

Cross-view Activity Recognition using Hankets

Binlong Li, Octavia I. Camps and Mario Sznajder *

Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115

<http://robustsystems.ece.neu.edu>

Abstract

Human activity recognition is central to many practical applications, ranging from visual surveillance to gaming interfacing. Most approaches addressing this problem are based on localized spatio-temporal features that can vary significantly when the viewpoint changes. As a result, their performances rapidly deteriorate as the difference between the viewpoints of the training and testing data increases. In this paper, we introduce a new type of feature, the “Hanket” that captures dynamic properties of short tracklets. While Hankets do not carry any spatial information, they bring invariant properties to changes in viewpoint that allow for robust cross-view activity recognition, i.e. when actions are recognized using a classifier trained on data from a different viewpoint. Our experiments on the IXMAS dataset show that using Hankets improves the state of the art performance by over 20%.

1. Introduction

Recognition of actions in video is central to many applications, including visual surveillance, assisted living for the elderly, and human computer interfaces [1, 4, 17, 28]. A significant portion of the most recent work in activity recognition [5, 19, 20, 12] has been inspired by the success of using bag of features (BoF) approaches for object recognition. Other approaches are based on time-series using trajectories or a combination of local features and trajectories [27, 23, 29, 14]. While these approaches are quite successful in recognizing actions captured from similar viewpoints, their performance suffer as the viewpoint changes due to the inherent view dependence of the features used by these methods.

In contrast, there is a smaller body of work addressing the problem of multi-view action recognition. Some of these approaches rely on geometric constraints [32], body joints detection and tracking [22, 21], and 3D mod-

els [30, 31, 8, 15]. More recent approaches transfer features across views [16, 7] or use self-similarities as quasi-view invariant features [10, 11]. However, the performances for these approaches are still far below the performances achieved for single view activity recognition.

1.1. Paper Contributions

In this paper, we propose Hankets – the Hankel matrix of a short tracklet – as a new feature to use with a BoF approach to recognize activities across different viewpoints. Hankets provide an alternative representation for activities that carries viewpoint invariance by capturing their dynamics instead of simple spatial gradient information. They are easy to extract and do not require camera calibration, 3D models, body joint detection, persistent tracking or spatial feature matching. Because building a codebook of Hankets requires comparisons of millions of these features, we also propose a simple and fast to compute dissimilarity score that can be used for this purpose. We tested the proposed approach with the IXMAS dataset [30] and our experiments show a performance improvement of 20% over the state of the art. A somewhat similar approach using bags of dynamic systems was proposed in [24] for view-invariant dynamic texture recognition. However, their approach used dense cubes of pixels, required nonlinear dimensionality reduction, system identification and solving a Lyapunov equation. In contrast, our approach uses tracklets, does not require system identification or prior knowledge of the dynamics involved and only requires computing matrix traces.

The paper is organized as follows. Section 2 gives a brief summary of background material on dynamical systems and Hankel matrices. Section 3 gives the details of the proposed approach and section 4 discusses experimental results comparing the proposed approach against previously reported results. Finally, section 5 gives final remarks.

2. Background: Hankel Matrices

Dynamic systems have been recently used in a wide range of computer vision applications, including dynamic texture recognition, target tracking, and activity recogni-

*This work was supported in part by NSF grants IIS-0713003 and ECCS-0901433, AFOSR grant FA9550-09-1-0253, and the Alert DHS Center of Excellence under Award Number 2008-ST-061-ED0001.



Figure 1. Sample frames of activities observed from five different view points from the IXMAS dataset.

tion. Their main appeal is that they can capture the essence of the temporal evolution of the data in a compact way that it is suitable for both analysis (i.e. dynamic texture or activity recognition) and synthesis (i.e. prediction of target location for tracking, synthesis of dynamic textures, etc.). While sometimes the dynamic model is assumed *a priori*, for example brownian motion for tracking applications, it is often more desirable to estimate the dynamic model directly from the available data as explained below.

Given a temporal sequence of a measurement vector $\mathbf{y}_k \in R^n$, the goal is to model its temporal evolution as a function of a relatively low dimensional state vector $\mathbf{x}_k \in R^d$ that changes over time. The simplest dynamical model is a linear time invariant (LTI) system of the form:

$$\begin{aligned} \mathbf{y}_k &= C\mathbf{x}_k + \mathbf{w}_k \\ \mathbf{x}_k &= A\mathbf{x}_{k-1}, \quad \mathbf{x}_o \text{ given} \end{aligned} \quad (1)$$

where both the state and the measurement equations are linear, the matrices A and C are constant over time, and where $\mathbf{w}_k \sim N(0, Q)$ is uncorrelated zero mean Gaussian measurement noise. The dimension of the state vector, d , is the order (memory) of the system and is a measure of its complexity.

