Analyzing the Subspaces Obtained by Dimensionality Reduction for Human Action Recognition from 3d Data

Marco Körner and Joachim Denzler
Chair for Computer Vision
Friedrich Schiller University of Jena
Jena, Germany
Email: {marco.koerner, joachim.denzler}@uni-jena.de

***Abstract*—Since depth measuring devices for real-world scenarios became available in the recent past, the use of 3d data now comes more in focus of human action recognition. Due to the increased amount of data it seems to be advisable to model the trajectory of every landmark in the context of all other landmarks which is commonly done by dimensionality reduction techniques like PCA. In this paper we present an approach to directly use the subspaces (*i.e.* their basis vectors) for extracting features and classification of actions instead of projecting the landmark data themselves. This yields a fixed-length description of action sequences disregarding the number of provided frames. We give a comparison of various global techniques for dimensionality reduction and analyze their suitability for our proposed scheme. Experiments performed on the CMU Motion Capture dataset show promising recognition rates as well as robustness in the presence of noise and incorrect detection of landmarks.**

***Keywords*-Human Action Recognition; Manifold Learning; Dimensionality Reduction; PCA; Kernel PCA; Isomap; Spectral Regression**

I. Introduction and Related Work

The recognition and analysis of actions and motions performed by humans became one of the most promising fields in computer vision research and lead to a wide variety of research topics. This family of problems aims to automatically determine human activities based on sensor observations and serves for a wide range of applications, *e.g.* human-machine interaction, surveillance, security and entertainment.

Since the complexity of classification tasks grows with the dimensionality of the input data, *manifold learning* techniques are commonly used to reduce the number of valid dimensions by finding an application-specific optimal projection to a lower-dimensional target space which might be more suitable for separating data clusters.

The usability of *Principle Component Analysis* (PCA) and *Independent Component Analysis* (ICA) on motion silhouettes was previously compared [1], [2]. Furthermore, *Locality Preserving Projections* (LPP) were utilized in combination with a special Hausdorff distance measure on silhouettes [3]. Other techniques for dimensionality reduction like *Locality Sensitive Discriminative Analysis* (LSDA) and *Local Spatio-Temporal Discriminant Embedding* (LSTDE) were compared in [4], while Tensor PCA was employed for reducing the dimensionality of the parameter space in [5].

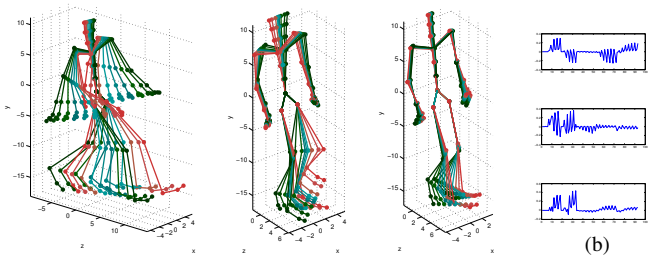
In the field of 3d human action recognition, far less work exist. The approach in [6] employs *Laplacian Eigenmaps* to recognize human actions from 3d points delivered by full-body ToF scans. *Hierarchical Gaussian Process Latent Variable Modeling* (H-GPLVM) combined with *Conditional Random Fields* (CRF) was used to classify actions from *CMU Motion Capture* data in [7].

Most of the approaches mentioned above attempt to use dimensionality reduction to project the data points into a more feasible target coordinate system with lower dimensionality and perform classification on these projected data. In contrast, we propose to use the target coordinate system bases themselves to extract features, similar to [8]. Following this scheme, any sequence of an arbitrary action can be represented by a small set of basis vectors and an mean shape which is independent from the number of frames. This allows to reduce computational time as well as memory needed for the classification data. In Sec. II we present an overview about various global dimensionality reduction techniques. Our proposed scheme for representation and classification of actions in 3d data will be presented in Sec. III. Experiments performed on the CMU MoCap dataset gave promising results compared to others (Sec. IV).

II. Global Approaches for Dimensionality Reduction

Dimensionality reduction is widely used for recognition of shapes and actions. Most approaches employ the embedded data points to find a representation with lower dimensionality. In contrast, we propose to use the projection parameters themselves for classification, as described in the following.

