Efficient Pose-based Action Recognition

Abdalrahman Eweiwi[1], Muhammed S. Cheema[1], Christian Bauckhage[1,3], Juergen Gall[2]

[1] *Bonn-Aachen International Center for IT, University of Bonn, Germany*
[2] *Computer Vision Group, University of Bonn, Germany*
[3] *Multimedia Pattern Recognition Group, Fraunhofer IAIS*

Abstract. Action recognition from 3d pose data has gained increasing attention since the data is readily available for depth or RGB-D videos. The most successful approaches so far perform an expensive feature selection or mining approach for training. In this work, we introduce an algorithm that is very efficient for training and testing. The main idea is that rich structured data like 3d pose does not require sophisticated feature modeling or learning. Instead, we reduce pose data over time to histograms of relative location, velocity, and their correlations and use partial least squares to learn a compact and discriminative representation from it. Despite of its efficiency, our approach achieves state-of-the-art accuracy on four different benchmarks. We further investigate differences of 2d and 3d pose data for action recognition.

1 Introduction

Human action recognition has recently drawn an increasing interest in computer vision owing to its applications in many fields including human computer interaction, surveillance and multimedia indexing. This interest has derived a rapid development in terms of the problem scale, the efficiency of the proposed algorithms, and even the data representations of human actions. Early approaches for action recognition used the human pose as a high level representation of actions and used joint trajectories for action and gait recognition [1, 2]. However, in these days, obtaining accurate measurements of body poses and joint locations required special setups that were often tedious and very expensive.

Consequently, efforts deviated toward alternative low and mid level representations of pose, motion, visual appearance, or particular combinations of them for better action models. For instance, [3, 4] rely majorly on motion cues to identify similar action sequences under static or moving camera setups. Others utilized the human appearance as the basic building blocks in discriminating actions [5–7]. The introduction of interest points in video sequences [8, 9] led towards a successful adaption of the bag-of-words model for human action recognition [10–12]. Despite encouraging results of low and mid level features for action recognition on several datasets, they suffer from variations of view point, subject, scale, and appearance. Moreover, they lack of a semantic meaning making

the interpretation of the results sometimes difficult. In contrast, high level representations (e.g. 3d pose-based) abstract away most variation factors and can provide a semantic interpretation of the results.

The recent advances in both depth sensors and human pose estimation have recently rekindled interest in high level human representations for action and behavior analysis [13]. Although current algorithms for pose estimation from monocular images [14], depth sensors [15], or multi-view setups [16] still have some limitations in terms of accuracy, several recent studies [16–20] point to the utility of pose estimation for improving the accuracy of action recognition systems. [17] utilize polar coordinates of joints in a sparse reconstruction framework to classify human actions in realistic video datasets. Their evaluation clarifies the implication of accurate pose estimation on action recognition and identifies the potential of current pose estimation approaches for improving action recognition. Similar observations are reported in [16, 18] on larger and more complex datasets. In particular, [18] showed that in some scenarios high-level features extracted by a current pose estimation algorithm [14] already outperform a state-of-the-art low level representation based on dense trajectories [11].

Some of the most successful approaches for action recognition from 3d pose data perform feature mining for training. For instance, [20] propose to learn a set of the most distinctive joints. While [21] weight poses of actions based on a mutual information criteria, [19] mine for most occurring temporal and spatial structures of body joints for classification. Mining meaningful poses [21], joints [20], or temporal and spatial joints structures [19], however, is usually time consuming.

In this work, we propose an algorithm for pose-based action recognition that is faster and more efficient for training and testing than existing works. Yet, it achieves on popular datasets for action recognition from 3d pose or RGB-D videos like [22, 23], state-of-the-art performance and outperforms other related pose-based approaches. The efficiency is achieved by simplicity in design. Each joint is modeled by a single feature vector that encodes only the most essential information to characterize an action: the relative location of the joint, the velocity of the joint, and the correlation between location and velocity. Inspired by [17], the information over a short video clip is encoded by histograms. Based on these features, a compact and discriminative representation is learned using partial least squares (PLS) [24–28]. The representation can then be used with any classifier like SVM or Kernel-PLS (KPLS) [29].

In our experimental evaluation, we show that for a high-level representation based on 3d pose an expensive training approach as in [19–21] is not necessary to achieve very accurate recognition results. We further investigate the performance of our approach for action recognition from 2d pose data. Since 2d pose data is ambiguous and not view-invariant, the performance drops in comparison to 3d pose data. However, given some training pairs of 2d pose and corresponding 3d pose, a mapping from 2d to 3d can be learned using a standard regression approach like Kernel partial least squares regression [29]. Although the regression does not provide very accurate 3d poses, our experiments show that features

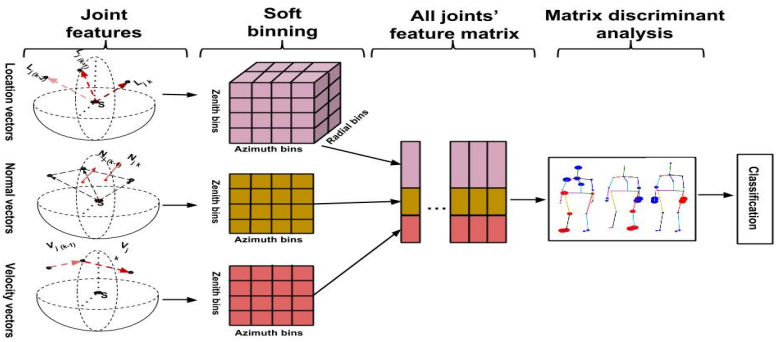


Fig. 1. Overview of the framework.

