Spotting Dynamic Hand Gestures in Video Image Sequences Using Hidden Markov Models

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

In this paper a new and general stochastic approach to find and identify dynamic gestures in continuous video streams is presented. Hidden Markov Models (HMMs) are used to solve this combined problem of temporal segmentation and classification in an integral way. Basically, an improved normalized Viterbi algorithm allows to continuously observe the output scores of the HMMs at every time step. Characteristic peaks in the output scores of the respective models indicate the presence of gestures. Our experiments in the domain of hand gesture spotting provided excellent recognition results and very low temporal detection delays.

1. Introduction

Human gestures appearing in a natural environment are *movements* rather than static postures. Consequently, a vision-based gesture recognition requires the classification of image *sequences*. Some previous works already demonstrated that a HMM-based recognition can cope with the variability and complexity of gestures or other human movements (e. g. [1, 5, 6, 9, 10, 11]). All these approaches use manually segmented image sequences.

Our new method, however, works with continuous video streams and allows an *automatic* temporal segmentation and a classification at the same time (see fig. 1). This so called *gesture spotting* is related to the procedure of *keyword spotting* in speech recognition. Our algorithm is an adapted and improved version of a context free keyword spotting method [4]. The first version of our approach was introduced in [7]. In this paper we present new procedures to increase recognition rates and to decrease the temporal detection delays. It is also explicitly shown that our HMM-based spotting method can suppress non-meaningful movements.

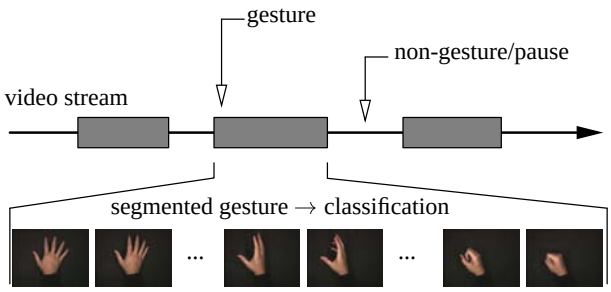

Figure 1: Spotting gestures in a continuous video stream

Every gesture is represented by a HMM $\lambda_1, \lambda_2, \ldots, \lambda_M$ (see fig. 2), which has to be trained with manually segmented video sequences. 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 model. Using a normalized Viterbi algorithm (see secs. 2.2 and 2.3), the output score of a HMM increases if it describes the momentary input video stream well, otherwise it decreases (example see fig. 3). The maximum score of the matching HMM is reached at the end of a gesture. Consequently, a peak finding algorithm can indicate which gesture appeared in the video stream at what time. To ob-

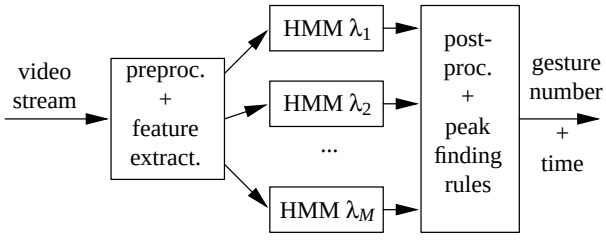

Figure 2: System overview

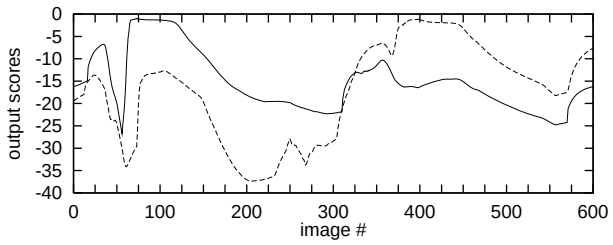

Figure 3: Example output scores of two models (solid and dashed); respective gestures end at image #100 and #400

tain a reliable peak detection, several postprocessing steps and a set of decision rules have to be applied (see secs. 2.4 and 2.5).

2. System description

2.1. Spatial segmentation and feature extraction

A color histogram based segmentation method calculates a sequence of *binary images* $f_{\mathrm{b},t}(x,y)$ containing only the hand shape which forms a sequence of regions R_t in the image plain [5, 6]. Afterwards, the image sequence is transformed into a sequence of *feature vectors* $\mathbf{v}_t$.

