Feature Representations for the Recognition of 3D Emblematic Gestures

Jan Richarz and Gernot A. Fink

Intelligent Systems Group, Robotics Research Institute
TU Dortmund University, Dortmund, Germany
{jan.richarz,gernot.fink}@udo.edu

Abstract. In human-machine interaction, gestures play an important role as input modality for natural and intuitive interfaces. The class of gestures often called "emblems" is of special interest since they convey a well-defined meaning in an intuitive way. We present an approach for the visual recognition of 3D dynamic emblematic gestures in a smart room scenario using a HMM-based recognition framework. In particular, we assess the suitability of several feature representations calculated from a gesture trajectory in a detailed experimental evaluation on realistic data.

Key words: 3D dynamic gesture recognition, human-machine interaction, smart rooms, time-series analysis

1 Introduction

In building interfaces for Human-Machine-Interaction (HMI), different facets of natural inter-human interaction should be taken into account to realize intuitive interfaces. This includes the analysis of speech and gesture, as well as gaze, facial expression and body language. While some of these modalities may be very subtle and subject to considerable variations between users, speech and gestures are much more explicit. Thus, they have been studied extensively as important cues for interpreting user intents and realizing human-centered interfaces. In this publication, we focus on the automatic visual recognition of dynamic arm gestures. For natural interaction, there should be as few constraints as possible imposed on the user. In particular, users should be able to interact with the interface from anywhere in the environment, which requires view- and position-invariant recognition. To achieve this, we aim at recognising gestures in 3D space using a (potentially arbitrary) multi-camera setup.

Since the term gesture has been used in very different meanings (including fingertip motion and full-body actions), some clarification is needed. In linguistics and semiotics, a variety of gesture taxonomies exist (cf. eg. [1]). Generally, three major classes of gestures can be identified, with speech-accompanying subconscious gesticulation at one end of the spectrum, artificial well-defined sign languages at the other, and emblems in between. The first is inherently multimodal [2] and difficult to interpret due to its subconscious nature. Sign language typically lacks intuitiveness and requires special user training.

Emblems are gestural actions that are well-defined and convey a certain meaning on their own, but are understood intuitively since they are established within a certain cultural region. Therefore, they are especially suited for natural HMI. We focus on one-armed emblems performed by cooperative users.

Dynamic arm gestures are defined by subsequent movements of a few prominent points (e.g. joint positions) relative to the body. Thus, given a spatio-temporal track of these points, recognition is a problem of time-series or trajectory analysis. Results from other work on gesture analysis (cf. Sec. 2) suggest that, for emblems, this problem reduces to analysis of the hand trajectory. Indeed, measurements like in [3] indicate that, for simple stroke-like arm movements the trajectories of the joints and hand are qualitatively similar. Furthermore, analyzing typical emblematic gestures shows that they tend to be composed of a relatively small set of basic movements. This suggests strong similarities to the field of on-line handwriting recognition, where the track of one point (the pen tip) is recognized based on basic units (characters or strokes) and some features describing their general spatio-temporal evolution (cf. e.g. [4]).

Accordingly, we exploit findings from this field and investigate whether 3D emblematic arm gestures can be recognized using approaches inspired by on-line handwriting recognition. Since the latter is a 2D problem, the concepts must either be transferred to 3D, or the 3D gesture trajectory has to be projected to some appropriate 2D frame. We will investigate both possibilities in the following. In particular, we propose representing a gesture by projection on its principal plane of motion, which we call the action plane. For the acquisition of gesture trajectories, we build upon our previous work on 3D pointing gesture recognition [5] and saliency-based view selection in multi-camera setups [6].

2 Related Work

The relevance of gestures for natural HMI – either as exclusive cue or as part of multi-modal systems – is undisputed. However, most work in the field focuses either on the recognition of specially crafted artificial gesture alphabets and sign language [7, 8] or on the interpretation of full-body movements, generally referred to as action recognition (cf. e.g. [9] for a recent survey, [10, 11]). While the shortcomings of artificial gesture alphabets regarding their intuitiveness have already been mentioned, full-body action recognition is related closely to emblematic gesture analysis, but typically operates on a higher level of abstraction: Instead of creating an input modality for HMI, it rather aims at analysing human behavior in surveillance settings, or for scene understanding. Approaches from the field may, however, also be suitable for gestural interfaces.