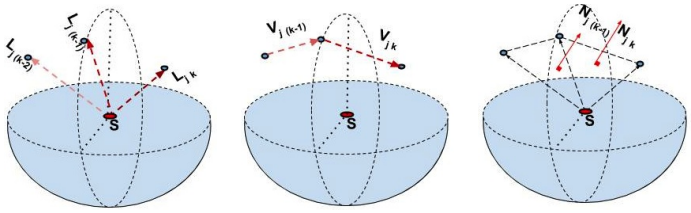
computed on the regressed 3d poses outperform the features computed from the 2d poses directly. This indicates that 3d pose estimation instead of 2d pose estimation from monocular videos has the potential to improve action recognition.

2 High Level Pose Representation

For representing actions by a high level pose-based representation, a sequence of extracted 2d or 3d pose per frame is given. In order to be flexible and learn the importance of a single joint, our representation consists of a feature for each joint as depicted in Figure 1. Each joint feature, which are discussed in Section 2.1 in more detail, models the distributions of the locations, velocities, and geometric orientation of the movements within a video clip or fixed number of frames as histograms. The histograms for each joint are then concatenated to build the feature matrix and matrix discriminant analysis is performed to obtain a set of of discriminant eigenvectors, which are used as high-level representation of the video clip. The representation can then be used with any classifier for classification.

2.1 Joint Features

To increase the robustness of the features to variations caused by different body shapes or even foreshortening in case of 2d pose, the relative joint positions and other vectors are converted into a spherical coordinate system. 2d vectors from 2d poses are represented by the length r and the orientation angle $\theta \in [0, 360]$. For a 3d skeleton representation, the horizontal orientation or azimuth $\alpha \in [0, 360]$ and the vertical orientation or zenith $\phi \in [0, 180]$ are used. A vector $v = (x, y, z) \in \mathcal{R}^3$ is then converted into spherical coordinates (r, α, ϕ) by:



(a) Location vectors (b) Velocity vectors (c) Normal vectors

Fig. 2. Illustration of the locations feature f_l, velocities feature f_v, and the normals feature f_n for a single joint j. For each frame k or frame pair $(k, k+1)$, the vectors l_{jk}, v_{jk}, and n_{jk} are converted into spherical coordinates and added to a histogram as shown in Figure 1.

$$r = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

$$\phi = \frac{180}{\pi} \times \left(\arccos\left(\frac{z}{r}\right)\right) \tag{2}$$

$$\alpha = \frac{180}{\pi} \times (\text{atan2}(y, x) + \pi) \tag{3}$$

Using spherical coordinates, we use three features that represent distributions over a fixed set of K frames as 3d or 2d histograms. For each feature, we indicate if it applies to 2d and 3d poses or both:

Joint locations feature f_l (2d and 3d): The f_l features resemble their 2d counterparts presented in [17]. But with 3d skeletons, our representation includes the azimuth α and zenith ϕ angles along with the joint displacement r from a reference point. For each sequence, we establish a local coordinate system whose origin is located at the *spine* s, which naturally corresponds to the center of the body. For a given location x_{jk} of a joint j at frame k, we quantize the polar coordinates (r, α, ϕ) of the joint location vector $l_{jk} = x_{jk} - s$ into a 3d histogram $(R \times O_{lv} \times O_{lh})$, where R, O_{lv}, O_{lh} are the number of bins for radius, vertical, and horizontal angle. The location vectors of all frames but of a single joint are accumulated in a single 3d histogram. The joint location vectors for three frames and one joint are illustrated in Figure 2 (a). Thus, the locations feature f_l consists of J 3d histograms, where J is the number of joints. In case of 2d pose, the histograms are 2d corresponding to the 2d coordinates (r, θ) for each 2d vector.

Joint velocities feature f_v (2d and 3d): The joint locations features do not encode any temporal information, which is important for classifying actions. Given the locations of a joint j at successive frames l_{jk} and $l_{j(k+1)}$, we convert the velocity vector $v_{jk} = l_{j(k+1)} - l_{jk}$ into spherical coordinates without radius

(α, ϕ). The radius is not taken into account to be invariant to different execution speeds of an action among subjects. The velocity vectors for all $K-1$ frame pairs are then added to the 2d histogram $O_{vv} \times O_{vh}$, where O_{vv} and O_{vh} are the numbers of bins for vertical and horizontal angle. The velocity vectors for two frame pairs are illustrated in Figure 2 (b). The velocities feature f_l therefore consists of J 2d histograms. The features f_l are in many cases complimentary to the f_v features. While f_v captures the velocity distributions of all joints, f_l captures the location distributions of all joints.