Experiments showed that simple but fast calculable feature vectors for binary images $f_{\mathrm{b}}(x,y)$ can be build out of shape describing moments. The general moments of binary images are defined as [2]:

$$m_{pq} = \sum_x \sum_y x^p y^q \cdot f_{\mathrm{b}}(x,y) = \sum_{(x,y)\in R} x^p y^q. \qquad (1)$$

The area A and the center of mass $(\bar{x},\bar{y})$ of the shape R can be expressed as:

$$A = m_{00}, \quad \bar{x} = \frac{m_{10}}{m_{00}}, \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}}. \qquad (2)$$

This defines the normalized central moments

$$\mu_{pq} = \frac{1}{A^{(p+q+2)/2}} \sum_{(x,y)\in R} (x^p - \bar{x})(y^q - \bar{y}). \qquad (3)$$

The Hu moment invariants (HMIs) h_i are combinations of the normalized central moments [2]. N_H different HMIs up to a particular order H indicating the maximum $p+q$ of the contained μ_{pq} can be constructed. They are invariant to translation, rotation, size changes, and reflection.

The feature vector $\mathbf{v}_t$ at time t is formed out of the Hu moment invariants $h_{i,t}$, the differences of the HMIs of successive images $\Delta h_{i,t}$, the difference of the shape areas ΔA_t, and of the centers of mass $(\Delta\bar{x}_t, \Delta\bar{y}_t)$:

$$\begin{aligned} \mathbf{v}_t &= [\Delta A_t, \Delta\bar{x}_t, \Delta\bar{y}_t, \\ &\quad h_{1,t}, \ldots, h_{N_H,t}, \Delta h_{1,t}, \ldots, \Delta h_{N_H,t}]^{\mathrm{T}}. \end{aligned} \qquad (4)$$

2.2. Normalized Viterbi algorithm

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) $f_{s_i}(\mathbf{v}_t)$ in the states s_i, $i = 1,\ldots,N$ are weighted sums of the prototypes.

The models are trained with the standard Viterbi algorithm [3, 8]. Using the state pdfs $F_{s_i,t} = \log f_{s_i}(\mathbf{v}_t)$ and the transition probabilties $A_{s_j,s_i} = \log a_{s_j,s_i}$, the Viterbi algorithm recursively accumulates and maximizes the local score $D_{s_i,t}$ for every HMM state:

$$D_{s_i,t} = \max_j [D_{s_j,t-1} + A_{s_j,s_i}] + F_{s_i,t}. \qquad (5)$$

The output score, which is the score $D_{s_N,t}$ of the last state, is crucial to the continuous *recognizing* process. But the standard Viterbi algorithm of eq. (5) cannot be used since, depending on the average state pdfs $F_{s_i,t}$, 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 [7].

For that reason, a local path length $L_{s_i,t}$, which allows recombining paths to have different lengths, is introduced [4]:

$$\begin{aligned} D_{s_i,t} &= \max_j \left[\frac{D_{s_j,t-1} \cdot L_{s_j,t-1} + A_{s_j,s_i} + F_{s_i,t}}{L_{s_j,t-1} + 1} \right], \\ L_{s_i,t} &= L_{s_k,t-1} + 1 \quad \text{with } k = \text{index of best } s_j. \end{aligned} \qquad (6)$$

2.3. Triggering new Viterbi paths

At any time t a new path may start in state s_1 in competition with the normal continuation of the Viterbi path from the preceeding state s_1. This *triggering* happens if a score threshold D_{tr} is higher than the local s_1-score at time $t = t_{\mathrm{tr}}$:

$$D_{\mathrm{tr}} > D_{s_1,t_{\mathrm{tr}}}. \qquad (7)$$

A new path always starts with length

$$L_{s_1,t_{\mathrm{tr}}} = 1. \qquad (8)$$

It should trigger at the beginning of a new gesture so that the local scores of the respective model can increase quickly. Apart from the following trigger method T1, two improved trigger strategies were newly introduced (see T2 and T3):

T1: According to [4] the trigger threshold is $D_{\mathrm{tr}} = 0$. A new path starts with the score of the trigger threshold:

$$D_{s_1,t_{\mathrm{tr}}} = D_{\mathrm{tr}} = 0. \qquad (9)$$

Since the threshold is constant, this method triggers *passively* if the local s_1-score is *lower* than the score threshold.

