Multi-user posture and gesture classification for 'subject-in-the-loop' applications

Giso GRIMM[1,2] and **Joanna LUBERADZKA**[2] and **Volker HOHMANN**[1,2]
[1] HörTech gGmbH, Marie-Curie-Str. 2, D-26129 Oldenburg, Germany
[2] Research group "digital hearing devices",
Department of Medical Physics and Acoustics,
Medizinische Physik und Cluster of Excellence Hearing4all
g.grimm@uni-oldenburg.de

Abstract

This study describes a posture classification method for a marker-free depth camera. The method consists of an object identification procedure, feature extraction, and a naïve Bayesian classification approach with a supervised training. Point clouds obtained from the depth camera are split into objects. For each object a set of features is extracted. A method of feature pre-processing is proposed and compared against a statistical orthogonalisation method. Using a manually labelled training data set, the probability distributions for the Bayesian classification are obtained. As a result of the classification, the most likely gesture is assigned to each object in real time. Classification performance was tested on a separate data set and reached about 80%.

Three different applications are described: Automatic estimation of user postures to estimate the influence of hearing devices on user behaviour in communication situations, the control of an interactive audio-visual art installation, and interactive light control on a dance-floor setup with multiple dancers. Classification performance in these applications was measured and discussed.

Keywords

gesture classification, behaviour analysis, hearing devices, interactive art, subject-in-the-loop

1 Introduction

With the development of assistive technologies, there is a growing need for robust automatic identification of human postures and gestures. Gesture recognition is used for improving the human-machine communication, e.g., in hand gesture-based device control [Freeman and Weissman, 1997; Richarz et al., 2008]. Another use case is the classification of gestures and postures that describe the subject's behaviour or provide information on the current state of the subject [Busso et al., 2008; Melo et al., 2015]. Automatic recognition of various postures has potential applications in research areas where the test subject's behaviour is analysed. As an example from the hearing research, in typical communication situations, leaning forward while listening is associated with a high listening effort, whereas sitting more relaxed indicates a lower effort [Paluch et al., 2015]. Manual labelling of user behaviour in similar tasks is usually time consuming and is not sufficient in case of the 'subject-in-the-loop' experiments, where the measurement is controlled by the responses of the test subject. Interaction between the subject reactions and the measurement procedure is desired when aiming at more realistic experimental conditions, but can also provide additional performance measures from the experimental feedback loop. 'Subject-in-the-loop' experiments require a real-time classification of gestures and postures. This differs from conventional behavioural experiments where a post-hoc analysis of the data is possible. Besides research applications, machine control functions based on natural postures are possible, e.g., a hearing device could increase the noise reduction efficiency when the user's change in posture indicates a higher listening effort. Such an application would require a body-worn motion tracking sensor, e.g., accelerometer and gyroscope embedded into a hearing device.

Gesture and posture recognition tools are also applied in music and arts [Ciglar, 2008; Donnarumma, 2011]. Typically, an artist controls music generation and modification tools with gestures, resulting in a mixture of dance and music performance. The classification system proposed here is designed to be useful for music and art applications with multiple users. One application is an audio-visual installation, where the postures of the audience influence the sound and vision. Another potential real-time application is the live interactive light and music control system for a dance-floor.

Real-time analysis of postures and gestures from depth images is commonly achieved via skeleton modelling [Shotton et al., 2013]. In the

applications of this study, such a high level posture model is not required, because only a limited number of posture and gesture classes need to be discriminated. Furthermore, these applications require a computationally fast method of classification. For this, a naïve Bayesian classifier as used in this study. This simple classification method can deal with a low-dimensional data and requires only a limited amount of training data [Ashari et al., 2013; Gupte et al., 2014]. For discriminating only a small set of classes, low level features describing the coarse point cloud distributions and the velocities of certain point cloud areas can be used. However, to fulfil the implicit statistical assumptions of the naïve Bayesian classifier, and to identify the most relevant application-specific feature sets, a pre-processing of features may be beneficial.