Joint movement normals feature f_n (3d only): The joint movement normals feature models the correlation of location and velocity, which corresponds to the cross product between the location vector l_{jk} and the velocity vector v_{jk} or the cross product of the locations of two consecutive frames as $n_{jk} = l_{jk} \times l_{j(k+1)}$. Up to a scaling factor, n_{jk} corresponds to the normal of the plane spanned by the three points s and the joint positions at the two frames k and $k+1$. Since the length of the normal vector is anyway one, we convert n_{jk} into spherical coordinates (α, ϕ) without r. The normals of the $K-1$ frames are quantized as the velocities feature into a 2d histogram and we obtain J 2d histograms for f_n. The movement normals for two frame pairs are illustrated in Figure 2 (c). All three features model only the most essential information to characterize an action: the relative locations of the joints, the velocities of the joints, and the correlations between locations and velocities. However, combined with a discriminative approach to learn a basis for the features, which is detailed in Section 2.2, we achieve state-of-the-art performance and outperform features that are much more expensive to compute.

Normalization and soft-binning To reduce any binning artifacts and to be more robust against style variations, we perform soft-binning. This is achieved by adding a quantized vector to all neighboring bins. The weights for the bins are given by a Gaussian kernel with $\sigma = 1$. To handle sequences of different length, the histograms are normalized by the L2-norm.

Temporal pyramid In addition, a temporal pyramid can be used. Instead of having a single histogram per video clip, it can be subdivided into smaller temporal segments. Since the videos in the datasets are short, we use a pyramid with only two layers. The second layer divides the video in three equally sized parts. The three histograms of the second layer and the histogram of the first layer are then concatenated.

2.2 Learning Discriminative Action Features

Not all joints have the same importance for action recognition as illustrated in Figure 3. It is therefore important to learn a compact and discriminative representation for action recognition. Since we have defined the features per

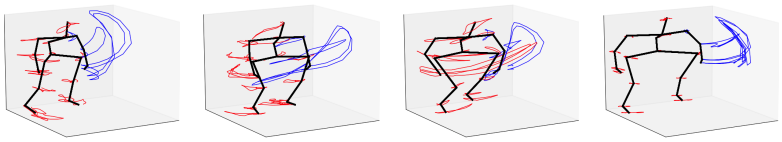


Fig. 3. Examples of joint trajectories for the *hammering* action from *MSR-Action3D* dataset [22]. The action can be well described by the trajectories of the left arm (blue), while the other joints (red) are less relevant. The trajectories also show the variations among subjects in the dataset.

joint, we can define a pose feature $\mathbf{f_p} \in \mathbb{R}^D$ as a weighted sum of all joint features $\{f_j \in \mathbb{R}^D\}_{j=1}^J$:

$$\mathbf{f_p} = \sum_{j=1}^{J} w_j f_j, \tag{4}$$

which can be expressed in matrix form as:

$$\mathbf{f_p} = \mathbf{F}\mathbf{w}, \tag{5}$$

where columns of $\mathbf{F} \in \mathbb{R}^{(D \times J)}$ corresponds to the joint features as illustrated in Figure 1 and $\mathbf{w} \in \mathcal{R}^J$ defines their corresponding weights. The weights $\mathbf{w}$ can be learned by partial least squares (PLS) [24], which has been recently adopted in computer vision for different applications including pose estimation and regression [25], image classification [26], pedestrian detection [27], and multi-view learning [28].

Given M training samples $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^M$ where $\mathbf{x}_i \in X$ and $\mathbf{y}_i \in Y$, PLS learns two linear projections $s_i = \mathbf{w}^T(\mathbf{x}_i - \overline{\mathbf{x}})$ and $t_i = \mathbf{v}^T(\mathbf{y}_i - \overline{\mathbf{y}})$ that maximize the sampling covariance between $\mathcal{X} = \{x_i\}_i$ and $\mathcal{Y} = \{y_i\}_i$ [26]:

$$\max \left\{ \frac{\left(\frac{1}{M}\sum_i s_i t_i\right)^2}{(\mathbf{w}^T\mathbf{w})(\mathbf{v}^T\mathbf{v})} \right\}, \tag{6}$$

where $\overline{\mathbf{x}}$ and $\overline{\mathbf{y}}$ are the corresponding means. When $\mathcal{Y}$ contains only class labels as in our case, $\mathbf{v}$ is not relevant and only $\mathbf{w}$ is estimated, which is equivalent to solving an eigenvalue problem [24, 26]:

$$\mathbf{\Sigma_b}\mathbf{w}^* = \lambda \mathbf{w}^*. \tag{7}$$

In our case, $\mathbf{\Sigma_b}$ is given by

$$\mathbf{\Sigma_b} = \frac{1}{M}\sum_{k=1}^{K} M_k \left[(\overline{\mathbf{F_k}} - \overline{\mathbf{F}})\right]^T \left[(\overline{\mathbf{F_k}} - \overline{\mathbf{F}})\right], \tag{8}$$

where M_k denotes the videos for class k, $\overline{\mathbf{F}}$ the mean feature matrix, and $\overline{\mathbf{F_k}}$ the mean feature matrix for class k. For the P largest eigenvalues λ, we use the

