FEATURE EXTRACTION METHODS FOR CONSISTENT SPATIO-TEMPORAL IMAGE SEQUENCE CLASSIFICATION 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 general and efficient approach for representing and classifying image sequences by Hidden Markov Models (HMMs) is presented. A consistent modeling of spatial and temporal information is achieved by extracting different low level image features. These implicitly convert the image intensities into probability density values, while preserving the geometry of the image. The resulting so called *image density functions* are contained in the states of the HMM. First results of applying the approach to the classification of dynamic hand gestures demonstrate the performance of the modeling.

1. INTRODUCTION

The first stage in preparing image sequences for Hidden Markov modeling is the transformation of the spatio-temporal image sequence into a pure time sequence. Two possible principles can be found in the literature:

(I) All the spatial information of the image is described by a *single* high level, property-based feature vector, thus transforming the image sequence into a sequence of feature vectors (e. g. in [1, 2]). This method is not universal, since it requires a-priori knowledge, and it uses the flexibility of the HMM only for the temporal processing.

(II) For each image, a feature vector sequence is generated by moving a vertical (horizontal) feature selection window along the image in horizontal (vertical) direction. The resulting low level feature vectors are lined up for the whole image sequence (e. g. in [3, 4]). That way, the HMM only has influence on the modeling of one spatial dimension resulting in asymmetric behaviour and problems in image normalization.

The new approach described in this paper eliminates the above diffculties using a symmetric spatial modeling based on low level features allowing simple measures for normalization of image position and orientation. The fundamental idea is to represent the two-dimensional intensity function of an image (or the average of a collection of similar images) with a two-dimensional probability density function — the *image density function* (IDF) — in a state of the HMM. In this geometry preserving representation, coordinate points of the image function correspond to those of the IDF, whereas the intensity values of the image are transformed to probability density values.

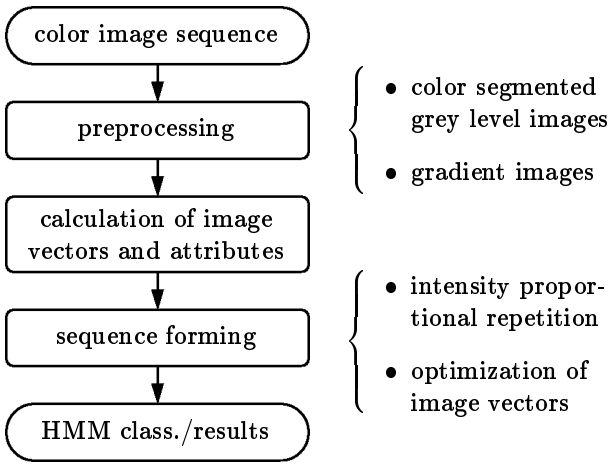

Figure 1: System overview

The overview in fig. 1 shows, that the system works with spatially segmented grey level images as well as unsegmented gradient images. The amount of data of these images has to be reduced significantly by representing the images with so called *(geometric) image vectors* and their *attributes* (see sec. 2). In the next step, two possibilities are used to form feature sequences out of these image vectors, so that the regular training algorithm of the HMM is forced to build internal IDF-representations of the images (see sec. 3).

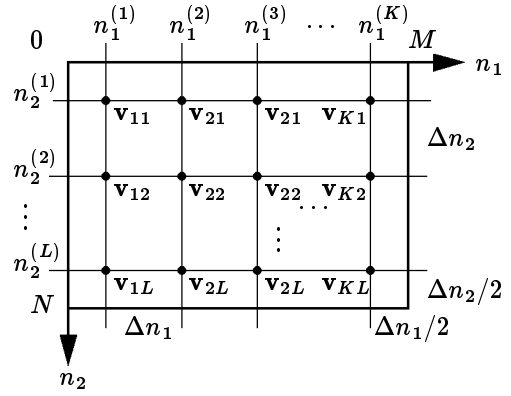

Figure 2: Initial placement of image vectors $\mathbf{v}_{ij}$

2. IMAGE VECTORS AND ATTRIBUTES

2.1. Initial placing

To reduce the amount of data, the discrete image function $f(n_1, n_2) = f(\mathbf{n})$ with the pixel dimensions $M \times N$ is represented by image vectors whose initial positions form a regular $K \times L$ grid:

$$\mathbf{v}_{ij}^{\text{init}} = \begin{bmatrix} n_1^{(i)} \\ n_2^{(j)} \end{bmatrix} = \begin{bmatrix} \Delta n_1 (i - 1/2) \\ \Delta n_2 (j - 1/2) \end{bmatrix}. \qquad (1)$$

