AN INTEGRAL STOCHASTIC APPROACH TO IMAGE SEQUENCE SEGMENTATION AND CLASSIFICATION

Peter Morguet and Manfred Lang

Institute for Human-Machine-Communication
Munich University of Technology
Arcisstr. 21, D-80290 Munich, Germany
{mor, lg}@mmk.e-technik.tu-muenchen.de

ABSTRACT
Finding and identifying characteristic or meaningful image sequences in a continuous video stream is a challenging task with many applications. This paper presents a new and efficient approach to these temporal segmentation and classification problems based on Hidden Markov Models (HMMs). The basic principle consists in continuously observing the output scores of the HMMs at every time step. Peaks, which appear in the individual HMM output scores, allow to determine in an integral way which image sequence occurred at what time. The application of our method to the spotting of connected dynamic hand gestures provided excellent recognition results and a high temporal accuracy.

1. INTRODUCTION
The validity and performance of a HMM-based approach to the classification of isolated image sequences have been recently demonstrated by some works (e.g. [1, 5, 6, 7, 8, 9]). All of these works deal with manually labeled image sequence material. However, many applications in a natural environment — like our gesture recognition task — require an automatic temporal segmentation. The corresponding task is now to detect so called key image sequences in a continuous video sequence and to identify them. With our stochastic approach these two problems can be solved as an integral process.

This problem is related to the procedure of keyword spotting in speech recognition where HMMs are successfully used for a long time [3]. But there is a main difference: continuous speech is composed of a defined and countable number of keywords and non-keywords, whereas in the case of a continuous video stream a defined number of key image sequences is embedded in a background of an indefinite number of movements and transitions.

For that reason, many spotting approaches in speech processing that explicitly model non-keywords or even integrate a language model are difficult to transfer to the image processing domain. Our new approach is rather an adapted and improved version of a context free spotting method using HMMs only for the key image sequence modeling [4].

Every isolated key image sequence is represented by a HMM $\lambda_1, \lambda_2, \ldots, \lambda_M$, which has to be trained with manually segmented video sequences (see fig. 1). The features of the continuous video stream (see sec. 2.1) are fed into the HMMs producing a characteristic course of the output score at the respective HMMs; the HMMs can be considered as "continuous filters". Using one of our modified Viterbi algorithms (see sec 2.2), the output score of a HMM increases if it describes the momentary input video stream well, otherwise it decreases (see fig. 2). The maximum score of the matching HMM is reached at the end of a key image sequence so that a peak finding algorithm can indicate both the time of occurrence and the index of the appropriate key sequence. Since in practice the peak detection in the pure output signal is too unreliable, a smoothing step and a set of decision rules was added (see sec. 2.3).

2. SYSTEM DESCRIPTION

2.1. Preprocessing and feature extraction

A color histogram based segmentation method calculates binary images solely containing the hand shape. Afterwards the image sequences are transformed into feature vectors. Many possibilities to generate features out of images are de-
scribed in the literature [1, 5, 6, 7, 8, 9]. Our experiments showed that simple but fast calculable feature vectors for binary images can be built out of the Hu moments $h_t$, up to order $H$ [2], the difference of the shape areas $\Delta h_t = h_t - h_{t-1}$, the difference of the centers of mass $\Delta x_t = x_t - x_{t-1}$ and $\Delta y_t$. The resulting feature vector at time $t$ is

$$v_t = [\Delta A_t, \Delta x_t, \Delta y_t, h_{1,t}, \ldots, h_{N_{hi},t}, \Delta h_{1,t}, \ldots, \Delta h_{N_{hi},t}]^T. \quad (1)$$

2.2. Normalized Viterbi algorithms

The used HMMs are semi-continuous since those models are a good compromise between few training data and accuracy of modeling [3]. Semi-continuous HMMs have a codebook of mixture density functions (or prototypes) calculated for the whole training data. The specific probability density functions (pdfs) $f_{s_i}(v_t)$ in the states $s_t$, $i = 1, \ldots, N$ are weighted sums of the prototypes.