In this paper, methods of point cloud processing (sections 2.1 to 2.3), feature pre-processing (section 2.4) and classification (section 2.5) are described. In section 2.6, the training conditions in three different applications – posture classification for hearing research, multi-user control of an audio-visual art installation, and individualised light control for a dance-floor – are explained. Classification performance in the different applications with the proposed pre-processing methods are given in section 3 and discussed in section 4.

2 Methods and apparatus

For this study, one or more subjects were tracked using a Microsoft kinect depth camera. Although the final applications of this gesture and posture classification approach significantly differ, they have all the same structure, which is depicted in Fig. 1. First, the camera data was filtered for a more robust point cloud estimation and background removal. In a second step, the point cloud was split into multiple objects. For each object, a set of features was extracted, and based on this feature set, the posture or gesture of each object was classified. The point cloud processing and classification was implemented in the openMHA hearing device signal processing platform [Herzke et al., 2017; Grimm et al., 2009; Grimm et al., 2006]. Training and data analysis was implemented in Matlab. These processing blocks are described below.

2.1 Noise reduction and background removal

The Microsoft kinect depth camera is an optical sensor which measures the depth through the

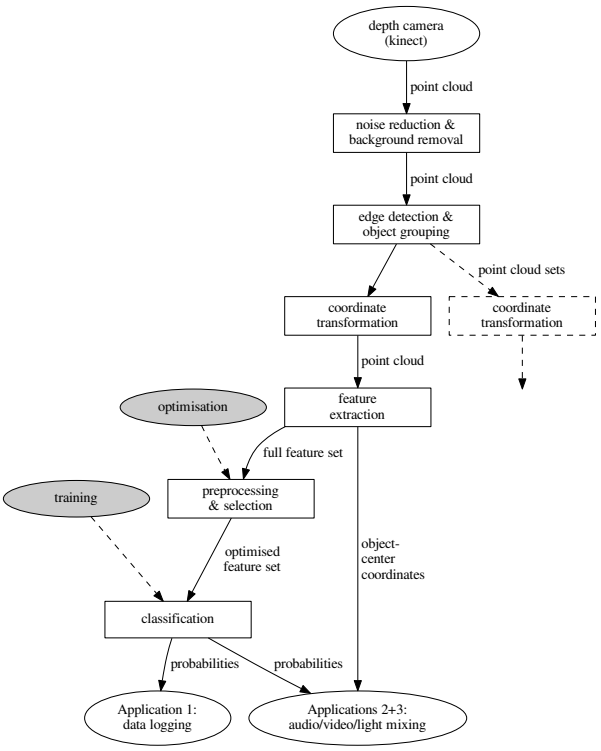

Figure 1: Structure of the proposed gesture and posture classification framework.

parallax of an infrared laser grid. It provides a depth value d for each pixel position (k, l). Invalid values (e.g., occlusion, absorption) get the depth value $d = 0$. In this study, the depth was scanned with a frame rate of 10 Hz.

Absorbing objects and objects with a very uneven surface, e.g., curly hair, typically result in invalid data points for many frames. To increase robustness in such conditions, invalid values were replaced by the last available valid value, if a value was measured within the last second.

For the classification of objects it was essential to separate them from the background. Therefore, in an initial phase without subjects in the sensing area, the background depth was measured, and all depth values close to the background were removed. After this step, only those data points remain which were assumed to belong to a relevant object.

2.2 Edge detection and object grouping

An assumption for the object grouping was that all objects have a spatial separation, i.e., either the depth was not continuous or the objects were separated by background pixels. This allows to use a simple first-order gradient edge detection algorithm using the depth data. A

pixel (k, l) was an object boundary if the depth gradient was above a threshold d_t:

$$(d_{k,l} - d_{k+1,l})^2 + (d_{k,l} - d_{k,l+1})^2 > d_t^2 \quad (1)$$

To construct objects, a generic flood fill algorithm [Torbert, 2012] was applied to identify all pixels within a closed boundary. These pixels were marked on an object map with their object number. The set of pixels (k, l) belonging to one object was $\mathbb{P}$, which was then used for further object-specific processing if the number of elements of $\mathbb{P}$, p, has a sufficient size.