Regarding the classification of emblematic dynamic gestures, the dominant approach is to represent gestures as trajectories in some reference frame and classify them with probabilistic graphical models encoding temporal relationships. In particular, (Hidden) Markov Models ((H)MM) have been used extensively. Good results have been achieved on gestures representing arabic digits [12] using only trajectory orientation information. In [13], bimanual movements are classified by

combining the trajectory with a shape descriptor of the hand, whereas [14] use the centroid positions of hand candidates and their mean optical flow. [15] transform the spatiotemporal trajectory to discrete symbols with a Self-Organizing Map, and classify the symbol sequence together with optical flow features in a MM framework. Combinations of 2D hand trajectories and associated inertial sensor data have also been used [16, 17]. Gaussian density features extracted at visual interest points are applied in [18], and gestures are classified using a protocol learning strategy.

In on-line handwriting recognition (cf. [4] for an overview of the field), state of the art recognizers are typically either also based on HMM [19] or on connectionist approaches [20]. However, the features used to describe time-series of points are much more diverse. Examples include velocity and curvature along with shape-describing features of short trajectory segments [20] or Hu moments [21]. [22] and [19] use pen pressure, vicinity, curliness and features relating the trajectory to the baseline. Appearance-based descriptors and higher-level structural features, like ascenders, descenders and crossings, are also frequently combined with online trajectory features (e.g. in [20][19]). Some of these features lack a straighforward resemblance for the task of gesture recognition. E.g., pen pressure is not available, and features referring to a baseline (like ascenders and descenders) are difficult to apply, since, opposed to handwriting, it is not clear what the baseline of a gesture should be. However, a multitude of interesting features for trajectory representation remain, and impressive results have been published in the field. To the best of our knowledge, no previous work exists applying similar features to 3D dynamic arm gesture recognition, and we will demonstrate their suitability in this work.

3 Visual Recognition of 3D Emblematic Gestures

As stated before, our goal is the automatic recognition of one-armed dynamic emblems performed by cooperative, but untrained users. Restricting the interaction space to a predefined area or camera setup, as well as requiring the user to wear markers or tracking gear, would impose severe limitations on the general applicability of such a system. Furthermore, the pose or orientation of the user with respect to the interface should not be restricted. Therefore, we aim at a 3D recognition framework based on visual cues utilizing off-the-shelf cameras in a principally arbitrary multicamera setup. Figure 1 shows an overview of the proposed approach. We will describe the individual components in the following.

The key assumption is that emblematic arm gestures may be analysed using the trajectory of the active hand alone, which means that no expensive full-body model tracking is required. While this seems like a strong assumption, its validity is indicated by the good results reported in the related literature. The first step is the extraction of 2D spatiotemporal hand and head trajectories in the individual camera images. These are then combined to a 3D trajectory. Also, the estimation of the action plane from the trajectory points and the representation of projected trajectories is shown. The main contribution lies in the assessment of different

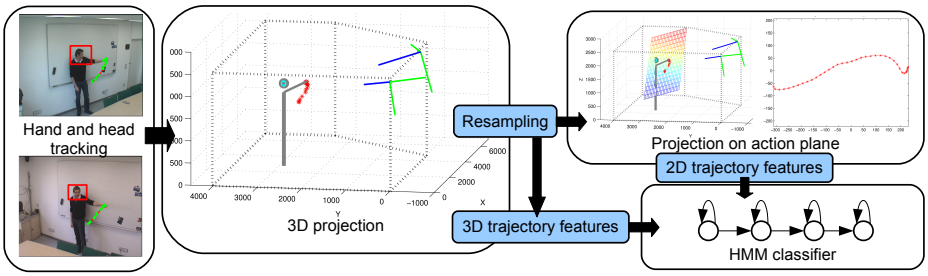

Fig. 1. Overview of the proposed approach.

alternate feature representations inspired by on-line handwriting recognition, which will be done in a detailed experimental evaluation on a realistic data set. We conclude with a discussion of the results.