Using a hierarchical model as in [9], any arbitrary skeleton configuration at time step $1 \leq t \leq N_\mathrm{f}$ can be parameterized as a vector of landmarks $\boldsymbol{l}^t = \left((x,y,z)_1^t, \ldots, (x,y,z)_{N_\mathrm{j}}^t\right) \in \mathbb{R}^{3 \cdot N_\mathrm{j}}$, while N_j is the number of joints in the model. To enforce stationarity, the actor's skeleton is normalized in an anatomically correct fashion by setting the global rotation $(\theta_1^t, \theta_2^t, \theta_3^t)$ and translation (x_1^t, y_1^t, z_1^t) to zero. Combining all



(a) (b)

Figure 1: The first three motion components of a `walking` action (a) and their corresponding eigenvectors (b). Colors indicate the weighting of the eigenvectors added to the mean shape (black: $w_k^t = -2\lambda_k^2$, blue: $w_k^t = 0$, red: $w_k^t = 2\lambda_k^2$). Note the anti-symmetric motion directions of the limbs in the first two components and the symmetric one in the third component.

Table I: Comparison of the mean shapes and the first three eigenvectors of the action classes in our dataset. Note that similar actions have similar higher-order eigenvectors.

Action	$\boldsymbol{l}_\mu$	$\boldsymbol{v}_1^{\boldsymbol{L}}$	$\boldsymbol{v}_2^{\boldsymbol{L}}$	$\boldsymbol{v}_3^{\boldsymbol{L}}$
`walk`				
`run`				
`march`				
`sneak`				
`hopp`				
`jump`				
`golf`				
`salsa`				

zero-mean skeleton configurations at every time step yields the matrix

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{l}^1 - \boldsymbol{l}_\mu \\ \vdots \\ \boldsymbol{l}^{N_\mathrm{f}} - \boldsymbol{l}_\mu \end{pmatrix} \in \mathbb{R}^{N_\mathrm{f} \times 3 \cdot N_\mathrm{j}}, \quad \boldsymbol{l}_\mu = \frac{1}{N_\mathrm{f}} \sum_{i=1}^{N_\mathrm{f}} \boldsymbol{l}^i. \qquad (1)$$

Although the sequences have a varying number of frames N_f, we aim to find a representation of fixed length. This section presents a selection of algorithms to estimate *global linear* transformations in order to reduce the dimensionality of our data and capture their variances.

Principal Component Analysis (PCA). *PCA* aims to find a low-dimensional representation for all data points $\boldsymbol{X}$ which preserves the maximum amount of variance of the original data. For this purpose, a linear basis transformation $\boldsymbol{T}$ to maximize the variances in every coordinate direction is estimated by solving the eigenproblem $\mathrm{cov}(\boldsymbol{X} - \boldsymbol{X}_\mu)\boldsymbol{T} = \lambda \boldsymbol{T}$ for the k_ev largest eigenvalues. Performing PCA on matrix (1) yields the matrix $\boldsymbol{P}_{\boldsymbol{L}} = \left(\boldsymbol{v}_1^{\boldsymbol{L}} | \cdots | \boldsymbol{v}_{3N_\mathrm{j}}^{\boldsymbol{L}}\right) \in \mathbb{R}^{3N_\mathrm{j} \times 3N_\mathrm{j}}$ of eigenvectors sorted descendingly according to their corresponding eigenvalues (*cf.* Fig. 1 and Tab. I). Every new skeleton configuration $\boldsymbol{l}'$ can then be expressed by a linear combination $\boldsymbol{l}' = \boldsymbol{l}_\mu + \boldsymbol{P}_{\boldsymbol{L}} \boldsymbol{b}_{\boldsymbol{l}'}$ of the eigenvectors in the data matrix columns and the frame-specific *shape parameters* $\boldsymbol{b}_{\boldsymbol{l}'}$ (*i.e.* the *motion components*) added to the constant *mean shape* $\boldsymbol{l}_\mu$.

Probablistic PCA (P-PCA). While the computational complexity of PCA grows cubically with the data dimensionality, approximative approaches were derived. *Probabilistic PCA*, for example, gives an EM-based reformulation of the standard PCA [10].

