3D Pose from Motion for Cross-view Action Recognition via Non-linear Circulant Temporal Encoding

Ankur Gupta*, Julieta Martinez*, James J. Little, Robert J. Woodham
University of British Columbia
{ankgupta, julm, little, woodham}@cs.ubc.ca

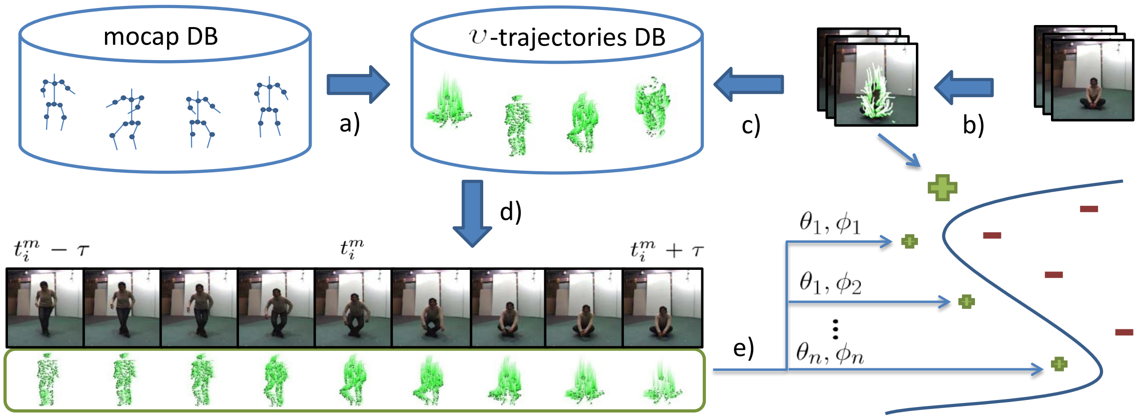


Figure 1: Our novel approach to cross-view action recognition. a) We begin with a large collection of unlabelled mocap data and synthesize fixed length 2D trajectories using orthographic projections (called υ-trajectories). b) Similarly, we generate dense trajectories for training videos. c) Then, we match the video trajectories to the υ-trajectory database using Non-linear Circular Temporary Encoding to achieve alignment in both time and pose. d) An alignment for an example is shown. e) We use the aligned mocap sequence to synthesize multi-view training data (with different azimuthal ϕ and polar θ angles) for the same action class as the original video. This additional data acts as a means to transfer knowledge across views.

Abstract

We describe a new approach to transfer knowledge across views for action recognition by using examples from a large collection of unlabelled mocap data. We achieve this by directly matching purely motion based features from videos to mocap. Our approach recovers 3D pose sequences without performing any body part tracking. We use these matches to generate multiple motion projections and thus add view invariance to our action recognition model.

We also introduce a closed form solution for approximate non-linear Circulant Temporal Encoding (nCTE), which allows us to efficiently perform the matches in the frequency domain. We test our approach on the challenging unsupervised modality of the IXMAS dataset, and use publicly available motion capture data for matching. Without any additional annotation effort, we are able to significantly outperform the current state of the art.

1. Introduction

We focus on the problem of action recognition from a novel viewpoint: given labelled training data from a particular camera angle (training view), we demonstrate an approach that recognizes the same action observed from a different camera view (test view) without having access to any test view data or labels during training time. In our case, this viewpoint-invariance comes at no additional labelling cost: a large collection of unlabelled, publicly available mocap data is used as is.

Human motion, as observed in video sequences, is a 2D projection of a highly articulated and agile human body. The change in the relative position of the subject and the camera alters these projections considerably. This is why under mild to extreme camera angle changes the performance of state-of-the-art action recognition techniques drops considerably, especially when no training examples in the test views are available (see Figure 2 in [14]). This motivates the development of techniques that add viewpoint-invariance to action recognition models, ideally without

*Indicates equal contribution.

having access to the test data during training time, which is the hardest but also more realistic scenario.

Approaches to cross-view action recognition range from geometry-based techniques [16, 18, 20] to purely statistical approaches [6, 14, 15, 34]. Geometry-based techniques reason about 2D or 3D body part configurations but often require robust pose estimation at each frame. Another possibility is to search for a transformation that aligns the observed video to a large number of view-dependent motion descriptions, or 3D motion models [3, 30]. However, to generate these descriptors labelled multi-view video data is needed in the first place.

On the other hand, more general statistical techniques can be applied to model the viewpoint changes of local feature-based representations. The goal is to find a transformation in feature-space that makes different views comparable [6, 14, 15, 34]. Unfortunately, these approaches are either not applicable or perform poorly in the case when no supervision is available for knowledge transfer [14, 34].

In this work, we address all the limitations mentioned above. We propose a novel approach that does not require exact tracking nor detection of body parts, but rather matches videos with a database of mocap sequence projections. We use trajectory features, which can be easily generated from mocap as well as from videos, thus we also avoid the need for labelled multi-view video data that search-based approaches require. Later, we use the mocap matches to add view invariance to our model. Moreover, all these advantages are obtained at no additional labelling cost.

