Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

# Linear-Projection-Based Classification of Human Postures in Time-of-Flight Data

Folker Wientapper, Katrin Ahrens, Harald Wuest, Ulrich Bockholt
Department for Virtual and Augmented Reality
Fraunhofer IGD
Darmstadt, Germany
{folker.wientapper, katrin.ahrens, harald.wuest, ulrich.bockholt}@igd.fraunhofer.de

***Abstract*—This paper presents a simple yet effective approach for classification of human postures by using a time-of-flight camera. We investigate and adopt linear projection techniques such as Locality Preserving Projections (LPP), Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA), which are more widespread in face recognition and other pattern recognition tasks. We analyze the relations between LPP and LDA and show experimentally that using LPP in a supervised manner effectively yields very similar results as LDA, implying that LPP may be regarded as a generalization of LDA. Features for offline training and online classification are created by adopting common image processing techniques such as background-subtraction and blob detection to the time-of-flight data.**

***Index Terms*—Human Pose Estimation, Linear Projection, Locality Preserving Projections (LPP), Linear Discriminant Analysis (LDA), PCA, Machine Learning, Classification, Time-of-Flight Camera, Ambient Assisted Living (AAL).**

## I. Introduction

Human pose estimation has been an active research area during the last decade with the aim of monitoring human behavior or supporting new human computer interaction paradigms. As one example, in Ambient Intelligence and Ambient Assisted Living (AAL) environments there is a need for capturing user related context information in order to enable the system to intelligently react to the specific needs of the user. In particular, Ambient Assisted Living is concerned with supporting elderly people in their daily activities in order to provide more comfort and security and to enable them to live more independently. An unsolved problem is the detection of risky situations, e.g. when an elderly person has been fallen. In such cases relatives or care givers should be alarmed by the system in order to rapidly provide help. Regarding the detection of those situations, cameras installed at home together with intelligent algorithms provide a non-invasive solution as the persons do not have to be equipped with sensors on their body. Another application is the monitoring of activities over time in order to detect *e.g.* unusual behavior patterns. Postures of persons evolving dynamically as a low level source of information may be joined with other sources to give important clues about the activities being performed.

In this paper we are interested in the problem of classifying postures according to a finite set of discrete states such as ’Sitting’, ’Standing’, ’Bending’, etc. In contrast to the main research on human pose estimation (see [1] and [2] for in-depth reviews), where postures are expressed in a parameterized fashion by joint angles of limbs and the general body orientation, *no continuous* pose is estimated. Moreover, by classifying postures directly according to semantic states, we more adequately support ontological reasoning mechanisms often found on higher levels in AAL-architectures. In [3] we describe a possible solution for embedding a human posture classification component within an AAL system for activity monitoring.

Compared to related work in the field of discrete posture classification for domestic services [4], [5], [6], our approach is characterized by its simplicity and effectiveness. Firstly, we do not require multiple conventional cameras in order to resolve ambiguities arising from self-occlusion of the body parts or to handle difficulties caused by poor segmentation. Instead, we advocate the use of time-of-flight cameras which produce a three-dimensional, geometric representation of the scene. This reveals better segmentation results and provides additional information for ambiguity handling. Secondly, we do not use a particular model of the human body in order to estimate a complete, parameterized body configuration before classification. In fact, we apply only a few processing steps to the time-of-flight data to obtain low resolution feature images which, in turn, are used for training and online classification. Regarding the latter we adopt linear projection based machine learning approaches, which are particularly well known in the field of face recognition [7], [8]. One of these techniques, Locality Preserving Projections (LPP) [9], has already been adopted in the context of human monitoring by Wang and Suter [10], [11]. However their approach differs from ours in that they use LPP to obtain a parameterization of the image manifold and, in turn, use the dynamic evolution of the projected feature vectors to classify time-dependent activities.

The remainder of this paper is organized as follows. In section II, a brief overview concerning the particular advantages of using TOF-cameras instead of conventional types is given. Further, we explain the processing steps applied to the time-of-flight data for obtaining feature vectors which are afterwards used for training and online classification. In section III Locality Preserving Projections (LPP) are shortly reviewed. LPP were originally proposed for (unsupervised) manifold learning. We explain how it can be applied to supervised classification problems and describe its relation to Linear Discriminant Analysis (LDA). In projected space the actual

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

classification is performed by applying a threshold on the Mahalanobis distance and temporal filtering techniques are used for smoothening jittering effects caused by noise. In section IV several projection methods are compared with real data. Section V concludes the paper with a summary.