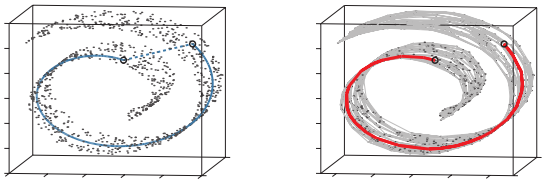
Kernel PCA (K-PCA). Instead of calculating the eigenvectors of the data covariance matrix, the *Kernel PCA* approach analyzes the kernel matrix $\boldsymbol{K} = \{k\}_{i,j}, k_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with kernel function $\kappa(\cdot,\cdot)$ [11]. Widely used kernel functions are for instance

$$\kappa_\mathrm{gauss}(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-|\boldsymbol{x}-\boldsymbol{y}|^2/2\sigma^2\right) \quad \textit{(Gaussian RBF)}, \qquad (2)$$
$$\kappa_\mathrm{poly}(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^\top \boldsymbol{y}\right)^d \quad \textit{(polynomial)}, \qquad (3)$$
$$\kappa_\mathrm{poly+}(\boldsymbol{x}, \boldsymbol{y}) = \left(\boldsymbol{x}^\top \boldsymbol{y} + 1\right)^d \quad \textit{(polynomial+)}. \qquad (4)$$

Using the linear kernel $\kappa_\mathrm{linear}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top \boldsymbol{y}$ yields the standard PCA.

Isometric Feature Mapping (Isomap). In addition to maximizing the variances along the coordinate axes, the preservation of local distances between data points yields another optimization criterion. The *Multidimensional Scaling (MDS)* approach optimizes the stress function $\phi(\boldsymbol{X}) = \sum_{i,j} \left(|x_i - x_j| - |y_i - y_j|\right)^2$ using Euclidean distances between the data points [12]. While this approach tends to produce short-circuits between layered data points (*cf.* dashed line in Fig. 2a), the *Isometric Feature Mapping* algorithm [13] minimizes geodesic distances between points along their underlying manifold (*cf.* solid line in Fig. 2a). The *Isomap* algorithm first builds a neighborhood graph, where every data point is connected by weighted edges with its k_nb nearest neighbors or all neighboring points within a specified margin ε_nb (*cf.* Fig. 2b) weighted by the local Euclidean distances. The sum of weights along the shortest path (*e.g.* obtained by Dijkstra's algorithm) equals the geodesic distance between two points within the manifold. This approach can be regarded as an instance of Kernel PCA using the kernel matrix $\boldsymbol{K} = \frac{1}{2}\boldsymbol{H}\boldsymbol{D}^2\boldsymbol{H}$, $\boldsymbol{H} = \boldsymbol{I}_N - \frac{1}{N}\boldsymbol{1}_n\boldsymbol{1}_n^\top$, where $\boldsymbol{D}^2$ is the matrix of squared pairwise geodesic point distances.

Neighborhood Preserving Embedding (NPE). In contrast to global linear techniques mentioned above, a variety of local nonlinear approaches exist. *Local Linear Embedding* (LLE) works similar to Isomap, but aims to solely preserve local geometric properties by approximating every data point by a linear combination of its k_nb nearest neighbors [14]. While this method estimates individual transforms for every



(a) (b)

Figure 2: (a) While *MDS* optimizes the Euclidean distances between data points (straight dashed line), *Isomap* employs the geodesic distance along the underlying manifold (solid line). (b) Approximation of geodesic distance by shortest path in the neighborhood graph. (Figures obtained from [13])

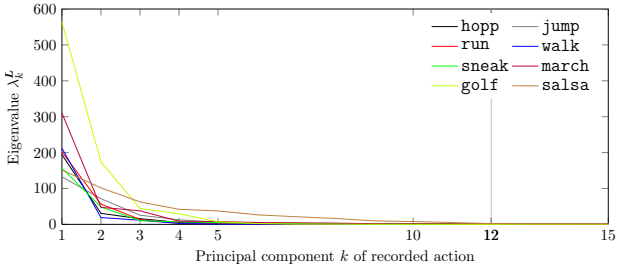


Figure 3: Evolution of PCA eigenvalues for different action classes. Eigenvalues are decreasing massively up to the third component, while they remain static for higher-order components.

data point, it does not fit in our proposed scheme. To overcome this problem we use *Neighborhood Preserving Embedding* to linearly approximate LLE by minimizing its underlying cost function [15].

Spectral Regression (SR). The computational complexity of eigen-decomposition grows cubically with the amount of data. Hence, the *Spectral Regression* framework reformulates the subspace learning problem as a two-step approach, namely graph embedding of the input data and regression for learning the parameters of projection functions [16]. Following this formulation, solely a small set of regularized least-square problems has to be solved, which runs with linear complexity. The smoothness of the parameter regression is controlled by adjusting the parameter α_{reg}. Since other graph embedding approaches (*e.g.* NPE or LPP) can be fit into this framework, SR was proven to approximate their results with high accuracy.