We present three main contributions. First, we introduce υ-trajectories, a simple, computationally inexpensive descriptor for mocap data that can be compared directly and efficiently with dense trajectories obtained from video sequences. Second, we derive an approximation to Circulant Temporal Encoding (CTE) that relaxes its linearity assumption. We use this Non-linear CTE to match human activities in videos to mocap examples, which allows us to estimate 3D pose from 2D videos directly. Third, we further combine these two ideas to automatically transfer knowledge across views in cross-view action recognition.

The rest of this paper is organized as follows: in Section 2 we discuss related work, Section 3 contains the details of the proposed approach, we present our experimental evaluations on the IXMAS dataset in Section 4, and Section 5 gives further discussion and outlines future work.

2. Background and Related Work

This works touches upon a number of different areas in computer vision. We briefly discuss the ideas most relevant to our approach.

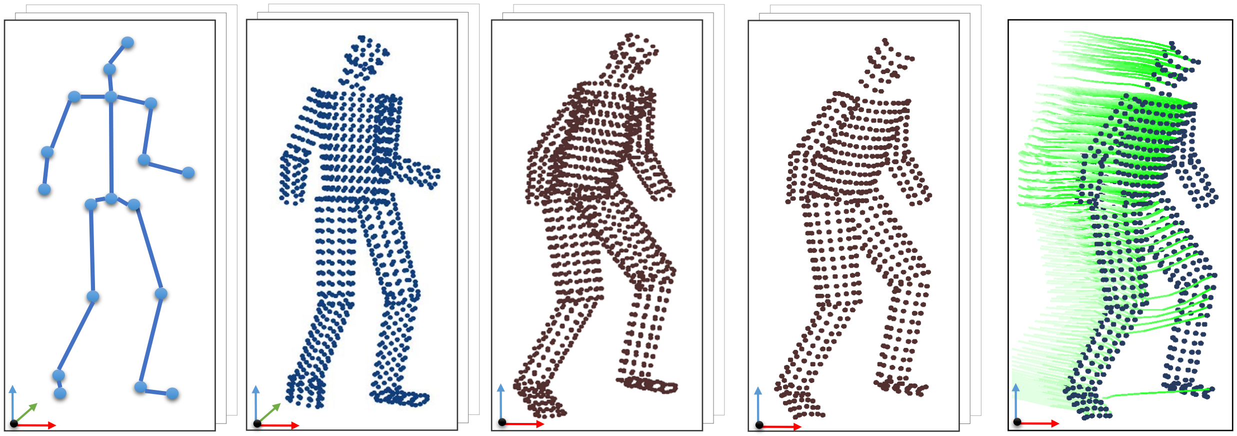
Pose from Monocular Video. The first part of our work deals with finding a 3D pose sequence from a monocular video of a single-actor. Due to the highly unconstrained nature of this problem, approaches in this field have focused on building statistical priors of motion models [29]. However, these approaches have the limitation of being action-specific. Andriluka *et al.* [2] track body parts exploiting recent advancements in 2D human pose detection and apply pose and motion priors to recover 3D pose in realistic videos. Other methods include using physics-based reasoning [4, 26] to resolve ambiguities and increase robustness.

By directly matching motion features in video to 2D projections of mocap data, we are able to avoid using any appearance information for body part tracking. Also, our non-parametric approach does not need to make any assumptions about a particular motion model. Moreover, for our application of cross-view action recognition we do not require the exact location of body parts to learn the action model. Rather, we are interested in finding similar actions in mocap examples.

View invariance for action recognition. Early efforts at view independence in action recognition were based on geometric reasoning. Parameswaran and Chellappa [16] propose Invariance Space Trajectories, a geometrical invariant that uses 5 points that lie in the same plane. Rao *et al.* [21] exploit the fact that discontinuities in motion trajectories are preserved across views. Both these methods assume that body joint trajectories are available. Another line of efforts focuses on engineering features that are inherently view independent. Junejo *et al.* [10] utilize self-similarity in motion that is preserved across views. Li *et al.* [13] show that dynamical properties of human motion can be used to generate a view-invariant representation. However, view-invariant features may lose discriminative information needed to distinguish different actions [32].

Lately, approaches based on transfer learning have gained a lot of attention. Farhadi *et al.* [6] transfer knowledge across views using random projections which are discriminative in both training and test views. A related method, the Bag of Bilingual Words (BoBW) [15], generates different dictionaries for each view and learns the mapping between words using correspondence labels. In both these cases, correspondence labels between views are needed. Li and Zickler [14] and more recently Zhang *et al.* [34] address this limitation by learning a virtual path between training and test view feature spaces in an unsupervised manner. These methods do not need labels in the test view, but still require access to the test view data at training time, which may not be available in certain realistic scenarios.

Another line of work that closely relates to our approach utilizes exemplar 3D poses or multi-view video data to model actions [3, 30]. Other recent methods include [33], where viewpoint is treated as a latent variable in a struc-



(a) Mocap sequence (b) Cylinder representation (c) Orthographic projection (d) Hidden point removal (e) ν-trajectories

Figure 2: υ-trajectories generation pipeline: (a) mocap data have a sequence of body joint locations over time. (b) We approximate human body shape on top of the joint locations using tapered cylinders to obtain a "Tin-man" model. (c) Sampled points on the surface of cylinders are projected under orthography for a fixed number of views and cleaned up using hidden point removal (d). (e) Connecting these points over a fixed time horizon, gives us the υ-trajectories.

tural SVM action classification framework, and [12], which builds a view-independent manifold for each action using non-linear dimensionality reduction. These approaches rely on the availability of annotated multi-view sequences, which limits their applicability. We want to relax this requirement by synthesizing the data required for training using unlabelled mocap examples.