T2: An *active* trigger threshold can be defined according to eq. (6) but with a constant smoothing length L_s:

$$D_{\mathrm{tr},t} = \left[D_{\mathrm{tr},t-1} \cdot L_s + A_{s_1,s_1} + F_{s_1,t}\right] \frac{1}{L_s+1}. \qquad (10)$$

If the pdf $F_{s_1,t}$ is rising at the beginning of a new gesture and if L_s is small compared to the current Viterbi path length $L_{s_1,t}$, the threshold $D_{\mathrm{tr},t}$ will increase faster than the local score $D_{s_1,t}$, and a new path will begin. An entry weight W is added to the local s_1-score at trigger time t_{tr}:

$$\hat{D}_{s_1,t_{\mathrm{tr}}} = D_{s_1,t_{\mathrm{tr}}} + W. \qquad (11)$$

It will help the new path to be continued through successive states.

T3: It is also possible to trigger a new path permanently at *every* time t. Using the entry weight W of method T2, the local s_1-score becomes

$$\hat{D}_{s_1,t} = D_{s_1,t} + W \qquad (12)$$

with a constant path length $L_{s_1,t} = 1$.

2.4. Score postprocessing

The output score $D^{(\lambda_i)}_{s_N,t}$ of a HMM λ_i can be quite ragged (see fig. 3). To simplify the peak search, the scores are smoothed by averaging them from $t-\tau_{\mathrm{sb}}$ to $t+\tau_{\mathrm{se}}$ resulting in:

$$\bar{D}^{(\lambda_i)}_{s_N,t} = \frac{1}{\tau_{\mathrm{se}}+\tau_{\mathrm{sb}}+1} \sum_{\tau=-\tau_{\mathrm{sb}}}^{\tau_{\mathrm{se}}} D^{(\lambda_i)}_{s_N,t+\tau}. \qquad (13)$$

After smoothing, the shape of the peaks can be *optionally* optimized adding the temporal score difference to the score itself using a mixture factor C_{mix} (see table 6 at sec. 5.3 for effect):

$$\bar{D}'^{(\lambda_i)}_{s_N,t} = \bar{D}^{(\lambda_i)}_{s_N,t} + C_{\mathrm{mix}} \cdot \left[\bar{D}^{(\lambda_i)}_{s_N,t} - \bar{D}^{(\lambda_i)}_{s_N,t-1}\right]. \qquad (14)$$

2.5. Peak finding rules

After that, four decision rules R1–R4 are applied to find a valid peak at time t_{p} [7]. The smoothed output score $\bar{D}^{(\lambda_i)}_{s_N,t_{\mathrm{p}}}$ or $\bar{D}'^{(\lambda_i)}_{s_N,t_{\mathrm{p}}}$ of model λ_i

R1: must be the maximum in an increasing score series from $t_{\mathrm{p}}-\tau_{\mathrm{pb}},\ldots,t_{\mathrm{p}}$ and a decreasing score series from $t_{\mathrm{p}},\ldots,t_{\mathrm{p}}+\tau_{\mathrm{pe}}$,

R2: must be greater than a model dependent rejection threshold $\bar{D}^{(\lambda_i)}_{\mathrm{thres}}$,

#	action	#	action
1	go to the front	7	reset
2	go to the left	8	grab
3	go to the rear	9	release
4	go to the right	10	grab on the left
5	take this	11	grab on the right
6	no	12	stop action

Table 1: Gesture catalog

R3: must have the highest score compared to the scores of all the other models λ_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:

$$\bar{D}^{(\lambda_i)}_{\mathrm{thres}} = \bar{D}^{(\lambda_i)}_{\mathrm{max}} - S_{\mathrm{rel}} \cdot \left[\bar{D}^{(\lambda_i)}_{\mathrm{max}} - \bar{D}^{(\lambda_i)}_{\mathrm{min}}\right]. \qquad (15)$$

3. Test data description

A catalog of 12 hand gestures (see table 1) forms the basis of the following tests. These gestures are planned to be used to visually control a three-dimensional graphics scene editor [5, 6]. 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 192×144 pixels.

The image material was devided in 20 training and 10 test gestures. *Continuous* training and test video sequences were generated out of the *isolated* training and test gestures 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 gestures. The synthesis allows to control specific characteristics of the continuous image sequences. Besides, the real user behaviour is perfectly approximated: observations showed that mostly distinct gestures with smooth transitions are performed. The 12 HMMs were trained with the isolated training gestures. The continuous training sequences were used to determine the model dependent minimum and maximum scores of eq. (15). Finally the continuous test sequences were used to evaluate the spotting system.