3.1 2D Trajectory Acquisition

In order to extract gesture trajectories, persons and their gesturing hands have to be detected first. We apply a detector based on Histograms of oriented Gradients (HoG) and a Multi-Layer Perceptron classifier which does not rely on skin color or face structure, but uses the shape of the head-shoulder line instead. Therefore, it is able to detect persons under a large variety of poses and viewing angles. The centers of the detection rectangle in subsequent frames form the head trajectory. Hand candidates are found combining motion detection with a personalized skin color model trained on-line [5]. The result is a series of spatial image coordinates for head and hands, along with temporal information. These are postprocessed using Gaussian smoothing to eliminate detection jitter, and short tracks of duplicate points are removed.

3.2 3D Combination

Given spatiotemporal trajectories from at least two cameras, the original 3D gesture trajectory can be reconstructed. First, the individual trajectories have to be aligned. We use a simple greedy aggregation algorithm taking into account temporal differences and reconstruction errors of pairs of data points. For the somewhat idealised data we use here (cf. Sec. 5.1), this is sufficient. Note that, in a multi-camera setting, a view selection algorithm can be applied choosing the two "best" views according to some global criteria [6]. Aligned trajectory points are then projected to 3D by ray casting.

It should be pointed out that our cameras are not synchronised. Therefore, and because of detection inaccuracies and the discretization of the image plane, the rays will generally be skewed. Thus, the projection is calculated as follows: Given two aligned points $\mathbf{p}_i^t = (p_{xi}^t, p_{yi}^t)$, $\mathbf{q}_j^t = (p_{xj}^t, p_{yj}^t)$ from cameras i and j at time t (we omit these indices in the following for readability), their corresponding

3D rays $\mathbf{r}_p = \mathbf{c}_i + \gamma\mathbf{p}'$ and $\mathbf{r}_q$ are obtained by backprojection using the camera calibration matrices. Here, $\mathbf{c}_i$ is the projection center of camera i, $\mathbf{p}'$ is the ray's directional vector, and similar for $\mathbf{r}_q$. The two directional vectors along with one of the projection centers define a plane $\mathcal{P} : \mathbf{n}^T\mathbf{x} - \mathbf{n}^T\mathbf{c}_i = 0$ with $\mathbf{n} = \mathbf{p}' \times \mathbf{q}'$. It contains one of the rays and is parallel to the other with distance d. Let $\mathbf{v}$ be the intersection point calculated after translating both rays into the plane. The reconstructed 3D point $\mathbf{u}$ is then given by linear interpolation

$$\mathbf{u} = (1 - \frac{\alpha_i}{\alpha_i + \alpha_j}) \cdot \mathbf{v} + (1 - \frac{\alpha_j}{\alpha_i + \alpha_j}) \cdot (\mathbf{v} + d\mathbf{n}) \tag{1}$$

where α_i and α_j are some confidence measures for the point positions in the individual images. Setting them to equal values yields the mean point along the direction of $\mathbf{n}$ where the two rays are closest. Candidate selection or rejection can be done based on d. The resulting 3D trajectory finally is resampled and smoothed using curvature-aware impulse resampling [22].

3.3 The Action Plane

Classifying gesture trajectories without seriously limiting the amount of allowed variation according to, e.g., viewpoint, gesturing speed or spatial expansion, cannot be performed reliably on raw spatial coordinates. Normalization to some common reference and abstraction from the absolute positions is necessary. A simple approach to achieve this are derivative features that do not encode the absolute values of trajectory points, but their consecutive changes. However, these may still depend on the external alignment of the trajectory in 3D space.

Another possibility we investigate here arises from our observation that, for most natural emblems, the 3D trajectory exhibits an inherent planar characteristic. This suggests that 3D emblem trajectories may be represented without too much loss of information by projecting them on an appropriate plane. A similar assumption has been made in [16] to compensate for camera pan and tilt. Opposed to them, however, in our setup the plane may be oriented arbitrarily in space, and will only rarely coincide with any of the image planes. We call this concept the action plane in the following, and show how an estimation of such a plane can be derived and used as a common reference for normalization.