Synthetic Data in Vision. The problem of collecting large realistic datasets to capture intra-class variations also occurs in human pose detection. Pishchulin *et al.* [19] render 3D human models on natural backgrounds to generate training data for joint detection and pose estimation using a small amount of real data. Shotton *et al.* [23] use synthetic depth images to expand their training set to learn body part labels from a single depth image. Another related technique, proposed by Chen and Grauman [5], uses unlabelled video data to generate examples for actions in still images.

Dense Trajectories. Dense trajectories have proven to be effective features for action recognition in the wild. First proposed as an auxiliary to visual descriptors [27], they were also found to perform well as local descriptors on their own [8, 27, 28]. In this work we introduce υ-trajectories, a version of dense trajectories that can be computed directly from mocap data. Previous approaches that have used mocap data for pose or action recognition usually require realistic rendering [19, 24], which attempts to approximate the visual richness of the real world. However, photo-realistic rendering is still a hard and computationally expensive problem. We avoid this by using entirely motion-based features. This has three advantages: first, it reduces the computational overhead compared to previous approaches, second, it allows us to compute trajectories at any arbitrary resolution (see Section 3.1), and third, it minimizes assumptions about visual appearance (*e.g.* clothing, lighting).

Non-linear Circulant Temporal Encoding. Also related to our work are large scale retrieval techniques. In this context, Revaud *et al.* recently proposed Circulant Temporal Encoding (CTE) [22], a method for large-scale event retrieval that incorporates previous findings on large-scale image search [9]. Furthermore, temporal consistency is enforced by interpreting filtered circular convolution as a Regularized Least Squares problem, and solving it efficiently in the frequency domain [7]. CTE assumes that individual frame descriptors can be compared with a linear kernel. While this works well for the VLAD descriptors used in [22], dense trajectories have been found to perform better when aggregated using BoW [8]. It is well known that BoW are better compared using the χ^2 distance [25]. Therefore, in Section 3.3, we derive an approximation to CTE that incorporates non-linear distance metrics.

3. Solution Methodology

We define υ-trajectories as an orthographic projection of curves generated by tracking points on the surface of a 3D model. In this section we show how to generate υ-trajectory features using mocap data. We then proceed to describe Non-linear CTE, which we use to align videos and mocap data. Finally, we also describe our action recognition pipeline based on data augmentation using the aligned 3D motion examples.

3.1. Generating υ-trajectories

Mocap sequences provide a series of body joint positions over time. We approximate body parts by fitting cylinders with bones (connections between body joints) as axes, and then putting a dense grid on the surface of each cylinder (see Figure 2 (b)).

Generating multiple projections. We project this 3D grid under orthography for a fixed number of viewpoints. With orthographic projection, there are only two parameters to vary: the azimuthal angle $\phi \in \Phi = \{0, \pi/3, 2\pi/3, pi, 4\pi/3, 5\pi/3\}$, and the polar or zenith angle $\theta \in \Theta = \{\pi/6, \pi/3, \pi/2\}$, as we assume that a camera looking up is unlikely. (see Figure 2 (c)). When computing trajectories from video sequences, making use of multiple spatial scales is essential to avoid scale artifacts [27]. However, since we do not create an image from the 3D model, we are not limited in spatial resolution.

Hidden point removal. We account for self-occlusions by removing points that should not be visible from a given viewpoint. First we perform back-face culling, and on the remaining points we use a freely available off-the-shelf implementation of the method by Katz *et al.*[1] [11]. This gives us a set of filtered points corresponding to each projection (see Figure 2 (d)).

Trajectory generation and postprocessing. We connect these filtered 2D points in time over a fixed horizon of $\mathcal{T}$ frames to obtain υ-trajectories. To make video and mocap data comparable, we make sure that the sampled frame rate for mocap is the same as for the videos used in the experiments, and use the same $\mathcal{T}$ when computing trajectories on videos. We use $\mathcal{T} = 15$, which has been found to work well in the past [27]. It was also found in [27] that denser trajectories increase the recognition accuracy. While in a video the trajectory density is inherently limited by the pixel resolution, with mocap data we can track points more densely. To balance the trade-off between accuracy and computational overhead, we compute 50 trajectories per frame. Figure 2 provides an overview of the υ-trajectory generation pipeline. Our code to generate υ-trajectories from mocap sequences is freely available online.[2]

3.2. Describing video inputs with dense trajectories

We describe our input videos with dense trajectories. We use a dense grid, computing trajectories every other pixel, as this increases accuracy at no additional memory cost after BoW aggregation [27]. We used the freely available implementation of Wang *et al.*[3]

3.3. Aligning video and mocap sequences via Non-linear CTE

The amount of synthesized υ-trajectory data scales linearly with the total length of the mocap sequences, as well as with the number of projections. This can result in a large database of feature descriptors that must be searched efficiently. For this reason, we extend Circulant Temporal Encoding [22], earlier applied to large scale video retrieval, to match video and mocap sequences.