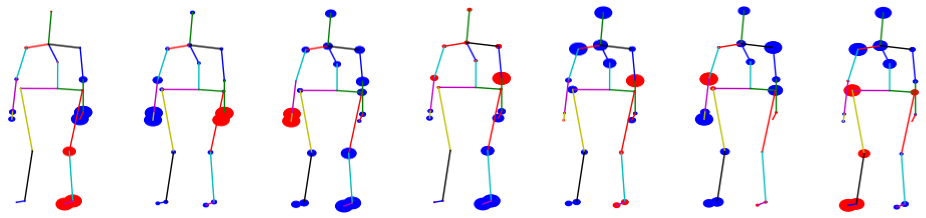


Fig. 4. The first 7 discriminative projections of joint features extracted using PLS from *MSR-Action3d*. Blue indicates positive weights and red negative weights for $\mathbf{w}_1, \ldots, \mathbf{w}_7$. Notice that only few part combinations in this dataset are relevant while some joints like the hips are irrelevant for human actions, which is indicated by the small size of of the joints.

corresponding eigenvectors $\mathbf{w}$ as representation. Hence, the final feature $\mathbf{f_p}$ of a video sample i is given by projecting its feature matrix $\mathbf{F_i}$ to the learned P largest projections as $\mathbf{f_p} = [(\mathbf{F_i w}_1)^T (\mathbf{F_i w}_2)^T \cdots, (\mathbf{F_i w}_P)^T]^T$ where $\mathbf{f_p} \in \mathbb{R}^{D*P}$.

Figure 4 depicts the first seven eigenvectors learned using PLS on the *MSR-Action3D*. Notice that most of the eigenvectors focus on joints that are relevant and can discriminate between the performed actions. So in this dataset, only a few body part combinations are relevant where some joints like the hips are irrelevant for the human actions, which is indicated by the small size of the joints.

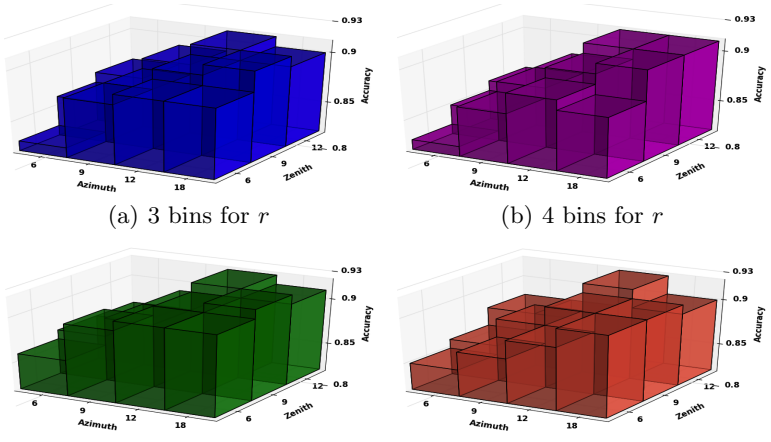
2.3 Classification

The obtained action features $\mathbf{f_p}$ can be classified using any off-the-shelf classifier like SVM. In our experiments, we use a non-linear classifier based on PLS, namely Kernel-PLS (KPLS) [27, 29]. As training data, we have for each video clip the label and the feature vector $\mathbf{f_p}$ which are transformed so that all its entries are positive. While the features define the set $\mathcal{X}$, the class labels are encoded by the set $\mathcal{Y}$. As kernel, we use the intersection kernel defined as $K_{i,j} = \sum_l \min\left(\mathbf{f_{p_i}}(l), \mathbf{f_{p_j}}(l)\right)$.

3 Datasets and Experiments

We choose four challenging datasets to evaluate our approach for human action recognition. The datasets are *MSR-Action3D* [22], *3D Action Pairs* [23], *MSR-DailyActivity3D*[1], and *TUM Kitchen* dataset [30]. For all the experiments in the following sections, we used the same parameters (number of bins) to construct our pose features. Empirically, we evaluated the impact of feature quantization and measured the average classification accuracy over three different splits only for the *MSR-Action3D* dataset for various quantizations of length r, azimuth α,