The models are trained using the standard Viterbi algorithm [3]. Using the state pdfs $F_{s_i,t} = \log f_{s_i}(v_t)$ and the transition probabilities $A_{i,j,s_t} = \log a_{s_i,s_t}$, the Viterbi algorithm recursively accumulates and maximizes the local score $D_{s_t,t}$ for every HMM state:

$$D_{s_t,t} = \max_j [D_{s_j,t-1} + A_i,s_t] + F_{i,t}. \quad (2)$$

The output score, which is the score of the last state $D_{s_N,t}$, is crucial to the continuous recognizing process. But the standard Viterbi algorithm of eq. (2) cannot be used since depending on the average state pdfs $F_{s_j,t}$ the output score will permanently increase or decrease on the average. To stabilize the average score, it has to be normalized to its respective Viterbi path length. Two normalization procedures have been examined:

(N1) If the score $D_{s_t, t}$ is just normalized to the total path length $L_{\text{total}} = t$, the contribution of a new Viterbi step will decline according to the increasing total time $t$. For that reason, a new normalization method was introduced using a constant length $L_a$, which can be recursively formulated as:

$$D_{s_t,t} = \left[\max_j [D_{s_j,t-1} \cdot L_a + A_{s_t,s_j}] + F_{s_t,t}\right] \frac{1}{L_a + 1}. \quad (3)$$

It can be shown that this normalization leads to an exponentially decreasing influence of the “older” scores in the Viterbi path which will stabilize the output score (example see fig. 2a). If the length $L_a$ is zero, the output score is build without the preceeding path history.

(N2) In addition to the local score $D_{s_t,t}$ a local path length $L_{s_t,t}$ can be introduced:

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

$$L_{s_t,t} = L_{s_j,t-1} + 1 \quad \text{with } k \text{ is index of best } s_j. \quad (4)$$

This allows recombining paths to have different lengths. According to [4] it is now possible that at any time $t$ a new path with $L_{s_t,t} = 1$ and $D_{s_t,t} = 0$ may start in competition with the normal continuation of the Viterbi path from the preceeding state $s_1$. But using this initialization, the score history is compared to an arbitrary constant $D_{s_1, t} = 0$. To be able to control the “trigger” probability of a new path, a new adjustable entry weight $D_{s_t,t} = W$ was defined (example see fig. 2c).

2.3. Smoothing and peak finding rules

The output scores of an HMM $\lambda_t$ using either normalization method can be quite ragged (see fig. 2a and c). To simplify
the peak search, the scores are smoothed by averaging the scores from $-\tau_{sb}$ to $\tau_{re}$:

$$D^{(l_i)}_{SB,t} = \frac{1}{\tau_{we} + \tau_{sb} + 1} \sum_{t=-\tau_{sb}}^{\tau_{we}} D^{(l_i)}_{SB,t+1}$$  \hspace{1cm} (5)$$

(examples see fig. 2b and d). After that, four decision rules R1–R4 are applied to find a valid peak at time $t_p$. The smoothed output score $D^{(l_i)}_{SB,t}$ of model $l_i$

**R1:** must be the maximum in an increasing score series from $t_p - \tau_{ph}$, ..., $t_p$, and a decreasing score series from $t_p$, ..., $t_p + \tau_{ph}$,

**R2:** must be greater than a model dependent rejection threshold $D_{thres}$,

**R3:** must have the highest score compared to the scores of all the other models $l_j$, and

**R4:** must have a *minimum temporal distance* of $t_{dist}$ to the last valid peak found.

The model dependent threshold of rule R2 is expressed by a single relative rejection threshold $S_{rel}$ with the help of the model specific maximum and minimum scores:

$$D^{(l_i)}_{thres} = D^{(l_i)}_{max} - S_{rel} \cdot \left[D^{(l_i)}_{max} - D^{(l_i)}_{min}\right],$$  \hspace{1cm} (6)$$

The smoothing process and the rules R1–R4 turned out to be very important for the robustness and effectiveness of the spotting process (see sec. 4).

### 3. TEST DATA DESCRIPTION

The gesture spotting system is planned to be a part of a three-dimensional graphics scene editor that can be visually controled by hand and head gestures [5, 6]. Performing a number of “Wizard of Oz” experiments, a catalog of 12 commonly used hand gestures could be determined (see table 1). These gestures form the basis of the following tests.

In the experimental setup, the camera was mounted above a uniformly colored table area looking downward to the right hand of the user. Each of the 12 gestures was recorded 30 times and stored as an isolated key image sequence. All gestures were performed by a single person. Each image sequence contained 70 non-interlaced images at the European rate of 50 images (fields) per second. The final size of the images was 192 x 144 pixels.

The image material was devided in 20 training and 10 test sequences. Continuous training and test sequences were generated out of the isolated training and test sequences by linking them together using filler sequences of the length $L_{fill}$ (in images). The filler sequences contained linearly interpolated feature vectors that smoothly connected successive key image sequences. The “Wizard of Oz” experiments showed that this is a very good approximation of the real user behaviour: mostly distinct gestures with smooth transitions were performed.

