Dynamics-Based Invariants for Video Analytics

A Dissertation Presented

by

Binlong Li

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

August 2013
Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction.

Albert Einstein
Abstract

Northeastern University

Department of Electrical and Computer Engineering

Doctor of Philosophy in Electrical Engineering

Dynamics-Based Invariants for Video Analytics

by Binlong Li

Cameras are ubiquitously everywhere and hold the promise of significantly changing the way we live and interact with our environment. A major roadblock to achieve this potential is the curse of dimensionality: often the actionable information is a very small fraction of a vast amount of data, which is difficult to extract in real-time. In this thesis we propose to address this issue by exploiting dynamics-based invariants as an information encapsulating paradigm. The approach is inspired by the fundamental fact that visual data comes in streams: videos are temporal sequences of frames, images are ordered sequences of rows of pixels and contours are chained sequences of edges. We make this ordering explicit by treating the data streams as outputs of dynamic systems that have associated quantities which are invariant to affine transformations, initial conditions, and viewpoint changes. These invariants provide compact representations of the dynamic information in the data, yet they can be efficiently extracted while avoiding identifying the underlying models. The power of the proposed framework is illustrated by applying it to several problems in dynamic scene understanding: activity recognition, shape representation, and multi-camera tracking.
3.2.1 Cell Candidates Identification ........................................... 20
3.2.2 Dynamic Invariance Based Association .............................. 21
   3.2.2.1 Cell trajectory initialization ................................... 22
   3.2.2.2 Dynamic invariance based association ....................... 23
3.3 Experimental Results ......................................................... 25
   3.3.1 Performance Metrics .................................................. 26
   3.3.2 Phantom Measurement ................................................. 26
   3.3.3 Vivo Measurement ..................................................... 27
3.4 Summary .................................................................................. 28

4 Activity Recognition using Dynamic Subspace Angles .............. 30
   4.1 Introduction ......................................................................... 31
   4.2 Proposed Approach ........................................................... 32
      4.2.1 Dynamic Subspaces Angles (DSA) ................................ 33
      4.2.2 Activity Recognition Using DSA ................................. 35
         Training Procedure ...................................................... 35
         Testing Procedure ..................................................... 37
   4.3 Experiments ......................................................................... 38
      4.3.1 KTH Database ............................................................ 38
         4.3.1.1 Feature Extraction ............................................. 38
         4.3.1.2 Hankel Matrix Assembly .................................... 40
         4.3.1.3 DCC ................................................................. 40
         4.3.1.4 SVM Training .................................................... 41
         4.3.1.5 Tests and Performance Evaluation ....................... 42
         Performance Evaluation .............................................. 42
         Benefits of Using DCC .............................................. 42
         Benefits of Using Hankel Matrices ............................... 43
      4.3.2 TV Interaction Database .............................................. 44
         4.3.2.1 Feature Extraction ............................................. 44
         4.3.2.2 Hankel Matrix Assembly, DCC, and SVM Steps ...... 45
         4.3.2.3 Performance Evaluation .................................... 45
   4.4 Summary ................................................................................ 45

5 Cross-view Activity Recognition using Hankelets .................. 47
   5.1 Introduction ......................................................................... 47
   5.2 Action Representation Using Hankelets .............................. 48
      5.2.1 Local Dynamic Features: Hankelets ............................ 49
      5.2.2 Comparing Hankelets ................................................. 50
      5.2.3 Codebook of Hankelets .............................................. 51
5.2.4 Bags of Hankelets and Activity Recognition . . . . . . . . . . . . . 52
5.2.5 Bi-Lingual Hankelets . . . . . . . . . . . . . . . . . . . . . . . . . . 53
5.2.6 Cross-view Activity Recognition . . . . . . . . . . . . . . . . . . . . 55
  Training Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
  Testing Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
5.3 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
  5.3.1 Implementation Details . . . . . . . . . . . . . . . . . . . . . . . . . 56
  5.3.2 Single View Using BoHks . . . . . . . . . . . . . . . . . . . . . . . . 57
    5.3.2.1 KTH dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
    5.3.2.2 UCF Sports Dataset . . . . . . . . . . . . . . . . . . . . . . . . . 57
  5.3.3 Cross-view Using BoHks . . . . . . . . . . . . . . . . . . . . . . . . . 58
  5.3.4 Cross-view Using BoBHks . . . . . . . . . . . . . . . . . . . . . . . . 59
5.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

6 Multi-camera Human Tracking 62
  6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
  6.2 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
    6.2.1 Region Covariance . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
      6.2.1.1 Flip Invariant Region Covariance . . . . . . . . . . . . . . . . . 64
      6.2.1.2 Implementation Details . . . . . . . . . . . . . . . . . . . . . . . 64
  6.3 Appearance representation using region covariance . . . . . . . . . . . . 65
    6.3.1 Proposed framework . . . . . . . . . . . . . . . . . . . . . . . . . . 65
      6.3.1.1 Spatial-temporal estimation for re-identification . . . . . . 65
    6.3.2 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
      6.3.2.1 Cross-view Validation . . . . . . . . . . . . . . . . . . . . . . . 66
        Flip Invariant Region Covariance Comparsion . . . . . . . . . . . . . 66
        Object vs. Random Backgroud . . . . . . . . . . . . . . . . . . . . . . 67
        Experiment Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
      6.3.2.2 Real Transportation Securit Administration (TSA) Dataset
        Validation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
        Same Person Different Camera (SPDC) . . . . . . . . . . . . . . . . . 69
        Different Person Different Camera (DPDC) . . . . . . . . . . . . . . 69
  6.4 Conclusion and future work . . . . . . . . . . . . . . . . . . . . . . . . . . 70

7 Conclusion 71
# List of Figures

1.1 Sample frames of cells observed in microscopic fluorescence images. From sample frames, we could easily notice how noisy the videos are. .......... 4

2.1 Hankelets 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-antidiagonals. ................................. 12

3.1 Sample frames of cells observed in microscopic fluorescence images. For better observation, the cells have been marked with red arrow. From sample frames, we could easily notice how obscure the cells are comparing to the highly corrupted videos. ................................. 19

3.2 The overall algorithm flow chart of the two-step automated analysis of wide field noisy fluorescence image sequence, as well as sample frames after each step. In step 1, the algorithm identifies cell candidates from noisy image sequence, and connect/merge those candidate cells into trajectories based on their dynamic invariance across time in step 2. ................................. 21

3.3 Sample frames of background subtraction output of videos in Figure 3.1. For each video, the first row is the original video frames, and the second row is the corresponding background subtraction results. ................................. 22

3.4 Sample frames of dynamics association based on background subtraction results of videos in Figure 3.1. For each video, the first row is the original video frames, and the second row is the corresponding dynamics association results. For better visualization, associated trajectories have been overlaid on clean artery images and marked as green or yellow. ................................. 25

3.5 (a) sample frame of tissue-mimicking optical flow phantoms with fluorescent micro. (b-f) example set of fluorescence images. (g-k) fluorescence images with white light images and cell track overlaid. ................................. 27

3.6 Performance statistics, in terms of sensitivity v.s. FAR while adopting different thresholds. with a low threshold, such as 99.93% of maximum, the performance could result in a sensitive better than 90% with relatively high FAR as 1.5 / min. Incresing the threshold, such as 99.99% of maximum, the sensitive is 0.65 and the FAR could be decreased as low as 0.04 / min. ......... 28
<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Overview of Activity Recognition Using Dynamic Subspace Angles</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Feature Extraction: Three frames from the KTH database for hand waving and walking and their responses to a Gabor 3D spatial-temporal filter. The two strongest clusters are tracked over time (red and green trajectories).</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Singular values of the Hankel matrices for all the videos in the training and validating sets.</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>Classification performance for the training and validation sets as the number of basis vectors $r$ used to train the SVM are varied. The best performance was obtained for $r = 1$.</td>
<td>41</td>
</tr>
<tr>
<td>4.5</td>
<td>Advantage of using Hankel matrices over using DCC with sub-sequences to recognize KTH activities.</td>
<td>43</td>
</tr>
<tr>
<td>4.6</td>
<td>(a) Sample frames from the TV Interaction Database [75]. (b) HOG features.</td>
<td>45</td>
</tr>
<tr>
<td>5.1</td>
<td>Sample frames of activities observed from five different view points from the IXMAS dataset.</td>
<td>48</td>
</tr>
<tr>
<td>5.2</td>
<td>Dense tracklets for the sample frames in Figure 5.1 (code provided by the authors of [12]). Red dots indicate point positions in the current frame, while tracklets are shown in green.</td>
<td>50</td>
</tr>
<tr>
<td>5.3</td>
<td>The histogram of dissimilarities for a typical cluster in the dictionary of Hankelets resembles a Gamma distribution.</td>
<td>51</td>
</tr>
<tr>
<td>5.4</td>
<td>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.</td>
<td>53</td>
</tr>
<tr>
<td>6.1</td>
<td>Manually selected regions across different views. Red box, stands for person object region. Blue box, stands for random background regions.</td>
<td>66</td>
</tr>
<tr>
<td>6.2</td>
<td>Region Covariance Compaison across different cameras. It could be noticed that if the angle between two cameras are relatively small, the dissimilarity score is also small, vice versa. The dissimilarity score is improved for flip invariant modification comparing to the original region covariance definition.</td>
<td>67</td>
</tr>
<tr>
<td>6.3</td>
<td>Templates from all cameras compare to all regions from camera 0. We can notice that the object region is always the most similar region comaped to the other random regions. Note, the object score column is equal to the first row of Figure 6.2.</td>
<td>67</td>
</tr>
<tr>
<td>6.4</td>
<td>Templates from all cameras compare to all regions from camera 1. We can notice that the object region is always the most similar region comaped to the other random regions. Note, the object score column is equal to the second row of Figure 6.2.</td>
<td>68</td>
</tr>
</tbody>
</table>
6.5 Templates from all cameras compare to all regions from camera 2. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the third row of Figure 6.2.

6.6 Templates from all cameras compare to all regions from camera 3. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the fourth row of Figure 6.2.

6.7 Templates from all cameras compare to all regions from camera 4. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the fifth row of Figure 6.2.

6.8 Histogram of pairwise dissimilarity score between the same person within different viewpoints. Each subfigure represents a specific colormap. (a): RGB (b) HSV (c) LAB. The $x$ axis represents the dissimilarity score, and the $y$ axis represents number of comparison within that particular range.