III. Employing Projection Spaces for Classification

A. Discriminability of Basis Vectors

When performing dimensionality reduction like PCA on sequential data $\boldsymbol{L}$, the result shows the most important directions of variance in the data. The eigenvectors $\boldsymbol{v}_k^{\boldsymbol{L}}$ corresponding to the largest eigenvalues $\lambda_k^{\boldsymbol{L}}$ are supposed

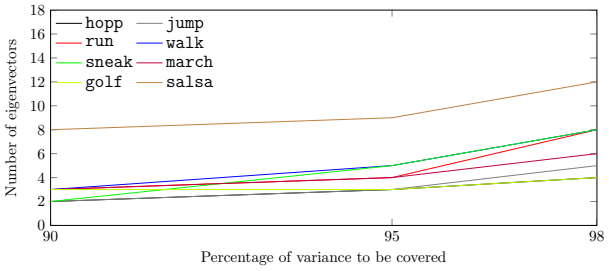


Figure 4: Variance covering facilities of our proposed representation scheme. Note that in most cases 90% of variance can be modeled by using $2-3$ basis vectors.

to encode most of the information, while the eigenvectors corresponding to the smaller eigenvalues model only minor changes in the data as well as noise. As already shown in Tab. I, the higher-order eigenvectors of similar action classes share a common appearance, while they behave contrary for different action classes.

Fig. 3 depicts the evolution of the eigenvalues for all action classes in our dataset. After a rapid descent up to the third principal component, the eigenvalues converged strongly towards zero. Beyond a certain number of components, there was no substantial contribution to the data. This argues for a well determined ordering of the first few eigenvectors while higher order eigenvectors may be permuted.

As depicted in Fig. 4, in most of the cases two to three eigenvectors are sufficient to cover 90% of the variances occurred during perfoming an exemplary action sequence. Solely the action classes with high variances in all directions need more discriminability, which can be handled by using a larger amount of eigenvectors.

Since the amount of represented variance within the landmark sequences decreases massively according to the evolution of the corresponding eigenvalues, the number of basis vectors can be restricted to achieve a substantial reduction of dimensionality. In Fig. 5 the Euclidean backprojection errors

$$\varepsilon_{\text{action}}(\boldsymbol{l}') = \left\| \left(\boldsymbol{l}' \cdot \widetilde{\boldsymbol{P}}_{\boldsymbol{L}^{\text{action}}} \right) \cdot \widetilde{\boldsymbol{P}}_{\boldsymbol{L}^{\text{action}}}^{\top} - \boldsymbol{l}' \right\|_2 \qquad (5)$$

of a `hopp` sequence obtained by projecting every frame into the reduced eigenspace $\mathbb{V}_{\text{action}}$ and back to Euclidean space $\mathbb{R}^3$ are displayed. While the projection into a related basis system like `jump` cause small errors, unrelated projections result in noticeable larger errors.

B. Feature Vector Design

The observations described in the previous section justify the assumption that the mean vector together with the first basis vectors obtained by the approaches described before contain all information about the variance within an action sequence. Due to the discriminative properties of the basis vectors obtained by the methods proposed before it becomes

Table II: Action classes selected from CMU MoCap dataset used in our experiments.

Class	walk	run	march	sneak	hopp	jump	golf	salsa
Samples	38	28	14	15	14	9	11	30
Actors	9	9	4	5	4	3	2	2
Avg. frame number	1283	853	6426	4200	602	1325	8626	5224

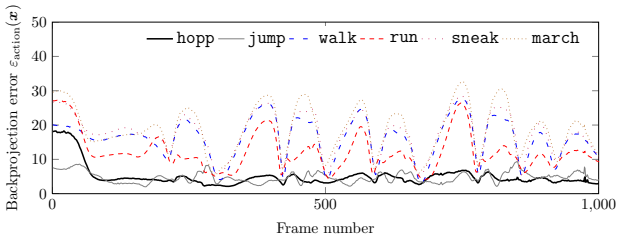


Figure 5: Back projection errors obtained by transforming a hopp sequence from Euclidean space into action-specific eigenspaces and back to Euclidean space.

