A Universal HMM-Based Approach to Image Sequence 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

In this paper a universal approach to the classification of video image sequences by Hidden Markov Models (HMMs) is presented. The extraction of low level features allows the HMM to build an internal image representation using standard training algorithms. As a result, the states of the HMMs contain probability density functions, so called *image density functions*, which reflect the structure of the underlying images preserving their geometry. The successful application of the approach to both the recognition of dynamic head and hand gestures demonstrates the universal validity and sensitivity of our method. Even sequences containing only small detail changes are reliably recognized.

1. Introduction

Vision based gesture recognition has many applications in natural and intuitive human-machine-communication, telecommunications, and robotics. Since gestures are dynamic processes, they require methods for the classification of image *sequences*.

Statistical approaches using HMMs applied to image sequence modeling mainly differ in the way they extract features from the images. There are specific disadvantages: Property-based feature extraction methods [1, 2] do not make full use of the HMM modeling abilities, since they produce only one feature vector per image, and are not universal, since they require a-priori knowledge. Common low level feature extraction methods [3, 4] forming feature vectors out of vertical or horizontal image stripes result in an asymmetric spatial behaviour and have difficulties in image normalization.

Our new and efficient approach, in its first version presented in [5], overcomes the above difficulties using spatially symmetric low level features which represent only a small image area. For that reason the HMM is allowed to model all spatial and the temporal dimensions in a consistent way. The main idea is to choose features on the basis of intensity or gradient images that force an ordinary HMM training algorithm to represent similar successive images as so called image density functions (IDFs) in a respective HMM state. If edge images are used, an IDF indicates the existance and the average value of an edge point at a certain image position (see fig. 1).

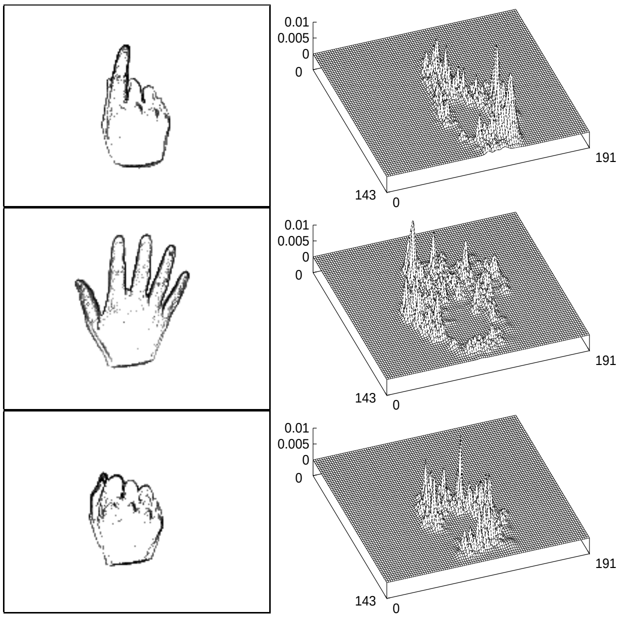

Figure 1: Typical edge images taken out of hand gesture sequences (see table 3) and corresponding image density functions after HMM training (see sec. 3)

The procedure is in detail explained in sec. 2. The setup of the experiments and the obtained results are described in sec. 3 and 4.

2. Description of the Algorithm

To generate a geometry preserving density representation of the image $f(\mathbf{n}) = f(n_1, n_2)$, the training algorithm of a HMM has basically to be provided with feature vectors consisting of two-dimensional image coordinates. The simplest feature set would theoretically contain all pixel coordinate vectors of all images of a sequence in a row. If the vectors in this set appear a number of times that is proportional to a corresponding pixel value $f(\mathbf{n})$[1], the training algorithm will approximate the desired IDFs [5].