Coordinate transformation. At this stage the objects were defined by a set of pixels with a certain depth from the camera. For a robust feature extraction, these have to be transformed into a world coordinate system. In the first step, pixel data was transformed into a camera coordinate system $\mathbf{x_c} = (x_c, y_c, d)^T$, with the horizontal distance from the camera axis x_c, the vertical distance y_c and the distance from the camera d. These coordinates were linearly approximated by

$$(x_c, y_c) = \alpha(k - k_0, l - l_0)d_{k,l}. \quad (2)$$

(k_0, l_0) was the central pixel of the camera. World-coordinates $\mathbf{x} = (x, y, z)^T$ (x distance along camera axis, y to the left, z upwards) were calculated by rotation and translation of the camera-coordinates. These point clouds $\mathbb{P}$ were the basis of further feature extraction of each object. The object centre was $\overline{\mathbf{x}} = \langle \mathbf{x} \rangle_{\mathbb{P}}$, i.e., the mean of all points in the point cloud $\mathbb{P}$.

Temporal alignment of objects. At this point, the order of detected objects depends on the first object pixel position in the camera plane. This is not a robust measure, thus the object order may change from frame to frame. However, to allow for analysis of time related features, the objects were re-ordered based on a similarity measure of distance d and the object size ratio r between consecutive frames. The distance between the objects o and q at the time indices t and $t - 1$ was defined as $d_{o,q}(t) = ||\overline{\mathbf{x}}_o(t) - \overline{\mathbf{x}}_q(t-1)||$. The size ratio was $r_{o,q}(t) = e^{|\ln(p_o(t)) - \ln(p_q(t-1))|}$. Then the coherence matrix $\mathbf{C}(t)$ between two objects was defined by its elements

$$c_{o,q}(t) = r_{o,q}(t) e^{-\gamma d_{o,q}(t)} \quad (3)$$

with a weighting coefficient $\gamma = 10$. For a re-sorting of objects, the columns of $\mathbf{C}$ were ordered to maximise the elements on the diagonal, corresponding to a maximal temporal coherence.

2.3 Feature extraction

A list of all extracted features and their labels can be found in Table 1. Features corresponding to the object in the global coordinate system as well as features describing size and distribution of the point cloud $\mathbb{P}$ relative to its centre were extracted. The object rotation was estimated from the ratio of depth to width. Two methods of calculating point cloud distribution were tested: In the first method, weighted averages across $\mathbb{P}$ were calculated. For example, the average left bottom position was estimated by using a weight w with

$$w = \begin{cases} (z - z_{max})^2 + (y - \langle y \rangle)^2 & y > \langle y \rangle \\ 0 & \text{otherwise} \end{cases}. \quad (4)$$

To account for dynamic properties, which may be important for gesture classification, in addition to the above mentioned point cloud distribution related features, the absolute value of their temporal derivatives was calculated.

These features define the time-variant feature vector $\mathbf{f}(t)$ which was used as an input of feature pre-processing.

2.4 Feature pre-processing and optimization

Before the actual classification, the features $\mathbf{f}$ were pre-processed with a method $\mathcal{P}$ to maximise the classification performance,

$$\hat{\mathbf{f}}(t) = \mathcal{P}\{\mathbf{f}(t)\}. \quad (5)$$

The pre-processing method $\mathcal{P}$ was a combination of temporal low-pass filtering with the time constant τ, selection of optimal feature set $\mathbb{F}$, and PCA.

The pre-processing method $\mathcal{P}$ was iteratively optimised. In each iteration cycle m, the training of the classifier was done based on the pre-processed training data set, whereas the classification performance to which this pre-processing method $\mathcal{P}_m$ led, was computed using the test data set, pre-processed in the same way as the training data. The pre-processing method $\mathcal{P}_m$, which gave the best classification performance was chosen as the final pre-processing method for classification.