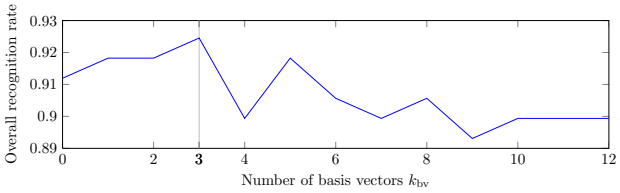


Figure 6: Effect of increasing the number of basis vectors used for building the feature vector on recognition rates obtained by k-NN classification.

evident to use them for classification of action sequences instead of employing the projected sequences themselves as proposed for other approaches. Hence, we design a feature vector $\boldsymbol{y}_{\boldsymbol{L}'} = \left(\boldsymbol{l}'_{\mu}, \boldsymbol{e}_1^{\boldsymbol{L}'}, \ldots, \boldsymbol{e}_{k_{\text{bv}}}^{\boldsymbol{L}'}\right)$ by combining the mean shape $\boldsymbol{l}'_{\mu}$ and the first k_{bv} basis vectors $\boldsymbol{e}_i$ (*e.g.* the first eigenvectors $\boldsymbol{v}_i^{\boldsymbol{L}}$ obtained by PCA). In order to overcome the ambiguity of the basis vector's signs, we enforce the maximal component to be positive. The main advantages of this representation scheme is the independence of the sequence lengths on the one hand as well as the low computational time needed to obtain these basis vectors on the other hand. For simplicity, we used the k *Nearest Neighbor* (k-NN) framework for classification and Euclidean distances $d(\boldsymbol{y}_{\text{test}}, \boldsymbol{y}_{\text{target}}) = \|\boldsymbol{y}_{\text{test}} - \boldsymbol{y}_{\text{target}}\|_2$.

In Fig. 6 one can see that in a first experiment the highest recognition rate is obtained by using $k_{\text{bv}} = 3$ basis vectors. Due to this observation as well as those displayed in Fig. 3 and Fig. 4 we fixed this parameter in further experiments.

IV. Experiments

In order to evaluate the proposed methods, we selected eight different actions from the CMU MoCap dataset performed by different actors, as shown in more detail in Tab. II. While we chose common actions with slightly different executions like walk, run, march and sneak or hopp and jump, we also took complex motions like salsa and golf into account.

A. Performance Comparison

We achieved results for the different projection techniques as shown in Tab. III by exhaustive leave-one-out tests. The K-PCA approach with polynomial kernel performed best, followed by the polynomial+ kernel. The Gaussian-kernelized PCA as well as the probabilistic PCA and Isomap yield results similar to the standard PCA algorithm while NPE performed worst. Since SP is designed to approximate graph embedding techniques, it performs similar to Isomap. The parameters used in these experiments were optimized to obtain best results. Due to these observations, we concentrate on K-PCA for further evaluation.

The confusion matrix shown in Tab. IV depicts the result of the K-PCA experiment using the polynomial kernel in more detail. All action classes were recognized correctly in more than 80% of the cases, while 5 classes gave recognition rates of nearly 100%. Solely the action class run has been confused with the semantically related classes walk and sneak due to their similar variations during execution.

The algorithms were implemented in Matlab. We obtained average run times of 30 ms for both feature extraction and testing performed on a standard desktop computer (Intel(R) Core(TM)2 Quad CPU Q9300 running with 2.50 GHz and 8 GB of RAM). Hence, the usage of standard techniques for dimensionality reduction makes our approach suitable for real-time applications.

B. Robustness to Noise

In real-world applications the input data for action classification are not ideal. Hence we modeled the influence