[1] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/



(a) 3 bins for r (b) 4 bins for r

(c) 5 bins for r (d) 6 bins for r

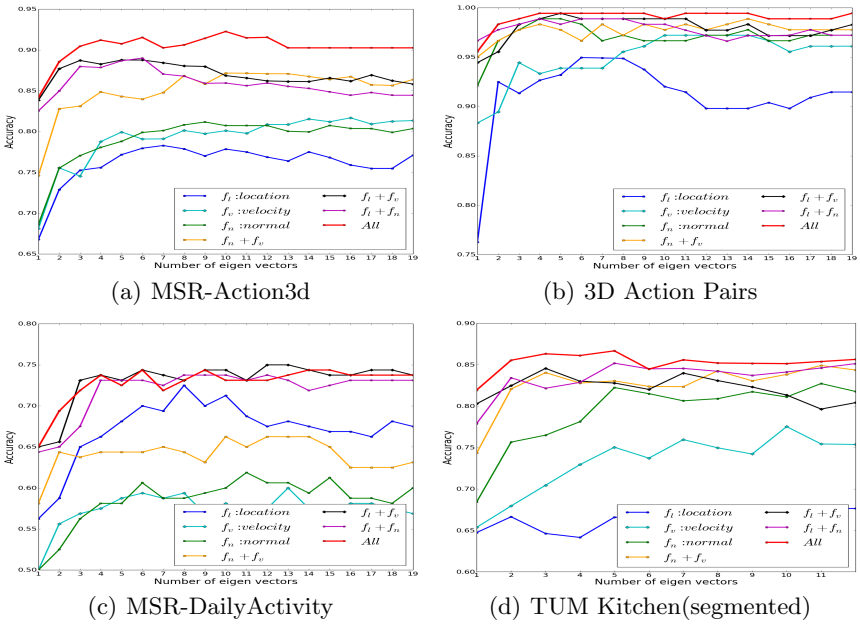
Fig. 5. Recognition accuracy for different feature quantizations for length r, azimuth α, and zenith ϕ. The plots show the accuracy when the number of bins changes. There are several configurations that give a good performance. Among them we use 5, 18, and 9 as the number of bins for length, azimuth, and zenith, respectively.

and zenith ϕ of the joint locations, joint velocities, and joint movement normals. The results are shown in Figure 5. While several configurations give a good performance, we chose 5, 18, and 9 as the number of bins for length, azimuth, and zenith, respectively. Our experiments use these configurations for feature extraction on all pose datasets. For all experiments, we learn the classifier parameters using 5-fold cross validation. This also includes the number of eigenvectors.

3.1 MSR-Action3D

The MSR-Action3d dataset is an action dataset captured with a RGB-D camera and designated for gaming-like interactions. It consists of 567 temporally segmented action sequences and contains 20 actions, each performed $2-3$ times by 10 different subjects. The actions are: *high-arm-wave, horizontal-arm-wave, hammer, hand-catch, forward-punch, high-throw, draw-x, draw-tick, draw-circle, hand-clap, two-hand-wave, side-boxing, bend, forward-kick, side-kick, jogging, tennis-swing, tennis-serve, golf-swing, pick-up and throw.* We exclude 10 sequences as in [20] and operate on the X,Y screen coordinates along with their corresponding depth.

For evaluation, we follow the work in [20, 23] and consider two evaluation tasks: (i) The cross-subject setup where we train our model using the actions of subjects 1,3,5,7,9 and report the results on the rest [20]. (ii) The second task reports the system performance on the average accuracy on all **252 (5-5)** cross-subject splits [23]. Using the first task, Figure 6 (a) shows the individual contribution of each joint feature with respect to the number of projection vectors obtained by PLS. The combinations of the three features f_l, f_v, and f_n,



(a) MSR-Action3d (b) 3D Action Pairs

(c) MSR-DailyActivity (d) TUM Kitchen(segmented)

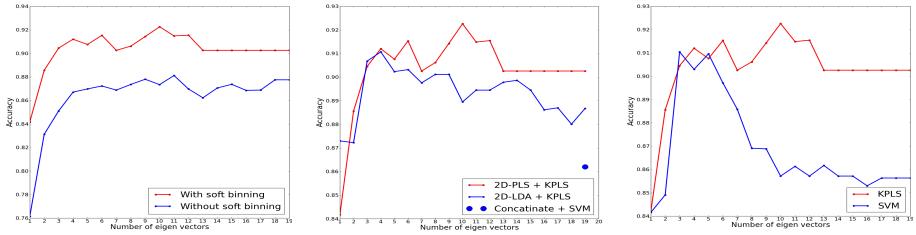
Fig. 6. Recognition accuracies for different numbers of eigenvectors and various feature combinations.

which capture joint location, velocity, and their correlation, clearly boost the performance in comparison to each single feature or feature pair. Using all three features and 10 PLS-projections, an accuracy of **92.3**% is achieved.

We further evaluated the impact of soft-binning in Figure 7 (a). Without soft-binning the descriptor is more sensitive to style variations and binning artifacts. Soft-binning therefore improves the results by a margin.

Figure 7 (b) compares PLS with linear discriminant analysis (LDA) [31, 32]. While PLS relies only on the between-class covariance matrix [26], which results in (8), LDA also takes the intra-class covariances for each class into account. In practice, however, the matrix based on intra-class covariances can be often singular, specially in cases where the number of training samples is less than the feature dimension. This can be observed in Figure 7 (b) where the performance drops when the number of eigenvectors increases. In contrast, PLS does not suffer from singularities. However, both approaches perform better than a baseline that concatenates the joint features described in Section 2.1 without learning a compact representation as described in Section 2.2 and that uses a SVM for classification. For the SVM and KPLS, we use an intersection kernel. Figure 7 (c) also compares KPLS and SVM using our representation described in Section 2.2 for a varying number of eigenvectors.