6.9 Histogram of pairwise distance between different person from different cameras. Each subfigure represents a specific colormap. (a): RGB (b) HSV (c) LAB.
List of Tables

4.1 Comparison of overall performance for KTH dataset using experimental protocol defined in [55] ................................. 42
4.2 Comparison of overall performance for KTH dataset using experimental protocol defined in [55] as reported in [61] ................................. 42
4.3 Inter-class confusion matrix for KTH (testing sets) using the proposed approach. ......................................................... 43
5.1 Classification accuracy for KTH (testing sets) using Hankelets. .......... 55
5.2 Comparison of overall performance for KTH dataset using experimental protocol defined in [55] ................................. 57
5.3 Comparison of overall performance for UCF Sports dataset using experimental protocol defined in [61] ................................. 58
5.4 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 [28], respectively. ..... 59
5.5 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. ................................. 60
5.6 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, [28]’s approach, [29]’s approach, [30]’s approach, and [84]’s approach, respectively. The overall average accuracies are 90.57%, 75.3%, 58.1%, 59.5% and 74.4%, respectively. ................................. 61
Chapter 1

Introduction

Cameras are ubiquitous everywhere and hold the promise of significantly changing the way we live and interact with our environment. Cameras can be installed almost everywhere, such as airports, train stations, mobile phones or even personal eye glasses. With the volume of captured data increasing, the ability to effectively analyze and utilize this information in order to make intelligent decisions has been a major challenge for the whole computer vision community. Related applications could range widely from large scene intelligent surveillance system to mobile phone APPs. All these applications require more advanced algorithms in terms of recognition accuracy, computation efficiency and data scaling.

Following one or multiple targets of interests across spatial and temporal space is a primary computer vision topic. It could be a foundation to most of computer vision applications such as: security surveillance, medical diagnostics, automatic industrial production and many others. Even though tracking has been addressed since the beginning of image processing[1], this topic is still far from accomplished. There are still scenarios, even the most advanced tracker could not handle. In this thesis, we will introduce a new tracking algorithm and apply it to a challenging scenario: macroscopic fluorescence images with high noise level. Due to the noisy level present in the images, the state-of-the-art tracking algorithms could not be directly applied. Instead, the new object dynamic invariance-based tracking system has been introduced. The experiments show that this tracking system could produce acceptable results with up to 90% cell tracking / counting accuracy and 1.5 per minute false alarm rate.
With the development of tracking algorithms, objects of interest could be localized and separated from the background. This makes the dynamics-based single person activity analysis feasible. Inspired by the success of object recognition in still images [2, 3, 4, 5], frame level probabilistic and machine learning algorithms have been widely exploited for activity recognition [6, 7, 8, 9, 10, 11, 12, 13]. However, the temporal information has been either discarded completely or not fully utilized in these approaches. In other words, the dynamic information and correlations among frames, has been partially ignored, which is a key advantage of videos over still frames. This disadvantage could lead to recognition difficulties for long and complex activities. This thesis will explore the underlying dynamical information and exploit the potential improvement in terms of accuracy and efficiency for activity recognition.

Due to subject’s self-occlusion, projective transformation and camera field-of-view limitation, single camera could only capture a limited spatial and temporal field. In order to fully recover the 4D information (3D + time) of the original scenario, a system of multiple sensors has been introduced. However, effectively processing this additional information in order to maximize activity recognition improvements is an ongoing problem. As discussed in Section 1.1.3, the most natural way to utilize information from multiple views would be developing a view invariant-feature, which intuitively carries multiple view-invariant information, and applies the features directly into state-of-the-art algorithms, such as bag-of-word[2]. In this thesis, we will introduce a new dynamics-based view-invariant feature, its comparing metrics and clustering algorithms and attempt to solve a multi-camera activity recognition challenge.

Nowadays, with the rising security concerns and the limitation of sensor’s field of view, it is common to deploy large scale camera networks across important public facilities, such as airports, train stations and government buildings. These days, a single public facility surveillance system with hundreds of video sensors is common. Massive amounts of data have been collected and recorded through these systems, and it would be tedious work for security officers to view all these videos to pick out any suspicious behavior. Therefore, it would be hugely benefited to develop the ability to effectively process and explore a big data volume in order to automatically discovery and locate suspects within a reasonable responding period and associate the identity throughout the cameras network. This thesis will attempt to offer some preliminary research efforts in terms of identity association by applying viewpoint-invariant appearance features across a camera network.
In conclusion, this thesis will address the topics of tracking/counting cells in chaos biology scenario, recognizing human activities in single and multiple views and associating suspicious identities across sensor network by exploring the object dynamics invariance. In this thesis, several feature are extracted by state-of-the-art algorithms, such as 3D Gabor filter [6] and dense tracklets [12]. These will be discussed in details in Section 2. The following provides a more detailed explanation of each aforementioned problem:

1.1 Problem Statement

1.1.1 Cell Counting / Tracking

Cell enumeration is useful for pre-clinical disease research and diagnostic, tumor research and many others. Recently, some research have been done to address this challenge [14, 15, 16]. Even though, according to the application requirements, batch and off-line process is acceptable, accuracy and robustness of the developed algorithm is highly desirable.

In order to analyze cell behavior for a sparse cell population in an extremely noisy image sequences (sample frames in Figure 1.1), our algorithm would be required to accurately tracking each individual cell across sequential frames. Effectively tracking an obscured cell within in very noisy video poses a major challenge to us.

1.1.2 Single View Activity Recognition

The goal of single view activity recognition is to accurately classify actor's action among other possible types of activities by utilizing collected training videos which are generated from one sensor. It is useful for video retrieving, intelligence organizations, video-based searching, assistant living for the elderly and detecting suspicious activity within surveillance system. In recent years, a considerable amount of research has been done to addressed this problem, as evidenced by several extended survey papers [17, 18, 19, 20]. In this problem, training and testing information are both captured by a single camera system. Researchers
need to develop an effective machine learning algorithm to explore the underlying distinguishable characteristics between different classes of activity.

1.1.3 Multiple View Activity Recognition

Activity recognition in video is central to many applications, including visual surveillance, assisted living, and human computer interfaces [17, 18, 19, 20]. A significant portion of the recent work in activity recognition [6, 7, 8, 9] 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 [10, 11, 12, 13].

While these approaches are quite successful in recognizing actions captured from similar viewpoints, their performance suffers as the viewpoint changes due to the inherent view-dependence of the features used by these methods. For this problem, training and testing are performed within separate cameras. In contrast, there is a smaller body of work addressing the problem of multi-view activity recognition.
A multi-camera system would be sparsely deployed over the 3D space around the action area in order to maximize the coverage of the actor, who performs in the center of the action scene. Because some algorithms utilize a 3D reconstruction, sensors would generally be calibrated before hand and kept stationary afterward.

Some previous approaches rely on geometric constraints [21], body joints detection and tracking [22, 23], and 3D models [24, 25, 26, 27]. More recent approaches transfer features across views [28, 29] or use self-similarities as quasi-view invariant features [30, 31]. However, the performance of these approaches is still far below the performance achieved by single view activity recognition.

1.1.4 Tag and Track

A fully developed security system within a large public facility, such as an airport, train station or sport stadium, could easily contain hundreds of cameras. With various sensor specifications and installations, such as PTZ, stationary, view-point and illumination variances, tracking a suspect across multiple cameras with non-overlapping field of view, could easily consume an entire work force of security officers. This challenge has been named: Tag and Track.

Developing an effective solution that relieves operators from this tedious work is attracting more and more interest in the computer vision community. Some research in simplified artificial scenarios has been addressed previously [32, 33, 34, 35], but these approaches generally focus only on appearance features, which are sensitive to environment-effects and lead to features with high dimensionality. These disadvantages make it very difficult to become a robust and effective algorithm.
1.2 Challenges

1.2.1 Noisy Video Tracking without Appearance Information

Current approaches to track objects of interest can be classified into three major categories: appearance features only [36, 37, 38], dynamic model only [39, 40, 41, 42] and appearance features with a dynamic model [43]. These algorithms are developed for videos with an average noise level.

However, as shown in Figure 1.1, the objects of interest is very obscure with little distinguishable appearance features. Because the cell behavior is scenario dependent, it is impossible to assume a simple dynamic model, such as constant velocity or acceleration. Based on these two observations, effectively exploring the dynamics model underneath and using this model as a distinguishable feature to associate tracks across frames would pose a major challenge for an algorithm developer.

1.2.2 Non-parametric Dynamic Information Exploring

Current approaches to modeling and recognizing actions of a single subject can be classified into three major classes: nonparametric, volumetric, and parametric time-series approaches [20]. Nonparametric methods rely on features extracted at the frame level that are later matched against stored templates. The templates can be 2D (e.g. motion history images), 3D (e.g. generalized cylinders in the joint space-time $(x, y, t)$ domain), or use dimensionality reduction methods (e.g. PCA or manifold embeddings). Volumetric approaches process the video data as a volume of pixels and use local features that are 3D generalizations of standard image features, such as corners and spatial-temporal filter responses. A significant portion of the most recent work in activity recognition has been inspired by the success of using local features for object recognition [6, 7, 8, 9].

However, both non-parametric and volumetric approaches are limited by the inherent localness of the features used and the lack of strong relations among features across frames. In contrast, parametric time-series approaches use dynamical models of the motion to exploit
temporal relations across frames. Thus, they are better equipped to model and recognize complex activities with longer duration. Examples of parametric approaches include hidden Markov models (HMMs) and linear dynamical systems which can be considered as a generalization of HMMs, where the state vector can take continuous values in $\mathbb{R}^d$ where $d$ is the dimensionality of the state space. However, a drawback of these methods is that they must assume a dynamical model, which is often too simplistic. In addition they must estimate the model parameters from extensive experimental data, often corrupted by noise. Therefore, effectively exploring the dynamical model would be a major challenge for researchers.

### 1.2.3 View Point Invariant Features

Recently, with widely deployed security camera networks in large public areas, researchers have shown increasing interest in the multiple-view activity recognition. These approaches either rely on various kinds of information, such as geometric constraints [21], body joints detection and tracking [22, 23], 3D models [24, 25, 26, 27] and transfer features across views [28, 29], or use self-similarities as quasi-view invariant features [30, 31]. The challenge lies in effectively transferring the information across different views or developing a view-invariant feature.

### 1.2.4 Significant variances in both time and appearance

Due to the multi-camera topology and various environment conditions, as explained in Section 1.1.4, the challenge of tag-and-track always involves significant appearance changes and inconsistent temporal transitions between viewpoints.