S_{rel}		L_{fill}				average
		35	70	105	140	
w/o	r	0.91	0.87	0.87	0.86	0.878
	f	29.76	39.29	40.71	42.26	38.005
	t_{d}	10.73	12.51	14.08	15.41	13.183
0.05	r	0.88	0.85	0.85	0.84	0.855
	f	13.10	18.75	20.00	19.05	17.725
	t_{d}	10.71	12.64	14.24	15.58	13.293

Table 2: Trigger method T1 (recognition rates r, false accept rates f, and average detection delays $\bar{t}_{\text{d}}$, different L_{fill} and S_{rel})

S_{rel}		W			
		0	45	90	135
w/o	r	0.838	0.905	0.933	0.950
	f	40.625	35.150	32.470	38.633
	t_{d}	13.068	10.503	9.828	9.555
0.08	r	0.838	0.905	0.915	0.855
	f	22.785	14.090	8.408	6.905
	t_{d}	13.068	10.523	9.858	9.313

Table 3: Trigger method T2 (r, f, and $\bar{t}_{\text{d}}$, $L_{\text{s}} = 3$, different W and S_{rel}, average for all L_{fill})

S_{rel}	W	r	f	$\bar{t}_{\text{d}}$
w/o	0	0.850	40.668	13.275
	30	0.945	30.078	10.288
	60	0.990	26.623	9.279
	90	0.990	28.318	8.383
	120	1.000	33.453	7.760
	150	1.000	42.710	7.205
	180	1.000	66.223	6.498
	210	0.983	90.193	6.078
0.06	0	0.838	22.220	13.278
	30	0.935	9.880	10.340
	60	0.985	3.378	9.253
	90	0.988	0.625	8.358
	120	0.990	0.150	7.708
	150	0.975	0.000	7.093
	180	0.938	0.475	6.193
	210	0.913	5.490	5.730

Table 4: Trigger method T3 (r, f, and $\bar{t}_{\text{d}}$, different W and S_{rel}, average for all L_{fill})

4. Evaluation criteria and parameter settings

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 ± 35 images around t_{g}. The temporal detection delay is $t_{\text{d}} = t_{\text{p}} - t_{\text{g}}$. The *recognition rate* r is the "ratio of correctly recognized gestures to the total number of key gestures". $\bar{t}_{\text{d}}$ is the *average detection 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" (in analogy to keyword spotting, e. g. see in [4]).

The HMMs had 256 prototypes and 25 states, the feature vectors contain Hu moments up to the order $H = 2$, the smoothing and peak detection intervals are $\tau_{\text{sb}} = \tau_{\text{pb}} = 30$, $\tau_{\text{se}} = \tau_{\text{pe}} = 1$, the minimum temporal peak distance is $t_{\text{dist}} = 10$. The tables 2–6 show results obtained without applying a rejection threshold S_{rel} and with an optimally adjusted rejection threshold for the respective methods and parameters. The lengths of the filler sequences were $L_{\text{fill}} = 35$, 70, 105 and 140 to simulate the usual transition durations between gestures.

5. Experimental results

5.1. Comparison of trigger methods

Table 2 shows the results applying the passive trigger method T1. While a recognition rate of 88% for $L_{\text{fill}} = 35$ is acceptable, the false accept rate is very high, especially for longer filler sequences.

The active trigger method T2 performs better at an optimal entry weight of $W = 90$ (see table 3). But a recognition rate of 91.5% at a false accept rate of about 8.4 fa/kg/h indicates that the triggering is not reliable enough.

The results of trigger method T3 in table 4 show that a permanent trigger gives the best results: at an optimal entry weight of $W = 120$ the recognition rate is 99% at a false accept rate of only 0.15 fa/kg/h.

5.2. Suppressing non-meaningful gestures

Table 5 shows the results of method T3 using only the *first half* of the gesture models (see table 1) for recognition. Since the test data still contained *all* gestures, these results prove that the HMM-spotting can distinguish valid from invalid movements very effectively: this time a recognition rate of even 100% and a false accept rate of 0 can be reached ($W = 150$).