## II. Using time-of-flight cameras for feature extraction

### A. Advantages of time-of-flight cameras

Common to many of the human pose estimation approaches is the assumption that the camera is placed at a fixed spot in the room. This allows for applying a background-foreground segmentation technique to obtain a silhouette image of the person. In fact, most pose estimation approaches rely heavily on a good segmentation result. However, despite many years of research (see [12] for a review on background-segmentation), in practical, realistic environments conventional cameras still pose certain difficulties to obtaining a clear silhouette. These difficulties include the following:

- noise in the image,
- similar colors in the background and of the clothes worn by persons,
- shadows that result in additional, unwanted variations between the background model and the live images,
- changing lighting conditions (*e.g.* when someone switches a light on or off) and,
- completely dark environments, as in the case of monitoring the behavior of people at night, where conventional cameras are not applicable, at all.

Due to these difficulties we propose to use images coming from time-of-flight (TOF) cameras [13], instead. TOF cameras produce a three dimensional, geometrical representation of the scene by emitting infra-red light and measuring the time the light takes to returning back to the camera.[1] With TOF cameras most of the abovementioned difficulties are overcome, as the representation of the scene is based on its geometry and not on its visual appearance, thus making it more robust and more applicable in realistic scenarios.

### B. Feature extraction

In this section we briefly review the different preprocessing steps perfomed on the time-of-flight data in order to obtain "features", *i.e.* robust, vector-based descriptions of fixed size that implicitly contain the posture information, while beeing invariant to the person's location in the room and in the image. Although TOF-cameras directly produce a cloud of 3D-coordinates of the perceived scene as output, each depth value is also associated to a pixel-coordinate. Thus, many of the conventional image processing techniques may be applied to the TOF-data in a similar fashion, including background-subtraction. However, compared to grey-level images from conventional cameras, the pixel values (depth-values) have much higher and very heterogeneous noise.

The first step consists of a background-subtraction and a subsequent connected-component merging, where the background may either learned once beforehand or recursively during online evaluation. In order to account to the high noise level of the TOF data, we adopted a recently proposed algorithm [14] that can estimate recursively the mean and variance for each pixel in the background. This serves for deciding whether the deviation in depth is significant enough to interpret it as foreground or background pixel (see figure 1, top-right). Next, the identified foreground pixels are grouped into connected regions ('blobs'). This step is used to remove any false positives arising especially due to data noise. A simple threshold is applied to the size of the connected components, *i.e.* blobs below a certain size are simply discarded. Furthermore, the largest blob among the remaining ones is assumed to be the one corresponding to a person.

Afterwards the subimage is cropped and resampled, *i.e.* based on the previous region-of-interest (ROI) detection, a low resolution sub-image with fixed size and constant aspect ratio is cropped in such a way that it contains as much foreground as possible. As a result the sub-image is always centered around the largest foreground region (see figure 1, bottom-left). Depth values are computed in relation to the mean of the ROI, whereupon points closer to the camera are assigned a higher value and farther points or points in the background obtain a value close or equal to zero. Values inbetween pixel locations are approximated by bi-linear interpolation. The cropped and resampled sub-image is interpreted as a high dimensional feature vector, $\mathbf{x}_t$, by stacking the rows of the image such that each pixel value represents one element of the vector (*e.g.* a 20-by-24 image results in a 480 dimensional vector $\mathbf{x}_t$). The feature vectors are subsequently used as input for both, offline training as well as online projection and classification.

## III. Linear projection techniques for classification

We investigated machine learning approaches that rely on a linear projection from high dimensional data, $\mathbf{x} \in \mathbb{R}^L$, to a low dimensional representation, $\mathbf{y} \in \mathbb{R}^M, M \ll L$:

$$\mathbf{y} = \mathbf{V}\mathbf{x}, \quad \mathbf{V} \in \mathbb{R}^{M \times L} \tag{1}$$

The appealing property of such methods is that the projection itself only requires a relatively small amount of multiplications and additions. Moreover, the actual classification, *i.e.* the assignment of a class label, $c = 1, 2, ..., C$, to every input $\mathbf{x}_t$, can be performed with significantly lower computational effort in projected space, making the whole process applicable to problems that need real-time evaluation. The various projection methods under consideration differ only in the way how the projection matrix is computed from the training data. Of course, the method should ideally be designed in such a way that a minimal amount of discriminative information is lost in the process of projecting the feature vectors. Moreover, most of the discriminant power should be contained in the first few leading row vectors of the projection matrix, such that data maps into a space with significantly reduced dimensionality.