Orthogonalisation. The naïve Bayesian classifier used in the current work assumes

name	label
global coordinates:	
number of pixels p	`n.n`
mean position $\mathbf{x}$	`n.x`, `n.y`, `n.z`
median position	`n.xmed`, `n.ymed`, `n.zmed`
rotation	`n.rot`
local coordinates:	
size	`n.sx`, `n.sy`, `n.sz`
thickness	`n.r`
segment positions	`o.lx` `o.lz`, `o.rx`, `o.rz`, `o.lby`, `o.rby`
segment thickness	`n.r1`, `n.r2`, `n.r3`
z-quantiles	`n.z25`, `n.z50`, `n.z75`
velocities:	
object velocity	`o.vz`
size changes	`o.vsy`, `o.vsz`, `o.vsx`
vertical segment velocities	`o.vlz`, `o.vlz`, `n.vz1`, `n.vz2`, `n.vz3`
horizontal segment velocities	`n.vxy1`, `n.vxy2`, `n.vxy3`
angular velocity	`n.vrot`

Table 1: List of features per identified object. The features were calculated by two different implementations, as indicated by the prefix `o` and `n`.

conditional independence of all the features. This means, that adding features which are highly correlated with other features might degrade the performance of the classification. Therefore, an orthogonalisation of the feature space is required. In this study, two orthogonalisation methods were tested.

A principle component analysis (PCA) is a generic orthogonalisation method. A transformation matrix is estimated, which is then applied to the feature vector before classification. To avoid a dominance of large-scale features, all features were scaled to ensure a standard deviation of one before calculating the PCA coefficients.

As an alternative method, a feature selection method is proposed. First, the individual classification performance of each feature from the full feature set was computed, by training the classifier only on the given feature. Classification performance was then measured on the test data set. The features were then sorted by their individual classification performance. Starting with the best performing feature, features from the sorted feature set were added successively to the optimal feature set. This procedure was repeated until no further increase of classification performance was observed. Although this feature set is optimised for classification performance, it does not guarantee that it is orthogonal in a statistical sense.

Low-pass filtering. Low-pass filtering of the features across time results in a smaller feature variance within a class and thus a better class separation, which as a consequence leads to a better classification performance. On the other hand, with long time constants the classifier is not able to track transitions between the classes. The time constant τ can be adapted to the expected frequency of class transitions in the test data, or to increase classification performance and stability. The optimal τ was determined by a one-dimensional grid search, with and without PCA and feature selection.

2.5 Classification

To accomplish the gesture classification task, a Gaussian Naïve Bayesian Classifier was implemented. This approach assumes a set of conditionally independent and normally distributed features. Each class c_h, where $h = 1, ..., N_c$ is the class index, and N_c is the total number of classes, represented a different gesture or posture. $\hat{\mathbf{f}}$ is a data vector with extracted features $\hat{f}_j$, where $j = 1, ..., N_{\hat{f}}$ is the feature index, and $N_{\hat{f}}$ is the number of features. Considering the independence assumption, Bayes formula can be written in the following form:

$$p(c_h|\hat{\mathbf{f}}) = \frac{p(\hat{\mathbf{f}}|c_h)p(c_h)}{p(\hat{\mathbf{f}})} = \frac{\prod_{j=1}^{N_{\hat{f}}} p(\hat{f}_j|c_h)p(c_h)}{p(\hat{\mathbf{f}})}, \tag{6}$$

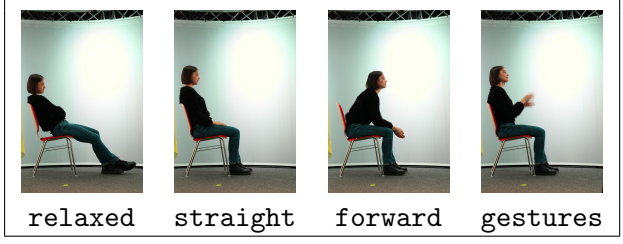

Figure 2: Labels of the project 1 (research application).

which means that the overall class conditional probability $p(\hat{\mathbf{f}}|c_h)$ can be computed by multiplying the conditional probabilities for each feature $p(\hat{f}_j|c_h)$.