Suppose we have a 3D trajectory $\mathbf{T} = [\mathbf{t}_1 ... \mathbf{t}_n]$ with n points $\mathbf{t}_i = (t_{xi}, t_{yi}, t_{zi})$. We seek a plane $\mathcal{P} : \mathbf{n}^T\mathbf{x} - \lambda = 0$ that best approximates $\mathbf{T}$. This can be formulated as a least-squares regression problem with the objective function

$$f(\mathbf{n}) = \sum_{i=1}^{n} (n_x t_{xi} + n_y t_{yi} + n_z t_{zi} - \lambda)^2 \rightarrow Min \tag{2}$$

assuming that $\mathbf{n}$ is normalized to unit length. This is a well-known problem, and the sought plane normal $\mathbf{n}$ is given by the Eigenvector corresponding to the smallest Eigenvalue of $\mathbf{\Psi} = \mathbf{M}^T\mathbf{M}$, $\quad \mathbf{M} = \{t_{x,i} - \bar{t}_x \quad t_{y,i} - \bar{t}_y \quad t_{z,i} - \bar{t}_z\}$, with the data mean $\bar{\mathbf{t}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{t}_i$.

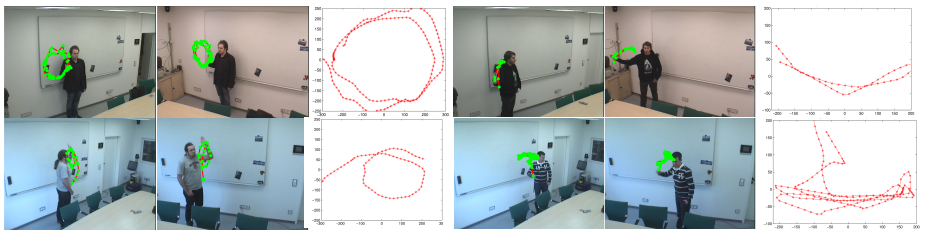

Fig. 2. Examples of action plane projections. Original images with overlaid hand trajectories and their 2D representations for "circle" (left) and "horizontal wave" (right).

Calculating a regression plane in this way may result in a solution strongly influenced by outlier points. Therefore, the above procedure is carried out on the concensus set obtained from RANSAC. Since $\mathbf{n}$ and $-\mathbf{n}$ correspond to the same global orientation of the plane and the sign depends on the choice of points, $\mathbf{n}$ is forced to always point towards the mean of head detections. In our experiments, this yields a good estimate of a gesture's principal plane of motion. When the above procedure is applied incrementally to online data, a smoothness constraint should be applied to the orientation of $\mathbf{n}$ to avoid abrupt changes. One possibility is adding a penalty term taking into account the angle between two consecutive plane normals in the model selection phase of RANSAC.

Projecting $\mathbf{T}$ onto $\mathcal{P}$ requires an 2D orthonormal coordinate system in $\mathcal{P}$. An obvious choice are the remaining two Eigenvectors of $\mathbf{M}$. This results also in a normalization of the global gesture orientation, which may not always be intended. Therefore, we also evaluate a solution where one coordinate axis is forced to be parallel to the ground plane. The trajectory mean $\bar{\mathbf{t}}$ is chosen as coordinate origin. Figure 2 shows some projection results.

3.4 Classification with Hidden Markov Models

HMMs are a popular tool for time-series analysis because of their ability to model temporal relationships between samples in a sound probabilistic framework and provide an integrated approach for segmentation and classification. Their properties are well understood and efficient algorithms exist for training and decoding. Therefore, they have been widely used, and their discriminative power has been demonstrated on a wide variety of tasks.

We use an open source HMM toolbox [23] to train one model for each gesture. The number of states in each model is initialised automatically according to the minimum observation length of the respective gesture class. Emission probabilities are modelled by Gaussian mixture densities with diagonal covariances. The resulting codebook is shared among models and states, i.e. we have semi-continuous HMMs. Classification is done according to the maximum path probability calculated by Viterbi alignment.