All the previous works [32, 33, 34, 35, 44, 45, 46] are based on various types of appearance features, such as color histogram [44, 46], principal axis histogram [45] and multiple feature-based representation [35, 46]. Due to the large variations for intra-class object appearance, single appearance feature is not enough to associate a person across a large surveillance system with complicated scenarios. To address this problem, researchers tend to combine more features as descriptors in order to increase the discrimination. So far, with the increased
feature dimensionality, neither performance nor speed meets the requirements for real-life applications.

Therefore, effectively associating a suspect across a non-overlapping camera network would be an interesting challenge for researchers to pursue.

### 1.3 Contribution

The main contributions of our work have been described as the following:

**Time-series approach for activity recognition**, which does not need to identify a model. Instead, it exploits the dynamic information encoded in the structure of Hankel matrices built from the data. Our contribution hypothesizes that the temporal data is the output trajectory of an underlying, *unknown* linear (possibly slowly varying) dynamical system. In this context, different realizations of the same activity correspond to trajectories of the same system in response to different initial conditions. Derived from realization theory, exploiting the fact that these trajectories are constrained to evolve in the same subspace (directly determined from the experimental data) allows us to measure the similarity between activities by simply computing the angle between the associated subspaces. We also showed that trajectories corresponding to the same activity live in the same subspace and that the dynamic subspace angles (DSA) – i.e. the canonical correlations between the associated subspaces – can be used to classify the activities.

**Hankelet**, the Hankel matrix of a short tracklet, is a new feature to use with a bag-of-features (BoF) approach to recognize similar activities from different viewpoints. Hankelets provide an alternative representation of activities with 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 nor spatial feature matching.
**Dynamic system dissimilarity score**, is a proposed method of quickly and easily computing a dissimilarity score between two different Hankelets. Because building a Hankelets codebook requires millions of comparisons, the new proposed dissimilarity score could greatly reduce the computation complexity to a reasonable level without sacrificing performance.

### 1.4 Organization

The following provides a brief explanation of each chapter:

**Chapter 2**, explains the background dynamics system, auto regression information and some feature extraction algorithms, which are the foundation of theories developed in this thesis.

**Chapter 3**, introduces a simple but effective cell counting / tracking algorithm for microscopic fluorescence image sequences.

**Chapter 4**, presents a dynamic-system-based subspace angles algorithm for activity recognition. Canonical correlation of linear subspace will be introduced. The algorithm has been tested for single person and multiple person activity recognition.

**Chapter 5**, introduces a brand new view-invariant feature: Hankelets. Corresponding comparison metrics and schemes with mathematical proof are provided. Outstanding performance compared to state-of-the-art algorithms has been achieved.

**Chapter 6**, includes some preliminary experiment results from cross-camera person association with region covariance feature and a pedestrian location estimation framework.

**Chapter 7**, will draw the conclusion for the entire thesis.
Chapter 2

Background

2.1 Dynamical Systems and Hankel Matrix

Dynamical systems are a powerful tool to work with temporally ordered data. They have been used in several applications in computer vision, including tracking, human recognition from gait, activity recognition, and dynamic texture. The main idea, is to use a dynamical system to model the temporal evolution of a measurement vector $y_k \in \mathbb{R}^n$ as a function of a relatively low dimensional state vector $x_k \in \mathbb{R}^d$ that changes over time. For example, depending on the application, the measurement vector $y_k$ can represent the coordinates of a tracked target at time $k$, or the pixel values of an image captured at time $k$. Then, the dynamical model can be used both, as a generative model, for example to predict the location of the target in the next frame or to generate a new video sequence of dynamic texture, or as a nominal model, for example to characterize activities or dynamic textures for recognition or classification.

The simplest dynamical model is a linear time invariant (LTI) system of the form:

$$
\begin{align*}
    y_k &= Cx_k \\
    x_k &= Ax_{k-1} + w_k, \quad x_0 \text{ given}
\end{align*}
$$

(2.1)

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

It should be noted that an important limitation to the practical use in computer vision of models of the form (2.1), is that one must assume or estimate the dimensions and values of the matrices \( A \) and \( C \) and the initial vector \( x_o \). Further, given a finite number of measurements of \( y_k \), the set of triples \( (A, C, x_o) \) that could have generated this data is not unique\(^1\). Finally, attempting to jointly identify the dynamics \( (A, C) \) and the initial condition \( x_o \) leads to computationally challenging non convex problems. To avoid these difficulties, in this proposal, we will not work directly with the model representation (2.1). Instead, motivated by subspace identification methods [48], we will work with the block Hankel matrices of its output sequences as defined next.

Given a sequence of output measurements from the system (2.1), \( y_o, \ldots, y_{r+s} \), its associated (block) Hankel matrix \( H_{y}^{s,r} \) is:

\[
H_{y}^{s,r} = \begin{bmatrix}
y_o & y_1 & y_2 & \cdots & y_r \\
y_1 & y_2 & y_3 & \cdots & y_{r+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
y_s & y_{s+1} & y_{s+2} & \cdots & y_{r+s}
\end{bmatrix}
\]

(2.2)

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 2.1. As explained in [13], this special structure of this matrix is what encapsulates the dynamic information of the system. In particular, a well known result from realization theory [49, 50] is that, under mild conditions, the rank of the Hankel matrix is the order \( n \) of the system \( \text{rank}(H_{y}^{s,r}) = n \) provided that \( r, s \geq n \).

As we show in Section 4.2, the special structure of this matrix encapsulates the dynamic information of the system.

\(^1\)This is related to the concepts of consistency set and diameter of information [47], Chapter 10.
Figure 2.1: Hankelets 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-antidiagonals.

### 2.2 Hankel Matrix with Auto Regressor

 Furthermore, writing \(y_k\) using an \(n^{th}\) order autoregressive model of the form:

\[
y_k = a_1 y_{k-1} + a_2 y_{k-2} + \ldots + a_n y_{k-n}
\]

and setting \(r = n\) in (2.2), 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,

\[
H_y^{s,n} \left[ a^T - 1 \right]^T = 0
\]

In this section, 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 [13] by factoring the Hankel matrix into \(H = \Gamma X\) where \(\Gamma\) is the system
observability matrix

\[ \Gamma = \left[ C^T \ldots (CA^m)^T \right]^T \quad \text{and} \quad X = \begin{bmatrix} x_o & \ldots & x_m \end{bmatrix} \]

is a matrix with columns given by the state trajectories.

**Dynamic Subspace Invariance to Affine Transformations.** The columns of two Hankel matrices corresponding to a trajectory and its affine transformation, span the same linear subspace which is orthogonal to the auto regressor vector of the trajectories \( \begin{bmatrix} a^T & -1 \end{bmatrix}^T \in R^{n+1} \). This property can be easily shown [51] by writing \( Y_k = \sum a_i Y_{k-i} \) and using the fact that affine transformations \( y_k = \Pi Y_k \) are linear. Then,

\[ y_k = \Pi \sum a_i Y_{k-i} = \sum a_i \Pi Y_{k-i} = \sum a_i y_{k-i} \]

and hence the two given Hankel matrices \( H_y \) and \( H_Y \) share the same auto regressor.

### 2.3 Canonical Correlations of Linear Subspaces

Recognition and classification problems can often be posed as a vector classification task, where an unknown vector (for example, a rasterized image) has to be assigned to one of a set of training classes represented as a linear subspace learned from a set of labeled vectors (images). The separation between classes can be measured using canonical correlations, also known as principal or canonical angles, which are defined as follows:

Given two subspaces \( F \) and \( G \) such that

\[ p = \dim(F) \geq \dim(G) = q \geq 1, \]

the cosine of the smallest principal angle \( \theta_1(F,G) = \theta_1 \in [0, \pi/2] \) between \( F \) and \( G \) is defined by

\[ \cos \theta_1 = \max_{u \in F} \max_{v \in G} u^T v \quad \|u\|_2 = \|v\|_2 = 1 \]
Assuming that the maximum is obtained at \( u = u_1, v = v_1 \), then \( \theta_2(F, G) \) is defined as the smallest angle between the orthogonal complement of \( F \) with respect to \( u_1 \) and that of \( G \) with respect to \( v_1 \), and so forth, until one of the subspaces is empty. Then, the canonical correlations are defined as:

\[
\cos \theta_k = \max_{u \in F} \max_{v \in G} u^T v = u_k^T v_k \quad \|u\|_2 = \|v\|_2 = 1
\]

subject to the constraints

\[
u_j^T u = 0, \quad v_j^T v = 0 \quad , j = 1, 2, \ldots, k - 1
\]

When the subspaces are defined as the range of two matrices \( A \) and \( B \), the canonical correlations can be computed by performing a singular value decomposition (SVD) as follows \[52\]. Let \( P_A \) and \( P_B \) be unitary bases for the subspaces spanned by \( A \) and \( B \) and let \( M = P_A^T P_B \). Then, the canonical correlations between \( A \) and \( B \) are given by the singular values of \( M \).

Intuitively, the canonical correlations measure the angles between the closest vectors from the two subspaces. A high canonical correlation value corresponds to a small subspace angle and to subspaces that are close to each other. On the other hand, a small canonical correlation corresponds to a subspace angle near \( \pi/2 \) or subspaces that are close to orthogonal. Thus, in classification applications, classes that have higher canonical correlations are more separated and easier to discriminate. Indeed, given a set of training classes represented by subspaces, it is possible to improve the classification performance and robustness to noise by using discriminant canonical correlations (DCC) \[53\]. This is accomplished by first applying a linear transformation to the given data in a way to maximize the canonical correlations of with-in classes while minimizing the canonical correlations of between-classes.

### 2.4 Spatial-temporal Feature Detection

Spatial-temporal feature detection has been naturally derived from spatial case. Instead of operation on image \( I(x, y) \), feature detection has to operate on a sequence of images, \( I(x, y, \)
t), likewise, the detection would have temporal extension. There are two major 3D feature
detector that we know of, one is the extended Harris corner detector [54] and the other is 3D
gabor filter [6].