[1] More precisely, a TOF camera emits sinusoidal light impulses and measures the phase shift of the returning light for each pixel.

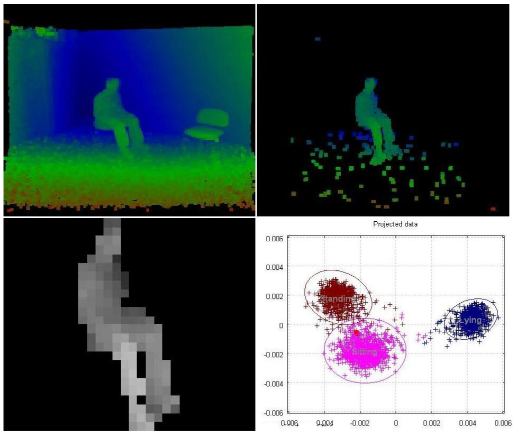

Fig. 1. Illustration of the different processing steps for the online posture classification (best viewed in color). From top-left to bottom-right: Color encoded visualization of the 3D raw data coming from a TOF-camera; result after background-subtraction; cropped feature-image with the depths grey-valued encoded relative to the mean of the ROI; projected training data for three states, "Standing", "Sitting and "Lying" (brown, pink and blue), and the projection of the current feature image on the left (red point above the "Sitting" caption). The ellipses represent the Mahalanobis distance threshold for each class.

In order to learn the projection, a set of reference images needs to be captured, in advance. For supervised methods they need to be annotated manually according to their corresponding discrete states. This may seem a laborious task at a first glance, but in fact, for discrete target outcomes one may capture long sequences with the person being only in one single state, such that the labeling can be done in bundles.

The two most popular projection methods are PCA and LDA, which are particularly widespread in face recognition, but also *e.g.* in character or digit classification. Considering the former, the projection vectors (rows of $\mathbf{V}$) are often termed Eigenfaces (PCA) [7] or Fisherfaces (LDA) [8], due to their ghost-like resemblance to faces when interpreted as images. The projection of each approach relies on a specific optimality criterion. PCA is unsupervised and seeks to maximize the global variance of the projected data, whereas LDA incorporates the class labels in order to maximize the ratio between the inter- and intra-class variance. Recently, He and Niyogi [9] proposed another method, Locality Preserving Projections, which aims at preserving neighborhoods from high dimensional space to projection space as best as possible. However, as the authors themselves point out, neighborhoods of points do not necessarily have to be defined on the original data, only. Any external source may equally well serve for constructing 'locality'. In this paper we investigate the application of LPP to classification by defining the neighborhood according to all points that share same class labels.

Once the optimal projection has been obtained, it can be applied to the online feature vectors. The projected online features are then classified according to the closest mean vector of the projected training data. Furthermore we apply a threshold on the Mahalanobis distance between the online feature vector and the covariance of the projected training data of each class (see figure 1, bottom-right). If the distance exceeds a threshold, the state is considered to be "Unknown".

### A. Supervised Locality Preserving Projections

LPP was originally formulated as an alternative method for manifold learning (LLE, ISOMAP, Laplacian Eigenmaps) to obtain a low dimensional parameterization of the underlying manifold in the high dimensional input data space. The leading projection vectors are computed in such a way that feature vectors being 'close' in the original high dimensional space remain 'close' together in the low dimensional space after projection, according to a suitable definition of 'closeness'. This idea can be formalized by the following objective function,

$$\begin{aligned} &\arg\min_{\mathbf{v}} \sum_{k_1,k_2}^{K} \left(\mathbf{y_{k_1}} - \mathbf{y_{k_2}}\right)^{\mathbf{2}} \mathbf{W}_{k_1,k_2} \\ &= \arg\min_{\mathbf{v}} \mathbf{v^T X L X^T v} \qquad (2) \end{aligned}$$