Unfortunately, an important limitation to the practical use of models of the form (1) in computer vision, is that one must assume or estimate the dimensions and values of the matrices A and C and the initial vector \mathbf{x}_o . Further, given a finite number of measurements of \mathbf{y}_k , the set of triples (A, C, \mathbf{x}_o) that could have generated this data is not unique¹ and trying to jointly identify the dynamics (A, C) and the initial condition \mathbf{x}_o leads to computationally challenging non convex problems. These difficulties can be avoided by working with the *Hankel matrices* of the data as proposed in [14] and summarized below.

Given a sequence of output measurements from the system (1), $\mathbf{y}_o, \dots, \mathbf{y}_{r+s}$, its associated (block) Hankel matrix

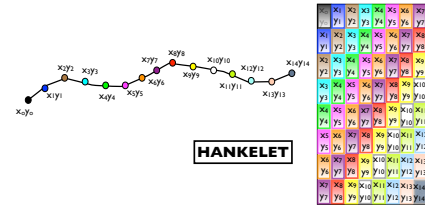


Figure 2. Hankellets represent a tracklet by stacking the coordinates $(x_i, y_i)'$ of the data points from overlapping subsequencies into a Hankel matrix that has constant block-antiagonals.

$H_{\mathbf{y}}^{s,r}$ is:

$$H_{\mathbf{y}}^{s,r} = \begin{bmatrix} \mathbf{y}_o & \mathbf{y}_1 & \mathbf{y}_2 & \dots & \mathbf{y}_r \\ \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \dots & \mathbf{y}_{r+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{y}_s & \mathbf{y}_{s+1} & \mathbf{y}_{s+2} & \dots & \mathbf{y}_{r+s} \end{bmatrix} \quad (2)$$

Note that the columns of the Hankel matrix correspond to overlapping subsequences of the data, shifted by one, and that the block anti-diagonals of the matrix are constant as visualized in Figure 2. As explained in [14], this special structure of this matrix is what encapsulates the dynamic information of the system. In particular, a well known result from realization theory [9, 18] is that, under mild conditions, the rank of the Hankel matrix is the order n of the system $\text{rank}(H_{\mathbf{y}}^{s,r}) = n$ provided that $r, s \geq n$. Furthermore, writing \mathbf{y}_k using an n^{th} order autoregressive model of the form:

$$\mathbf{y}_k = a_1\mathbf{y}_{k-1} + a_2\mathbf{y}_{k-2} + \dots + a_n\mathbf{y}_{k-n} \quad (3)$$

and setting $r = n$ in (2), it is easy to see that the last column of the Hankel matrix is a linear combination of the previous ones and that the coefficients of this combination are precisely the coefficients of the auto regressor. That is,

$$H_{\mathbf{y}}^{s,n} \begin{bmatrix} \mathbf{a}^T & -1 \end{bmatrix}^T = 0$$

In this paper, we will use two useful properties of the Hankel matrix:

Dynamic Subspace Invariance to Initial Conditions. The columns of two Hankel matrices corresponding to two trajectories of the same dynamical system in response to potentially different initial conditions span the same linear subspace, in the absence of noise. This property can be easily shown [14] by factoring the Hankel matrix into $H = \Gamma X$ where Γ is the system observability matrix

$$\Gamma = [C^T \quad \dots \quad (CA^m)^T]^T \text{ and } X = [\mathbf{x}_o \quad \dots \quad \mathbf{x}_m]$$

is a matrix with columns given by the state trajectories.

Dynamic Subspace Invariance to Affine Transformations. The columns of two Hankel matrices corresponding

¹This is related to the concepts of consistency set and diameter of information [25], Chapter 10.

to a trajectory and its affine transformation, span the same linear subspace which is orthogonal to the auto regressor vector of the trajectories $[\mathbf{a}^T \ -1]^T \in R^{n+1}$. This property can be easily shown [2] by writing $\mathbf{Y}_k = \sum a_i \mathbf{Y}_{k-i}$ and using the fact that affine transformations $\mathbf{y}_k = \Pi \mathbf{Y}_k$ are linear. Then,

$$\mathbf{y}_k = \Pi \sum a_i \mathbf{Y}_{k-i} = \sum a_i \Pi \mathbf{Y}_{k-i} = \sum a_i \mathbf{y}_{k-i}$$

and hence the two given Hankel matrices $H_{\mathbf{y}}$ and $H_{\mathbf{Y}}$ share the same auto regressor.