The idea of extended Harris corner detector [54] is simple but elegant. Gradients could be
found not only in spatial domain but also along the temporal domain by requiring large vari-
ances in the 3D (x, y, t) space. In order to model a spatial-temporal image sequence, an
anisotropic Gaussian kernel is defined as below:

\[
g(x, y, t; \sigma^2_t, \tau^2_t) = \frac{\exp\left(-\frac{(x^2 + y^2)}{2\sigma^2_t} - \frac{t^2}{2\tau^2_t}\right)}{\sqrt{(2\pi)^3\sigma^4_t\tau^4_t}}
\]

Similar to spatial domain, the spatial-temporal second-moment matrix, a 3-by-3 matrix, is
averaged with a Gaussian function \( g(\cdot; \sigma^2_t, \tau^2_t) \)

\[
\mu = g(\cdot; \sigma^2_t, \tau^2_t) \ast \begin{pmatrix}
L_x^2 & L_xL_y & L_xL_t \\
L_xL_y & L_y^2 & L_yL_t \\
L_xL_t & L_yL_t & L_t^2
\end{pmatrix}
\]

Eigenvalues of \( \mu, \lambda_1, \lambda_2, \lambda_3 (\lambda_1 \leq \lambda_2 \leq \lambda_3) \) as significant values, indicate the presence of an
interested point. The extended Harris corner function has been defined as below:

\[
H = \det(\mu) - k\text{trace}^3(\mu) = \lambda_1\lambda_2\lambda_3 - k(\lambda_1 + \lambda_2 + \lambda_3)^3
\]

The extended Harris corner detector has been successfully applied for activity recognition
[9, 55]. However, in certain problem domains, the true spatial-temporal corners are fairly rare,
such as human facial expression and other fine/detailed motions. Even though, sparseness is
desirable in general case, too rare could lead to difficulties for recognition frameworks as
discussed in [56].

Therefore a new spatial-temporal feature detector has been proposed in [6]. It has been ex-
plicitly designed to detect too many features rather than too few. Based on the success of
recognition frameworks, which could deal with irrelevant or misleading features [56], too
many features should not cause any recognition issues. The response function has the form of

\[ R = (I \ast g \ast h_{ev})^2 + (I \ast g \ast h_{od})^2 \]

where \( g(x, y; \sigma) \) is the 2D Gaussian smoothing kernel, only applied along spatial domain, and \( h_{ev} \) and \( h_{od} \) are a quadrature pair [57] of 1D Gabor filter for temporal domain. This quadrature pair is defined as:

\[
\begin{align*}
    h_{ev}(t; \tau, \omega) &= -\cos(2\pi t \omega)e^{-t^2/\tau^2} \\
    h_{od}(t; \tau, \omega) &= -\sin(2\pi t \omega)e^{-t^2/\tau^2}
\end{align*}
\]

The detector is designed to induce strong response whenever a periodic frequency component appears. However, in general, any area with distinct appearance characteristics and a complex motion can introduce a high response. Regions with undergoing simple translation motion, such as moving or smoothed edge will induce a gradual change. Homogeneous or static area will cause no response at all.

## 2.5 Dense Trajectories

Due to different characteristics of 2D spatial domain and 1D temporal domain, it is intuitive to handle them as interesting points tracked through image sequences, rather than spatial-temporal interest point detection. This has been addressed in different manners, such as tracking Harris3D interest points [58] with the standard KLT tracker [59], matching SIFT [56] descriptors between consecutive frames[60]. Observed from a recent activity recognition evaluation [61], dense sampling outperforms the state-of-art interest point detectors. Therefore, in this section, a dense trajectories sampling algorithm [12] will be introduced.

In [12], dense trajectories are extracted for 8 spatial scales spaced by a factor of \( 1/\sqrt{2} \) with 8 pixels sampling step. Each point \( P_t(x_t, y_t) \) at frame \( t \) is tracked to frame \( t + 1 \) by median filtering in dense optical flow field \( \omega = (u_t, v_t) \).
The final shape of a trajectory is represented and normalized as below:

\[ S = \frac{(\Delta P_t, \ldots, \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\Delta P_j\|} \]

where \( \Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t) \).
Chapter 3

Rare-cell tracking in vivo fluorescence flow cytometry

Most of previous cell tracking approaches are based on probability distribution models, which could only estimate the number of active cells, without individual cell trajectory extracted for cell behavior research. These algorithms could only be applied on low noise level video dataset. Their performance would easily be downgraded on more challenging high noise level datasets. In this chapter, we will introduce a new dynamics based cell counting algorithm with dynamical information as the critical cue for appearance-less cell tracking in challenging image sequences with high noise level. Our experiments on a biological vivo cell dataset show that using dynamics could achieve up to 90% accuracy with 1.5 / min false alarm rate in this challenging dataset.

3.1 Introduction

There are many clinical biomedical researches that require high-sensitive detection and enumeration of sparse circulating cell populations in the blood stream of small animals [63, 64, 65, 66, 67, 68]. With the development of microscopic fluorescence sensors, it’s possible to capture microscopic fluorescence video for a large skin area. For further biology research,
Figure 3.1: Sample frames of cells observed in microscopic fluorescence images. For better observation, the cells have been marked with red arrow. From sample frames, we could easily notice how obscure the cells are comparing to the highly corrupted videos.

how to effectively extract the statistical information, such as active cells’ number and behavior habits, from captured image sequences, is a critical challenge for the community. In recent years, a few researches have been done to address this challenge. Some of these approaches are based on the theory of Random Finite Set (RFS) to estimate the cell enumeration[14] in low noise level videos [16]. More recent approaches used dynamic probabilistic model to roughly track a massive number of cells[15]. However, these approaches either suffer from lacking of individual cell trajectory or high noise level.

Due to the large imaging area of microscopic fluorescence capturing system and the rareness of fluorescently labeled circulating cells, active cells are relatively small with obscure appearance within the highly noisy context (as shown in Figure 3.1). It was critical to effectively extract cell locations and associate them through time. Due to cells’ obscure appearance, the best chance to recover the association would be borrowing information from their dynamics.

In this chapter, we propose a new dynamic invariance based tracking system, which estimates dynamic information from connected initial trajectories. Due to the obscure cell appearance, dynamic model carries more distinguishable information than general appearance features. Because a simple dynamic model usually does not works well and the fact that estimating a
dynamic model requires initial trajectory, a simple but effective cell-identification algorithm that could extract cells from a noisy video will also be introduced. We test the proposed approach with a new fluorescence dataset and our experiments show a performance with up to 90% detection accuracy with 1.5 / min false alarms. Comparing to other approaches, our algorithm could effectively enumerate and track individual cells from very noisy image sequences by using online dynamic analysis, without requiring any prior knowledge of the dynamic model nor pre-assumed cell distribution.

The chapter is organized as follows. Section 3.2 gives the details of the proposed approach and Section 3.3 discusses experimental results. Finally, Section 3.4 gives final remarks.

### 3.2 Cell Tracking Using Dynamics

In this section, we describe a two-step approach to extract tracks of active cells from a wide field noisy fluorescence image sequence. The overall algorithm flow chart is shown in Figure 3.2. In the first step, the algorithm identifies candidate cells from individual images in the noisy sequence, and in the second step, the candidate cells will be connected and merged across frames into cell tracks based on their dynamic invariance across time.

#### 3.2.1 Cell Candidates Identification

The primary feature used in our approach is cell candidate blob, foreground of each cell in the image sequence. To obtain cell candidates blobs from a noisy image sequence, a foreground detection from the denoised image sequence will be performed (Step-1A in Figure 3.2), by applying the state-of-art background subtraction algorithms, such as MoG [69], rPCA [70] and the basic mean value subtraction algorithm.

A median filter has been applied on the background subtraction output to filter out small noisy regions. In order to convert the background subtracted results into a foreground binary image sequence, thresholding is required. How to select the threshold for different scenarios is crucial since it alters the algorithm’s performance with respect to accuracy and false alarm rate.
In our algorithm, the threshold is selected as a function of percentile of all pixel intensities in the image sequence; due to sparseness of cells, empirically, threshold in range of 99.93% - 99.99% works well for mice in vivo dataset. Sample background subtraction results are shown in Figure 3.3.

### 3.2.2 Dynamic Invariance Based Association

Following the operations in Section 3.2.1, the binary foreground image sequence contains a set of circulating candidate cells, which in practice outnumbered the actual number of cells due to the high noise level. Therefore in this section, cells will be separated from background noise and associated as trajectories by applying dynamic invariance based analysis.
Figure 3.3: Sample frames of background subtraction output of videos in Figure 3.1. For each video, the first row is the original video frames, and the second row is the corresponding background subtraction results.

3.2.2.1 Cell trajectory initialization

1. Candidate level association

From a practical perspective, the ratio between cell speed and frame rate controls cell moving distance between frames. In practice, this distance is also related to cell size. The bigger cells tend to moving faster than smaller ones. Therefore, in our experiments, candidate cells in consecutive frames would be connected. Empirically, for each cell, a candidate association search will be performed within a disk of a radius, which equals to two times of the cell’s diameter, in the next frame. After the candidate level association, each potential cell would have a initial tracklet.
2. Tracklet level association

Because in our application, cells were relatively dim, gaps for a given cell’s track, lasting for up to 10 frames, were regularly observed. In order to recover the potentially full length track, each tracklet would search for other candidate tracklets’ starting point within the spatial and temporal neighborhood of the querying tracklet’s ending point. In our experiments, the spatial and temporal searching radius have been set as 15 pixels and 15 frames respectively. The tracklet level association provides a good initialization trajectory for each cell, which will be further analyzed based on the dynamic information carried with these initial trajectories. Each trajectory is represented as in Definition 1.

**Definition 1.** A set of trajectories: \( \{ S_{t_1}^{t_1}, S_{t_2}^{t_2}, ..., S_{t_n}^{t_n} \} \), \( t_n \) stands for trajectory \( n \) with starting time \( t_n \), where \( t_1 \leq t_2 \leq ... \leq t_n \)

### 3.2.2.2 Dynamic invariance based association

Depending on different circumstances, such as cells within artery v.s. not in artery, behaviors of each cell could be very different, which means each cell’s dynamic model may be very different as well. A cell candidate searching, with homogenous radius in spatial and temporal domain, as discussed in Section 3.2.2.1, may fail. Therefore, dynamics analysis based association is required.

**Definition 2.** Immediate velocity of trajectory \( n \) (\( S_{t_n}^{t_n} \)):

\[
\nu_{t_n} = \frac{\| p_{t_n}^{l_n} - p_{t_{n-1}}^{l_{n-1}} \|}{t_{\text{inter}}} \]

where \( p_{t_n}^{l_n} \) and \( p_{t_{n-1}}^{l_{n-1}} \) represent the last two points of \( S_{t_n}^{t_n} \) and \( l_n \) represents the number of points of \( S_{t_n}^{t_n} \).