Since in practice this plain feature set is too large to handle, the number of features has to be significantly reduced. To achieve this, a regular $K \times L$ grid of so called *image vectors* $\mathbf{v}_{ij}^{\text{init}}$ is positioned over the image. Up to two attributes are attached to every image vector. The first attribute n_{ij}^{init} contains the average intensity or the average magnitude of the gradient respectively. The second attribute d_{ij}^{init} contains the average orientation of the gradient if required. All mean values are calculated in the nearest neighborhood N_{ij} implicitly defined by an Euclidian distance measure D:

$$\begin{aligned} N_{ij} &= \{\mathbf{n} | \mathrm{D}[\mathbf{n}, \mathbf{v}_{ij}] < \mathrm{D}[\mathbf{n}, \mathbf{v}_{kl}] \\ &\quad \text{for all } k, l \text{ with } k \neq i \text{ and } l \neq j\}, \qquad (1) \\ n_{ij} &= \frac{1}{|N_{ij}|} \sum_{\mathbf{n} \in N_{ij}} f(\mathbf{n}), \qquad (2) \\ d_{ij} &= \frac{1}{|N_{ij}|} \sum_{\mathbf{n} \in N_{ij}} \delta(\mathbf{n}). \qquad (3) \end{aligned}$$

$|N_{ij}|$ is the number of pixels in the neighborhood of $\mathbf{v}_{ij}$.

To put more information in the individual *positions* of the image vectors, they have to be placed in an optimal way considering their attributes. This is done by a variation of the k-means algorithm [6]: Instead of clustering randomly positioned feature points, the task here is finding an optimal representation of the regularily positioned image pixels taking into account their randomly distributed intensities or gradients. The iteration step to calculate the optimal image vectors $\mathbf{v}_{ij}^{\text{opt}}$ results in the calculation of new image vectors at time $t+1$ as the specific centers of mass of the old neighborhoods $N_{ij}^{(t)}$ at time t. The average orientation attribute d_{ij} is only needed for gradient images together with the restricted summation areas introduced with eq. (10). The complete optimization algorithm is:

1. initialization:

$$\begin{aligned} \mathbf{v}_{ij}^{(0)} &= \mathbf{v}_{ij}^{\text{init}} \quad \text{implying} \qquad (4) \\ n_{ij}^{(0)} &= n_{ij}^{\text{init}} \quad \text{and} \qquad (5) \\ d_{ij}^{(0)} &= d_{ij}^{\text{init}} \quad \text{from eqs. (1)–(3);} \qquad (6) \end{aligned}$$

2. iteration:

$$\mathbf{v}_{ij}^{(t+1)} = \frac{1}{\sum_{\mathbf{n} \in N_{ij}^{(t)}} f(\mathbf{n})} \sum_{\mathbf{n} \in N_{ij}^{(t)}} \mathbf{n} \cdot f(\mathbf{n}) \qquad (7)$$

implying new $N_{ij}^{(t+1)}$ and $d_{ij}^{(t+1)}$ from eqs. (1) and (3);

3. if $\mathrm{D}\left[\mathbf{v}_{ij}^{(t+1)}, \mathbf{v}_{ij}^{(t)}\right] > \varepsilon$ for all i, j repeat step 2, else go to step 4;

4. the optimal vectors and intensity attribute are obtained at the last time step $t = T-1$:

$$\begin{aligned} \mathbf{v}_{ij}^{\text{opt}} &= \mathbf{v}_{ij}^{(T-1)} \quad \text{and} \qquad (8) \\ n_{ij}^{\text{opt}} &= n_{ij}^{(T-1)}. \qquad (9) \end{aligned}$$

In the case of gradient images the iteration converges faster if the orientation attribute d_{ij} is used to select only edge pixels $f(\mathbf{n})$ whose orientation $\delta(\mathbf{n})$ differs less than a threshold Δd from d_{ij}. Consequently the summation areas in eq. (7) change to

$$\mathbf{n} \in N_{ij}^{(t)} \text{ and } \left|\delta(\mathbf{n}) - d_{ij}^{(t)}\right| \leq \Delta d \qquad (10)$$

while those in eqs. (2) and (3) remain unchanged. The orientation angle difference is defined to be less or equal ±180 degrees. At the end of the iteration process the image vectors are concentrated near brighter image areas or are located on significant edges respectively.