The 12 HMMs were trained with the isolated training sequences. The continuous training sequences were used to determine the model dependent minimum and maximum scores, which are needed to calculate the absolute thresholds in eq. (6). Finally the continuous test sequences were used to evaluate the spotting system.

### 4. EVALUATION CRITERIA AND EXPERIMENTAL RESULTS

A gesture that ends at time $t_g$ is defined as correctly recognized if the system indicates it at a time $t_p$ that lies within an interval of $\pm 35$ images around $t_g$ (arbitrary defined as half the length of a key gesture). The temporal detection delay is $t_d = t_p - t_g$. The **recognition rate** $r$ is the “ratio of correctly recognized gestures to the total number of key gestures” in a continuous sequence. $t_{dist}$ is the **average detection delay** of correctly recognized gestures. The **total average delay** $t_d$ between occurrence and detection of a gesture is $t_d = t_d + \tau_{sb}$. The false accept rate $f$ is measured in $fa/kg/h = \text{“number of wrongly accepted gestures/number of key gestures/hour”}$ (in analogy to keyword spotting, e.g. see in [4]).

The HMMs had 256 prototypes and 25 states for all results shown. The feature vectors according to eq. (1) contain Hu moments up to the order $H = 2$. The smoothing and peak detection intervals ($\tau_{sb}$, $\tau_{we}$, $\tau_{ph}$, $\tau_{pe}$), the rejection threshold $S_{rel}$ and the minimum temporal peak distance $t_{dist}$ were extensively varied to empirically find the optimal results (see tables 2–4 for the respective values). The length of the filler sequences reached from $L_{fill}$ = 35 to 140 to simulate the usual transition durations between gestures.

Table 2 shows the results applying normalization method N1 used in eq. (3). While a recognition rate of 95% for $L_{fill}$ = 35 is acceptable, the false accept rate increases significantly for longer filler sequences even if a low rejection threshold $S_{rel}$ is used (the increasing recognition rate for a lower $S_{rel}$ is due to the high $t_{dist}$).

<table>
<thead>
<tr>
<th>#</th>
<th>action</th>
<th>#</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>go to the front</td>
<td>7</td>
<td>reset</td>
</tr>
<tr>
<td>2</td>
<td>go to the left</td>
<td>8</td>
<td>grab</td>
</tr>
<tr>
<td>3</td>
<td>go to the rear</td>
<td>9</td>
<td>release</td>
</tr>
<tr>
<td>4</td>
<td>go to the right</td>
<td>10</td>
<td>grab on the right</td>
</tr>
<tr>
<td>5</td>
<td>take this</td>
<td>11</td>
<td>grab on the right</td>
</tr>
<tr>
<td>6</td>
<td>no</td>
<td>12</td>
<td>stop action</td>
</tr>
</tbody>
</table>

Table 1: Gesture catalog
Table 2: Normalization N1: recognition rates $r$, false accept rates $f$ and average detection delays $t_d$ resulting from different fill lengths $L_{fill}$ and rejection thresholds $S_{rel}$ ($L_n = 15$,  
$f = 5.95$ (5.41), $105, 6143, 66.07, 45.640$, $t_d = 4.34, 5.61, 5.19, 6.09, 5.308$, $r = 0.95, 0.78, 0.85, 0.83, 0.853$, $f = 2.38, 16.96, 54.29, 32.94, 43.48$, $t_d = 4.19, 4.85, 5.10, 6.12, 5.065$, $0.05, 0.893$, $0.85$, $7.48$, $4.34$, $2.38$).

On the other hand, normalization method N2 shown in eq. (4) can be very efficient, provided that an appropriate entry weight $W$ is used (see table 3 for results at a constant $L_{fill}$). With an optimal weight of $W = 135$ and a rejection threshold of $S_{rel} = 0.05$, method N2 produces an average recognition rate of 99.5% with an average false accept error of 0 over the whole range of filler lengths (see table 4). Since both end times for the smooth and peak search intervals $t_{se}$ and $t_{pe}$ are 1 (this is the minimum value for $t_{pe}$), the total average time delay for gesture detection is $t_d = 8.29$ which is only about 0.17 seconds.

5. CONCLUSION

A new stochastic approach to the temporal segmentation and classification of image sequences was introduced. Using an improved HMM-based spotting method, both problems can be solved in an integral way. Applying this approach to the recognition of connected dynamic gestures provided very good recognition rates and low temporal detection delays. Depending on the underlying feature extraction, the approach is univerally applicable to many video spotting tasks.

6. REFERENCES