3. Action Representation Using Hankelets

In this section we describe an approach for cross-view activity recognition, where the system is trained using data from one viewpoint and is tested using data captured from a different viewpoint. Our approach is inspired by the activity recognition method using Hankel matrices proposed in [14] and motivated by the affine invariance property of Hankel matrices introduced in [2]. Indeed, the affine invariance property of Hankel matrices suggests that they should be good features to use for recognizing activities from different viewpoints. However, the original approach in [14] relies on Hankel matrices of video-long trajectories of features, such as cuboids or histogram of gradients (HOG), which are then compared using canonical correlations between their spanned subspaces, called dynamic subspace angles (DSA). A drawback of the DSA approach is that it requires persistent tracking through out the whole video, something that it is difficult to achieve in cluttered scenes with complex activities, and a problem that is exacerbated when considering videos from multiple viewpoints. On the other hand, the initial condition invariance property introduced in [14] suggests that one could use pieces of trajectories, i.e. tracklets, without loss of performance. Thus, we propose a modification of this approach, in the spirit of bag of words approaches, that instead uses many more, but densely distributed, and much shorter, tracklets. The advantage of using shorter tracklets is that they are easier to obtain, and by using large numbers of them, it is more likely that some of these tracklets may be visible from different viewpoints. However, before we can do this, we must address the issue of how to efficiently compare large numbers of (noisy) Hankel matrices since using the DSAs as proposed in [14] becomes prohibitive as the number of Hankel matrices increases².

3.1. Local Dynamic Features: Hankelets

The primary features used in our approach are Hankel matrices of relatively short tracklets that we call “Hankelets”. To obtain Hankelets from a video, we first obtain

² Each comparison requires estimates of the ranks of the Hankel matrices and three singular value decompositions.

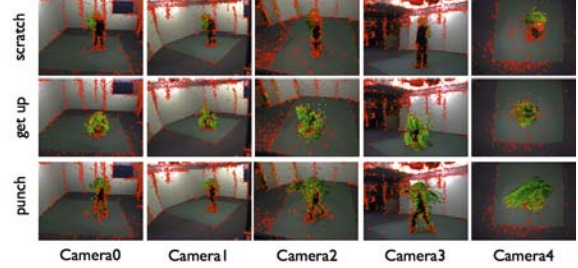


Figure 3. Dense tracklets for the sample frames in Figure 1 (code provided by the authors of [29]). Red dots indicate point positions in the current frame, while tracklets are shown in green.

densely distributed short (typically 15 frames) trajectories of features sampled on a grid, tracked at different scales. The trajectories consist of a set of temporally ordered 2D normalized velocities

$$\frac{1}{\sum_{j=t}^{t+L-1} \|\Delta \mathbf{p}_j\|} (\Delta \mathbf{p}_t, \Delta \mathbf{p}_{t+1}, \dots, \Delta \mathbf{p}_{t+L-1})$$

where $L + 1$ is the number of tracked frames and $\Delta \mathbf{p}_t = (x_{t+1} - x_t, y_{t+1} - y_t)^T$ is a vector with the two components of the velocity of the tracked feature at time t [29]. Both, static trajectories and trajectories with sudden large displacements are discarded. Finally, Hankelets are obtained by assembling the velocity trajectories into Hankel matrices using equation (2) and normalizing them using the Frobenius norm ($\|M\|_F^2 = \text{trace}(M^T M)$):

$$\hat{H}_{\mathbf{p}} = \frac{H_{\mathbf{p}}}{\|H_{\mathbf{p}} H_{\mathbf{p}}^T\|_F^{1/2}}$$

3.2. Comparing Hankelets

Given two Hankelets $\hat{H}_{\mathbf{p}}$ and $\hat{H}_{\mathbf{q}}$ we would like to determine if the corresponding trajectories were generated by the same dynamical system. In principle, one could use an idea similar to that proposed in [14] and define a distance between Hankelets in terms of the angles of the subspaces spanned by their columns. However, a difficulty here is that this approach requires accurately estimation of these subspaces from noisy data³. To avoid this problem, in this paper we will use the following *dissimilarity score* function to compare Hankelets:

$$d(\hat{H}_{\mathbf{p}}, \hat{H}_{\mathbf{q}}) = 2 - \|\hat{H}_{\mathbf{p}} \hat{H}_{\mathbf{p}}^T + \hat{H}_{\mathbf{q}} \hat{H}_{\mathbf{q}}^T\|_F \quad (4)$$

The intuition behind this definition is to exploit the triangle inequality to capture the degree of “alignment” of the column subspaces of $\hat{H}_{\mathbf{p}}$ and $\hat{H}_{\mathbf{q}}$ in a computationally efficient

³To illustrate this point, note that generically, if two tracklets p and q are corrupted by noise to $\hat{q} = q + \eta_q$ and $\hat{p} = p + \eta_p$, the corresponding Hankelets $\hat{H}_{\hat{\mathbf{p}}}$ and $\hat{H}_{\hat{\mathbf{q}}}$ will have full column rank. Hence the angle between the subspaces of the noisy Hankelets is zero, even if \mathbf{p} and \mathbf{q} correspond to different activities.