4 Trajectory Features

The features used for representing gesture trajectories are mostly motivated by [20]. Some changes are made to adapt them to the different characteristics of the data. In order to increase robustness against noise and detection errors, the features are calculated in a sliding window scheme. Let w be the window size, and let $\mathbf{O}_i = \mathbf{o}_i, ..., \mathbf{o}_{i+w-1}$ be the trajectory points in the ith sliding window and $\mathbf{o}_i^m$ the median point. Then, the features are calculated as follows:

Raw trajectory: Mean point of the window: $\bar{\mathbf{O}}_i = \frac{1}{w} \sum_k \mathbf{o}_k, k = i...i+w-1$

Normalized trajectory: $\hat{\mathbf{O}}_i = \bar{\mathbf{O}}_i / \bar{h}$, where $\bar{h}$ is the average height of the person calculated from the trajectory of head positions.

Normalized polar trajectory: $\mathbf{P}_i = \{|\mathbf{r}_i|, \phi_i\}$. For the 3D case, $\mathbf{r}_i = (\bar{\mathbf{O}}_i - \bar{\mathbf{H}}_i)/\bar{h}$, $\phi_i = \arctan(\sqrt{r_{xi}^2 + r_{yi}^2}/r_{zi})$, i.e. the radius between mean trajectory and mean head point inside the window normalized by the person's height and the elevation angle of their connecting line. Note that the azimuth angle would correspond to the global orientation of the person, so it is not included. For 2D, $|\mathbf{r}_i|$ and ϕ_i are polar coordinates in the plane relative to the coordinate origin.

Velocity: The mean velocity of data points in the window, i.e. $\mathbf{v_i} = \frac{1}{w} \sum_{k=i+1}^{i+w-1} (\mathbf{o}_k - \mathbf{o}_{k-1})/(t_k - t_{k-1})$ where t_k is the time associated with $\mathbf{o}_k$. The mean length of the velocity vectors is also included.

Curvature: Curvature is defined as the cosine and sine of the angle between the vectors from $\mathbf{o}_i^m$ to $\mathbf{o}_i$ and $\mathbf{o}_{i+w-1}$.

Vicinity: These features are intended to describe the general shape of a feature window. Let $\mathbf{d_i} = \mathbf{o}_{i+w-1} - \mathbf{o}_i$ be the vector connecting the window boundaries. The vicinity features comprise the vicinity aspect $\alpha = (d_{yi} - d_{xi})/(d_{yi} + d_{xi})$ for 2D data and three values with permutations of the vector components for 3D data, the cosine and sine of the angle between $\mathbf{d}_i$ and the x-axis or ground plane, respectively, the normalized trajectory length $l_i = \sum_{k=i+1}^{i+w-1} |\mathbf{o}_k - \mathbf{o}_{k-1}|/|\mathbf{d_i}|$ and the average sum of squared distances between trajectory points and $\mathbf{d_i}$.

Orientation change: For two subsequent windows $\mathbf{O}_i$ and $\mathbf{O}_j$, the orientation change is calculated as the cosine and sine of the angle between $\mathbf{d_i}$ and $\mathbf{d_j}$.

Head distance: The mean distance between points in $\mathbf{O}_i$ and the mean head position $\bar{\mathbf{H}}_i$, normalized by $\bar{h}$. This feature encodes some very weak representation of the spatial relation between gesturing hand and head.

All features together yield 20 and 25-dimensional feature vectors for 2D and 3D points, respectively, and twice the size including derivative features.

5 Experiments

The main goal of this publication is the investigation of alternative features regarding their applicability to gesture trajectory recognition, in order to derive a richer representation and optimize recognition results. To this purpose, we have conducted a detailed experimental evaluation on a realistic dataset.

Fig. 3. Selected (cropped) examples of the gesture set with original trajectories overlaid. Gestures marked with (R) are repetitive, (r) indicates a gesture that may or may not be repetitive. From left to right: "circle"(r), "come here"(r), "down", "go away"(r), "pointing", "stop", "up", "horizontal wave"(R) and "vertical wave"(R).

5.1 Experimental Setup and Data

The evaluation took place in a realistic setup inside a smart conference room equipped with several Sony EVI D70P pan-tilt-zoom cameras. The cameras are mounted on the ceiling and are calibrated, but not synchronized. Throughout the experiments, the same pair of cameras was used. In neutral position, their principal axes form an angle of approximately 90°, but their orientations were changed several times during data recording.