subject to the constraint (to avoid the trivial solution $\mathbf{v} = \mathbf{0}$),

$$\mathbf{v^T X D X^T v} = 1, \qquad (3)$$

which finally results in the following generalized eigenvalue problem (GEVP): [2]

$$\mathbf{X D X^T v} - \lambda \mathbf{X L X^T v} = \mathbf{0}. \qquad (4)$$

In the formulation above, $K$ denotes the number of training samples used and $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_K}] \in \mathbb{R}^{L \times K}$ is the matrix composed of the training data. $\mathbf{W} \in \mathbb{R}^{K \times K}$ is a sparse symmetric weighting matrix defining the similarity (or closeness) between two data entities, $\mathbf{D} \in \mathbb{R}^{K \times K}$ is the corresponding diagonal matrix with its entries being the sum of the rows (or columns) in $\mathbf{W}$, $\mathbf{D}_{k,k} = \sum_k \mathbf{W}_k$, and the difference of both, $\mathbf{L} = \mathbf{D} - \mathbf{W}$, is called the Laplacian matrix.

In the original formulation of LPP [9], where it is presented in the context of manifold learning, the construction of $\mathbf{W}$ is *unsupervised* and purely based on the feature data $\mathbf{X}$. Two possible strategies are presented:

1) *$\varepsilon$-neighborhood*: $\mathbf{W}_{k_1,k_2} = 1$, if $\|\mathbf{x}_{k_1} - \mathbf{x}_{k_2}\|^2 < \varepsilon$ and $\mathbf{W}_{k_1,k_2} = 0$ otherwise.
2) *$k$ nearest neighbors*: $\mathbf{W}_{k_1,k_2} = 1$, if $\mathbf{x}_{k_2}$ is among the k nearest neighbors of $\mathbf{x}_{k_1}$ and vice versa, $\mathbf{W}_{k_1,k_2} = 0$ otherwise.

Additionally, all positive entries in $\mathbf{W}$ could be weighted with a kernel function. In [9] a Gaussian kernel is proposed, *i.e.* $\tilde{\mathbf{W}}_{k_1,k_2} = \mathbf{W}_{k_1,k_2} \cdot e^{-\frac{\|\mathbf{x}_{k_1} - \mathbf{x}_{k_2}\|^2}{t}}$, with $t \in \mathbb{R}$ chosen experimentally.

Now, the choice of the matrix $\mathbf{W}$ determines the actual characteristics of the projection. In fact, it even may be interpreted as a unifying element between different projection methods, since many of them may effectively be constructed by different rules regarding the choice of the entries $\mathbf{W}_{k_1,k_2}$.

[2] The GEVP formulation can be derived *e.g.* with the method of Lagrange.

Opposed to the unsupervised construction of $\mathbf{W}$ above, in this paper we propose to utilize the class labels $c_k$ of the training samples, directly, which results in a supervised variant of LPP. Specifically, we design $\mathbf{W}$ according to:

$$\mathbf{W}_{k_1,k_2} = \begin{cases} \frac{1}{n_c} & \text{if } c_{k_1} = c_{k_2} \\ 0 & \text{if } c_{k_1} \neq c_{k_2} \end{cases}, \tag{5}$$

where $n_c$ denotes the number of training samples per class $c$. Note that $\mathbf{D}$ becomes the identity matrix $\mathbf{I}_K$.

The solution of (4) yields $L$ eigenvalues, $\lambda_l$, and corresponding eigenvectors, $\mathbf{v}_l$. The projection matrix $\mathbf{V} \in \mathbb{R}^{M \times L}$ is constructed by using the $M$ eigenvectors that correspond to the $M$ highest eigenvalues. Then, the matrix of projected training samples, $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_K}] \in \mathbb{R}^{M \times K}$, is simply given by the product,

$$\mathbf{Y} = \mathbf{V}\mathbf{X}. \tag{6}$$

When constructing the matrix $\mathbf{W}$ according to (5), one should note that it is of reduced rank, as it consists of only $C$ independent rows (or columns). There are two important consequences of this property:

1) The generalized eigenvalue problem in (4) becomes numerically unstable and will not converge properly. However, a common regularization strategy is simply to add a small amount of the identity matrix (white noise) to $\mathbf{XLX}$ before solving the eigenvalue problem,

$$\mathbf{XDX^T v} - \lambda(\mathbf{XLX^T} + \epsilon^2 \cdot \mathbf{I}_K)\mathbf{v} = 0, \tag{7}$$

where $\epsilon^2$ might be a fraction of the norm $\|\mathbf{XLX^T}\|_F$.

