Continuous hand gesture recognition requires the detection of gestures in a video stream and their classification. In this paper two continuous recognition solutions using Hidden-Markov-Models (HMMs) are compared. The first approach uses a motion detection algorithm to isolate gesture candidates followed by a HMM recognition step. The second approach is a single-stage, HMM-based spotting method improved by a new implicit duration modeling. Both strategies have been tested on continuous video data containing 41 different types of gestures embedded in random motion. The data has been derived from usability experiments with an application providing a realistic visual dialog scenario. The results show that the improved spotting method in contrast to the motion detection approach can successfully suppress random motion providing excellent recognition results.

1. INTRODUCTION

Gestures are an efficient communication modality in many everyday situations. To study and optimize the possible use of gestures in human-machine interaction a visual dialog system, which is exclusively controlled by gestural commands, has been developed (see fig. 1 and sec. 2). The central task of the system is the temporal segmentation and classification of image sequences.

Since the HMM-based recognition of isolated image sequences works satisfactorily (e.g. see in [1, 4, 5, 8, 9]), it suggests itself to add an independent motion detection to achieve the temporal segmentation. The advantage of this first approach is the low additional computational cost of the implemented detection which is based on the image features (see sec. 3.2). Since, of course, any kind of motion is detected, the efficiency of this approach can only be measured in a realistic dialog scenario (see secs. 2 and 4).

The second approach is an integrated, HMM-based spotting method which was introduced in [6]. Tests using synthetically connected gesture material out of a small size catalog proved the ability to ignore non-meaningful and to indicate meaningful movements [7]. For the first time, the newly improved spotting algorithm (see sec. 3.3) has been applied to true continuous video data and tested against the above two-stage solution (see sec. 6).

2. THE VISUAL DIALOG APPLICATION

The used visual dialog application is a gesture controlled 3-D scene editor. The editor allows to create three-dimensional objects, to change their position and orientation, and to destroy them. Certain object attributes, like color and size, can be varied. Finally, the orientation and the observation distance of the whole scene can be changed.

The objects in the scene editor are manipulated indirectly: the editor contains a graphically represented agent which is supposed to be the communication partner. This agent receives gestural directives from the user, interprets them, and carries out the desired actions to a certain extent independently. These actions and the internal state of the application are represented by graphical animations and appearance changes. Complex or abstract actions, like gen-
erating and destroying objects or changing their attributes, are assigned to graphical objects in the scene. As a consequence, those actions can be consistently selected and grabbed like any other scene object.

3. DESCRIPTION OF ALGORITHMS

3.1. Spatial segmentation and feature extraction

Previous tests showed that the hand shape provides enough information to distinguish image sequences [6, 7]. The hand shape is obtained using a segmentation method trained on skin color. The segmented binary images are transformed into Hu moment invariants (HMIs) [2]. The HMIs \( h_i \) up to order \( H \), the differences of the HMIs of successive images \( \Delta h_{ij} \), the difference of the shape areas \( \Delta A_i \) and of the centers of mass \( \Delta \bar{x}_i, \Delta \bar{y}_i \) form a feature vector \( \mathbf{v}_i \) at time \( t \):

\[
\mathbf{v}_i = [v_{1,i},v_{2,i},\ldots,v_{2NH,i},\bar{x}_i,\bar{y}_i]^T = [\Delta A_i, \Delta \bar{x}_i, \Delta \bar{y}_i, h_{1,i},\ldots,h_{NH,i}, \Delta h_{1,i},\ldots,\Delta h_{NH,i}]^T. \tag{1}
\]

The mean value \( \mu_{vi} \) and standard deviation \( \sigma_{vi} \) of the feature vector elements \( v_{i,t} \) are used to calculate an unbiased and normalized feature vector \( \mathbf{v}'_i \) with comparable elements:

\[
\mathbf{v}'_i = \frac{\mathbf{v}_i - \mu_{vi}}{\sigma_{vi}}. \tag{2}
\]

The feature vector \( \mathbf{v}'_i \) is the basis of the motion detection, the isolated recognition and the continuous spotting process.

3.2. Approach 1: motion detection and HMM-based isolated recognition

The differential feature vector elements \( \mathbf{v}'_i \) reflect motion or shape changes in the image sequence. The absolute value of a feature vector built from these differential elements is defined as the motion value of the image sequence at time \( t \):

\[
m_t = \sqrt{\Delta A_t^2 + \Delta \bar{x}_t^2 + \Delta \bar{y}_t^2 + \Delta h_{1,t}^2 + \ldots + \Delta h_{NH,t}^2}. \tag{3}
\]

To indicate the begin \( b_t \) of a coherent segment of motion, this motion value has to be above a certain motion threshold \( m_{thres} \) for the next \( \tau_{min} \) subsequent images of the sequence. The end of a motion segment \( e_t \) is reached if the motion value stays below the threshold \( m_{thres} \) for at least the following \( \tau_{min} \) images. This is necessary since many gestures contain short motion pauses at turning points. A minimum total length of the motion segment \( \tau_{min} \) and a minimum distance to the last detected motion \( \tau_{min} \) help to find only motion segments that are possible gesture candidates.