A set of nine emblematic command gestures was chosen such that they either represent natural gestures that are commonly used, or their meaning can be understood intuitively. Some examples are shown in Fig. 3. The potential meanings they convey can be used in a variety of scenarios, e.g. directing a mobile robot, steering computational attention, or controlling services of the smart room. The set contains short one-stroke as well as more complicated repetitive gestures, and gestures that can be both. Note that the pointing gesture is not purely emblematic, since it can only be interpreted with additional context.

Short sequences of still images were recorded from 17 different people each performing one to three instances of each gesture with their right as well as their left arm. The sequences were captured with a resolution of 378 by 278 pixels at 20 Hz. No instructions on gesture speed, absolute or relative position, etc., were given. The subjects were allowed to move freely inside the cameras' fields of view, including their orientation with respect to the cameras. Thus, the dataset is quite challenging since it contains multiple viewpoints as well as considerable variations in gesture appearance, speed and trajectory diameter. In total, it contains 51217 images and 799 gesture instances.

The positions of hands and heads were annotated semi-automatically. First, the 2D head and hand detection algorithm was applied. The generated hypotheses were then inspected manually. Missing detections were added and erroneous hypotheses were corrected. In general, if a hypothesis from the detector was remotely correct, it was kept. The gesture instances were furthermore segmented manually, with considerable variations in starting and end points as well as number of repetitions. From the 2D trajectories obtained in this way, the 3D projections were computed using the described algorithm. The resampled trajectory lengths vary between 16 and 364 data points.

Table 1. Classification accuracy in % for single features (left) and respective derivative (Δ) features (right). The best results for each trajectory type are highlighted.

Feature	3D	2D-Ground	2D-PC	Δ3D	Δ2D-Ground	Δ2D-PC
Raw trajectory	59.3	65.0	59.4	**85.1**	62.7	**61.1**
Norm. cart. trajectory	80.0	**65.8**	57.1	44.2	21.3	23.7
Norm. polar trajectory	81.9	61.8	58.8	73.7	**66.5**	60.6
Curvature	40.3	40.3	39.8	48.6	48.2	47.1
Headdistance	61.3	61.3	**61.3**	27.0	27.0	24.2
Orientation change	44.6	46.7	45.6	48.6	48.2	49.9
Velocity	**83.0**	61.5	59.2	76.8	55.9	50.9
Vicinity	74.7	59.7	61.2	71.1	53.2	57.7

5.2 Results

First, each feature type is evaluated separately in order to assess the performance of individual features. Two types of HMM topologies (Linear left-right and Bakis) with different parameter sets were trained and evaluated using 17-fold cross-validation. In each iteration, 16 persons were used for training, and the remaining one for testing. Thus, the reported results are user-independent. For feature extraction, sliding window sizes of 5, 7 and 9 were applied, with 50% window overlap. Table 1 summarizes the best results for 3D trajectory features and 2D projection features using both described coordinate system choices (first coordinate axis parallel to ground plane, denoted as "Ground", and first axis chosen according to first principal component, denoted as "PC").

The best classification results in all cases were achieved using a Bakis model and a window size of five. For the 3D case, using the derivative of the raw trajectory yields a classification accuracy of 85.1%. Compared to this, the 2D features perform poorly, with best results of 66.5% and 61.3%, respectively. This may indicate that our assumption about the inherent planar nature of 3D emblematic gestures is invalid. On the other hand, this assumption is backed up by the low average reconstruction error of the projection (Tab. 2). Thus, the performance loss is more likely to be caused by the loss of positional information in relation to the body as a result of the projection and normalization. This is further indicated by the fact that choosing the 2D coordinate system according to the principal components of the 2D trajectory, thereby normalizing out the trajectory's global orientation, further degrades performance. In this case, the only feature type that encodes some weak relative positional information, the head distance, performs best.

For the remaining two trajectory types, the best results are achieved with derivative representations of the hand trajectory. This, on the one hand, confirms that classifying gestures based on their trajectory alone is indeed a suitable approach, on the other hand it shows that some abstraction from the raw trajectory is needed. As mentioned before, using derivatives of the trajectory is a very simple possibility of abstracting from the absolute spatial positions. Using

Table 2. Average reconstruction errors for projected 2D gesture trajectories.