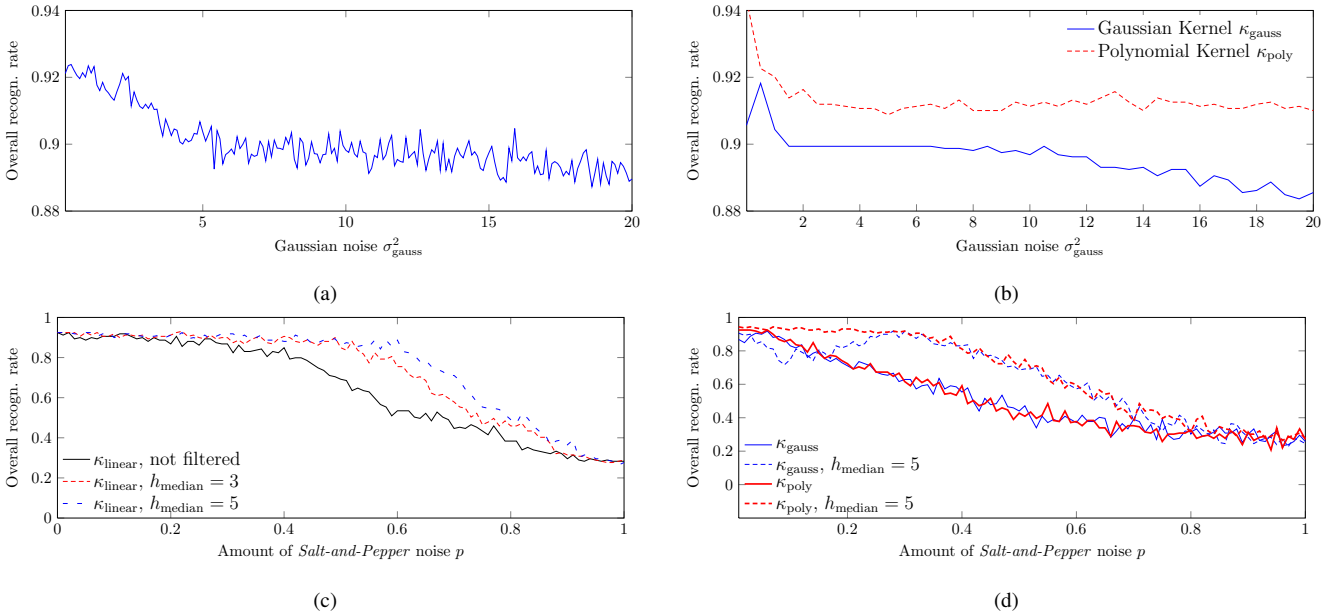


Figure 7: Influence of (a),(b) zero-mean Gaussian noise and (c),(d) *Salt-and-Pepper* noise to the recognition rates. In order to reduce the performance drop in (c),(d) a temporal median filter of size h_{median} was applied on the data as a preprocessing step.

Table III: Recognition rates obtained while using different techniques for dimensionality reduction.

Method		Parameters	Recognition Rates	
			Average	Overall
PCA		—	89.85%	92.45%
P-PCA		$s = 50$	88.06%	91.19%
K-PCA	*gauss*	$\sigma = 0.0005$	87.11%	90.57%
	poly	$d = 9$	**92.07%**	**94.34%**
	poly+	$d = 5$	91.18%	93.71%
Isomap		$k_{\text{nb}} = 7$	84.75%	90.60%
NPE		$k_{\text{nb}} = 15$	74.32%	78.52%
Spectral Regression		$\alpha_{\text{reg}} = 0.0065$	84.74%	90.60%

Table IV: Confusion matrix of exhausting leave-one-out test using Kernel PCA features with polynomial kernel κ_{poly}.

Training	Testing							
	walking	running	marching	sneaking	hopping	jumping	golfing	salsa
walking	**100**	0	0	0	0	0	0	0
running	22	**78**	0	11	0	0	0	0
marching	3	0	**97**	0	0	4	0	0
sneaking	0	0	0	**100**	0	0	0	0
hopping	13	7	0	0	**80**	0	0	0
jumping	0	0	7	7	0	**86**	0	0
golfing	0	0	0	0	0	0	**100**	0
salsa	0	0	3	0	0	0	0	**97**

of additive, zero-mean normally distributed and uncorrelated *Salt-and-Pepper* noise to quantitatively evaluate the robustness of our approach and compared the results of the standard PCA with K-PCA using the polynomial kernel. As can be seen in Fig. 7a and b, Gaussian noise added to the input data did affect the classification results just slightly for both PCA as well as K-PCA. In order to find the principal components, noise added to the data only affects the eigenvectors corresponding to the smaller eigenvalues, while the inherent and consistent information of movement over time is still captured by the eigenvectors corresponding to the larger eigenvalues. A similar behavior can be observed in the case of adding uncorrelated Salt-and-Pepper noise to input data. As shown in Fig. 7c, while the recognition rates were decreasing with the amount of added noise, median filters applied along the time dimension were able to drastically reduce these effects. In case of PCA, an amount of 60% Salt-and-Pepper noise can be compensated by a 5-frame median filter which only results in a small decrease in the recognition rates, whereas K-PCA seems to be more sensitive to Salt-and-Pepper noise. However, 5-frame median filtering still compensates up to 40% of uncorrelated outliers in the data without a substantial loss of accuracy.