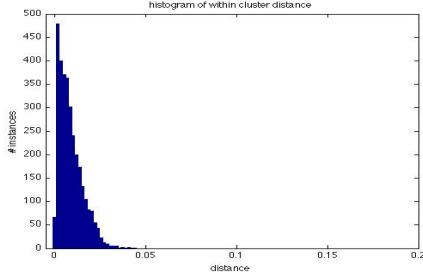


Figure 4. The histogram of dissimilarities for a typical cluster in the dictionary of Hankelets resembles a Gamma distribution.

way, while de-emphasizing the effect of directions potentially associated with noise. Note that, from the fact that the Hankelets are normalized, it follows that $d \geq 0$. Next, consider the singular value decompositions $\hat{H}_p = U_p \Sigma_p V_p^T$, $\hat{H}_q = U_q \Sigma_q V_q^T$, and define $\hat{U}_p \doteq U_p \Sigma_p$, $\hat{U}_q \doteq U_q \Sigma_q$ and $\Theta_{p,q} \doteq \hat{U}_p^T \hat{U}_q$. It is easy to see that, due to the normalization of the Hankelets, $\|\Theta_{p,q}\|_F \leq 1$, and that in terms of this matrix, d can be rewritten as:

$$d(\hat{H}_p, \hat{H}_q) = 2 - \sqrt{2 + 2\|\Theta_{p,q}^T \Theta_{p,q}\|_F}$$

Thus, $d = 0$ if and only if $\Theta_{p,q} = 1$, or, equivalently, $\langle \hat{U}_p, \hat{U}_q \rangle = \text{trace}(\hat{U}_p^T \hat{U}_q) = 1$. Further, from the definition of \hat{U}_p and \hat{U}_q it follows that directions corresponding to small singular values of \hat{H}_p and \hat{H}_q have little effect in d . Thus, $d \approx 0$ for Hankelets corresponding to noisy measurements of the same dynamical system.

3.3. Codebook of Hankelets

Like in the traditional bag of features framework, millions of low level features from training data need to be clustered to build a codebook or dictionary. The algorithm most commonly used for this step is the K-means algorithm. However, in our case the local features are Hankelets representing linear dynamic systems and computing their mean would be meaningless. Thus, we modified the K-means algorithm to work using the set of dissimilarities $D = \{d_{pq} = d(\hat{H}_p, \hat{H}_q)\}$ between all pairs of Hankelets. Then, a Hankelet is assigned to the cluster with the smallest dissimilarity between its “representative” and the given Hankelet, where the “representative” of each cluster is selected as follows. Let $D_w = \{d_{w1}, d_{w2}, \dots, d_{wn_w}\}$ be the dissimilarity scores for all the Hankelets in cluster w with respect to an arbitrarily selected Hankelet in the same cluster and μ_w be their mean. Then, the Hankelet in the cluster that has the dissimilarity score closest to μ_w is selected as the “representative” or “center” of the cluster. Figure 4 shows a typical histogram of the dissimilarity scores with respect to the center of a cluster. As seen there, the distribution closely resembles a Gamma distribution, with a large

number of Hankelets with very small dissimilarities and the number of Hankelets exponentially decreasing for increasing dissimilarities. Thus, we will represent each cluster w by its representative Hankelet \hat{H}_w and a Gamma pdf:

$$p(d|w) = \begin{cases} \frac{a^b d^{b-1}}{(b-1)!} e^{-ad} & \text{for } d \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

with mean $\mu_w = b/a$ and variance $\sigma_w^2 = b/a^2$ estimated from the data. Furthermore, each cluster w has a prior probability $P(w)$ where

$$P(w) \approx \frac{\text{Number of Hankelets in cluster } w}{\text{Total Number of Training Hankelets}}$$

3.4. Bags of Hankelets and Activity Recognition

Each activity video is represented with a Bag of Hankelets (BoHk) – i.e. a histogram of words from the dictionary of K Hankelets. This is done by assigning to each Hankelet in the video the label of the cluster with the maximum *a posteriori* probability:

$$\text{label}(\hat{H}) = \arg \max_w p(d(\hat{H}, \hat{H}_w)|w)P(w) \quad (5)$$

where \hat{H}_w is the representative Hankelet of cluster w and $P(w)$ is the prior probability of cluster w . Then, the entire video is represented by a BoHk given by the histogram of these labels. Finally, activities can be recognized by training a support vector machine (SVM) using BoHks from training data as feature vectors.

3.5. Bi-Lingual Hankelets

Now consider the problem of activity recognition when videos from multiple view points are available. In this case, we seek to relate knowledge about an activity as seen from one view to knowledge of the same activity as seen from a different view.

As shown in [2] the dynamic subspace associated to a Hankel matrix, and hence to a Hankelet, is invariant with respect to affine transformations. Thus, Hankelets of corresponding features across views can be explained by the same regressor and hence have small dissimilarity, provided that the cameras are far enough⁴ to disregard perspective distortion effects. Using the multi-lingual analogy introduced in [16], one can think of Hankelets of trajectories as “bi-lingual” words that have the same “meaning” in the languages of the two viewpoints.