First, we compute the υ-trajectories of each orthographic projection of a mocap sequence and then obtain n frame descriptors $\mathbf{v}_i \in \mathbb{R}^d$ by aggregating the υ-trajectories in standard bags of features. Later, we concatenate the n frame descriptors of each sequence to obtain a matrix representation $\mathbf{v} = [\mathbf{v}_{.1}^\top, \dots, \mathbf{v}_{.d}^\top]^\top = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{d\times n}$. We construct a database, $\mathcal{V}$, of these descriptors using all the mocap sequences available in the mocap dataset.

At query time, we obtain dense trajectories for each video. We similarly obtain per-frame descriptors by aggregating the trajectories into bags of features $\mathbf{z}_i \in \mathbb{R}^d$, and concatenate the descriptors into a video representation $\mathbf{z} = [\mathbf{z}_{.1}^\top, \dots, \mathbf{z}_{.d}^\top]^\top = [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{d\times n}$.

Equation 1 of [22] first considers the *correlation* similarity measure to compare $\mathbf{v} \in \mathcal{V}$ and $\mathbf{z}$:

$$s_\delta(\mathbf{z}, \mathbf{v}) = \sum_{t=-\infty}^{\infty} \langle \mathbf{z}_t, \mathbf{v}_{t-\delta} \rangle . \tag{1}$$

This assumes that dot product is a good similarity measure between $\mathbf{v}$ and $\mathbf{z}$. However, this is not the case for the BoW that describe our frames. We therefore define a kernelized similarity

$$\begin{aligned} s_\delta(\mathbf{z}, \mathbf{v}) &= \sum_{t=-\infty}^{\infty} k(\mathbf{z}_t, \mathbf{v}_{t-\delta}) \qquad (2) \\ &= \sum_{t=-\infty}^{\infty} \langle \mathbf{\Psi}(\mathbf{z}_t), \mathbf{\Psi}(\mathbf{v}_{t-\delta}) \rangle , \qquad (3) \end{aligned}$$

where $\mathbf{\Psi}(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^{d'}$ is a transformation that represents $\mathbf{x}$ in the reproducing kernel Hilbert space of k. Using column notation and the convolution theorem, we can rewrite (3) as

[1] http://www.mathworks.com/matlabcentral/fileexchange/16581-hidden-point-removal

[2] https://github.com/jltmtz/mocap-dense-trajectories

[3] http://lear.inrialpes.fr/people/wang/dense_trajectories

$$s(\mathbf{z},\mathbf{v}) = \mathcal{F}^{-1}\left(\sum_{i=1}^{d} \mathcal{F}(\boldsymbol{\Psi}(\mathbf{z}_{\cdot i}))^{*} \odot \mathcal{F}(\boldsymbol{\Psi}(\mathbf{v}_{\cdot i}))\right), \tag{4}$$

where $\odot$ denotes element-wise multiplication. Here, we note that if k is a homogeneous additive kernel, then we can rewrite $\boldsymbol{\Psi}$ [25] as

$$\boldsymbol{\Psi}(\mathbf{x}) = \left[\sqrt{x_1 L\kappa}, \ldots, \sqrt{x_n L\kappa}\right], \tag{5}$$

where L is the sampling period, and κ is the sampling spectrum given by the inverse Fourier transform of the kernel signature, $\mathcal{K}$. To obtain an approximation, $\hat{\boldsymbol{\Psi}}(\mathbf{x}) \approx \boldsymbol{\Psi}(\mathbf{x})$, of the same dimensionality as $\mathbf{x}$, we assume $\kappa \approx \hat{\kappa} = \kappa(0)$ which is a constant [25]. Therefore, due to the linearity of the Fourier transform, we can rewrite (4) as

$$s(\mathbf{z},\mathbf{v}) \approx \sqrt{L\hat{\kappa}}\mathcal{F}^{-1}\left(\sum_{i=1}^{d} \mathcal{F}(\sqrt{\mathbf{z}_{\cdot i}})^{*} \odot \mathcal{F}(\sqrt{\mathbf{v}_{\cdot i}})\right), \tag{6}$$

where $\sqrt{\mathbf{x}}$ denotes the element-wise square root of $\mathbf{x}$. By adding a filter and a regularization parameter, λ, we achieve the final expression [7, 22]

$$s^{\lambda'}(\mathbf{z},\mathbf{v}) \approx \frac{1}{d}\mathcal{F}^{-1}\left(\sum_{i=1}^{d} \frac{\mathcal{F}(\sqrt{\mathbf{z}_{\cdot i}})^{*} \odot \mathcal{F}(\sqrt{\mathbf{v}_{\cdot i}})}{\mathcal{F}(\sqrt{\mathbf{z}_{\cdot i}})^{*} \odot \mathcal{F}(\sqrt{\mathbf{z}_{\cdot i}}) + \lambda'}\right), \tag{7}$$

where $\lambda' = \lambda / \sqrt{L\hat{\kappa}}$. This result incorporates the finding that computing the square root of a BoW leads to better retrieval performance [17], and offers another perspective on the signed square root heuristic for BoW [9]. In what follows, we refer to (7) as Non-linear Circulant Temporal Encoding (nCTE).