2) Furthermore, we observe[3] that only the first $C$ eigenvectors have discriminative capability and the remaining project the feature vectors to fully overlapping regions.

### B. Review of LDA

Linear Discriminant Analysis (LDA) seeks to maximize the ratio between the projected between-class scatter covariance, $\mathbf{S}_b$, and the within-class covariance, $\mathbf{S}_w$, (Fisher criterion):

$$\arg\max_{\mathbf{v}} \frac{\mathbf{v^T S}_b \mathbf{v}}{\mathbf{v^T S}_w \mathbf{v}}, \tag{8}$$

where

$$\mathbf{S}_b = \sum_{c=1}^{C} n_c(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})^T, \text{ and} \tag{9}$$

$$\mathbf{S}_w = \sum_{c=1}^{C}\sum_{i=1}^{n_c}(\mathbf{x}_i{}^c - \bar{\mathbf{x}}_c)(\mathbf{x}_i{}^c - \bar{\mathbf{x}}_c)^T. \tag{10}$$

Here, $\bar{\mathbf{x}}$ denotes the mean of all training samples and $\bar{\mathbf{x}}_c$ the mean for class $c$. This leads to the following generalized eigenvalue problem:

$$\mathbf{S}_b\mathbf{v} - \lambda\mathbf{S}_w\mathbf{v} = \mathbf{0}. \tag{11}$$

If we set the similarity matrix, $\mathbf{W}$, according to (5) again, we can rewrite the generalized eigenvalue problem of (11) to

$$\mathbf{X}(\mathbf{W} - \mathbf{1}_K/K)^2\mathbf{X^T v} - \lambda\mathbf{X}(\mathbf{I} - \mathbf{W})^2\mathbf{X^T v} = \mathbf{0}, \tag{12}$$

with $\mathbf{1}_K$ being a $K$ by $K$ matrix with all values equal to one. This is in turn equivalent to:

$$\mathbf{X\tilde{D}X^T v} - \lambda\mathbf{XLX^T v} = \mathbf{0}, \tag{13}$$

where we defined $\tilde{\mathbf{D}} = \mathbf{W} - \mathbf{1}_K/K$.[4] From here, we observe that the only difference with regard to the GEVP of the supervised LPP, (4) and (5), is the first term, where $\tilde{\mathbf{D}}$ appears instead of $\mathbf{D} = \mathbf{I}$. Note that LDA can be derived with a different constraint, (3), but with the same objective function like LPP, (2), thus both methods share the same fundament. Moreover, according to our experience, this difference will only introduce a scaling, shift and rotation of the projected feature vectors, but does not affect their intra- vs. inter-scattering behavior considerably. Stated differently, LPP provides a more general framework than LDA, comprising an unsupervised manifold learning method in its original sense as well as a supervised discriminating projection like LDA.

### C. Classification in projected space

The main advantage of projecting the feature vectors into a lower dimensional space lies in the reduced computational effort required for the actual classification step. Typically, based on the projected training features, a set of linear decision surfaces is constructed [15]. Alternatively the classification can simply be performed by nearest-neighbor (NN) search with respect to the means. The problem with these methods is that they assume a closed set of possible output classes, *i.e.* the output is *always* mapped to one of the predefined classes.

In practice, we would like the algorithm to signalize, when beeing confronted with unknown situations. It would be unfeasible to train the learning method with negative examples, since any selection would hardly be exhaustive enough. Alternatively, we propose to classify on the basis of the Mahalanobis distance, by incorporating both, the class-specific means $\bar{\mathbf{y}}_c$ and the covariances $\mathbf{R}_c$ of the projected training vectors $\mathbf{y}$:

$$\bar{\mathbf{y}}_c = \frac{1}{n_c}\sum_{i \in \{k|c_k = c\}}^{K} \mathbf{y}_i, \tag{14}$$

and

$$\mathbf{R}_c = \frac{1}{n_c - 1}\sum_{i \in \{k|c_k = c\}}^{K} (\mathbf{y}_i - \bar{\mathbf{y}}_c)(\mathbf{y}_i - \bar{\mathbf{y}}_c)^T. \tag{15}$$