The choice between intensity and edge images, the two kinds of image vectors $\mathbf{v}_{ij}^{\text{init}}$ and $\mathbf{v}_{ij}^{\text{opt}}$, and the possibility to let each vector appear once or in proportion to the respective average intensity attribute n_{ij}^{init} or n_{ij}^{opt} result in several possibilities to form different feature sequences. Experiments show that the best sequence can be built out of gradient images with the optimal vectors $\mathbf{v}_{ij}^{\text{opt}}$ in combination with a repetition in proportion to the attribute n_{ij}^{opt} (see also [5]).

The *images* become invariant to translation and rotation if the *vectors* are normalized using the moment-based centers of mass and orientation angles.

3. Experimental Setup

The used HMMs are semi-continuous since those models are a good compromise between few training data and accuracy of modeling [6]. Semi-continuous HMMs have a codebook of mixture density functions (or *prototypes*) calculated for the whole training data. The covariance matrices of the prototypes are diagonalized. Training and recognition are carried out by the Viterbi algorithm.

[1] $f(\mathbf{n})$ stands for the intensity image or the magnitude part of the gradient image (edge image); the orientation part of the gradient image is called $\delta(\mathbf{n})$.

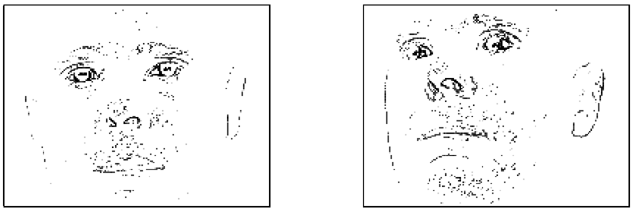

Figure 2: Two example images taken out of the head gesture sequences (s. table 1)

#	action/meaning	#	action/meaning
1	yes (normal)	6	maybe (start left)
2	yes (emphasized)	7	go to the rear
3	no (start right)	8	go to the right
4	no (start left)	9	go to the left
5	maybe (start right)		

Table 1: Head gesture catalog I

The classification was tested on sequences containing dynamic head and hand gestures respectively. The head was recorded from the front, the (right) hand from above. Head and hand were recorded in about full size (see figs. 1 and 2 for examples) against a uniformly colored background. The gestures are planned to be used for the visual control of a virtual object world. The following three different catalogs were used:

1. *Head gesture catalog I* containing 9 gestures that are mainly characterized by global position and orientation changes of the head (s. table 1).
2. *Head gesture catalog II* containing 5 gestures with internal changes exclusively in the area of the eyes (s. table 2).
3. *Hand gesture catalog* containing 12 gestures that are a combination of shape, position and orientation changes (s. table 3).

Each of the gestures was recorded 30 times; all gestures were performed from only a single person. Each image sequence contains 70 non-interlaced images at the European rate of 50 images (fields) per second. The final sizes of the images were 180×144 pixels for the head and 192×144 pixels for the hand images in YUV-mode.

#	action/meaning	#	action/meaning
1	look at right	4	blink left eye
2	look at left	5	blink both eyes
3	blink right eye		

Table 2: Head gesture catalog II

#	action/meaning	#	action/meaning
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 3: Hand gesture catalog

p	15×11	30×22	45×33	60×44
16	1.62	1.82	2.12	1.52
32	2.12	0.61	1.92	1.11
64	0.51	0.10	1.31	0.91
128	2.12	0.00	1.31	0.71

Table 4: Error rates (%) for head gesture catalog I (see table 1)

The edge images are the result of an absolute value calculation of a simple gradient operator applied to the Y-component of the unsegmented image with a subsequent threshold operation (see figs. 1 and 2 for examples). The gradient threshold for the hand gestures is uncritical, whereas the threshold for the head sequences is adjusted to a value so that only the eye and nostril areas remain visible (which is even less than in fig. 2). The segmented intensity images with a zero background were calculated with a color histogram based segmentation method out of the UV-components.