The grid intervals are $\Delta n_1 = N/K$ and $\Delta n_2 = M/L$ (see fig. 2). An additional attribute value n_{ij} is attached to every image vector. The attribute n_{ij} contains the average intensity of the image in the so called *neighborhood* N_{ij} of the image vector, which is defined with the nearest neighbor rule using the Euclidian distance measure d:

$$N_{ij} = \{\mathbf{n} | d(\mathbf{n}, \mathbf{v}_{ij}) < d(\mathbf{n}, \mathbf{v}_{kl}) \text{ for all } k, l \text{ with } k \neq i \text{ and } l \neq j\}; \qquad (2)$$

$$n_{ij} = \frac{1}{|N_{ij}|} \sum_{\mathbf{n} \in N_{ij}} f(\mathbf{n}). \qquad (3)$$

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

2.2. Optimal placing

With the initial vector placement, the image information is contained in the attributes n_{ij} and in the grid intervals. To put more information in the individual *positions* of the vectors, they have to be placed in an optimal way considering their attributes. This can be done by concentrating the vectors iteratively in brighter image areas using a variation of the k-means algorithm [5]. 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. Using the initialization in eq. (1), the iteration results in the calculation of new image vectors at time $t + 1$ as the specific centers of mass of the old neighborhoods at time t. The complete optimization algorithm is:

1. initialization: $\mathbf{v}_{ij}^{(0)} = \mathbf{v}_{ij}^{\text{init}}$ from eq. (1) implying $n_{ij}^{(0)}$ from eqs. (2) and (3);
2. iteration:
$$\mathbf{v}_{ij}^{(t+1)} = \frac{1}{\sum_{\mathbf{n} \in N_{ij}^{(t)}} f(\mathbf{n})} \sum_{\mathbf{n} \in N_{ij}^{(t)}} \mathbf{n} \cdot f(\mathbf{n}) \qquad (4)$$
implying new $n_{ij}^{(t+1)}$ from eqs. (2) and (3);
3. if $d\left(\mathbf{v}_{ij}^{(t+1)}, \mathbf{v}_{ij}^{(t)}\right) > \epsilon$ for all i, j repeat step 2, else go to step 4;
4. optimal vectors and attributes at last time step $t = T - 1$:
$$\mathbf{v}_{ij}^{\text{opt}} = \mathbf{v}_{ij}^{(T-1)} \text{ and} \qquad (5)$$
$$n_{ij}^{\text{opt}} = n_{ij}^{(T-1)}. \qquad (6)$$

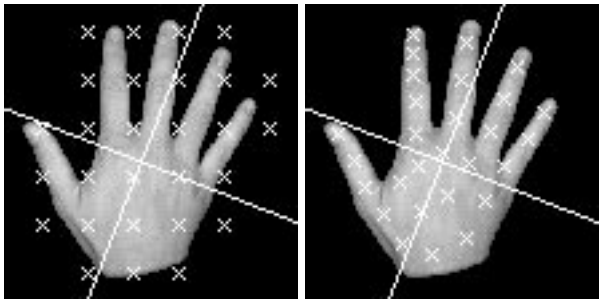

Figure 3: Initial and optimal image vectors with a non-zero attribute in a spatially segmented grey level image (11×8 grid)

Figs. 3 and 4 show the initially placed and optimized image vectors with a non-zero attribute for grey level and gradient images respectively.

2.3. Vector normalization

The *images* can be made translational and rotational invariant by normalizing the *vectors* using the moment based centers of mass and orientation angles. The resulting shifted coordinate systems are also shown in figs. 3 and 4. These normalization values have to be calculated only on the first image of a sequence. All the successive images are normalized relative to the first image.

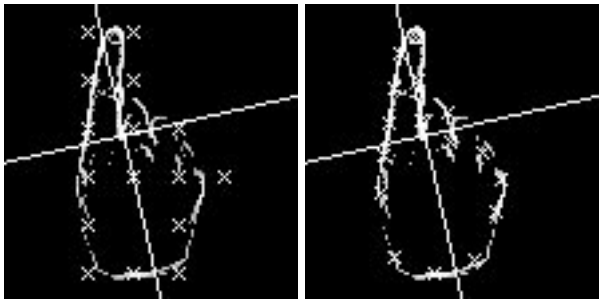

Figure 4: Initial and optimal image vectors with a non-zero attribute in a gradient image (11×8 grid)

3. SEQUENCE FORMING

An image is transformed into a feature sequence by sequentially emitting all image vectors with a non-zero attribute. That means these features are two-dimensional and contain the image coordinates of their location. The order of the feature emission within one image can be chosen randomly because the HMM should be able to collect all features of one or more *complete* images to build one probability image representation. The complete feature sequence is obtained by lining up the feature sequences of successive images in an image sequence. Two different strategies of sequence forming are applied which both force the HMM training algorithm to approximate IDFs. They are later compared with the so called *C-sequence* which contains each initial image vector just one time.

3.1. Repetition of initial image vectors

The so called *R-sequence* also consists of the initial image vectors. But to represent more of the image information, which is mainly contained in the vector attributes, each feature $\mathbf{v}_{ij}^{\text{init}}$ is repeated a number of times c_{ij} that is proportional to the value of the the attached attribute n_{ij}. The training algorithm of the HMM sees an emission at the location $\mathbf{v}_{ij}^{\text{init}}$ happen c_{ij} times, and since c_{ij} is proportional to the average intensity of the neighborhood N_{ij}, it will approximate the underlying image as an IDF. The number of repetitions is normalized so that the sum of feature emissions of each image is constant.

