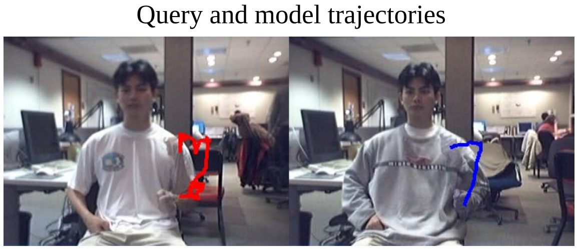

Figure 7: Example confusion between query digit 3 (left) and model digit 7 (right). In the final segment of the query digit 3 the elbow rather than the hand is falsely matched with the hand of model digit 7.

6. Conclusion and Future Work

Dynamic space-time warping (DSTW) is a general method for matching time-series, that can accommodate multiple candidate feature vectors at each time step. In this paper DSTW has been applied to the simultaneous localization and recognition of dynamic hand gestures. The algorithm can recognize gestures using a fairly simple hand detection module that yields multiple candidates. The system does not break down in the presence of a cluttered background, multiple moving objects, multiple skin-colored image regions, and users wearing short sleeves shirts.

Incorporation of translation invariance further increases the system flexibility by allowing the user to gesture in any part of the image. Scale invariance may be obtained through the commonly used image pyramid method, or alternatively by detecting certain body parts, like the head and shoulders, and measuring their size. Implementing scale invariance remains a topic for future investigation.

Another aspect of the problem that has not been addressed so far is temporal segmentation. In our experiments, the system knew the starting and ending frame of each gesture. In a real application, the user could indicate the start and end of a gesture, for example by using a distinct pose for the non-gesturing hand [8], or by pressing a key.

Our current implementation uses very simple features, both for hand detection and for DSTW matching. We expect accuracy to improve as we include more expressive features, such as appearance-based features like edges, orientation histograms [8], or optical flow correlation [6] for recognition. We are currently working on efficient and accurate methods for combining such features within a dynamic framework.

References

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

[2] 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.

[3] 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*, pages 82–89, 2001.

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

[5] T. Darrell and A. Pentland. Space-time gestures. In *Proc. CVPR*, pages 335–340, 1993.

[6] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ICCV*, pages 726–733, 2003.

[7] W. Freeman. Computer vision for television and games. In *Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 118, 1999.

[8] W. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *Automatic Face and Gesture Recognition*, pages 296–301, 1995.

[9] M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. ECCV*, pages 893–908, 1998.

[10] M. Jones and J. Rehg. Statistical color models with application to skin detection. *IJCV*, 46(1):81–96, January 2002.

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

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

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

[14] Palm. Grafitti alphabet. http://www.palmone.com/us/products/input/.

[15] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *PAMI*, 19(7):677–695, July 1997.

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

[17] C. Rasmussen and G. Hager. Probabilistic data association methods for tracking complex visual objects. *PAMI*, 23(6):560–576, June 2001.

[18] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *PAMI*, 20(1):23–38, January 1998.

[19] 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.

[20] Y. Sato and T. Kobayashi. Extension of hidden markov models to deal with multiple candidates of observations and its application to mobile-robot-oriented gesture recognition. In *Proc. ICPR*, pages II: 515–519, 2002.

[21] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *SCV95*, pages 265–270, 1995.

[22] J. Triesch and C. von der Malsburg. Robotic gesture recognition. In *Gesture Workshop*, pages 233–244, 1997.

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages I:511–518, 2001.

[24] Q. Yuan, S. Sclaroff, and V. Athistos. Automatic 2D hand tracking in video sequences. In *Proc. WACV*, 2005.