Remark on using different kernels. In (7), setting different values for $\hat{\kappa}$ is equivalent to using different kernel approximations [25]. For example, $\hat{\kappa} = 1$ amounts to using either Hellinger's or the χ^2 kernel, while $\hat{\kappa} = 2/\pi$ and $\hat{\kappa} = 2/\log 4$ are equivalent to using the intersection or the JS kernels respectively. This can be done efficiently at query time, without the expense of recomputing the entire database. We set $\hat{\kappa} = 1$ so as to use an approximate χ^2 kernel, and $L = 0.8$, as suggested in [25].

nCTE implementation details. In practice, we obtain bags of words by computing cluster centers exclusively with video data, using 2,000 words. Later we perform PCA on each $\mathbf{v}_i$ and keep the first 200 components. Unlike [22], we do not perform further high-frequency pruning, as it has the disadvantage of making alignments inherently ambiguous. Rather, we compute (7) directly. Since all $\mathbf{v}_i$ are guaranteed to be purely real, we can store only half of their Fourier representation. For our experiments, we use the CMU mocap database [1]. After calculating 18 different projections for each sequence (as indicated in Section 3.1), we can store the database containing approximately 162 hours of projected data in 30 GB, and perform the match in around 30 seconds on a 3.2 GHz machine using a single core.

3.4. Data augmentation and classification

For a video sequence $\mathbf{v}$, we compute (7) directly and obtain similarity scores for each entry in the mocap database. We simply keep the sequence and the alignment with the maximum score and define the tuple $\mathbf{m} = (f, \theta, \phi, t^m)$, where f represents the selected mocap sequence, θ and ϕ are the azimuthal and polar angles corresponding to the matched viewpoint, and t^m is the index of the temporally aligned frame in the matched mocap sequence where the match peak occurs.

Data augmentation. Let $\mathcal{M}$ be the set of all the matches for the videos in the training view. For every match $\mathbf{m} \in \mathcal{M}$, we pick a subset of frames in the mocap sequence f in the range $[t^m - \tau, t^m + \tau]$. Then we collect υ-trajectories for all 18 viewing angles (3 in Θ, 6 in Φ) and label them with the same class label as the video being matched. We later use these multi-view mocap examples to augment our training data. For an overview of the pipeline, see Figure 1.

4. Experiments

We chose the INRIA IXMAS dataset [31] for our experiments. The dataset has 11 actions such as waving, punching and kicking, performed three times by 10 subjects. Five synchronized camera views are available. Researchers have used this dataset for cross-view action recognition in three different evaluation modes [34]: a) *correspondence* mode, where apart from the annotations for source views, view correspondence for a subset of the test examples is also given, b) *semi-supervised* mode, where some of the examples in the test view, along with labels, are available at training time, and c) *unsupervised* mode, where no labelled examples in the test view are available. The latter is the most challenging in terms of classification accuracy reported in the literature so far, since the test view is completely novel.

In this paper, we present results for the unsupervised mode because it is the most challenging. This also means we make fewer assumptions about the available labels. All our reported results correspond to training and testing on different views, assuming that no videos, labels or correspondences from the test view are available at training time.