\( t_{\text{inter}} \) represents time interval between frames, which is related to capturing frame rate and constant for one video. Therefore, it is omitted in our implementation.
Chapter 3. Cell Counting using Dynamics

**Definition 3.** Searching Radius:

\[ \varphi(c, r) : \|p - c\|_2 \leq r \]

where \( c \) represents a center point, \( r \) is the radius around the center, and \( p \) is any point within the radius of \( r \) around center \( c \).

In this step, the searching area was determined by the cell’s projected position, and a wide range of velocity variance. Depending on the annotations in Definition 2 and 3, it leads to our final dynamics merging criterion in Equation 3.1, where \( t_m \geq t_n, \Delta t = \|t_m - t_n\| \) and \( p_{1m}^{tm} \) represents the first point in \( S_{tm}^m \).

\[
p_{1m}^{tm} \in (\varphi(p_{1n}^{tn} + 0.5v_{tn}\Delta t, 15)) \cup \\
\varphi(p_{1n}^{tn} + v_{tn}\Delta t, 15)) \cup \\
\varphi(p_{1n}^{tn} + 2v_{tn}\Delta t, 15))
\]

(3.1)

In practice, Equation 3.1 creates a group of searching areas to connect those mis-associated trajectories in Section 3.2.2.1. At the end of the association process, candidates of the following two types would be discarded.

1. any cell candidate which could not connect as a track (i.e. observed only once single frame).

2. any trajectory with almost zero velocity (i.e. stationary blinking points in the scene).

Some sample frames after dynamics association is shown in Figure 3.4.
Chapter 3. *Cell Counting using Dynamics*

3.3 Experimental Results

The proposed approach is tested on the bio-cell-counting dataset which consists of 2 types of measurements, phantom and vivo measurements. There are total 4 phantoms, each with 5 minutes video and 6 different sets of vivo measurements, each set with 60 videos. The whole dataset is captured by an advanced microscopic fluorescence sensor system, details of the dataset and capturing system are described in [68]. As noted in Section 3.2.1, a range of histogram thresholds from 99.93% to 99.99% have been tested in order to obtain a better performance measurement.

**Figure 3.4:** Sample frames of dynamics association based on background subtraction results of videos in Figure 3.1. For each video, the first row is the original video frames, and the second row is the corresponding dynamics association results. For better visualization, associated trajectories have been overlaid on clean artery images and marked as green or yellow.
3.3.1 Performance Metrics

In order to correctly evaluate our algorithm in terms of accuracy and false alarm, two standard performance metrics have been applied.

1. Sensitivity = \( \frac{TP}{TP + FN} \)

2. False Alarm Rate (FAR) = \( \frac{FP}{\text{minute}} \)

where \( TP \), \( FN \) and \( FP \) stand for true positive, false negative (a.k.a. miss detection) and false positive respectively. The ground truth (GT) number for each video has been manually counted by a well trained operator. Comparing to using injected cells’ number as ground truth, generating ground truth data in this way would get rid of the invisible-cells effects. We believe this is the correct way to estimate the performance of a computer vision algorithm. Finally, the GT would be used to compare with our algorithm output to obtain the final performance metrics.

3.3.2 Phantom Measurement

Our proposed algorithm is tested on CV-IVFC approach in tissue-mimicking optical flow phantoms (Figure 3.5a) with fluorescent micro. An example set of fluorescence images are shown in Figure 3.5 (b-f), along with white light images and cell track overlays in Figure 3.5 (g-k). By observation of Figure 3.5 (b-f), the appearance contrast was good in phantom model. This eases the challenge of cell identification and tracking.

Because the appearance contrast is high in this measurement, the threshold twisting does not have a huge impact on our experiment. Therefore, the threshold of 99.96\% of maximum intensity value out of all pixels in the video was used.

Overall performance over the four phantoms is: sensitivity = 0.993 and false alarm rate = 0.074 / minute. Even without comparison to other algorithms, this statistical data is very high and impressive.
3.3.3 Vivo Measurement

Our algorithm is also tested on CV-IVFC approach in nude mice injected with very low concentrations of fluorescently-labeled multiple myeloma cells. The example fluorescence image sequences and their processed result after each major step have been shown before in Figure 3.1, 3.3, 3.4, respectively. By observation of Figure 3.1, the appearance contrast in this measurement is fairly low and the image noise level is extremely high. This brings a big challenge for cell tracking and counting algorithm.

Due to the high noise level and low appearance contrast for this measurement set, a multiple thresholding selections have been adopted in order to test algorithm’s performance respect to different parameters. As shown in Figure 3.6, with a low threshold, such as 99.93% of maximum, the performance could result in a sensitivity better than 90% with relatively high FAR as 1.5 / min. Increasing the threshold, such as 99.99% of maximum, the FAR could be decreased to as low as 0.04 / min, with the sacrificed sensitivity as 0.65. Therefore, it turns out the threshold is a critical parameter for our proposed algorithm. This trade-off between sensitivity and FAR could be made by turning the threshold and eventually the choice is based on different application requirements.
Figure 3.6: Performance statistics, in terms of sensitivity vs. FAR while adopting different thresholds. With a low threshold, such as 99.93% of maximum, the performance could result in a sensitive better than 90% with relatively high FAR as 1.5 / min. Increasing the threshold, such as 99.99% of maximum, the sensitive is 0.65 and the FAR could be decreased as low as 0.04 / min.

3.4 Summary

In this chapter, we discussed a new proposed dynamic invariance based cell tracking/counting algorithm for very noisy biology microscopic fluorescently labeled image sequences. After cell trajectories initialization, the data association has been applied based on dynamics analysis for each individual cell trajectory. Our experiments showed that operated on the phantom measurement, our algorithm could enjoy a combination of high sensitivity and low FAR, and even under the high noise circumstances, such as vivo measurement, our algorithm could still enjoy an up to 90% sensitivity performance. Depending on the various threshold...
values, which could be adopted by users based on specific application requirement, our algorithm could make the trade-off between the sensitivity and FAR in terms of performance. Due to the success of our association algorithm on cell tracking, we would like to extend this algorithm with a more advanced dynamics analysis and apply it to pedestrian tracking in order to provide necessary information for activity recognition, which would be the major topic for the rest of this thesis.
Chapter 4

Activity Recognition using Dynamic Subspace Angles

Cameras are ubiquitous everywhere and hold the promise of significantly changing the way we live and interact with our environment. Human activity recognition is central to understanding dynamic scenes for applications ranging from security surveillance, to video gaming without controllers. Most current approaches to solve this problem are based on the use of local temporal-spatial features that limit their ability to recognize long and complex actions. In this chapter, we propose a new approach to exploit the temporal information encoded in the data. The main idea is to model activities as the output of unknown dynamic systems evolving from unknown initial conditions. Under this framework, we show that activity videos can be compared by computing the principal angles between subspaces representing activity types which are found by a simple SVD of the experimental data. The proposed approach outperforms state-of-the-art methods classifying activities in the KTH dataset as well as in much more complex scenarios involving interacting actors.
4.1 Introduction

Activity recognition from video is central to many applications, including visual surveillance, assisted living for the elderly, and human computer interfaces. In recent years, a large number of researchers have addressed this problem as evidenced by several extended survey papers on this topic [17, 18, 19, 20].

Current approaches to modeling and recognizing actions of single actors can be classified into one of three major classes: nonparametric, volumetric, and parametric time-series approaches [20]. Nonparametric methods rely on features extracted at the frame level that are then matched against stored templates. The templates can be 2D (e.g., motion history images), 3D (e.g., generalized cylinders in the joint space-time \((x, y, t)\) domain), or use dimensionality reduction methods (e.g., PCA or manifold embeddings). Volumetric approaches process the video data as a volume of pixels and use local features that are 3D generalizations of standard image features such as corners and spatial-temporal filter responses. Indeed, a significant portion of the most recent work in activity recognition has been inspired by the success of using local features for object recognition [6, 7, 8, 9]. However, both, non-parametric and volumetric approaches are limited by the inherent local nature of the features used and the lack of strong relations among features across frames. In contrast, parametric time-series approaches use dynamical models of the motions to exploit temporal relations across frames. Thus, they are better equipped to model and recognize complex activities that last longer. Examples of parametric approaches include hidden Markov models (HMMs) and linear dynamical systems which can be thought of as a generalization of HMMs, where the state vector can take continuous values in \(\mathbb{R}^d\) and where \(d\) is the dimensionality of the state space. However, a drawback of these methods is that they must assume a dynamical model, which is often too simplistic, and that they must estimate the model parameters from extensive experimental data, often corrupted by noise.

In this chapter, we propose a time-series approach for activity recognition that, in contrast with previous approaches, requires neither assuming nor identifying a dynamical model. Instead, we simply hypothesize that the temporal data is the output trajectory of an underlying, unknown linear (possibly slowly varying) dynamical system. In this context, different realizations of the same activity correspond to trajectories of the same system in response to different
Chapter 4. Activity Recognition using Dynamic Subspace Angles

initial conditions. Exploiting the fact, derived from realization theory, that these trajectories are constrained to evolve in the same subspace (directly determined from the experimental data) allows for measuring the similarity between activities by simply computing the angle between the associated subspaces. While the approach outlined above works well for low levels of noise, its performance degrades substantially as the noise level increases. To improve robustness against noise, rather than directly clustering activities based on the angle of the corresponding subspaces, we first apply a discriminative canonical correlation \[53\] transformation to simultaneously decrease the inter-class and increase the intra-class distances. Finally, the resulting subspaces are used to train a support vector machine (SVM) to classify the activities.

The main result of this chapter, shows that the proposed approach outperforms existing ones when tested using a standard database (KTH) containing video clips of different single actor activities. Further, our approach can also handle, without modifications, the much more difficult case where the scenes contain multiple actors and activities are characterized by the interaction between agents, rather than individual activities.

This chapter is organized as follows. Section 5.2 gives the details of the proposed approach and Section 5.3 discusses experimental results comparing the proposed approach against previously reported results on activity recognition. Finally, Section 4.4 gives final remarks and discusses future directions.