C. Comparison to other Work

Although human action recognition was widely investigated for 2d data, there is less work concerning the case of

having access to 3d data. A similar approach to classify human actions in 3d data was followed in [7], but they selected less action classes from the CMU MoCap dataset. While they distinguish only 3 action classes with variations in execution, recognition rates of 98.29% were obtained without taking the presence of noise into account. In [17], the same database has been used to create artificial 2d views and evaluating several distance metrics on the landmark points without modeling the shape at all. They observed recognition rates of about 90.5% in average when combining all their camera views for training and testing. The approach of [18] employed homography constraints and lead to an overall recognition rate of about 92%. Compared to those results, our approach performs similarly on the same data even in the presence of noise.

V. Summary and Outlook

We proposed a scheme to represent and classify human action sequences basing on dimensionality reducing projections. In contrast to other approaches, our approach employs the target coordinate system to represent the main directions of variance within a sequence of landmarks. For finding those projections we compared the usability of different techniques (*i.e. PCA*, *P-PCA*, *K-PCA*, *Isomap*, *NPE* and *SR*). It was pointed out that polynomial-kernelized PCA improves the linear PCA noticeably in experiments performed on the CMU MoCap dataset. We have also shown our approach's robustness in the presence of different kinds of noise. As our approach only uses linear operations and a fixed-length representation of the input data, it is able to perform in real-time. We also successfully applied our approach to various shape descriptors for 3d point clouds obtained by surface reconstruction from multi-view recordings.

To improve our results, the parameter vector $\boldsymbol{b}_{l'}$ could be used to build a self-similarity matrix instead of using the Euclidean landmark distances as proposed by [17]. Using recent real-time motion capture approaches [19], data delivered by depth imaging devices (*e.g.* Kinect) can directly be fed to our approach.

References

[1] A. Bottino, M. De Simone, and A. Laurentini, "Recognizing human motion using eigensequences," *WSCG*, vol. 15, 2007.

[2] M. Yamazaki, Y.-W. Yen-Wei Chen, and G. Xu, "Human action recognition using independent component analysis," in *Int. Symp. on Intel. Techn. in Comp. Games and Simul.*, Shiga, Japan, 2007.

[3] L. Wang and D. Suter, "Learning and matching of dynamic shape manifolds for human action recognition," *TIP*, vol. 16, no. 6, pp. 1646–1661, 2007.

[4] K. Jia and D.-Y. Yeung, "Human action recognition using local spatio-temporal discriminant embedding," in *CVPR*, 2008, pp. 1–8.

[5] M.-F. Sun, S.-J. Wang, X.-H. Liu, C.-C. Jia, and C.-G. Zhou, "Human action recognition using tensor principal component analysis," in *IEEE Int. Conf. on Comp. Science and Inf. Techn.*, Chengdu, China, 2011, pp. 487–491.

[6] L. A. Schwarz, D. Mateus, and N. Navab, "Recognizing multiple human activities and tracking full-body pose in unconstrained environments," *Pattern Recognition*, vol. 45, no. 1, pp. 11–23, 2012.

[7] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia, "Discriminative human action recognition in the learned hierarchical manifold space," *IVC*, vol. 28, no. 5, pp. 836–849, 2010, best of Automatic Face and Gesture Recognition 2008.

[8] C. W. Anderson and J. A. Bratman, "Translating thoughts into actions by finding patterns in brainwaves," in *14th Yale Workshop on Adaptive and Learning Systems*, 2008, pp. 1–6.

[9] M. Körner, D. Haase, and J. Denzler, "Scale-independent spatio-temporal statistical shape representations for 3d human action recognition," in *ICPRAM*, vol. 1, 2012, pp. 288–294.

[10] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[11] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," in *ICANN*, vol. 1327. Springer, 1997, pp. 583–588.

[12] T. F. Cox and M. Cox, *Multidimensional Scaling*, 2nd ed. UK: Chapman and Hall/CRC, 2000.

[13] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[14] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[15] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *ICCV*, vol. 2, 2005, pp. 1208–1213.

[16] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *ICCV*, 2007, pp. 1–8.

[17] I. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-independent action recognition from temporal self-similarities," *TPAMI*, vol. 33, no. 1, pp. 172–185, 2011.

[18] Y. Shen, N. Ashraf, and H. Foroosh, "Action recognition based on homography constraints," in *ICPR*, 2008, pp. 1–4.

[19] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, 2011, pp. 1297–1304.