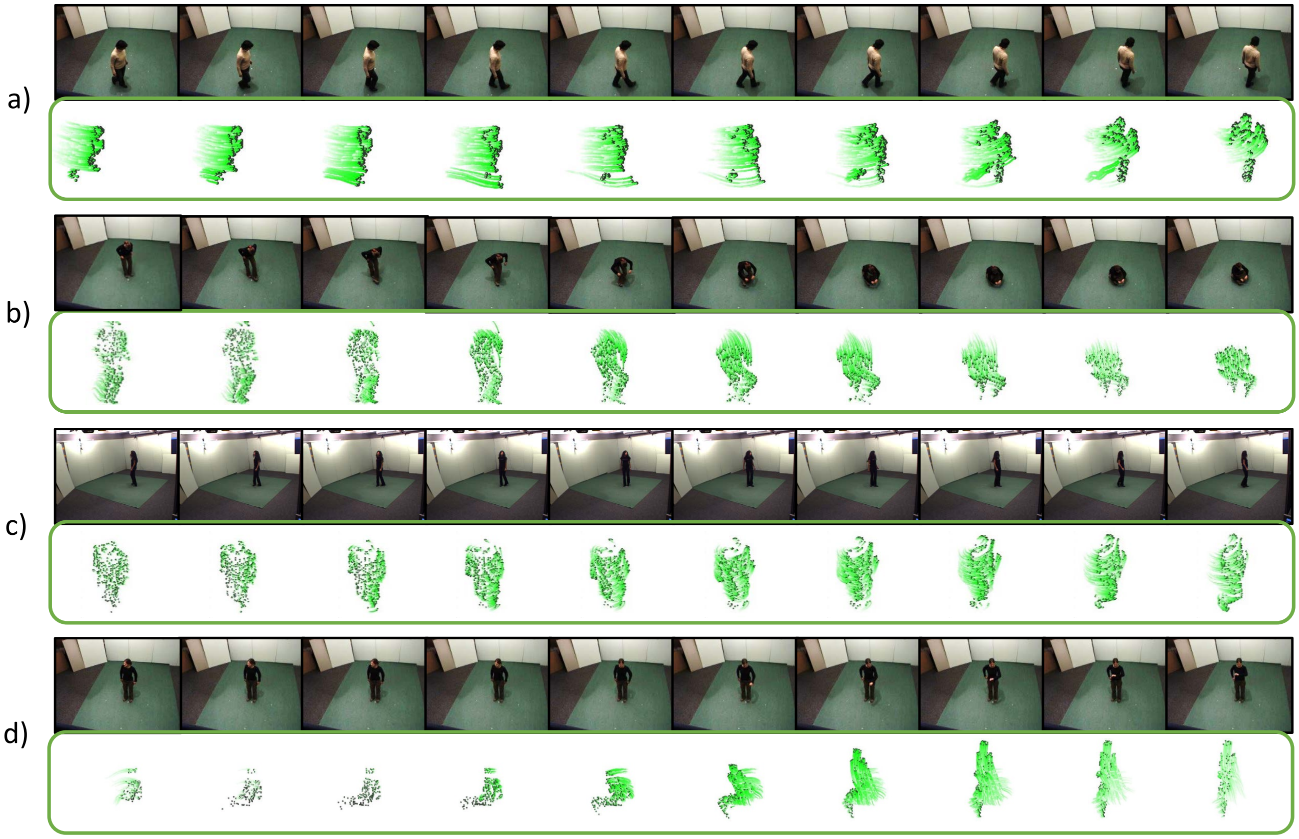
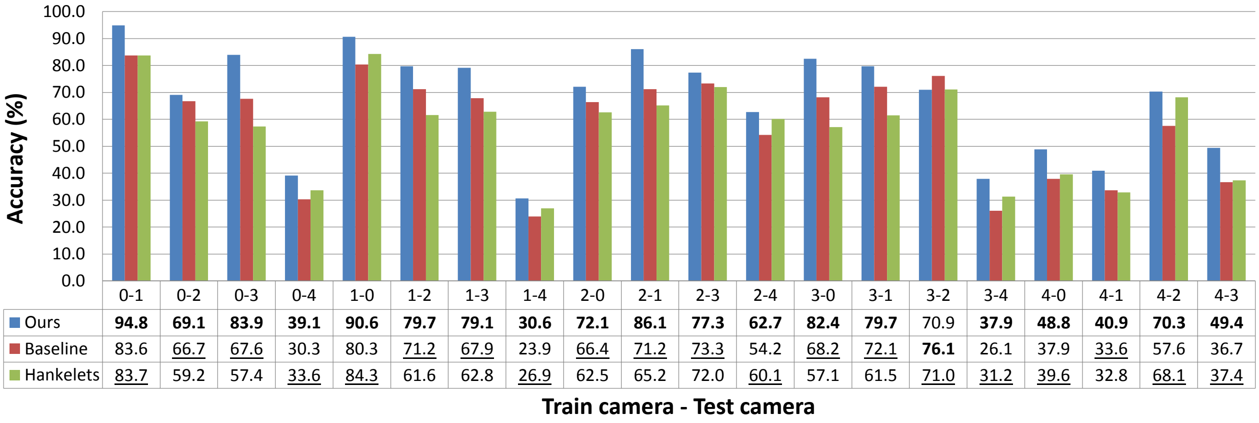


Figure 3: Typical alignments produced by nCTE. The above four sequences include video and the corresponding v-trajectories generated from mocap (in green boxes). a), b) and c) are typical good alignments, where both the action and the viewpoint are well matched. The fourth sequence is a typical failure case: the movement of the hand of the actress causes it to align with a sequence of a person standing up.



	0-1	0-2	0-3	0-4	1-0	1-2	1-3	1-4	2-0	2-1	2-3	2-4	3-0	3-1	3-2	3-4	4-0	4-1	4-2	4-3
🟦 Ours	**94.8**	**69.1**	**83.9**	**39.1**	**90.6**	**79.7**	**79.1**	**30.6**	**72.1**	**86.1**	**77.3**	**62.7**	**82.4**	**79.7**	70.9	**37.9**	**48.8**	**40.9**	**70.3**	**49.4**
🟥 Baseline	83.6	66.7	67.6	30.3	80.3	71.2	67.9	23.9	66.4	71.2	73.3	54.2	68.2	72.1	**76.1**	26.1	37.9	33.6	57.6	36.7
🟩 Hankelets	83.7	59.2	57.4	33.6	84.3	61.6	62.8	26.9	62.5	65.2	72.0	60.1	57.1	61.5	71.0	31.2	39.6	32.8	68.1	37.4

Figure 4: We compare the performance of our approach (nCTE + data augmentation) with our baseline and Hankelets (Table 3 in [13]), the current state-of-the-art. Every column corresponds to one train-test view pair. The best result is shown in bold, and the second best is underlined. Results are averaged over all classes.