Table 1 compares our approach with the state-of-the-art on this dataset. In this case, the number of eigenvectors is estimated on the training data by 5-



(a) (b) (c)

Fig. 7. Performance evaluation on *MSR-Action3D* dataset. (a) Impact of soft binning (b) Comparison of LDA and PLS (c) Comparison of KPLS and SVM classifiers

Table 1. Recognition accuracy for *MSR-Action3D* dataset. The methods use different data modalities where **S** denotes skeleton data and **D** depth. **TP** denotes the use of a temporal pyramid.

Method	[33]	[20]	[34]	[23]	[35]	[21]	[19]	[36]	**Ours**	**Ours(TP)**
Modality	**D+S**	**D+S**	**D**	**D**	**D**	**S**	**S**	**S**	**S**	**S**
Accuracy(%)	92.67	88.2	86.5	88.36	89.3	91.7	90.22	89.48	**91.5**	**90.1**

fold cross validation. Our approach achieves an accuracy of **91.5**%. It performs comparable to the state-of-the-art [33] and performs better than most other skeleton-based approaches. The temporal pyramid does not improve the results since the dataset contains short, well-defined actions where temporal invariance is beneficial. We verify further the robustness of our features against different subjects by evaluating our features on all **252 (5-5)** possible splits. In this task we achieved a mean accuracy of **88.38**% and standard deviation of **0.027** compared to **82.15**%±**4.18** in [23] establishing our method's robustness against cross-subject variations for human action recognition.

The training and testing time on the *MSR-Action3D* standard split is 27 and 14 seconds, respectively. More precisely, the classification time required for a video clip comprising 55 frames is 161ms where the feature extraction step takes 148ms. The approach [21] provides comparable results in terms of classification time, however, the training time is much more expensive since each frame is classified by a kNN classifier. We also compared with the recent approach [36], which uses dynamic time warping and requires many mappings between Lie group and tangential space. Using the provided source code [36], classification of a single video clip of 58 frames requires around 20 seconds. All the experiments were conducted on an Intel Core i7 CPU with 3.40GHz and 8Gbyte RAM. This shows that our approach is both very efficient for training and testing.

3.2 3D Action Pairs Dataset

This dataset emphasizes on particular scenarios where motion and shape cues are highly correlated. It comprise of six pairs of actions, such that within each

Table 2. Recognition accuracy for 3D Action Pairs. The methods use different data modalities where **S** denotes skeleton data and **D** depth. **TP** denotes the use of a temporal pyramid.

Method	MMTW [33]	Actionlets [20]	HON4D[23]	**Ours**	**Ours(TP)**
Modality	**D+S**	**D+S**	**D**	**S**	**S**
Accuracy(%)	97.22	82.22	96.67	**92.0**	**99.4**

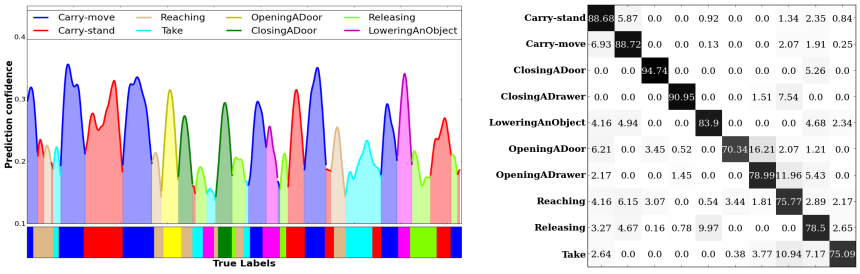
pair the motion and the shape cues are similar, but their temporal correlations vary. The action pairs are: *Pick up a box/Put down a chair, Lift a box/Place a box, Push a chair/Pull a chair, Wear a hat/Take off hat, Put on a backpack/Take off a backpack, and Stick a poster/Remove a poster.* We evaluate our framework using the same cross-subject evaluation protocol as in *MSR-Action3D*.

Figure 6 (b) shows the individual performance of each feature for different projections. For the datasets, the correlation features f_n outperform the location and velocities features since they capture temporal-spatial correlations of the action classes better. The best performance is, however, achieved when all features are used.

We compare our approach with the state-of-the-art on this dataset in Table 2. As for the other dataset, the number of eigenvectors is estimated on the training data by 5-fold cross validation. Our algorithm achieves **92.0**%. When a temporal pyramid is used, it achieves **99.4**% and outperforms the other methods. The performance boost of the pyramid can be explained by the classes. These are activities that consist of smaller sub-actions in a specific order, which can be well captured by the temporal pyramid.

3.3 MSRDailyActivity

This dataset has been captured with an RGB-D camera to mimic daily human activities in a living room. There are 16 different actions, each performed by 10 subjects twice, once standing and the other while sitting. The actions are: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lie down on sofa, walk, play guitar, stand up, sit down.* The standard task for this dataset aims at cross subject evaluation as in *MSR-Action3D*, where we train on the odd numbered subjects and test on the rest. Figure 6 (c) shows the individual accuracies of the different features. Unlike *MSR-Action3D* and *3D Action Pairs* datasets, the joints location feature (f_l) in this dataset outperforms both velocity and normal features. This is because many actions in this dataset are of static or merely static nature (e.g. *call cellphone*, *play game*, *use laptop*). However, our combined features outperform the individual features and achieve an overall accuracy of **70.0**%. With a temporal pyramid, the accuracy is further improved to **73.1**% accuracy. Compared to previous work [20], our method outperforms their results of **68.0**% by **5.1**%.