The abovementioned computations can be done once in the offline training phase. During online computations we first determine the closest mean vector of the projected feature, $\mathbf{y}_t = \mathbf{V}\mathbf{x}_t$, by nearest-neighbor search. Next, having found the class with the closest mean, $\bar{\mathbf{y}}_c$, we validate the classification result by applying a threshold on the Mahalanobis distance,

$$d_c(\mathbf{y}_t) = (\mathbf{y}_t - \bar{\mathbf{y}}_c)^T R_c^{-1}(\mathbf{y}_t - \bar{\mathbf{y}}_c), \tag{16}$$

*i.e.* if $d_c(\mathbf{y}_t)$ exceeds a threshold, $\rho$, we reject the result.[5]

[3] The observation is based on simulations. A formal proof is omitted here.

[4] Notice that $\mathbf{W}^2 = \mathbf{W}$, $\mathbf{W} \cdot \mathbf{1}_K = \mathbf{1}_K$ and $\mathbf{1}_K^2 = K \cdot \mathbf{1}_K$, which yields $(\mathbf{W} - \mathbf{1}_K/K)^2 = \mathbf{W} - \mathbf{1}_K/K$ and $(\mathbf{I} - \mathbf{W})^2 = \mathbf{I} - \mathbf{W} = \mathbf{L}$.

[5] If we assume a Gaussian distribution of the projected data, $\rho = 1.0$ would correspond to the standard deviation $\sigma$. In practice, we set $\rho = 3.0$.

As described so far, the classification is performed for each time instant independently, without incorporating temporal relations inbetween the frames. In fact, since our posture classification approach runs in real-time with more than 15 frames per second on a conventional PC, it is a reasonable assumption that the monitored person maintains its posture during several frames before changing to another. In this case temporal filtering techniques can be applied to the projected features prior classification in order to reduce jittering effects caused by noisy measurements of the camera. This can be achieved by low-pass filtering with a suitable ARMA-filter or a Kalman-filter. In practice, we obtained satisfactory results by applying a second order auto-regressive filter of the form:[6]

$$\mathbf{y}_t = (1 - a_1 - a_2)\mathbf{V}\mathbf{x}_t + a_1\mathbf{y}_{t-1} + a_2\mathbf{y}_{t-2}. \qquad (17)$$

## IV. Results and validation

We validated our approach with real data, both used for training and evaluation.[7]

In the first scenario we trained the system to distinguish between three different states, namely 'Sitting', 'Standing' and 'Lying' (see figure 2). For the 'Sitting' state people sat onto chairs at predefined places and turn themselves 180° around their axis back and forth. The places of the chairs were changed from time to time in order to increase the generalization ability for multiple perspectives. The 'Standing'-state was trained by letting them walk randomly across the room. For the 'Lying' state people had to turn themselves completely (*i.e.* 360°) while lying on the floor. The samples also include cases with people lying sideways or having arms and legs spread apart. The whole dataset comprised 8000 TOF images, of which one-third was used for training and the rest for evaluation.

It should be noted that in this first scenario the training data has a high *intra-class variation* between postures of the same class, *i.e.* a sitting person may be turned to the left, to the right or directly facing the camera. Despite these variations the feature vectors belonging to the same classes are projected to similar locations and in the low dimensional pose space as can be seen in figure 3 (top, middle-left) for the two dominant projection directions (2nd and 3rd eigenvector). Moreover, also the validation data below still forms well separated clusters at the same locations with marginal stronger variation.

Furthermore, we compared the Supervised LPP algorithm to other linear projection techniques. Figure 3 (middle) shows the projection using the first two eigenvectors of LDA. As can be seen, the projected data is a scaled and shifted copy of the result obtained by Supervised LPP, which confirms the presumption that both learning methods are strongly interrelated. The application of Principal Component Analysis yields less clear separations. Figure 3 (right) also shows the results obtained by using the unsupervised LPP-method. Interestingly, the projected data forms half-circles, which on the one hand reflects the different orientations in the postures and on the other hand owes to the fact that LPP was originally designed for retrieving the underlying manifold structure of the data.