4.2 Proposed Approach

We propose to model activities as the responses of unknown LTI systems with unknown initial conditions. In this scenario, two realizations of the same activity are explained by a single dynamical system with different initial conditions, while different activities are explained by different dynamical systems. Then, the problem of activity recognition reduces to:

Comparison of Output Trajectories Problem: *Given two temporal sequences, decide whether or not they can be explained as two output trajectories of the same dynamical system, possibly with different initial conditions.*
In the sequel, we show that the answer to this question can be found in the Hankel matrices of the output sequences under consideration. In particular, we show that in the noise-less case, all the output trajectories of a system, regardless of the initial condition, lie in a single subspace that can be used to represent the corresponding activity and that is easily determined from the Hankel matrix of the experimental data from a single realization. Based on this result, we propose to classify unknown activities by using canonical correlations to compare their associated subspaces to those obtained from training labeled data. To further improve robustness to noise and variations due to different actors performing the activities, classification is actually done by first transforming the data using the method proposed in [53] followed by a support vector machine (SVM) that selects the best matching subspace. The full details of the classification algorithm are given in Section 4.2.2.

4.2.1 Dynamic Subspaces Angles (DSA)

In this section we present the key observation that motivates this proposal: in the absence of noise, all the output trajectories of the dynamical system (2.1) lie on a single subspace that can be determined from the experimental data from a single realization.

**Theorem:** The principal angles between the subspaces spanning the columns of the Hankel matrices corresponding to trajectories of the same dynamical system in response to potentially different initial conditions are zero.

**Proof:** Let $H_y$ be a block Hankel matrix of an output trajectory of the dynamical system (2.1) when $w_k \equiv 0$. Then, using (2.1) we have

$$y_k = Cx_k = CAx_{k-1} = CA^2x_{k-2} = \cdots = CA^kx_o$$
Thus, the Hankel matrix $H_y$ can be rewritten as:

$$H_y = \begin{bmatrix} y_0 & y_1 & y_2 & \cdots & y_m \\ y_1 & y_2 & y_3 & \cdots & y_{m+1} \\ y_2 & y_3 & y_4 & \cdots & y_{m+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{m-1} & y_m & y_{m+1} & \cdots & y_{2m-1} \\ y_m & y_{m+1} & y_{m+2} & \cdots & y_{2m} \end{bmatrix} = \Gamma X \quad (4.1)$$

where

$$\Gamma = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^m \end{bmatrix} \quad \text{and} \quad X = \begin{bmatrix} x_0 & x_1 & \cdots & x_m \end{bmatrix}$$

and $X$ is a matrix containing the state trajectories as its columns. From (4.1) it follows that regardless of the initial condition, the columns of $H_y$ and $\Gamma$ span the same subspace. Hence, the principal angles between subspaces spanned from the columns of the Hankel matrices of output trajectories from the same system must be zero. q.e.d.

The significance of this result, as explained next, is that output trajectories can be compared in terms of dynamic subspaces angles (DSA), defined as the canonical correlations between the subspaces spanned by their Hankel matrices.

Note that in realistic situations, $w_k \neq 0$. In this case, the angle between subspaces for two realizations of the same activity will not be zero. However, since this angle is a continuous function of the entries of $H_y$, angles corresponding to (noisy) trajectories of the same system will still be small, when compared against the subspace angles of different systems.

To illustrate this effect, consider a simple one-dimensional version of (2.1) with $C = 1$, $A = 1$ and initial condition $x_0 = 1$. Thus, $y_k = x_k = 1$ for all $k$. Assume now that, due to noise, the first three measurements of $y$ yield $y_o = 0.95$, $y_1 = 0.975$ and $y_2 = 1.012$. It can be easily shown that the minimal order of a system required to generate these measurements is two.
Indeed, simple algebra shows that the sequence $y_k$ could have been generated by the triple

$$C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 2.5 & 1 \\ -1.5 & 0 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 0.95 \\ -1.4 \end{bmatrix}$$

Hence, a moderate amount of noise (less than 5%) can lead to substantial error in estimating the dynamics and initial condition. On the other hand, the column subspaces of the nominal and noisy Hankel matrices are given by:

$$U_{\text{nom}} = \begin{bmatrix} -0.71 & -0.71 \\ -0.71 & 0.71 \end{bmatrix}, \quad U_{\text{noisy}} = \begin{bmatrix} -0.70 & -0.72 \\ -0.72 & 0.70 \end{bmatrix}, \quad (4.2)$$

with the angle between the first vector of each subspace $\sim 0.014$. Thus, using subspace angles instead of identifying and comparing dynamical models provides substantial robustness against errors in estimating $y$. Applying a canonical correlation maximizing transformation [53] to the subspaces prior to computing the angles results in an even larger separation, allowing for successfully classifying activities from realistic video sequences, where $y$ is affected by noise, and correspondence errors.

### 4.2.2 Activity Recognition Using DSA

In this section we describe the details for training and testing a system to recognize activities using DSA, given a labeled database consisting of $c$ classes of actions, with $N_k$ sample videos for class $k$, with $k = 1, \ldots, c$ and $N = \sum_k N_k$. Figure 4.1 shows a diagram illustrating all the required steps.

**Training Procedure**

1. **Feature Extraction and Tracking**

   The first step is to collect temporal features from all the videos in the training database. The proposed method can be used with different type of features, the only requirement is that they are temporally ordered. They can be, for example, a set of point features
and/or dimensions of bounding boxes tracked across frames, or HOGs values in a set of bins across time. In the sequel we will use $y^{(i)}_{ij} \in \mathbb{R}^n$ to denote the feature vector extracted from video $i$ at frame $j$.

2. Hankel Matrices Assembly

Next, the measurements for each video are collected in a Hankel matrix representing its temporal information. Let $H_i^{(k)} \in \mathbb{R}^{mn \times (F_{ki} - mn)}$ denote the Hankel matrix for video $i$ of class $k$, $i = 1, \ldots, N_k$, and $k = 1, \ldots, c$ where $m$ is the number of row blocks and
$F_{ki}$ is the number of frames in the video. Thus, there are a total of $N$ Hankel matrices: 
\{\{H_1^{(1)}, \ldots, H_{N_1}^{(1)}\}, \ldots, \{H_1^{(c)}, \ldots, H_{N_c}^{(c)}\}\}.

3. DCC

The discriminant function for canonical correlations among the subspaces spanned by the columns of the Hankel matrices \{\{H_1^{(1)}, \ldots, H_{N_1}^{(1)}\}, \ldots, \{H_1^{(c)}, \ldots, H_{N_c}^{(c)}\}\} is computed using the algorithm in [53] (repeated here for completeness):

(a) Find an orthogonal column basis for each Hankel matrix: Let $H_i^{(k)} P_i^{(k)} T \approx P_i^{(k)} \Lambda_i^{(k)} P_i^{(k)T}$ where $P_i^{(k)} \in \mathbb{R}^{mn \times d}$ and $\Lambda_i^{(k)}$ are the eigenvalue and eigenvector matrices of the $d$ largest eigenvalues, respectively.

(b) Find an orthogonal transformation matrix $T \in \mathbb{R}^{mn \times q}$ with $q \leq mn$:

i. $T \leftarrow I_{mn}$

ii. Do iterate the following:

iii. For all $i$ do QR-decomposition: $T^T P_i = \Phi_i \Delta_i \rightarrow P_i' = P_i \Delta_i^{-1}$

iv. For every pair $i, j$ do SVD $P_i'^T T^T P_j' = Q_{ij}' \Lambda_{ij}' Q_{ij}'^T$

v. Compute $S_b = \sum_{i=1}^N \sum_{l \in B_i} (P_i' Q_{li} - P_i' Q_{il})(P_i' Q_{li} - P_i' Q_{il})^T$, where $B_i = \{j | H_j \notin C_i\}$

vi. Compute $S_w = \sum_{i=1}^N \sum_{l \in W_i} (P_i' Q_{ki} - P_i' Q_{ik})(P_i' Q_{ki} - P_i' Q_{ki})^T$, where $W_i = \{j | H_j \in C_i\}$

vii. Compute $mn$ eigenvectors of $\{t_i\}_{i=1}^{mn} S_w^{-1} S_b$, $T \leftarrow [t_1, \ldots, t_{nm}]$

viii. End

ix. $T \leftarrow [t_1, \ldots, t_q]$

(c) Apply $T$ to the left orthogonal matrix of every Hankel matrix: $P_i'^{(k)} = T^T P_i^{(k)}$

4. SVM Training

Train a multi-class support vector machine using the first $r$ columns of $P_i'^{(k)}$, $i = 1, \ldots, N_k$, as sample feature vectors for class $k$, $k = 1, \ldots, c$.

**Testing Procedure** The steps to classify a video sequence are:
1. Collect features.

2. Assemble the Hankel matrix of the measurements, $H_y$.

3. Compute the svd of $H_y H_y^T = PD P^T$.

4. Apply the DCC Transformation $T$ to $P$: $P' = T^T P$.

5. Use the trained SVM to assign a label based on the first $r$ columns of $P'$.

### 4.3 Experiments

#### 4.3.1 KTH Database

The proposed approach was tested using six types of human activities (walking, running, boxing, hand waving, hand clapping, and jogging) from the widely used KTH activity dataset [55]. 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. Unfortunately, comparing performance results against results published in the literature is difficult, since different authors use different experimental protocols [71]. We chose to follow the most commonly used experimental protocol which was described in the original paper [55]. This protocol partitions the data into a training set (subjects: 1,4,11,12,13,14,15,16), a validating set (subjects: 17,18,19,20,21,23,24 25) to tune system parameters, and a testing set (subjects: 2,3,5,6,7,8,9,10,22) to evaluate performance.

#### 4.3.1.1 Feature Extraction

A variety of possible features can be used to capture the dynamics of the activities. For these experiments we chose to use the two strongest connected components of the response to a Gabor 3D spatial-temporal filter proposed in [6] and the width of the bounding box (measured with respect to the centroid of the person performing the activity), tracked across
Figure 4.2: Feature Extraction: Three frames from the KTH database for hand waving and walking and their responses to a Gabor 3D spatial-temporal filter. The two strongest clusters are tracked over time (red and green trajectories).

The duration of the video. The response function has the form

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$$

where $g$ is a 2D Gaussian spatial smoothing kernel and $h_{ev} = -\cos(2\pi t \omega)e^{-t^2/\tau^2}$ and $h_{od} = -\sin(2\pi t \omega)e^{-t^2/\tau^2}$ are a quadratic pair of 1D Gabor filters. In our experiments we used $\sigma = 0.8$ and $\tau = 1.2$. Features were tracked using a LK tracker. Figure 4.2 shows sample frames for two activities, their filter response and the tracks for the two strongest clusters.