It should be noted that, in general, not all features visible in one view are visible in the other, due to self-occlusions and limited field of view overlap. Thus, videos of the same action but seen from different viewpoints can have very different BoHks. Nevertheless, when the field of view of the

⁴This is usually the case in most surveillance systems.

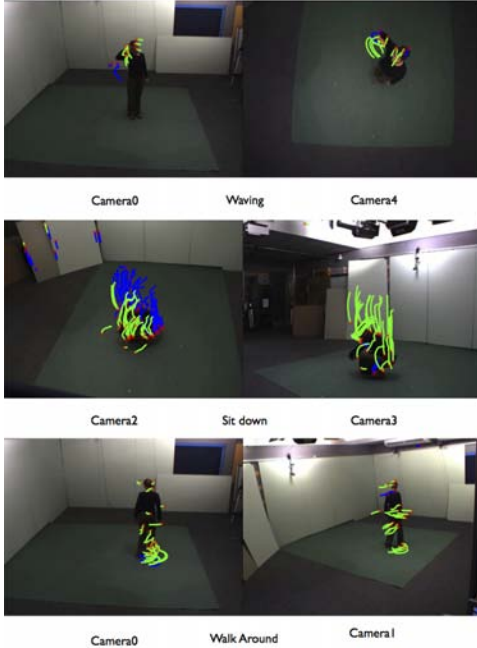


Figure 5. Bilingual Hankelets: Green and blue tracklets indicate bi-lingual and no-bilingual Hankelets, respectively, and red points indicate the ending position in the current frame for each tracklet. See text for discussion.

two cameras partially overlap, many features are likely to be visible from both views for at least a short period of time (for 30 fps videos, a tracklet only lasts 0.5 seconds). Thus, this problem can be easily overcome by restricting the label of bi-lingual to only those Hankelets that are visible from both points of view. Figure 5 shows examples from the IXMAS dataset for two views of three different activities, where green and blue tracklets indicate bi-lingual and not bi-lingual Hankelets, respectively. There, it is seen that even for very large differences of viewpoint (top example) or significant self-occlusion (middle example) there are many bi-lingual Hankelets. As shown in Figure 5 and corroborated by our experiments in section 4, bi-lingual words occur often enough that a dictionary made entirely of bi-lingual words is sufficiently rich to capture the meaning of the activities across different views.

Bi-lingual Hankelets can be easily learned from *unlabeled* videos captured simultaneously from the different viewpoints by matching Hankelets across views. A Hankelet from one viewpoint is assigned a match on the other view if both Hankelets start at the same time and their dissimilarity is less than a selected threshold. In the cases when there is more than one candidate match, the one with the smallest dissimilarity is selected. It should be emphasized that the videos do not need to be labeled and that the matching is purely done on the Hankelets, not their image location

or any other spatial features or geometrical constraints. Intuitively, the purpose of the matching process is to implicitly learn a rough mapping between the different views. Once bi-lingual Hankelets have been identified, it is possible to use them as a common vocabulary between the two views, so that a classifier trained on data from one view is capable of recognizing activities in the other view.

3.6. Cross-view Activity Recognition

In this section we describe the details for training and testing a system for cross-view activity recognition using bags of bi-lingual Hankelets (BoBHK), given a labeled database consisting of c classes of actions, with N_c sample videos for each class captured from two cameras, the “source” and the “target” viewpoints. For a better comparison to previous approaches the procedures follow the experimental protocol proposed in [7]. The protocol uses a *leave-one-action-class-out* strategy, which means that each time only one action is used for testing in the target view and that the data from that action is not used to learn the codebook of Hankelets.

Training Procedure

1. Learn Bi-lingual Hankelets. Extract Hankelets from the *unlabeled* source and target videos and match them using (4). Do not include any data of the activity to be tested in this step.

2. Build Codebook of Bi-lingual Hankelets. Using the modified K-means algorithm, build a dictionary with K words by clustering the bi-lingual Hankelets.

3. Label Hankelets in Source Data. Assign a label from the codebook to every Hankelet in all the source data using (5)⁵. Each video is represented using a BoBHK.

4. Train Classifier using Source Data. A SVM is trained to classify one activity against all others using the BoBHKs from labeled data from the source view.

Testing Procedure

1. Label Hankelets in Target Data. Assign a label from the codebook to every Hankelet in all the target data using (5)⁵. Each video is represented using a BoBHK.







2. Classify Target Data. Using the classifier trained on the source data, classify the data from the target view.

4. Experimental Results

The proposed approach was tested on the IXMAS multi-view action data set [30] which consists of 11 daily-life activities (check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, and pick up.).

⁵If the a posteriori probability is below a threshold, the Hankelet is not used.

Table 1. Classification accuracy for KTH (testing sets) using Hankelets.