Semi-continuous, left-to-right structured HMMS were used to classify the motion segments. The specific probability density functions \( F_{s_i,t} = \log f_{s_i}(\mathbf{v}'_i) \) in the states \( s_i \), \( i = 1,\ldots,N \) of the HMMS are defined by a codebook of \( L \) mixture density functions (or prototypes) calculated for the whole training data. The transition probabilities \( A_{s_i,s_j} = \log a_{s_i,s_j} \) describe the sequence of states.

Training and recognition are based on the Viterbi algorithm [3]. It recursively accumulates and maximizes the so-called local score \( D_{s_i,t} \) for every HMM state:

\[
D_{s_i,t} = \max_j [D_{s_j,t-1} + A_{s_j,s_i} + F_{s_i,t}], \tag{4}
\]

To classify a motion segment, the score accumulation starts at time \( t_b \) and results in the final score \( D_{s_k,t_e}^{(s)} \) in the last state of the respective models \( \lambda \) at the end of a detected motion. A final maximum likelihood decision provides the best matching model \( \lambda_k \). As a result, a gestural meaning is assigned to every detected motion segment.

3.3. Approach 2: HMM-based spotting

To spot gestures in a continuous video stream, the features are fed into the HMMS at every time step. To prevent the output score \( D_{s_k,t} \) from increasing or decreasing permanently, the local scores have to be normalized to its respective Viterbi path lengths [6, 7]. For that reason, the local path lengths \( L_{s_{j,t}} \) are stored along with the local scores, and a normalized Viterbi algorithm is formulated:

\[
D_{s_i,t} = \max_j \left[ \frac{D_{s_j,t-1} \cdot L_{s_j,t-1} + A_{s_j,s_i} + F_{s_i,t}}{L_{s_{j,t-1}} + 1} \right],
\]

\[
L_{s_{k,t}} = L_{s_{k,t-1}} + 1 \quad \text{with} \quad k = \text{index of best} \ s_j. \tag{5}
\]

Several methods to trigger new paths in the first state \( s_1 \) have been examined [7]. Optimal results are obtained if a new path with the length \( L_{s_{1,t}} = 1 \) starts permanently. The output score of the respective model will start to increase if an appropriate gesture appears; it will decrease after the gesture has ended. Consequently, peaks in the output score indicate the possible end of a gesture.

A smoothing process and several peak detection rules are necessary for a reliable peak detection requiring the following variables [7]: \( \tau_{ib} \) and \( \tau_{eb} \) define the smoothing interval, \( \tau_{pb} \) and \( \tau_{pe} \) the peak detection interval. A relative rejection threshold \( S_{rej} \), based on the model dependent absolute maximum and minimum output scores, and a minimum temporal peak distance \( \Delta_{dist} \) help to suppress irrelevant peaks.

The local path length can be manipulated allowing a simple duration modeling. Given the estimated average duration [3] of the whole HMM as \( \lambda^1 \):

\[
\bar{d} = \sum_{i=1}^{N} d_{s_i} \approx \frac{N}{N-1} \sum_{i=1}^{N-1} \frac{1}{1-a_{ii}}, \tag{6}
\]

\[1\text{The approximation in eq. (6) is necessary since in a left-to-right model the transition probability of the last state is always 1.} \]
number of variations

<table>
<thead>
<tr>
<th>functional category</th>
<th>number of variations</th>
</tr>
</thead>
<tbody>
<tr>
<td>displace</td>
<td>23</td>
</tr>
<tr>
<td>rotate</td>
<td>4</td>
</tr>
<tr>
<td>tilt</td>
<td>2</td>
</tr>
<tr>
<td>change size</td>
<td>4</td>
</tr>
<tr>
<td>stop</td>
<td>2</td>
</tr>
<tr>
<td>release</td>
<td>3</td>
</tr>
<tr>
<td>point</td>
<td>1</td>
</tr>
<tr>
<td>trigger action</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>41</td>
</tr>
</tbody>
</table>

Table 1: Gesture catalog

a new normalization length function can be defined as:

$$L'_{\text{norm}} = \begin{cases} L_{\text{norm}} & \text{for } L_{\text{norm}} < \bar{d} \\ v(L_{\text{norm}} - \bar{d}) + \bar{d} & \text{for } L_{\text{norm}} > \bar{d} \end{cases}$$ (7)

As a result, Viterbi paths are artificially “extended” if they are longer than the average model duration and if the normalization function parameter $v$ is greater than 1. This indirectly helps shorter Viterbi paths to grow since the score of longer paths is reduced. The normalization length function can diminish the error rate significantly (see sec. 6).