Gesture	RError (mm)	Gesture	RError (mm)	Gesture	RError (mm)
circle	18.8	comehere	15.2	down	15.0
goaway	13.7	pointing	9.6	stop	9.1
up	9.3	hor. wave	22.3	ver. wave	28.6

Table 3. Classification accuracy of feature combinations in %. Feature combinations are: FC1: Δ Raw traj. + velocity + vicinity + headdist; FC2: Δ Raw traj. + vel.; FC3: Δ Raw traj. + vic.; FC4: Δ Norm. polar traj. + vel. + vic.; FC5: Δ Norm. polar traj. + vel.; FC6: Δ Norm. polar traj. + vic.; FC7: vel. + vic.

Features	FC1	FC2	FC3	FC4	FC5	FC6	FC7
3D	82.4	84.4	82.2	83.1	**85.5**	80.4	81.0
2D Ground	64.0	**65.6**	63.0	64.8	**65.6**	62.7	65.0
2D 1st PC	63.3	64.2	62.5	63.7	**65.0**	62.7	63.2

the raw trajectory results in a severe performance loss (59.3% classification accuracy) for the 3D case, while in the 2D case, where the "raw" trajectory is already normalized due to the projection, the results are close to the best. Considering the alternative feature representations, the velocity profile and vicinity features also yield promising results on our data, while orientation change and curvature seem less suited for the task.

Following these findings, combinations of the best-performing trajectory representations with velocity and vicinity features were evaluated, along with the head distance, which seems to be beneficial in the 2D case. The results are summarized in Tab. 3. No improvement in classification accuracy could be achieved, and the results of most combinations are comparable. This suggests that the different feature types are highly correlated. Furthermore, the increased dimensionality of the features leads to a higher complexity of the model, and much more data is needed to accurately estimate the parameters, which may be detrimental to the classifier's performance. Indeed, opposed to the previous experiment, the best results were achieved with Linear models and bigger window sizes (7 for 2D Ground, 9 for the others), which corresponds to simpler models with less states.

This raises the question whether better performance can be achieved by decorrelating the features. In order to assess this, a third experiment was carried out. After normalizing the features to zero mean and unit variance in order to account for the different feature dynamics, Principal Component Analysis (PCA) was applied to the complete feature representation (all feature types + derivatives) of the data, and classifiers were trained using different numbers of Principal Components. Table 4 summarizes the results.

Using PCA features indeed resulted in a substantial improvement in classification accuracy. The best result was again achieved using the first 10 PC of

Table 4. Classification accuracy for PCA features in %.

No. of PC	1	2	3	4	5	6	7	8	9	10	12	15
3D	60.3	74.5	79.5	83.4	85.4	86.6	88.7	88.0	89.7	**90.4**	89.5	88.6
2D	48.7	63.1	69.3	71.5	72.1	73.2	73.1	76.0	76.5	76.7	76.3	**78.1**
2D+3D	48.1	65.8	76.3	83.0	83.6	83.7	84.4	86.6	86.0	85.6	85.4	**87.0**

3D features, which yielded 90.4% correct classifications, a relative improvement of 6.2%. The 2D features still perform substantially worse, with 78.1% accuracy (17.4% relative improvement). However, these findings clearly indicate that emblematic gesture recognition can benefit from the incorporation of alternative feature representations. Surprisingly, combining 2D and 3D features yielded worse results compared to 3D only. A possible reason for this is the high dimensionality (90) of the combined feature space. The amount of available data may not be sufficient for estimating reliable statistics.

6 Summary

We presented an approach to hand-trajectory based 3D emblematic arm gesture recognition for Human-Machine Interaction in a smart room. In particular, we evaluated several alternative feature representations inspired by approaches in on-line handwriting recognition, and demonstrated their suitability for the task in a detailed experimental evaluation on realistic data. It could be shown that the incorporation of the additional features indeed improved the recognition results, and very promising overall results were achieved. The experiments were conducted with offline data, but all presented concepts and algorithms can be applied incrementally to online data in a straightforward way. We plan to extend the recognition approach to a hierarchic system building on strokes or subgesture units, aiming for a more powerful and flexible recognizer.

Furthermore, we suggested that natural emblematic gestures have an inherent planar nature, and proposed representing them by projection on an estimate of this inherent plane. While the recognition results for the projected data were inferior in our experiments, the estimated plane might serve as a cue for inferring the addressee of a gesture – a question we will investigate in our future research.