3.2. Usage of optimized image vectors

On the other hand, the so called *O-sequence* is constructed with the optimized image vectors $\mathbf{v}_{ij}^{\text{opt}}$ which appear one time each. Since the optimized vectors concentrate around bright image areas, the HMM training algorithm will collect more events around bright image areas and give them a higher density value.

#	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

4. HMM AND TEST DATA DESCRIPTION

HMMs have been successfully applied in many fields such as speech [6] and more recently in handwriting recognition [7][1]. The used HMMs are semi-continuous since those models are a good compromise between few training data and accuracy of modeling [6, 5]. 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.

The classification is tested on image sequences containing dynamic hand gestures. A catalogue of 12 different gestures was defined and is planned to be used as a vocabulary in a visual dialog with a virtual object world (see table 1). 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 a single image sequence. All data was recorded 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 size of the images was 192×144 pixels in YUV-mode.

A color histogram based segmentation method calculates the grey level images with a zero background value out of the UV-components. The edge images are obtained by applying a simple gradient operator to the Y-component of the unsegmented image (see figs. 3 and 4 for examples).

5. EXPERIMENTAL RESULTS

20 of the sequences of each gesture were used for the training and the other 10 for recognition. The models

[1] At this point the authors would like to thank their colleague H.-J. Winkler since large parts of the HMM source code used for this work could be derived from his HMMs for handwriting recognition.

p	C-seq.	O-seq.	R-seq.	OR-seq.	col.vecs.
8	41.14	9.02	14.55	18.71	31.59
16	16.89	5.23	11.97	8.64	14.17
32	19.24	6.59	16.82	6.89	8.18

Table 2: Error rates (%) for grey level images, different number of prototypes p, no normalization

	t/1	tr/1	t/a	tr/a
p	O-seq.	OR-seq.	O-seq.	OR-seq.
8	3.26	8.64	14.55	17.58
16	0.98	1.97	5.68	7.80
32	0.00	0.83	0.61	4.39

Table 3: Error rates (%) for grey level images, different number of prototypes p, best sequences for the respective normalizations (t=translational, r=rotational, 1=relative to first image, a=absolute for each image)

used had $p = 8$, 16 and 32 prototypes and $s = 2$–15 states. All the results presented here are average values over $s = 5$–15 states (the error rates stabilize with 5 states or more) and over all 12 gestures. The initial image vectors were placed on a 6×4 grid; vectors with a zero attribute were discarded.

Table 2 shows the results for the grey level images for differently formed feature sequences (see sec. 3) and different numbers of prototypes p without applying any normalization. As a rule, the error rate decreases for an increasing number of prototypes. The O- as well as the R-method perform significantly (up to 4 times) better than the plain C-sequence. The best is the O-sequence; the combination of the methods (OR-sequence) is even slightly worse. The last column shows, that using the attributes of the initial image vectors as 4-dimensional column vector features (resulting in method (II) explained in the introduction, see [3, 4]) performs only about half as well as our method.

If normalization methods are used (see table 3), the error rate for the underlying test data goes down to 0%. Even if only the change of pure shapes in the image sequence are used for classification (that means absolute translational and rotational normalization for *each* image of the sequence), the combined OR-method has a error rate of only 4.39%.

The results for gradient images show a similar tendency but the absolute errors are less compared to grey level images (see table 4). Here the combined OR-sequences mostly lead to the best results; the results without normalization are now 4 times better than those of the column vector method applied to gradient images.

	—	t/1	tr/1	t/a	tr/a	—
p	OR-seq.					col.vecs.
8	3.03	0.83	8.41	10.23	28.64	16.44
16	2.05	0.00	2.20	4.32	6.14	46.29
32	1.59	0.00	0.08	0.15	2.88	6.14

Table 4: Error rates (%) for gradient images, different number of prototypes p, OR-sequences, different normalizations (abbreviations in table 3)

6. CONCLUSION

The results show that the concept of image density functions as geometry preserving image representations, which allow HMMs a consistent spatio-temporal modeling, are superior to previous used methods. Above that, they allow straight forward measures for image normalization. Since only low level image features are required, the approach is applicable to a wide range of image sequence recognition tasks.

7. 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: *Rocognizing Human Action in Time-Sequential Images using Hidden Markov Model.* IEEE Comp. Vision and Pattern Recog. 1992, pp. 379–385, 1992.

[4] M. Schuster, G. Rigoll: *Fast Online Video Image Sequence Recognition with Statistical Methods.* ICASSP 1996, Atlanta, Vol. 6, pp. 3450–3453, 1996.

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

[6] 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.

[7] H-J. Winkler: *HMM-based Handwritten Symbol Recognition using On-line and Off-line Features.* ICASSP 1996, Atlanta, Vol. 6, pp. 3438–3441, 1996.