					
Boxing	HClapping	HWaving	Walking	Running	Jogging
95.71	95.48	99.09	99.52	91.52	94.05

The activities were performed by 12 different actors and observed from 5 different viewpoints (four side views and one top view). While the focus of the proposed approach is for cross-view activity recognition, we also tested the performance of Hankelets to recognize activities from a single viewpoint to have a baseline. The test was performed using six types of human activities (walking, running, boxing, hand waving, hand clapping, and jogging) from the widely used KTH activity dataset [26].

4.1. Implementation Details

We use 15 frames long tracklets extracted with the code provided by the authors of [29]. A typical video has approximately 15,000 tracklets with an average of 20 tracklets per frame. The tracklets are then assembled into 16×8 Hankelets. To compute the dissimilarity scores between Hankelets we use a fast implementation of (4) that exploits the structure of the Hankel matrices (i.e. their anti-diagonal blocks are constant). For single view activity recognition we used a codebook of 300 Hankelets. For cross-view activity recognition, we only use bi-lingual Hankelets, i.e. Hankelets that appear both in the source and target views (approximately 80% of all Hankelets for most views). Bi-lingual Hankelets were clustered into codebooks with $K = 1,000$, with one codebook for each pair of views and each testing activity (to keep the testing activity out of the training data). For classification we use a one-against-all SVM with histogram Chi-Squared Kernel. The final results are reported in terms of average accuracy for all classes of activity for each view.

4.2. Single View using BoHks

We tested the use of BoHks to recognize the six activities in the KTH dataset. The activities were performed by 25 subjects in four scenarios: outdoors, outdoors with scale variation, outdoors with different clothing, and indoors. All sequences have an homogeneous background and were captured by a stationary camera. The experiments were done following the most commonly used experimental protocol, described in the original paper [26]. Table 1 shows the recognition accuracy for the six activities using Hankelets and Table 2 compares the overall performance to the state of the art. There we can see that using Hankelets alone, without any kind of spatial feature, resulted in very competitive accuracy with a small improvement of 0.87% over the

current state of the art.

Table 2. Comparison of overall performance for KTH dataset using experimental protocol defined in [26]

<i>Algorithm</i>	<i>Perf.</i>
Ours	95.89
Cao et al. [3]	95.02
Wang et al. [29]	94.2
Le et al. [13]	93.9
Li et al [14]	93.6

4.3. Cross-View using BoHks

In these experiments, we learn a codebook of 1000 Hankelets and train a one-against-all classifiers using data from videos captured from each of the five source views in the IXMAS dataset. Then, we test these classifiers on videos captured from the remaining four target views, without any data transfer between the views (i.e. we use all Hankelets, not just bi-lingual Hankelets). The results of the experiments are summarized in Table 3 where the rows and columns correspond to training (source) and testing (target) views, respectively, and the columns **Ours** and **A** show the average accuracy using BoHks and cuboids without model transfer as reported in [16], respectively. The average accuracy using BoHks is 56.4% while using cuboids is only 10.9%, that is an over 400% improvement. The vastly superior performance using Hankelets clearly shows the robustness of the proposed features with respect to changes in viewpoint. Not surprisingly, the top view (Camera 4) has the worst performance since this viewpoint is very different from the others. However, even for this view the BoHks perform three times as better than cuboids.

Table 3. Classification accuracy without data transfer between views. The rows and columns correspond to training (source) and testing (target) views, respectively. The columns **Ours** and **A** show the average accuracy using BoHks and cuboids without model transfer as reported in [16], respectively.

	Cam 0		Cam 1		Cam 2		Cam 3		Cam 4	
	Ours	A	Ours	A	Ours	A	Ours	A	Ours	A
Cam 0			83.70	14.40	59.20	10.69	57.37	10.61	33.62	19.09
Cam 1	84.27	16.12			61.58	11.11	62.75	7.41	26.93	9.22
Cam 2	62.52	10.27	65.17	11.80			71.96	12.90	60.14	8.08
Cam 3	57.05	11.15	61.45	8.59	71.04	9.98			31.24	9.30
Cam 4	39.60	8.80	32.84	8.46	68.12	9.22	37.36	10.06		

Table 4. Classification accuracy for cross-view activity recognition using BoBHks on the IXMAS dataset for each activity. Minimum and Maximum accuracy for a given source/target pair are indicated in yellow and green, respectively.