Finally, the feature vector $y \in \mathbb{R}^6$ for each frame consists of the following six numbers: the two coordinates for each of the two strongest clusters, the distance between the left side of the bounding box and the person’s centroid, and the distance between the person’s centroid and the right side of the bounding box.

---

1. All measurements are normalized with respect to the height of the bounding box to make them invariant to different people’s height.
4.3.1.2 Hankel Matrix Assembly

The Hankel matrices were assembled using the features from all the frames. The dimensions were chosen such that the Hankel matrices are as square as possible. Since the average number of frames per video in the database is 200 and the dimension of the feature vector $n = 6$, we chose $m = 24$ and hence, the Hankel matrices have $mn = 144$ rows and a variable number of columns.

4.3.1.3 DCC

The number of vectors for the Hankel matrix bases, $d$, was set to 21. Figure 4.3 shows a plot of the singular values for the Hankel matrices for the videos in the training and validating sets.
sets. The figure shows that the singular values decay quickly and that the energy for beyond the 21\textsuperscript{th} singular value is negligible.

4.3.1.4 SVM Training

We use a non-linear support vector machine with 2\textsuperscript{nd} degree inhomogeneous poly-kernel using a one-against-rest approach from the SVM toolbox [72]. The number of vectors $r$ from the transformed bases was chosen by varying it from 1 to 32 and evaluating the classification performance using the validation set. The best performance was obtained for $r = 1$ (See Figure 4.4).
4.3.1.5 Tests and Performance Evaluation

We conducted three tests with the KTH dataset. The first one, tested the trained system using the testing set as indicated by the experimental protocol in [55]. The second test was to evaluate the benefit of using DCC to increase performance. Finally, the third test was designed to test whether using the structure and information encoded in the Hankel matrices provided any added value over using DCC on a set of vectors formed by the tracked features in sub-sequences of the videos.

Performance Evaluation The overall accuracy rate of the proposed approach is 93.6%. Tables 5.2 and 4.2 show that this performance is better than previously reported performances using the same experimental protocol. The inter-class confusion matrix using the test set is given in Table 5.1.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>93.06</td>
</tr>
<tr>
<td>Wang et al. [61]</td>
<td>92.1</td>
</tr>
<tr>
<td>Laptev et al. [9]</td>
<td>91.8</td>
</tr>
<tr>
<td>Niebles et al. [8]</td>
<td>91.3</td>
</tr>
<tr>
<td>Wong et al. [73]</td>
<td>86.7</td>
</tr>
<tr>
<td>Schuldt et al. [55]</td>
<td>71.5</td>
</tr>
</tbody>
</table>

Benefits of Using DCC We measured the performance of the system when skipping the DCC step (in effect, setting the transformation $T$ as the identity). It was observed that the overall performance drops to 89.35% if DCC is not used.
TABLE 4.3: Inter-class confusion matrix for KTH (testing sets) using the proposed approach.

<table>
<thead>
<tr>
<th></th>
<th>Boxing</th>
<th>HClapping</th>
<th>HWaving</th>
<th>Walking</th>
<th>Running</th>
<th>Jogging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxing</td>
<td>97.22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.78</td>
<td>0</td>
</tr>
<tr>
<td>HClapping</td>
<td>2.78</td>
<td>94.44</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.78</td>
</tr>
<tr>
<td>HWaving</td>
<td>0</td>
<td>13.89</td>
<td>86.11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walking</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>97.22</td>
<td>0</td>
<td>2.78</td>
</tr>
<tr>
<td>Running</td>
<td>0</td>
<td>0</td>
<td>2.78</td>
<td>0</td>
<td>83.33</td>
<td>13.89</td>
</tr>
<tr>
<td>Jogging</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

FIGURE 4.5: Advantage of using Hankel matrices over using DCC with sub-sequences to recognize KTH activities.

Benefits of Using Hankel Matrices  

Kim et al. [53] proposed to use DCC for object and object category recognition where the data consist of rasterized images, captured under different settings such as different illuminations and viewpoints. One could try to use the same approach to perform activity recognition by applying it to the frames in the activity videos. In this case, the classes are the activities, and each frame is considered as a sample of an
activity. However, this approach would not fully exploit the temporal information since DCC is invariant to the ordering of the data. Thus, a better way of using DCC with temporal data would be to cut each video into small sub-sequences, rasterize each sub-sequence into a vector, and apply DCC to these vectors. In this way, each sample would capture a snippet of temporal information. Note that from an implementation point of view, the difference between this approach and the proposed one in this chapter is minimal: when we use Hankel matrices, each sample vector corresponds to a sub-sequence (a column of the Hankel matrix) that completely overlaps other sample sub-sequences (the previous and next columns of the Hankel matrix), except for two frames\(^2\). However, it should be emphasized that the effect of this seemingly minor difference is quite significant as shown in Figure 4.5, where the performance of the two approaches in classifying the KTH test set as the number of basis vectors \(d\) is varied are compared. There, the best performance using DCC alone is 81.48% while using Hankel matrices together with DCC achieves a performance of 93.6%.

It should be noted, that Kim and Cipollal [74] proposed a different generalization to DCC for handling temporal data through tensors. However, using their method to recognize activities requires significant down-sampling of the data due to large computational requirements and manual alignment of the videos.

### 4.3.2 TV Interaction Database

The proposed approach was also tested with the more challenging TV Interaction dataset [75] to classify two types of human interactions (hand shaking and high-five). The database has 50 videos of each type that are short clips from TV sitcoms. Figure 4.6 (a) shows sample frames illustrating the level of clutter and scene complexity in this database.

#### 4.3.2.1 Feature Extraction

Due to the level of clutter and ego-motion in the videos, we chose to use as features the histogram of gradients (HOG) and track the temporal evolution of the angles of the gradients.\(^2\)

\(^2\)In our experiments, this corresponds to an overlap of 22 frames out of a 24 frames sub-sequence.
We computed HOG for each actor (using the bounding boxes provided in the dataset) using a $16 \times 8$ grid, as shown in Figure 4.6 (b). Then, the feature vector for frame $k$ is a vector $y \in \mathbb{R}^{256}$ made up of the 128 HOG angles for each actor.

### 4.3.2.2 Hankel Matrix Assembly, DCC, and SVM Steps

The Hankel matrices were assembled using the HOG features using $m = 4$. Hence, they have $mn = 1024$ rows and a variable number of columns depending on the number of frames in the clip. For the estimation of $T$ during the DCC step, we used $d = 6$ and we kept one vector ($r = 1$) for the SVM training.

### 4.3.2.3 Performance Evaluation

We tested the classification performance following the experimental protocol used by the creators of the database [76] achieving an overall performance of 68% which is significantly higher compared to the performance of 54.45% reported in [76].

### 4.4 Summary

We proposed a new time-series approach for human activity recognition that does not need to identify a model. Instead, it exploits the dynamic information encoded in the structure of

![Figure 4.6: (a) Sample frames from the TV Interaction Database [75]. (b) HOG features.](image)
Hankel matrices built from the data. We showed that trajectories corresponding to the same activity live in a subspace and that the DSA – i.e. the canonical correlations between the associated subspaces – can be used to classify the activities. Our experiments show that both, using DCC to increase the separation between classes, and using Hankel matrices to capture the temporal information, result in significant improvements of the overall classification performance. The proposed approach was tested with the KTH database and the much more challenging TV Interaction database, achieving an overall performance of 93.6% and 68%, respectively, which are significantly higher than the highest reported performance using the same experimental protocols. In the future, we plan to explore the effect of using different types of features, multiple view activities and the possibility of using the dynamic subspaces as generative models.
Chapter 5

Cross-view Activity Recognition using Hankelets

Most approaches addressing activity recognition 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 chapter, we introduce a new type of feature, the “Hankelet” that captures dynamic properties of short tracklets. While Hankelets 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 Hanklets improves the state of the art performance by over 20%.

5.1 Introduction

In this chapter, we propose Hankelets – the Hankel matrix of a short tracklet – as a new feature to use with a BoF approach to recognize activities across different viewpoints. Hankelets 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 Hankelets 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 [24] 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 [77] 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 chapter is organized as follows. Section 5.2 gives the details of the proposed approach and Section 5.3 discusses experimental results comparing the proposed approach against previously reported results. Finally, Section 5.4 gives final remarks.
5.2 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 [13] and motivated by the affine invariance property of Hankel matrices introduced in [51]. 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 [13] 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 throughout 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 [13] 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 [13] becomes prohibitive as the number of Hankel matrices increases.

5.2.1 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 densely distributed short (typically 15 frames) trajectories of features sampled on a grid, tracked at different

\[1\] Each comparison requires estimates of the ranks of the Hankel matrices and three singular value decompositions.
scales. The trajectories consist of a set of temporally ordered 2D normalized velocities

\[
\frac{1}{\sum_{j=t}^{t+L-1} \| \Delta p_j \|} (\Delta p_t, \Delta p_{t+1}, \ldots, \Delta p_{t+L-1})
\]

where \( L + 1 \) is the number of tracked frames and \( \Delta p_t = (x_{t+1} - x_t, y_{t+1} - y_t)^T \) is a vector with the two components of the velocity of the tracked feature at time \( t \) \([12]\). 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 (2.2) and normalizing them using the Frobenius norm (\( \| M \|^2_F = \text{trace}(M^T M) \)):

\[
\hat{H}_p = \frac{H_p}{\| H_p H_p^T \|^{1/2}_F}
\]

### 5.2.2 Comparing Hankelets

Given two Hankelets \( \hat{H}_p \) and \( \hat{H}_q \) 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 \([13]\) 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 data\(^2\). To avoid this problem, in this

---

\(^2\)To 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 \( H_{\hat{p}} \) and \( H_{\hat{q}} \) will have full column rank. Hence the angle between the subspaces of the noisy Hankelets is zero, even if \( p \) and \( q \) correspond to different activities.
Chapter 5. Cross-view Activity Recognition using Hankelets

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

Proposal we will use the following dissimilarity score function to compare Hankelets:

\[ d(\hat{H}_p, \hat{H}_q) = 2 - \|\hat{H}_p\hat{H}_p^T + \hat{H}_q\hat{H}_q^T\|_F \]  

(5.1)

The intuition behind this definition is to exploit the triangle inequality to capture the degree of “alignment” of the column subspaces of \( \hat{H}_p \) and \( \hat{H}_q \) in a computationally efficient 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 \( \hat{H}_p = U_p\Sigma_pV_p^T \), \( \hat{H}_q = U_q\Sigma_qV_q^T \), 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:

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

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.
5.2.3 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 \( D_w = \{ d_{w1}, d_{w2}, \ldots, d_{wn} \} \) be the dissimilarity scores for all the Hankelets in cluster \( w \) with respect to an arbitrarily selected Hankelet in the same cluster and \( \mu_w \) be their mean. Then, the Hankelet in the cluster that has the dissimilarity score closest to \( \mu_w \) is selected as the “representative” or “center” of the cluster. Figure 5.3 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:

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

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

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

5.2.4 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
Figure 5.4: 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.

the label of the cluster with the maximum \textit{a posteriori} probability:

\[
\text{label}(\hat{H}) = \arg \max_w p(d(\hat{H}, \hat{H}_w)|w)P(w)
\]  

(5.2)

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.