Since the elements of $\hat{\mathbf{f}}$ are assumed to be normally distributed $p(\hat{f}_j|c_h) = N(\mu_{jh}, \sigma_{jh})$, the probability density function (PDF) of a feature j for a class h can be modelled by the mean μ_{jh} and standard deviation σ_{jh}. These parameters were estimated from the manually labelled training data. Also a flat prior probability was assumed, $p(c_h) = 1/N_c$.

In the current study, probabilities $p(c_h|\hat{\mathbf{f}}(t))$ were calculated for each object in each time frame. For estimating the classification performance, the confusion matrix was computed as an average posterior probability for each class. The classification performance was the geometric average across the diagonal of the confusion matrix.

2.6 Classification tasks and class labels.

The training was executed for three different classification tasks, corresponding to the use cases in hearing research, art and entertainment. In each training data set, data from nine test subjects (age from 23 to 44 years) was used. The recording of each gesture or posture lasted approximately 90 seconds for each subject.

In the first task ('project 1'), four classes with typical communication states were defined with an indention to track the subject's behaviour during the hearing experiment. There were three sitting postures with labels `relaxed`, `straight`, `forward`, and a class corresponding to gesticulation while talking, `gestures`.

The second task ('project 2') consisted of eight classes, either body movements or postures, which were used for controlling and mixing of sound and video art installation concerning different manifestations of water. The 'water' classes had the following labels: labels

Figure 3: Labels of the project 2 (audio-visual art installation).

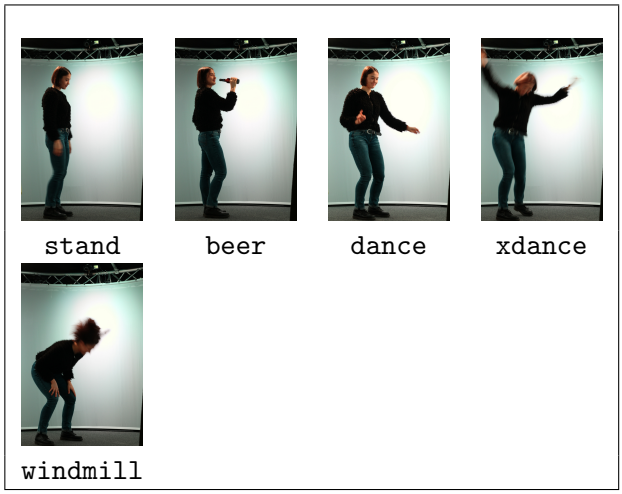

Figure 4: Labels of the project 3 (dance-floor light control).

`lake`, `rain`, `ice`, `waves`, `ocean`, `boil`, `steam` and `thunder`.

The third classification task ('project 3') contained five classes related to typical actions on a dance-floor at parties, to control the light according to individual behaviour of the dancers. The labels `stand` (standing or slowly walking), `beer` (drinking from a bottle), `dance` (dancing), `xdance` (excessive dancing) and `windmill` (rotating head) were used.

Images from Figures 2, 3 and 4 present the selection of classes for each project.