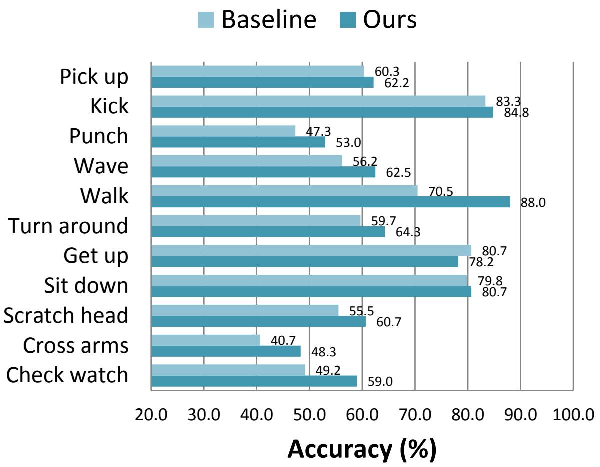


Figure 5: Per class performance gain using augmented data vs. the baseline.

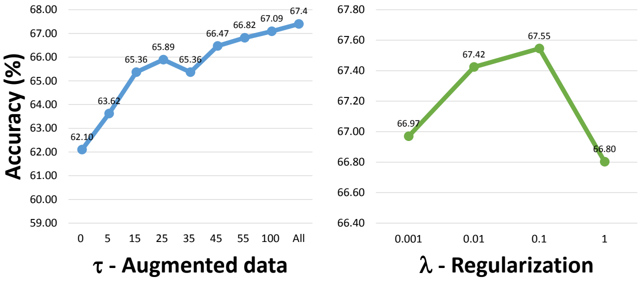


Figure 6: Impact on accuracy of different values of λ and τ.

Baseline. We describe each video in our dataset as a BoW of dense trajectories. We compute dense trajectories sampling every other pixel, and let each trajectory be 15 frames long as in [27]. We cluster the trajectories from the training view into 2,000 k-means clusters, and generate a BoW representation for each video. For classification, we use a non-linear SVM with a χ^2 kernel. We use all the examples (from a training view) as our training set in a one-vs-all SVM framework, and all the videos in the test view as our test set. This simple baseline already outperforms the state of the art. As reported in Table 1, dense trajectories perform significantly better than the overall accuracy results reported in the literature so far [13].

Recognition with augmented multi-view data. The pipeline using multi-view data is almost entirely similar to the one described above. The only difference between the baseline and our approach is the augmentation step: for every video sequence, the 18 projections of the matched mocap sequence are added as positive training examples. We show some typical good matches and error cases returned by nCTE in Figure. 3. We chose $\lambda = 0.01$ for all our experiments. For BoW aggregation, we use the same cluster centers as described above.

Combination	Accuracy
nCTE based matching + augmentation (ours)	**67.4%**
Baseline	62.1%
Hankelets [13]	56.4%

Table 1: Comparison of the overall performance of our augmented approach with our baseline, and the state of the art [13] on the unsupervised cross-view action recognition mode of the IXMAS dataset.

It is a common practice in data augmentation approaches to give different importance to the original and augmented data [5]. In our case, this importance is controlled by the slack penalty C of the SVM. Therefore, we use two different slack penalties $C_{orig} = 1$ and $C_{aug} < 1$ for original and augmented data respectively. This way, we account for a) the imbalance in the number of examples in original and augmented data, and b) the fact that the augmented data might contain errors.

The performance using this approach, as well as our baseline, is reported in Figure 4. We observe a consistent improvement in each train-test pair, with only one exception (pair 3-2). The accuracy improvements per class are reported in Figure 5. We note that walking receives the best performance gain, much in line with a) the amount of available mocap data (*i.e.* almost *every* mocap sequence contains the act of walking), and b) the periodic nature of walking, which makes it easy to match. However, our approach is also able to significantly improve results on harder classes like “scratch head” and “check watch”.

Although we make absolutely no use of annotations in our experiments, a quick search for the action classes in the description file of the CMU mocap dataset[4] makes it evident that not all the actions that we are trying to match have been annotated. Particularly, a search for “check watch”, “cross arms” and “scratch head” returns no results on the CMU mocap website. Nonetheless, we *are* still able to improve the results on these actions. This finding suggests that our approach can be used to discover unlabelled actions in large collections of mocap data.

Parameter tuning. Free parameters of our approach include the regularization parameter of nCTE, λ, the number of added mocap frames around the peak match, τ, and the SVM slack penalty for augmented data, C_{aug}. We report different results for these values in Figure 6. Increasing τ

[4] https://sites.google.com/a/cgspeed.com/cgspeed/motion-capture/cmu-bvh-conversion/bvh-conversion-release---motions-list

amounts to augmenting the number of frames – from the matched mocap sequence – that are added to the training data. We observe that larger values of τ yield marginally better accuracy, obtaining optimal performance when the whole matched mocap sequence is added. An important observation is that we can augment the training data with as few as 50 ($\tau = 25$) frames, with a marginal loss in performance. C_{aug} was set to 0.01 via cross-validation.

5. Conclusions and Future Work

We have demonstrated that unlabelled motion capture data can help improve cross-view action recognition. View independence comes from the large amount of multi-view training data which we match without any additional annotation effort. In the process, we have introduced an inexpensive, purely motion-based descriptor that makes mocap and video data directly comparable.