	Check Watch	Cross Arms	Scratch Head	Get Up	Sit Down	Turn Around
Ave:	92.1	87.9	91.7	90.9	90.4	90.5
MAX:	96.2	96.2	96.5	91.7	91.7	97.5
Min:	89.6	67.4	77.0	89.9	86.4	68.7
	Walk	Wave	Punch	Kick	Pick Up	
Ave:	88.4	91.8	94.0	92.1	86.5	
MAX:	97.7	96.0	99.2	94.7	94.7	
Min:	72.0	90.4	90.2	89.9	69.7	

%	Cam0_Train				Cam1_Train				Cam2_Train				Cam3_Train				Cam4_Train			
	Cam 1Test	Cam 2Test	Cam 3Test	Cam 4Test	Cam 0Test	Cam 2Test	Cam 3Test	Cam 4Test	Cam 0Test	Cam 1Test	Cam 3Test	Cam 4Test	Cam 0Test	Cam 1Test	Cam 2Test	Cam 4Test	Cam 0Test	Cam 1Test	Cam 2Test	Cam 3Test
Action1	89.6	90.9	93.4	90.9	90.9	90.9	91.2	90.9	90.9	94.9	92.2	91.2	91.7	93.2	96.2	90.9	93.7	90.4	93.7	93.7
Action2	96.2	92.9	92.4	89.4	86.1	94.2	76.3	67.4	83.3	92.9	73.2	76.8	93.7	92.9	94.7	82.8	92.7	91.2	96.2	92.7
Action3	96.2	91.7	94.2	87.4	93.9	90.9	87.1	77.0	92.2	93.7	96.5	92.7	93.9	96.0	96.2	90.4	90.7	90.7	92.4	90.9
Action4	91.7	90.9	90.7	89.9	91.2	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.7	90.9	90.9	90.7
Action5	91.7	88.9	90.2	90.9	90.9	89.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	90.9	86.4	88.4	90.9	91.4
Action6	96.5	91.4	89.4	90.9	97.5	92.4	91.4	92.7	91.7	91.2	92.4	92.7	87.1	90.7	91.2	68.7	90.4	90.7	92.4	88.9
Action7	96.2	82.3	83.3	72.0	97.7	89.9	89.9	75.8	91.2	91.9	90.9	90.7	85.9	91.2	93.7	82.8	91.2	90.9	90.9	90.7
Action8	93.7	90.9	96.0	90.9	92.4	90.9	91.9	90.9	90.9	90.9	90.9	90.9	94.7	94.7	91.2	90.9	90.4	90.9	91.4	90.9
Action9	98.5	92.7	96.0	90.9	99.0	93.4	98.2	90.2	96.0	90.7	90.9	97.7	99.2	93.9	90.9	90.9	90.9	90.9	97.5	90.9
Action10	93.7	91.9	92.9	93.4	93.9	91.4	92.4	90.9	91.2	91.4	91.9	91.2	93.4	91.4	90.9	91.9	91.4	91.9	94.7	89.9
Action11	94.7	91.2	91.4	84.1	91.4	90.9	90.9	81.8	87.9	90.9	89.4	87.9	82.3	69.7	69.7	76.8	87.9	89.6	91.7	89.4

Table 5. Comparison against state of the art of classification accuracy for cross-view activity recognition using model transferring on the IXMAS dataset. Columns Ours, A, B, C, and D correspond to our approach, [16]’s approach, [7]’s approach, [10]’s approach, and [6]’s approach, respectively. The overall average accuracies are 90.57%, 75.3%, 58.1%, 59.5% and 74.4%, respectively.

	Cam 0					Cam 1					Cam 2					Cam 3					Cam 4				
	(%)	Ours	A	B	C	D	Ours	A	B	C	D	Ours	A	B	C	D	Ours	A	B	C	D	Ours	A	B	C
Cam 0						94.8	79.9	72	77.6	79	91.2	76.8	61	69.4	79	91.7	78.8	62	70.3	68	88.1	74.8	30	44.8	76
Cam 1	92.2	81.2	69	77.3	72						91.1	75.8	64	73.9	74	90.4	78	68	67.3	70	83.5	70.4	41	43.9	66
Cam 2	91.3	79.6	62	66.1	71	92.1	76.6	67	70.6	82						89.9	79.8	67	63.6	76	90.3	72.8	43	53.6	72
Cam 3	90	73	63	69.4	75	90.2	74.1	72	70	75	89	74.4	68	63	79						86.5	66.9	44	44.2	76
Cam 4	90.7	82	51	39.1	80	90.6	68.3	55	38.8	73	93	74	51	51.8	73	90.4	71.1	53	34.2	79					
Ave.	91	79	61	63	75	91.9	74.7	67	64.3	77	91.1	75.3	61	64.5	76	90.6	76.4	63	58.9	73	87.1	71.2	40	46.6	72.5

4.4. Cross-View using BoBHks

In this section we report the results of experiments testing the effect of using dictionaries of bi-lingual Hangelets on the IXMAS dataset. As described in section 3.6 in this case, only bi-lingual Hangelets are used to build the codebook used to train the classifier in the source view. The main effect of this limitation is to focus the classifier on features that are likely to be visible from the target and source views. The summary of the classification accuracies for all source/target views combinations for each activity, together with the overall average, maximum and minimum accuracy

are given in Table 4. In average, the activity easiest to identify is “Punch” with an average accuracy of 94.4% and the hardest is “Pick Up” with an average accuracy of 86.5%. These results are not surprising, since Punch is one of the activities with the most exaggerated motions while Pick Up is affected by severe self occlusion. Table 5 gives side by side the average accuracy using BoBHks with the accuracy of previous approaches [16, 7, 10, 6]. The overall average accuracy using BoBHks is 90.57%, a 20.28% improvement over the state of the art performance reported in [16] and a 60.58% improvement over using BoHk.