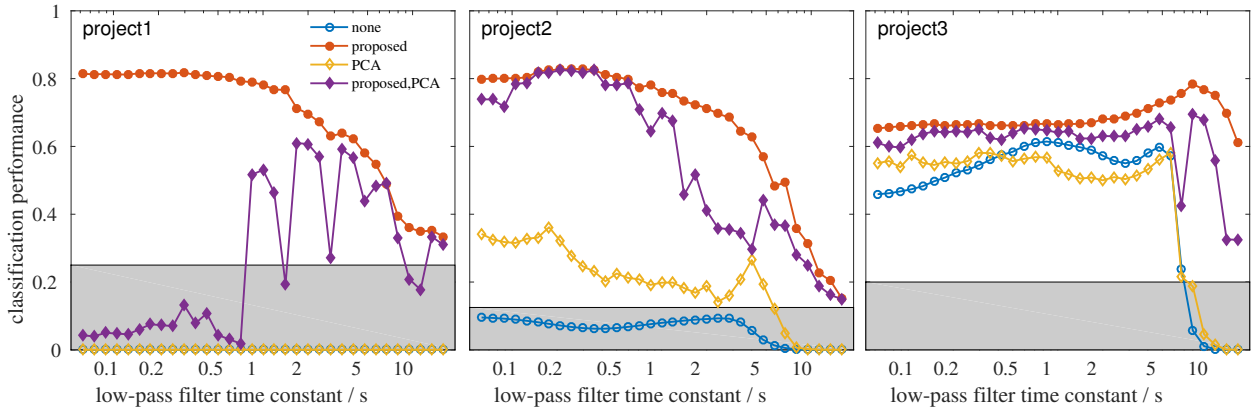

Figure 5: Classification performance as a function of feature low-pass filter time constant τ for the orthogonalisation methods 'none', 'proposed', 'PCA' and the combination of 'proposed' with 'PCA', in the three tested projects. The shaded area denotes the chance level.

3 Results

3.1 Influence of feature pre-processing on classification performance

Time constant optimisation. Figure 5 shows the classification performance as a function of feature low-pass filter time constant τ in all tested projects. The optimal value for project 1 was 297 ms, resulting in a classification performance of 81.8%. In project 2, the optimal time constant was 250 ms with a performance of 82.9%. In the third project, the time constant τ was 8 s, leading to a classification performance of 78.4%.

In all cases, the feature orthogonalisation improved the performance. The maximum performance was always achieved with the proposed method for feature selection. Using the PCA alone increased the performance only marginally. Both methods in combination do not give better performance results than the proposed method alone.

Feature selection. Figure 6 shows the performance of individual features in the three different projects. In project 1, the proposed feature selection method reduced the dimensionality to 12 features. The performance of individual features ranged from 19.1% to 44.4%. 42.7% of the selected features were velocity-related features. In project 2, a set of 17 features was found to be optimal; individual performance ranged from 13% to 28.4%. 35.3% of the features were velocity-related. In the last project, only 9 features were sufficient for optimal classification, with individual performance between 26.3% and 48.2%. In this case, 66.7% of the features were related to motion.

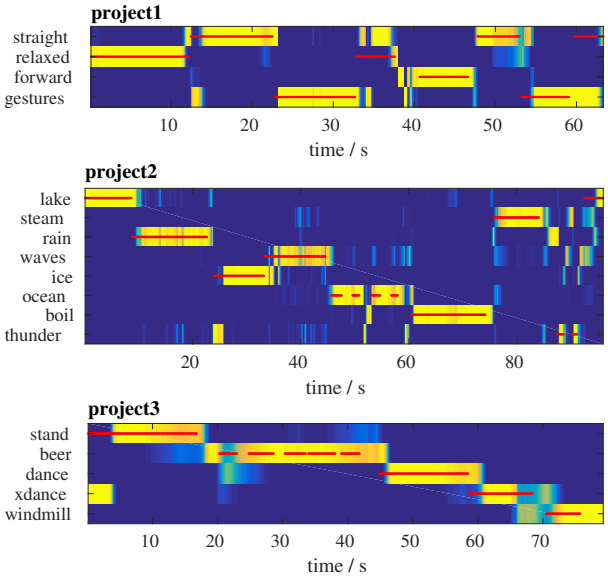

Figure 7: Posterior class probability as a function of time for the test data, with the labelled classes indicated by red lines.

3.2 Classification performance with optimised parameter sets

Figure 7 shows the posterior probabilities as a function of time. It can be noticed that classification errors mostly occurred at class transitions. With the longer time constants of project 3, a lag of classification at each transition can be seen.