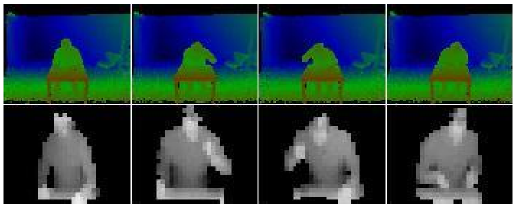

Fig. 4. Training samples of the 'Eating' dataset. Upper row: color encoded TOF-data. Lower row: feature images. The states from left to right are: 'ArmsOnTable', 'LeftArmToHead', 'RightArmToHead' and 'Cutting'

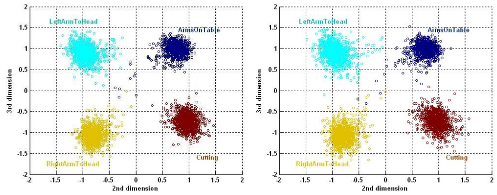

Fig. 5. Projected feature vectors for the 'Eating' dataset using the second and third eigenvectors of the Supervised LPP method for projection: training (left) and validation (right).

The results in figure 5 correspond to a different setup in order to detect different states of a person sitting at a table while eating. The posture classes 'ArmsOnTable', 'LeftArmToHead', 'RightArmToHead' and 'Cutting' (see figure 4) were again trained with several people of different sizes. Here also, approximately 8000 data samples were used, half of them for validation. Note that in this scenario there is a *low interclass* variation, *i.e.* the feature images seem to be fairly similar when comparing postures of different classes. However, the projected feature vectors showed in figure 5 again form well separated clusters, both in the training dataset and the validation dataset.

## V. Conclusion

In this paper we presented an approach for classifying human postures in time-of-flight images. As explained in detail, using time-of-flight cameras the feature extraction proves to be more reliable, as it is not necessary to make restrictive assumptions on *e.g.* stable lighting conditions or on the colors of the clothes worn by the persons. Furthermore we investigated linear projection techniques with a particular focus on a supervised version of Locality Preserving Projections for which we analyzed its relation to Linear Discriminant Analysis. Methods for classification and noise reduction in projected space were presented. Finally, we evaluated our approach on real data, where we showed that it is capable of capturing large intra-class and low inter-class variations, thus supporting a variety of human monitoring applications within an Ambient Assisted Living environment.

## Acknowledgment

This work has been partially funded by the EU IST Project PERSONA (FP6 contract N.045459).

[6] We used $a_1 = 0.2$ and $a_2 = 0.4$.

[7] Corresponding videos can be downloaded from ftp://ftp.igd.fraunhofer.de/outgoing/fowienta

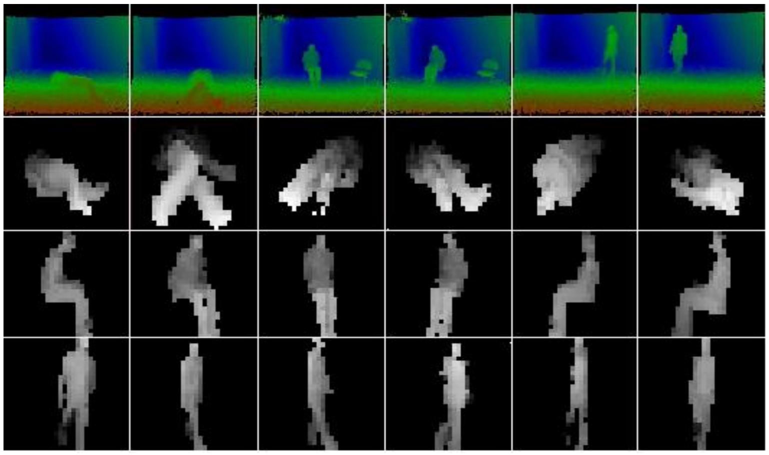

Fig. 2. First row: samples of color encoded TOF-data. Second to fourth row: feature images for the states 'Lying', 'Sitting' and 'Standing', respectively.