5. Conclusion

In this paper we proposed a new dynamics-based feature (Hankelet) for activity recognition and a simplified score to compare them. Hankelets are easily formed from very short tracklets which do not require persistent tracking, and capture dynamic information that is invariant to affine transformations. Our experiments show that Hankelets perform slightly better than the state of the art in the simple scenario when the training and testing data were captured from the same viewpoint. More importantly, Hankelets perform extremely well in the more challenging scenario when the viewpoints of the training and testing data are significantly different. Our experiments show that using Hankelets alone improve performance by over 400% compared to using cuboids on the IXMAS database. Finally, compared to other cross-view approaches that specifically address viewpoint changes, using a subset of Hankelets (i.e. bi-lingual Hankelets) to compensate for self-occlusions, results in an average accuracy of 90.57% that is an over 20% improvement over the best performance on the IXMAS dataset reported so far.

References

- [1] J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *CVIU*, 73(3):428–440, 1999. [1](#)
- [2] M. Ayazoglu, B. Li, C. Dicle, M. Sznaiier, and O. Camps. Dynamic subspace-based coordinated multicamera tracking. In *ICCV*, 2011. [3](#), [4](#)
- [3] L. Cao, Z. Liu, and T. Huang. Cross-dataset action detection. In *CVPR*, 2010. [6](#)
- [4] C. Cedras and M. Shah. Motion-based recognition: A survey. *Image and Vision Computing*, 13(2):129–155, 1999. [1](#)
- [5] P. Dollar, V. Rabaud, G. Cotteell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005. [1](#)
- [6] A. Farhadi, M. Tabrizi, I. Endres, and D. Forsyth. A latent model of discriminative aspect. In *ICCV*, 2009. [7](#)
- [7] A. Farhadi and M. K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008. [1](#), [5](#), [7](#)
- [8] D. Gravila and L. Davis. 2d model-based tracking of humans in action: A multi-view approach. In *CVPR*, 1996. [1](#)
- [9] B. Ho and R. Kalman. Effective construction of linear, state-variable models from input/output functions. *Regelungstechnik*, 14:545–548, 1966. [2](#)
- [10] I. Junejo, E. Dexter, I. Laptev, and P. Perez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008. [1](#), [7](#)
- [11] I. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *PAMI*, 2011. [1](#)
- [12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008. [1](#)
- [13] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011. [6](#)
- [14] B. Li, M. Ayazoglu, T. Mao, O. Camps, and M. Sznaiier. Activity recognition using dynamic subspace angles. In *CVPR*, 2011. [1](#), [2](#), [3](#), [6](#)
- [15] R. Li, T. Tian, and S. Sclaroff. Simultaneous learning of non-linear manifolds and dynamical models for high dimensional time series. In *ICCV*, 2007. [1](#)
- [16] J. Liu, M. Shah., B. Kuipers, and S. Savarese. Cross-View Action Recognition via View Knowledge Transfer. In *CVPR*, 2011. [1](#), [4](#), [6](#), [7](#)
- [17] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2):90–126, 2006. [1](#)
- [18] M. Moonen, B. D. Moor, L. Vandenberghe, and J. Vandewalle. On- and off-line identification of linear state space models. *Int. J. of Control*, 49:219–232, 1989. [2](#)
- [19] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008. [1](#)
- [20] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *ECCV*, pages 1–14, Jul 2010. [1](#)
- [21] V. Paramesmaran and R. Chellappa. View invariance for human action recognition. *IJCV*, 2006. [1](#)
- [22] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *IJCV*, 2002. [1](#)
- [23] M. Raptis and S. Soatto. Tracklet descriptors for action modeling and video analysis. In *ECCV*, 2010. [1](#)
- [24] A. Ravichandran, R. Chaudhry, and R. Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems. In *CVPR*, 2009. [1](#)
- [25] R. Sánchez Peña and M. Sznaiier. *Robust Systems Theory and Applications*. Wiley & Sons, Inc., 1998. [2](#)
- [26] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. [6](#)
- [27] P. Turaga and R. Chellappa. Locally time-invariant models of human activities using trajectories on the Grassmanian. In *CVPR*, 2009. [1](#)
- [28] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov 2008. [1](#)
- [29] H. Wang, A. Klaser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. [1](#), [3](#), [6](#)
- [30] D. Weinland, E. Boyer, and R. Ronfard. Action Recognition from Arbitrary Views using 3D Exemplars. In *ICCV*, pages 1–7, 2007. [1](#), [5](#)
- [31] P. Yan, S. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008. [1](#)
- [32] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005. [1](#)