The confusion matrix is shown in Figure 8. In project 1, the least confusions were achieved for the `forward` class. Typical confusions were between the classes `straight` and `relaxed`, as well as between `gestures` and `straight`. In project 2, the least confusions were found for

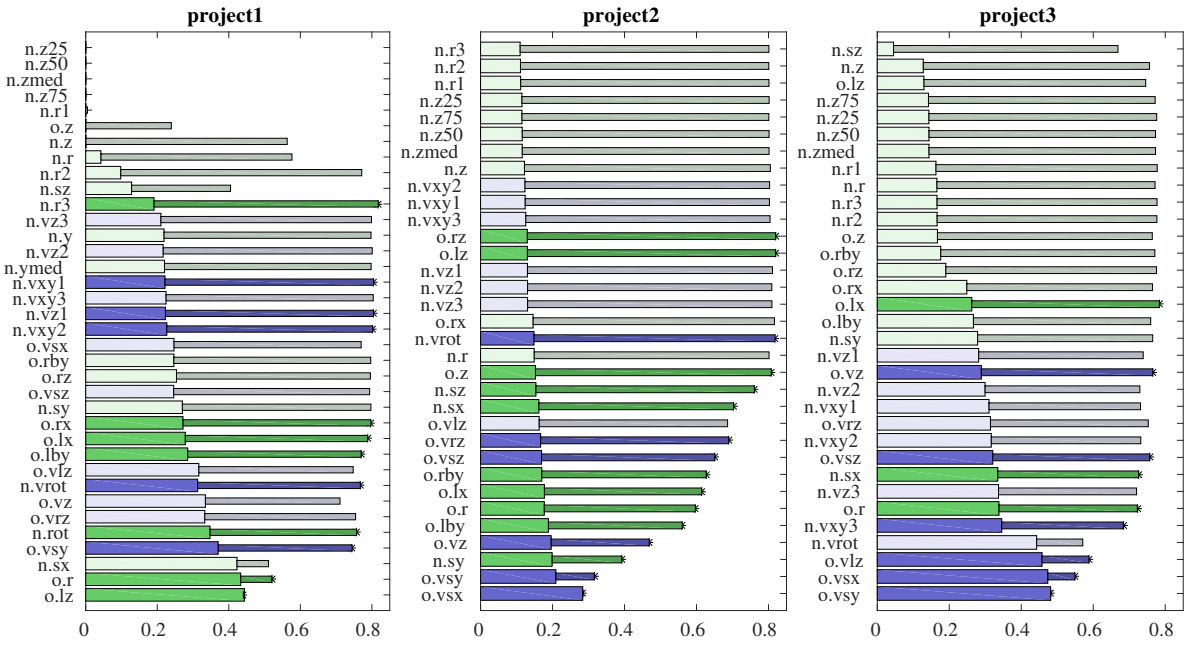

Figure 6: Classification performance of single features (thick bars) and the cumulative classification performance (thin bars). Stars denote the features which were selected by the proposed orthogonalisation method. Blue colours denote velocity-related features.

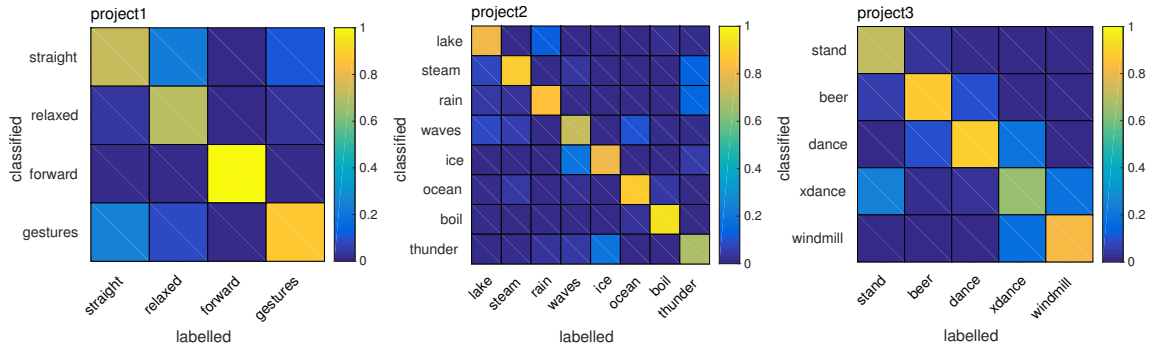

Figure 8: Confusion matrices (average posterior probability for each class in the test data set) of the three different projects.

the classes `boil`, `ocean` and `steam`. The class `thunder` was often confused with the classes `rain` or `steam`. In the third project, more confusions can be noticed. Most confusions can be found for the classes `xdance` and `windmill`, and between the classes `beer` and `dance`.