Regarding future work, we note that despite their similarity mocap and video trajectories are bound to differ in their distribution; we expect that domain adaptation between the two will help to improve matching accuracy. Also, the success of this approach depends on the availability of a large enough mocap dataset to cover a wide range of activities, so we would like to analyse mocap data to account for different action combinations. Incorporating geometry information is also expected to improve the discriminative power of υ-trajectories. Finally our approach currently does not deal with scenarios of camera motion, occlusion due to other objects in the scene, or multiple actors. These all are interesting areas of future work.

Acknowledgements. The authors would like to thank Michiel van de Panne and Ehsan Nezhadarya for fruitful discussions during the development of this work. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Institute for Computing, Information and Cognitive Systems (ICICS) at UBC, and enabled in part by WestGrid and Compute Canada.

References

[1] CMU Motion Capture Database. http://mocap.cs.cmu.edu/.

[2] M. Andriluka, S. Roth, and B. Schiele. Monocular 3D Pose Estimation and Tracking by Detection. In *CVPR*, 2010.

[3] A. F. Bobick and J. W. Davis. The Recognition of Human Movement Using Temporal Templates. *TPAMI*, 23(3), 2001.

[4] M. A. Brubaker and D. J. Fleet. The Kneed Walker for Human Pose Tracking. In *CVPR*, 2008.

[5] C. Chen and K. Grauman. Watching Unlabeled Video Helps Learn New Human Actions from Very Few Labeled Snapshots. In *CVPR*, 2013.

[6] A. Farhadi and M. K. Tabrizi. Learning to Recognize Activities from the Wrong View Point. In *ECCV*, 2008.

[7] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the Circulant Structure of Tracking-by-detection with Kernels. In *ECCV*, 2012.

[8] M. Jain, H. Jégou, and P. Bouthemy. Better Exploiting Motion for Better Action Recognition. In *CVPR*, 2013.

[9] H. Jégou and O. Chum. Negative Evidences and Co-occurences in Image Retrieval: The Benefit of PCA and Whitening. In *ECCV*, 2012.

[10] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent Action Recognition from Temporal Self-similarities. *TPAMI*, 33(1), 2011.

[11] S. Katz, A. Tal, and R. Basri. Direct Visibility of Point Sets. *TOG*, 26(3), 2007.

[12] M. Lewandowski, D. Makris, and J.-C. Nebel. View and Style-independent Action Manifolds for Human Activity Recognition. In *ECCV*, 2010.

[13] B. Li, O. Camps, and M. Sznaier. Cross-view Activity Recognition using Hankelets. In *CVPR*, 2012.

[14] R. Li and T. Zickler. Discriminative Virtual Views for Cross-view Action Recognition. In *CVPR*, 2012.

[15] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[16] V. Parameswaran and R. Chellappa. View Invariance for Human Action Recognition. *IJCV*, 66(1), 2006.

[17] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale Image Categorization with Explicit Data Embedding. In *CVPR*, 2010.

[18] P. Peursum, S. Venkatesh, and G. West. Tracking-as-recognition for Articulated Full-body Human Motion Analysis. In *CVPR*, 2007.

[19] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele. Articulated People Detection and Pose Estimation: Reshaping the Future. In *CVPR*, 2012.

[20] D. Ramanan and D. A. Forsyth. Automatic Annotation of Everyday Movements. In *NIPS*, 2003.

[21] C. Rao, A. Yilmaz, and M. Shah. View-invariant Representation and Recognition of Actions. *IJCV*, 50(2), 2002.

[22] J. Revaud, M. Douze, S. Cordelia, H. Jégou, et al. Event Retrieval in Large Video Collections with Circulant Temporal Encoding. In *CVPR*, 2013.

[23] J. Shotton, A. 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.

[24] M. Ullah and I. Laptev. Actlets: A Novel Local Representation for Human Action Recognition in Video. In *ICIP*, 2012.

[25] A. Vedaldi and A. Zisserman. Efficient Additive Kernels via Explicit Feature Maps. *TPAMI*, 34(3), 2012.

[26] M. Vondrak, L. Sigal, J. Hodgins, and O. Jenkins. Video-based 3D Motion Capture through Biped Control. *TOG*, 31(4), 2012.

[27] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[28] H. Wang and C. Schmid. Action Recognition With Improved Trajectories. In *ICCV*, 2013.

[29] J. Wang, A. Hertzmann, and D. M. Blei. Gaussian Process Dynamical Models. In *NIPS*, 2005.

[30] D. Weinland, E. Boyer, and R. Ronfard. Action Recognition from Arbitrary Views Using 3D Exemplars. In *ICCV*, 2007.

[31] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition Using Motion History Volumes. *CVIU*, 104(2), 2006.

[32] D. Weinland, R. Ronfard, and E. Boyer. A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition. *CVIU*, 115(2), 2011.

[33] X. Wu and Y. Jia. View-invariant Action Recognition Using Latent Kernelized Structural SVM. In *ECCV*, 2012.

[34] Z. Zhang, C. Wang, B. Xiao, W. Zhou, S. Liu, and C. Shi. Cross-View Action Recognition via a Continuous Virtual Path. In *CVPR*, 2013.