5.3. Decreasing temporal detection delays

Finally, table 6 shows that forming the output score according to eq. (14) can clearly diminish the detection delays. A mixture factor of $C_{\text{mix}} = 0$ indicates that the shape of the output score remains unchanged (compare to table 4 at an optimal entry weight of $W = 120$). Using a mixture factor of $C_{\text{mix}} = 3$, the recognition results are even slightly better while the detection delay decreases more than 55% down

S_{rel}	W	r	f	t_{d}
w/o	0	0.828	320.418	13.918
	30	0.913	281.430	12.948
	60	0.980	280.418	11.748
	90	0.980	288.393	10.830
	120	1.000	288.273	10.265
	150	1.000	331.073	9.743
	180	1.000	402.143	9.228
	210	1.000	446.668	8.808
0.08	0	0.828	54.940	13.918
	30	0.913	35.238	12.950
	60	0.980	16.785	11.725
	90	0.980	8.808	10.830
	120	1.000	4.998	10.265
	150	1.000	0.000	9.743
	180	0.980	20.120	9.183
	210	0.978	80.000	8.778

Table 5: Trigger method T3, first half of gestures (r, f, and $\bar{t}_{\text{d}}$, different W and S_{rel}, average for all L_{fill})

S_{rel}	C_{mix}	r	f	$\bar{t}_{\text{d}}$
w/o	0	1.000	33.453	7.760
	3	1.000	40.088	3.413
	6	0.970	40.313	0.753
	9	0.918	39.970	-0.055
	12	0.898	48.498	-0.608
0.08	0	0.990	0.150	7.708
	3	1.000	0.150	3.413
	6	0.950	2.323	0.773
	9	0.863	2.575	-0.378
	12	0.735	4.523	-1.175

Table 6: Trigger method T3, shape forming using eq. (14) (r, f, and $\bar{t}_{\text{d}}$, different C_{mix} and S_{rel}, average for all L_{fill}, $W = 120$)

to 3.4 images. If the mixing factor is further increased, the detection delay even turns negative but the recognition rates decrease significantly.

6. Conclusion

A new HMM-based method for the combined temporal segmentation and classification of video image sequences was introduced. The approach was applied to the spotting of dynamic hand gestures in continuous video image sequences and provided excellent recognition results. It could be demonstrated explicitly that our spotting method can successfully reject non-meaningful movements. The approach is universal and can be applied to register any meaningful image sequences in video streams.

7. References

[1] G. I. Chiou, J.-N. Hwang: *Lipreading from Color Motion Video*. ICASSP 1996, Atlanta, Vol. 4, pp. 2156–2159, 1996.

[2] M.-K. Hu: *Visual Pattern Recognition by Moment Invariants*. IRE Trans. Inform. Theory, vol. IT-8, pp. 179–187, Feb. 1962.

[3] X. D. Huang, Y. Ariki, M. A. Jack: *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[4] J. Junkawitsch, L. Neubauer, H. Hoege, G. Ruske: *A new keyword spotting algorithm with pre-calculated optimal thresholds*. ICSLP 1996, Philadelphia, pp. 2067–2070, 1996.

[5] P. Morguet, M. Lang: *Feature Extraction Methods for Consistent Spatio-Temporal Image Sequence Classification Using Hidden Markov Models*. ICASSP 1997, Munich, Vol. 4, pp. 2893–2896, 1997.

[6] P. Morguet, M. Lang: *A Universal HMM-Based Approach to Image Sequence Classification*. ICIP 1997, Santa Barbara, Vol. 3, pp. 146–149, 1997.

[7] P. Morguet, M. Lang: *An Integral Stochastic Approach to Image Sequence Segmentation and Classification*. ICASSP 1998, Seattle, Vol. 5, pp. 2705–2708, 1998.

[8] L. R. Rabiner: *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, Vol. 77, No. 2, pp. 257–286, Feb. 1989.

[9] G. Rigoll, A. Kosmala, M. Schuster: *A New Approach to Video Sequence Recognition Based on Statistical Methods*. ICIP 1996, Lausanne, Vol. 3, pp. 839–842, 1996.

[10] T. Starner, A. Pentland: *Visual Recognition of American Sign Language Using Hidden Markov Models*. International Workshop on Automatic Face- and Gesture-Recognition 1995, Zürich, pp. 189–194, 1995.

[11] J. Yamato, J. Ohya, K. Ishii: *Rocognizing Human Action in Time-Sequential Images using Hidden Markov Model*. IEEE Comp. Vision and Pattern Recog. 1992, pp. 379–385, 1992.