4. “WIZARD OF OZ” EXPERIMENTS AND TEST DATA DESCRIPTION

The 3-D scene editor (see sec. 2) was tested in “Wizard of Oz” experiments employing a human “wizard” to recognize the gestures and to control the scene editor remotely. During the experiments the test persons invented over 60 different gestures to operate the editor. The 41 most frequently used gestures have been selected forming a catalog which is the basis of all the following tests (see table 1).

All the training and test material is contained in a continuous video sequence 46 minutes in length containing gestures of a single person. The motion JPEG compressed version of the video takes about 50 images around a manually labeled gesture. This interval is chosen very large to be sure to obtain any gesture candidate for the subsequent classification. The recognition rate $r_d$ is the “ratio of correctly detected motion segments to the total number of valid gestures”. The false detected rate $f_d$ represents the relative number of wrongly detected motion segments. The multiple detection rate $r_{d,mult}$ is the relative number of motion segments that are repeatedly assigned to one gesture.

Similar to the above definition, a gesture that ends at time $t_g$ is defined as correctly recognized if the system indicates it at a time $t_g^2$ that lies within an interval of ±50 images around $t_g$. The temporal recognition delay is $t_d = t_p - t_g$. The recognition rate $r$ is the “ratio of correctly recognized gestures to the total number of valid gestures”. The multiple recognition rate $r_{mult}$ measures correctly but repeatedly recognized gestures. $t_d$ is the average recognition delay of correctly recognized gestures. The false accept rate $f$ is measured in fa/kg/h = “number of wrongly accepted gestures/number of key gestures/hour”.

The HMM and spotting parameters have been extensively varied to empirically find the optimal recognition results. These optimal parameters are the same in all the result tables 2–4: maximum order of HMIs $H = 2$, smoothing interval parameters $\tau_{pb} = 30$ and $\tau_{pe} = 0$, peak detection interval parameters $\tau_{pb} = 10$ and $\tau_{pe} = 1$, minimum temporal peak distance $t_{dist} = 10$, number of HMM states for isolated recognition $N = 5$ and for spotting $N = 15$.

6. EXPERIMENTAL RESULTS

6.1. Motion detection and isolated recognition

The motion detection parameters (see sec. 3.2) were determined by a numerical optimization process. Three different optimization strategies M1–M3 were used: M1 is emphasizing a maximum $r_d$, M2 a maximum $r_d$ combined with a minimum $r_{d,mult}$, and M3 a minimum $r_{d,mult}$ (see table 2). The appropriate detection results demonstrate that more than 90% of the possible gestures can only be detected if a high multiple detection rate is tolerated (see table 2, M1).

The classification results of the motion segments prove that a high multiple detection rate results in a high false recognition rate: obviously many motion segments are non-meaningful and cannot be correctly recognized (see table 3).

$^2t_p = t_g$ for motion detected gestures.

$^3$The number of different key gestures is 41 (see table 1).
A one-stage and a two-stage approach to continuous hand gesture recognition were compared. A realistic visual dialog scenario provided video data containing a typical mixture of meaningful and non-meaningful motion and a catalog of 41 different gestures. The results proved that the one-stage HMM-based spotting method is far superior to the straight-forward, two-stage algorithm that combines an independent motion detection with an isolated HMM recognition. The spotting could be further improved introducing independent motion detection with an isolated HMM recognition. The spotting results can even be improved if the normalization function parameter \( v \) is raised (see eq. (7) and table 3). Increasing \( v \) without using a rejection threshold, diminishes the false accept rate significantly (more than 40%) while the recognition rate is only reduced slightly (less than 4%). At a constant recognition rate of 90% the false accept rate can still be improved by 13%.

### 7. CONCLUSION

A one-stage and a two-stage approach to continuous hand gesture recognition were compared. A realistic visual dialog scenario provided video data containing a typical mixture of meaningful and non-meaningful motion and a catalog of 41 different gestures. The results proved that the one-stage HMM-based spotting method is far superior to the straight-forward, two-stage algorithm that combines an independent motion detection with an isolated HMM recognition. The spotting could be further improved introducing a new implicit duration modeling.

### References