4. Experimental Results

20 of the sequences of each gesture were used for the training and the other 10 for recognition. The models used had different numbers of prototypes p and different grid resolutions of $K \times L$ initial image vectors. All the presented results are averaged over $s = 5$–15 states (the error rates stabilize between 5 and 15 states) and over all gestures of a catalog. Image vectors with a zero attribute n_{ij} were discarded.

The following tables show results for gradient images only since they produce the lowest error rates. But at least the gestures from the head catalog I and the hand catalog are classified with acceptable error rates if intensity images are used (less than 7% without and 0.3% with translational vector normalization).

Table 4 shows the results for head gesture catalog I. The error rates typically decrease for an increasing number of prototypes. At the optimal grid resolution of 30×22 vectors an error rate of zero can be reached.

p	15×11	30×22	45×33	60×44
2	46.36	50.55	23.64	21.09
4	16.73	27.27	0.91	0.36
8	27.82	34.18	8.00	5.27
16	12.00	25.27	2.91	4.55

Table 5: Error rates (%) for head gesture catalog II (see table 2)

p						
2	4	8	16	32	64	128
71.04	60.97	19.09	15.06	10.65	9.35	6.43

Table 6: Error rates (%) for combined head gesture catalogs I and II (see tables 1 and 2) with constant 60×44 grid

The results for gesture catalog II show a different behaviour (see table 5): As a rule, the error rates decrease for a higher grid resolution but are always optimal for $p = 4$ prototypes. Obviously 4 prototypes are a good compromise between modeling accuracy and amount of training data to cover the two crucial eye areas. Despite of the expected difficulties in distinguishing sequences with very small changing areas the lowest error rate is less than 0.4%. The normalization of the vectors has only small effects on the error rates of the head gestures. But it can be very helpful if larger offsets between different sequences are expected.

Although the respective error rates of both head gesture catalogs are very low, merging the two catalogs is difficult since they require different optimal parameter settings. For that reason the lowest achievable error rate of the combined head gesture catalogs is about 6.4% (see table 6).

The results for the hand gesture catalog in table 7 are already satisfactory if only a low grid resolution of 6×4 vectors is used. If additionally a translational normalization of the vectors is applied, the error rates decrease to zero. Evidently the coherent structure of the hand is easier to model than the distributed structures of the internal face features. More hand gesture recognition results with further parameter variations can be found in [5].

Table 8 shows the effect of the restricted summation areas expressed by eq. (10) applied to the hand gesture catalog. As Δd decreases the number of iterations (which are mainly responsible for the execution speed of the algorithm) decreases significantly. Obviously the execution speed can be tripled while the error rate of the normalized sequences does not change.

p	4	8	16	32	64
pure	11.52	0.76	1.14	0.76	0.98
normalized	13.79	0.45	0.00	0.00	0.00

Table 7: Error rates (%) for hand gesture catalog (see table 3, constant 6×4 grid)

Δd	180	21	14	7
rel. iterations	100.00	54.62	45.10	27.18
pure	0.98	1.89	1.89	3.71
normalized	0.00	0.00	0.00	0.00

Table 8: Error rates (%) for hand gesture catalog (see table 3, constant 6×4 grid, $p = 64$ prototypes, Δd in degrees, average number of iterations in % relative to $\Delta d = 180$ degrees)

5. Conclusion

A new method for the classification of image sequences using Hidden Markov Models was presented. Low level feature vectors that significantly reduce the amount of redundant image information and that allow the HMM a complete spatio-temporal modeling cause the generation of image density functions in the HMM states. The results obtained from different recognition tasks prove that the algorithm is universally applicable and very sensitive.

6. References

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

[2] 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.

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

[4] G. Rigoll, A. Kosmala, M. Schuster: *A New Approach to Video Sequence Recognition Based on Statistical Methods*. ICIP 1996, Lausanne, Vol. III, pp. 839-842, 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] X. D. Huang, Y. Ariki, M. A. Jack: *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.