(a) (b)

Fig. 8. Evaluation results on the *TUM Kitchen* dataset (a) Sample frame-level prediction where the x-axis shows the time span of the video sequence with ground-truth annotations and the y-axis shows the predicted class label with confidences. (b) Confusion matrix of the unsegmented video sequence of the *TUM Kitchen* dataset (Best viewed in colors)

3.4 TUM Kitchen Dataset

The *TUM kitchen* dataset focuses on a home-monitoring scenario using a multi-view camera setup (4 cameras). The dataset provides 3d human pose data estimated by a markerless full-body tracker. Our evaluation criteria considers two tasks: (i) *segmented* test data and (ii) *unsegmented* test data as in [16]. On both tasks, we used the episodes 0-2,0-8,0-4,0-6,0-10,0-11,1-6 for testing and the remaining 13 for training. However, in the first task we assume that the videos are already segmented while in the second task we perform continuous classification. The evaluation criteria for the unsegmented case follows the protocol described in [16], where the average class accuracies is measured on a frame-level. We use the skeleton with 13 joints for evaluation and do not count the errors at the transition frames between annotations with a margin of 4 frames on both sides as in [16]. For the first task, Figure 6 (d) presents a detailed overview of the average recognition accuracies over all classes for each feature along with their combination. Our algorithm achieves for this task an average accuracy of **86.65**% over all classes.

On the second task, we evaluated the performance of our approach using a fixed sliding window of 30 frames that was determined empirically. This task is more challenging as the dataset stands for actions of arbitrary time stamps ranging from 10 to 150 frames. The evaluation considers the average accuracy on frame level over all classes. Our algorithm achieves an average accuracy of **82.5**% as compared to **80.03**% in [16]. Figure 8 (a) depicts the prediction of our model for an unsegmented action sequence from the *TUM* dataset. While Figure 8 (b) shows the confusion matrix for all classes.

3.5 2D vs. 3D Pose

Recent advances in pose estimation introduce new opportunities towards action recognition in challenging environments. For example, several applications in

[16–18] show that pose estimation is vital towards generalizable, robust, and efficient action recognition. However, most estimation methods reconstruct 2d poses from monocular views, failing to provide view-invariant descriptors for action recognition.

Table 3. Recognition accuracy (%) for the *TUM* dataset. We compare a 2d appearance based approach [16], 2d versions of our features, 3d features obtained by mapping the 2d pose to 3d, and 3d features computed from the provided 3d poses, which have been estimated using all four camera views.

Camera	Camera 1	Camera 2	Camera 3	Camera 4
HF + 2d appearance features [16]	68.00	70.00	68.00	65.00
KPLS + joint features from 2d pose	65.66	65.19	63.95	62.51
KPLS + joint features from 3d pose estimated from 2d pose of one camera view	77.61	77.78	78.23	78.47
KPLS + joint features from 3d pose estimated from all camera views	82.5			

We therefore compare action recognition using 3d pose features and 2d pose features. To this end, we project the 3d pose of each frame to the 4 camera views and perform action recognition with our 2d pose features as described in Section 2.1. The evaluation protocol is the same as for the unsegmented sequences of *TUM*.

Table 3 shows the classification accuracies obtained for each of the four camera views. As opposed to the previously reported result for 3d pose features, action recognition accuracies in 2d show a significant drop of almost **20%** in recognition rates on all 4 cameras. The performance is also lower than the one reported for the 2d appearance approach [16], which does not use high-level features but low-level features based on optical flow and gradients.

In order to investigate if the performance loss comes from the inherent depth ambiguity of 2d poses or the view sensitiveness of the representation based on 2d poses, we learn a mapping from 2d to 3d pose by non-linear regression. Given a set of 2d training poses of J joints $\mathbf{x} \in \mathbb{R}^{2*J}$ and their corresponding 3d poses $\mathbf{y} \in \mathbb{R}^{3*J}$, we linearly scale the individual body parts so that the distance between the central shoulder and the central hip joint is constant. Then we learn a mapping function $\Phi : \mathcal{R}^{2*J} \rightarrow \mathcal{R}^{3*J}$ that maps the observed 2d poses of one camera view to their 3d representation. The non-linear regression is implemented by KPLS [29] with a radial basis kernel, where $\mathcal{X} = \mathbb{R}^{2*J}$ and $\mathcal{Y} = \mathbb{R}^{3*J}$. The bandwidth of the kernel is $\sigma = 0.01$.

Instead of encoding 2d poses by a discriminative representation learned from 2d pose features, we can also predict for a single camera and each frame the 3d pose from the observed 2d pose. The predicted 3d poses are then used for learning the discriminative representation from 3d pose features as described in Section 2.2. Table 3 compares the obtained accuracies. Since the predicted

3d poses from a single camera are not as accurate as the 3d poses provided by the dataset, which have been estimated by a multi-view human pose estimation algorithm, also the recognition accuracy with the features computed from the predicted poses is lower. However, the representation based on the predicted 3d poses shows a significant performance boost over the corresponding representation based on 2d poses. It is also interesting to note that the performance is around 78% for all views while the 2d features show more performance variation among views. Furthermore, the 2d appearance-based approach of [16] is outperformed. This result underlines the benefit of view-invariant pose features and indicates that 3d pose estimation instead of 2d pose estimation from monocular videos has the potential to improve action recognition.