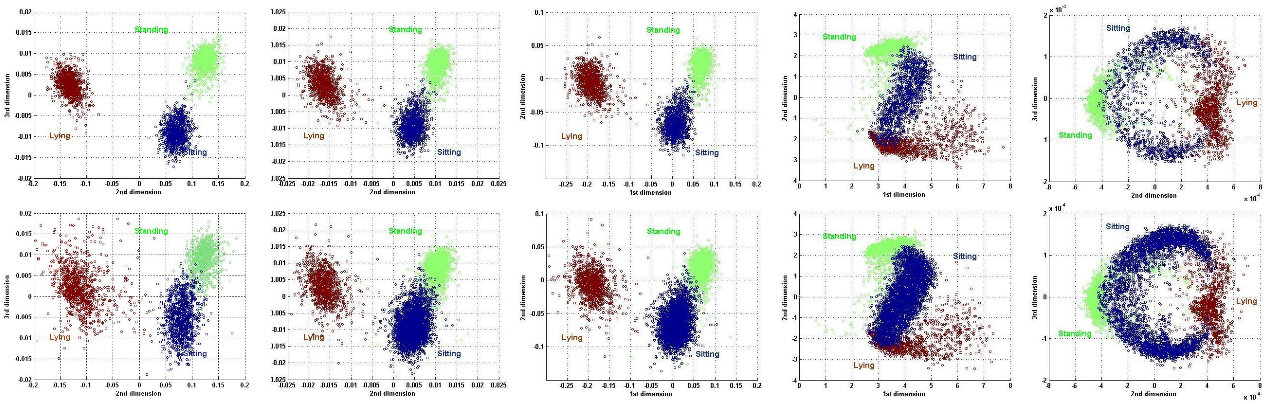

Fig. 3. Validation results for three states, 'Sitting' (blue), 'Standing' (green) and 'Lying' (red). Left: Supervised LPP trained with a tall, male person (top) and evaluated with another small, female person (bottom). All other results are based on a dataset with a mix of persons of which one third is taken for training (top) and the rest for evaluation (bottom). Supervised LPP (middle-left); Linear Discriminant Analysis (middle); PCA (middle-right); Unsupervised (original) LPP with a Gaussian kernel size of $t = 0.75L^2$ and an $\varepsilon$-neighborhood of $\varepsilon = 0.5L^2$).

## References

[1] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90–126, Nov. 2006.

[2] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, Oct. 2007.

[3] M. Amoretti, F. Wientapper, F. Furfari, S. Lenzi, and S. Chessa, "Sensor data fusion for activity monitoring in ambient assisted living environments," *S-Cube 2009, 1st International Conference on Sensor Systems and Software*, Sept. 2009.

[4] R. Cucchiara, A. Prati, and R. Vezzani, "Posture classification in a multi-camera indoor environment," *Image Processing, 2005. IEEE International Conference on*, vol. 1, pp. 725–728, Sept. 2005.

[5] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani, "Probabilistic posture classification for human-behavior analysis," *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 35, no. 1, pp. 42–54, Jan. 2005.

[6] S. Pellegrini and L. Iocchi, "Human posture tracking and classification through stereo vision and 3d model matching," *Journal on Image and Video Processing*, vol. 8, no. 2, pp. 1–12, Apr. 2008.

[7] M. Turk and A. Pentland, "Face recognition using eigenfaces," *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 586–591, Jun. 1991.

[8] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[9] X. He and P. Niyogi, "Locality preserving projections," in *In Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2003.

[10] L. Wang and D. Suter, "Learning and matching of dynamic shape manifolds for human action recognition," *Image Processing, IEEE Transactions on*, vol. 16, no. 6, pp. 1646–1661, 2007.

[11] ——, "Analyzing human movements from silhouettes using manifold learning," *Advanced Video and Signal Based Surveillance, IEEE Conference on*, p. 7, 2006.

[12] M. Piccardi, "Background subtraction techniques: a review," *Systems, Man and Cybernetics, IEEE International Conference on*, vol. 4, pp. 3099–3104 vol.4, Oct. 2004.

[13] MESA Imaging. (2009, February) http://www.mesa-imaging.ch/.

[14] F. Porikli and O. Tuzel, "Bayesian background modeling for foreground detection," in *Proceedings of the third ACM international workshop on Video surveillance & sensor networks*. ACM, 2005, pp. 55–58.

[15] H. Wang, S. Chen, Z. Hu, and W. Zheng, "Locality-preserved maximum information projection," *Neural Networks, IEEE Transactions on*, vol. 19, no. 4, pp. 571–585, Apr. 2008.