4 Discussion

The results show that a robust classification of gestures and postures based on a low-level feature set is possible, even with a naïve Bayesian classifier and a small feature space. The preprocessing of features indicated that a orthogonalisation of the feature space in a statistical sense is less important than the selection of features with an optimal class separation. However, it is still unclear whether another order of combination of orthogonalisation methods or a dimension-reduction in the PCA would further increase performance.

In this study, only number of low-level features was used. A high-level feature space, e.g., skeleton modelling, might be beneficial for robust classification of complex and high-level gestures. On the other hand, using such low-level features does not require any model assumptions. An intermediate solution could be an advanced segmentation of the point cloud.

5 Conclusions

In this study it was shown that even with a small and low-level point-cloud based feature space a robust classification of gestures and postures is possible. The tested applications covered research, art and entertainment, with four to eight classes in each application. The proposed method of feature-space optimisation by selecting a subset of the features was shown to result in better classification performance than a statistical orthogonalisation method. Low-pass filtering of features with application-specific time constants allowed for a trade-off between stable classification and fast reactions at class transitions. Classification performance of approximately 80% was achieved in all applications. Automatic classification of gestures and postures for hearing research applications with the 'subject-in-the-loop', i.e., with a behavioural feedback loop, seems feasible.

6 Acknowledgements

This study was funded by DFG research grant 1732 "Individualisierte Hörakustik" and by "klangpol Netzwerk Neue Musik Nordwest". We would like to thank all test subjects who participated in the training phase.

References

Ahmad Ashari, Iman Paryudi, and A Min Tjoa. 2013. Performance comparison between naïve bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool. *Int. J. Adv. Comput. Sci. Appl*, 4(11).

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.

Miha Ciglar. 2008. " 3rd. pole"-a composition performed via gestural cues. In *NIME*, pages 203–206.

Marco Donnarumma. 2011. Xth sense: a study of muscle sounds for an experimental paradigm of musical performance. In *ICMC*.

William T Freeman and Craig D Weissman. 1997. Hand gesture machine control system, January 14. US Patent 5,594,469.

Giso Grimm, Tobias Herzke, Daniel Berg, and Volker Hohmann. 2006. The Master Hearing Aid – a PC-based platform for algorithm development and evaluation. *Acta Acustica united with Acustica*, 92:618–628.

Giso Grimm, Tobias Herzke, and Volker Hohmann. 2009. Application of linux audio in hearing aid research. In Frank Neumann, editor, *Proceedings of the Linux Audio Conference*, pages 61–66, Parma, Italy. Istituzione Casa della Musica.

Amit Gupte, Sourabh Joshi, Pratik Gadgul, Akshay Kadam, and A Gupte. 2014. Comparative study of classification algorithms used in sentiment analysis. *International Journal of Computer Science and Information Technologies*, 5(5):6261–6264.

Tobias Herzke, Hendrik Kayser, Frasher Loshaj, Giso Grimm, and Volker Hohmann. 2017. Open signal processing platform for hearing aid research (openMHA). In *Proceedings of the Linux Audio Conference*.

Renato de Souza Melo, Andrea Lemos, Carla Fabiana da Silva Toscano Macky, Maria Cristina Falcão Raposo, and Karla Mônica Ferraz. 2015. Postural control assessment in students with normal hearing and sensorineural hearing loss. *Brazilian journal of otorhinolaryngology*, 81(4):431–438.

Richard Paluch, Matthias Latzel, and Markus Meis. 2015. A new tool for subjective assessment of hearing aid performance: Analyses of interpersonal communication. In *Proceedings of the International Symposium on Auditory and Audiological Research*, volume 5, pages 453–460.

Jan Richarz, Thomas Plotz, and Gernot A Fink. 2008. Real-time detection and interpretation of 3d deictic gestures for interaction-with an intelligent environment. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.

Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. 2013. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124.

Shane Torbert. 2012. *Applied computer science*. Springer.