4 Conclusion

We have presented a framework for action recognition from 2d and 3d poses. The approach is very efficient for training and testing and achieves state-of-the-art performances on several datasets. This has been achieved by focusing on the the most essential information that characterizes an action, namely the relative locations of the joints, the velocities of the joints, and the correlations between locations and velocities denoted as movement normals. Together with a discriminative approach to learn a basis for the features, we obtain an action representation that outperforms other representations that are much more expensive to compute. We finally compared 2d and 3d pose features and conclude that learning a mapping from 2d pose to 3d pose to obtain view-invariant features can boost the performance significantly.

Acknowledgment. This work was carried out in the project automatic activity recognition in large image databases which is funded by the German Research Foundation (DFG). The authors would also like to acknowledge the financial support provided by the DFG Emmy Noether program (GA 1927/1-1).

References

1. Campbell, L., Bobick, A.: Recognition of human body motion using phase space constraints. In: ICCV. (1995)
2. Bissacco, A., Chiuso, A., Ma, Y., Soatto, S.: Recognition of human gaits. In: CVPR. (2001)
3. Wu, S., Oreifej, O., Shah, M.: Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In: ICCV. (2011)
4. Efros, A., Berg., A., Mori, G., Malik, J.: Recognizing action at a distance. In: CVPR. (2003)
5. Thurau, C., Hlavac, V.: Pose primitive based human action recognition in videos or still images. In: CVPR. (2008)
6. Ikizler-Cinbis, N., Cinbis, R., Sclaroff, S.: Learning actions from the web. In: ICCV. (2009)

7. Eweiwi, A., Cheema, M., Bauckhage, C.: Discriminative joint non-negative matrix factorization for human action classification. In: GCPR. (2013)
8. Laptev, I.: On space-time interest points. IJCV **64** (2005) 107–123
9. Willems, G., Tuytelaars, T., Van Gool, L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: ECCV. (2008)
10. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR. (2008)
11. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision **103** (2013) 60–79
12. Xia, L., Aggarwal, J.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: CVPR. (2013)
13. Ye, M., Zhang, Q., Wang, L., Zhu, J., Yang, R., Gall, J.: A survey on human motion analysis from depth data. In: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications. Springer Berlin Heidelberg (2013) 149–187
14. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013) 2878–2890
15. Shotton, J., Girshick, R.B., Fitzgibbon, A.W., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., Blake, A.: Efficient human pose estimation from single depth images. IEEE Trans. Pattern Anal. Mach. Intell. **35** (2013) 2821–2840
16. Yao, A., Gall, J., van Gool, L.: Coupled action recognition and pose estimation from multiple views. International Journal of Computer Vision **100** (2012) 16–37
17. K. Tran, I.K., Shah, S.: Modeling motion of body parts for action recognition. In: BMVC. (2011)
18. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.: Towards understanding action recognition. In: ICCV. (2013)
19. Wang, C., Wang, Y., Yuille, A.: An approach to pose-based action recognition. In: CVPR. (2013)
20. Wang, J., Liu, Z., Liu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: CVPR. (2012)
21. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: ICCV. (2013)
22. Wanqing, L., Zhengyou, Z., Zicheng, L.: Action recognition based on a bag of 3d points. In: CVPRW. (2010)
23. Oreifej, O., Liu, Z.: Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: CVPR. (2013)
24. Barker, M., Rayens, W.: Partial least squares for discrimination. J. Chemometrics **17** (2003) 166173
25. Hajd, M.A., Gonzlez, J., Davis, L.: On partial least squares in head pose estimation: How to simultaneously deal with misalignment. In: CVPR. (2012)
26. Harada, T., Ushiku, Y., Yamashita, Y., Kuniyoshi, Y.: Discriminative spatial pyramid. In: CVPR. (2011)
27. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. In: ICCV. (2009)
28. Sharma, A., Jacobs, D.: Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In: CVPR. (2011)
29. Rosipal, R., Be, P.P., Trejo, L.J., Cristianini, N., Shawe-Taylor, J., Williamson, B.: Kernel partial least squares regression in reproducing kernel hilbert space. JMLR **2** (2001) 97–123

30. Tenorth, M., Bandouch, J., Beetz, M.: The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition. In: ICCV Workshops. (2009)
31. Li, M., Yuan, B.: 2d-lda: A statistical linear discriminant analysis for image matrix. Pattern Recognition Letters **26** (2005) 527 – 532
32. Bauckhage, C., Käster, T.: Benefits of separable, multilinear discriminant classification. In: ICPR. (2006)
33. Wang, J., Wu, Y.: Learning maximum margin temporal warping for action recognition. In: ICCV. (2013)
34. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3d action recognition with random occupancy patterns. In: ECCV. (2012)
35. Xia, L., Aggarwal, J.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: CVPR. (2013)
36. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: CVPR. (2014)