References

1. Kendon, A.: Current Issues in the Study of Gestures. In: The Biological Foundation of Gestures. Motor and Semiotic Aspects. Lawrence Erlbaum Assoc. (1986) 23–47
2. Eisenstein, J., Davis, R.: Visual and linguistic information in gesture classification. In: Proc. Int. Conf. on Multimodal Interfaces. (2004) 113–120
3. Atkeson, C.G., Hollerbach, J.M.: Kinematic features of unrestrained vertical arm movements. Journal of Neuroscience **5** (1985) 2318–2330

4. Plamondon, R., Srihari, S.N.: On-line and off-line handwriting recognition: A comprehensive survey. Trans. Pattern Analyis and Mach. Int. **22** (2000) 63–84
5. Richarz, J., Plötz, T., Fink, G.A.: Real-time detection and interpretation of 3d deictic gestures for interaction with an intelligent environment. In: Proc. Int. Conf. on Pattern Recognition. (2008)
6. Schauerte, B., et al.: Multi-modal and multi-camera attention in smart environments. In: Proc. Int. Conf. on Multimodal Interfaces and Workshop on Machine Learning for Multi-Modal Interaction. (2009)
7. Ong, S., Ranganath, S.: Automatic sign language analysis: A survey and the future beyond lexical meaning. Trans. Pattern Analysis and Mach. Int. **27** (2005) 873–891
8. Wang, Q., et al.: Viewpoint invariant sign language recognition. Computer Vision and Image Understanding **108** (2007) 87–97
9. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: A survey. IEEE Trans. on Circuits and Systems for Video Technology **18** (2008) 1473–1488
10. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still images. In: Proc. Int. Conf. on Computer Vision and Pattern Recog. (2008)
11. Rapantzikos, K., Avrithis, Y., Kollias, S.: Dense saliency-based spatiotemporal feature points for action recognition. In: Proc. Int. Conf. on Computer Vision and Pattern Recognition. (2009) 1454–1461
12. Elmezain, M., et al.: A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In: Proc. Int. Conf. on Pattern Recog. (2008)
13. Shamaie, A., Sutherland, A.: Bayesian fusion of hidden markov models for understanding bimanual movements. In: Proc. Int. Conf. on Automatic Face and Gesture Recognition. (2004)
14. Alon, J., et al.: A unified framework for gesture recognition and spatiotemporal gesture segmentation. Trans. Pattern Analysis and Mach. Int. **31** (2009) 1685–1699
15. Caridakis, G., et al.: SOMM: Self organizing markov map for gesture recognition. Pattern Recognition Letters **31** (2010) 52–59
16. Rett, J., Dias, J.: Gesture recognition using a marionette model and dynamic bayesian networks (DBNs). In: Proc. Int. Conf. on Image Analysis and Recognition. (2006) 69–80
17. Calinon, S., Billard, A.: Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. In: Proc. Int. Conf. on Machine Learning. (2005) 105–112
18. Kirishima, T., Sato, K., Chihara, K.: Real-time gesture recognition by learning and selective control of visual interest points. IEEE Trans. Pattern Analyis and Mach. Int. **27** (2005) 351–364
19. Schenk, J., Kaiser, M., Rigoll, G.: Selecting features in on-line handwritten whiteboard note recognition: SFS or SFFS? In: Proc. Int. Conf. on Document Analysis and Recognition. (2009) 1251–1254
20. Graves, A., et al.: A novel connectionist system for unconstrained handwriting recognition. Trans. Pattern Analyis and Mach. Int. **31** (2009) 855–868
21. Daifallah, K., Zarka, N., Jamous, H.: Recognition-based segmentation algorithm for on-line arabic handwriting. In: Proc. Int. Conf. on Document Analysis and Recognition. (2009) 886–890
22. Fink, G.A., Wienecke, M., Sagerer, G.: Video-based on-line handwriting recognition. In: Proc. Int. Conf. on Document Analysis and Recognition. (2001) 226–230
23. Fink, G.A., Plötz, T.: Developing pattern recognition systems based on markov models: The ESMERALDA framework. Pattern Recognition and Image Analysis **18** (2008) 207–215