5.2.5 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 [51] 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 enough\textsuperscript{3} to disregard perspective distortion effects. Using the multi-lingual analogy introduced in [28], 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 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 5.4 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 5.4 and corroborated by our experiments in Section 5.3, 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

\textsuperscript{3}This is usually the case in most surveillance systems.
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.

### Table 5.1: Classification accuracy for KTH (testing sets) using Hankelets.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boxing</td>
<td>95.71</td>
</tr>
<tr>
<td>HClapping</td>
<td>95.48</td>
</tr>
<tr>
<td>HWaving</td>
<td>99.09</td>
</tr>
<tr>
<td>Walking</td>
<td>99.52</td>
</tr>
<tr>
<td>Running</td>
<td>91.52</td>
</tr>
<tr>
<td>Jogging</td>
<td>94.05</td>
</tr>
</tbody>
</table>

### 5.2.6 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 [29]. 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 (5.1). 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 (5.2)$^4$. Each video is represented using a BoBHk.

---

$^4$If the a posteriori probability is below a threshold, the Hankelet is not used.
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 \((5.2)^4\). 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.

**5.3 Experimental Results**

The proposed approach was tested on the IXMAS multi-view action data set [24] 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.). 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 [55].

**5.3.1 Implementation Details**

We use 15 frames long tracklets extracted with the code provided by the authors of [12]. A typical video has approximately 15,000 tracklets with an average of 20 tracklets per frame. The tracklets are then assembled into \(16 \times 8\) Hankelets. To compute the dissimilarity scores between Hankelets we use a fast implementation of \((5.1)\) 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.

### 5.3.2 Single View Using BoHks

#### 5.3.2.1 KTH dataset

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 [55]. Table 5.1 shows the recognition accuracy for the six activities using Hankelets and Table 5.2 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>
<thead>
<tr>
<th>Algorithm</th>
<th>Perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>95.89</td>
</tr>
<tr>
<td>Cao et al. [78]</td>
<td>95.02</td>
</tr>
<tr>
<td>Wang et al. [12]</td>
<td>94.2</td>
</tr>
<tr>
<td>Le et al. [79]</td>
<td>93.9</td>
</tr>
<tr>
<td>Li et al [13]</td>
<td>93.6</td>
</tr>
</tbody>
</table>

Table 5.2: Comparison of overall performance for KTH dataset using experimental protocol defined in [55]
5.3.2.2 UCF Sports Dataset

We also tested the use of BoHks with MBH [80] appearance feature on UCF sports dataset [81], which contains 10 sports activities: swinging (pommel horse and on the floor), diving, kicking, weight-lifting, horse-riding, running, skateboarding, swing (high bar), golf and walking. The experiments were done following the most commonly used protocol [61], which added horizontally flipped version of each video added to the dataset and applies leave-one-out experimental setting. Table 5.3 compares the overall performance to the state-of-art. There we can see that using Hankelets with MBH appearance feature, our approach could handle challenging activity dataset with very large inner variance. It's worth to mention that [12] utilizes a feature combination (Trajectory + HOG + HOF + MBH), whereas our algorithm only uses our Hankelet + MBH. From the computation perspective, our algorithm could be more efficient.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>89.33</td>
</tr>
<tr>
<td>Wang et al. [12]</td>
<td>88.2</td>
</tr>
<tr>
<td>Kovashka et al. [82]</td>
<td>87.27</td>
</tr>
<tr>
<td>Klaser et al [83]</td>
<td>86.7</td>
</tr>
<tr>
<td>Wang et al. [61]</td>
<td>85.6</td>
</tr>
</tbody>
</table>

5.3.3 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 5.4 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 [28], 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 5.4: 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 [28], respectively.

<table>
<thead>
<tr>
<th>Cam 0</th>
<th>Cam 1</th>
<th>Cam 2</th>
<th>Cam 3</th>
<th>Cam 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>83.70</td>
<td>14.40</td>
<td>56.00</td>
<td>10.69</td>
</tr>
<tr>
<td>A</td>
<td>61.58</td>
<td>11.11</td>
<td>62.75</td>
<td>7.41</td>
</tr>
<tr>
<td>Ours</td>
<td>84.27</td>
<td>16.12</td>
<td>61.45</td>
<td>11.80</td>
</tr>
<tr>
<td>A</td>
<td>65.17</td>
<td>11.80</td>
<td>62.75</td>
<td>7.41</td>
</tr>
<tr>
<td>Ours</td>
<td>57.05</td>
<td>11.15</td>
<td>61.45</td>
<td>8.59</td>
</tr>
<tr>
<td>A</td>
<td>61.45</td>
<td>8.59</td>
<td>68.12</td>
<td>9.22</td>
</tr>
<tr>
<td>Ours</td>
<td>39.60</td>
<td>8.80</td>
<td>68.12</td>
<td>9.22</td>
</tr>
<tr>
<td>A</td>
<td>32.84</td>
<td>8.80</td>
<td>68.12</td>
<td>9.22</td>
</tr>
</tbody>
</table>

5.3.4 Cross-view Using BoBHks

In this section we report the results of experiments testing the effect of using dictionaries of bi-lingual Hankelets on the IXMAS dataset. As described in Section 5.2.6 in this case, only bi-lingual Hankelets 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 5.5. 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 5.6 gives side by side the average accuracy using BoBHks with the accuracy of previous approaches [28, 29, 30, 84]. The overall average accuracy using BoBHks is 90.57%, a 20.28% improvement over the state of the art performance reported in [28] and a 60.58% improvement over using BoHk.
In this chapter 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
Table 5.6: 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, [28]'s approach, [29]'s approach, [30]'s approach, and [84]'s approach, respectively. The overall average accuracies are 90.57%, 75.3%, 58.1%, 59.5%, and 74.4%, respectively.

<table>
<thead>
<tr>
<th>Cam 0</th>
<th>Cam 1</th>
<th>Cam 2</th>
<th>Cam 3</th>
<th>Cam 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(%)</td>
<td>Ours</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Cam 0</td>
<td></td>
<td>94.8</td>
<td>79.9</td>
<td>72.9</td>
</tr>
<tr>
<td>Cam 1</td>
<td>89.2</td>
<td>61.2</td>
<td>69.8</td>
<td>77.1</td>
</tr>
<tr>
<td>Cam 2</td>
<td>91.3</td>
<td>79.6</td>
<td>88.1</td>
<td>71.7</td>
</tr>
<tr>
<td>Cam 3</td>
<td>89.9</td>
<td>78.6</td>
<td>89.4</td>
<td>76.0</td>
</tr>
<tr>
<td>Ave.</td>
<td>91.7</td>
<td>79.6</td>
<td>89.4</td>
<td>76.0</td>
</tr>
</tbody>
</table>

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. Due to Hankelet’s superficial performance on cross view activity recognition problem, we would propose to generalize Hankelets in other topics, such as re-identification challenges.
Chapter 6

Multi-camera Human Tracking

6.1 Introduction

Due to the limited field of view and occasionally object occlusions, it is common to deploy a camera network in large public facilities, such as airport and government buildings. How to effectively analyze the "big data" collected from camera network is becoming an important topic for computer vision researchers. One of the very first and most important topic is: how to effectively associate the observations of cameras with non-overlapping field-of-view in such surveillance network.

This cross camera association challenge has also been proposal and studied in object recognition community in terms of developing more distinguishable appearance features among multiple candidates. The methods to solve this challenge can be divided into the following three groups[85]: (a) descriptive methods [35, 86, 87, 88], (b) discriminative methods [89, 90, 91, 92, 93, 94, 95], (c) metric learning methods [96, 97]. Descriptive methods attempt to extract discriminative and stable visual features across cameras to distinguish between different pedestrians. However, due to large intra-class variations between cameras, stable and powerful visual features is hard to computer under realistic conditions. In order to overcome this disadvantage, discriminative methods have been proposed to learn the inter / intra-class structure given by the data. However, this method is prone to over-fitting models to
the training data. A mild way between the previous two methods is the metric learning, which leads to complex optimization problems associated with high computational costs. The same drawback as discriminative method, metric learning would also fall into over-fitting problem according to the provided training data. This chapter will demonstrate the advantage of integrating a state-of-art descriptive feature with a Bayesian probabilistic estimation framework.

This chapter is organized as follows. Section 6.2 gives a brief summary of the descriptive feature: region covariance. Section 6.3 provides a Bayesian probabilistic framework to estimate both the target’s spatial and temporal location across a non-overlapping camera network. Finally, Section 6.4 leads to the conclusion and further work discussion.

6.2 Background

6.2.1 Region Covariance

Region Covariance, was proposed in [98] to capture the characterization of a spatial region of interest with $d$ specific features by encoding the covariance between each of the $d$ features. In [33], region covariance is validated as superior regards of rotation, illumination and other appearance variations, when compared to other standard local features.

Region covariance, as a positive semi-definite symmetric matrix, it’s lying on a Riemannian manifold instead of Euclidean space [99]. Based on the differential geometry analysis framework in [100], the comparison between two different region covariance matrices requires special operations: the most commonly used metric, Eq. 6.1, has been adopted for matrices comparison [99].

$$\rho(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)}$$ (6.1)
6.2.1.1 Flip Invariant Region Covariance

Due to appearance variance for different viewpoints, regular region covariance does not perform well across different views. In order to employ view-invariant characteristics with the region covariance matrix, we propose the following three modifications to address this issue.

1. Relative $x$ indexing: as the relative distance from the center of the bounding box, instead of absolute index starting from the left boundary.

2. Absolute value of the first order derivative in $x$ (horizontal) direction, this will enhance horizontal flip invariant.

3. Resize cropped image window into fixed size. Here we set the standard window size as: $[105 \ 65]$ according to our experiment video resolution.

6.2.1.2 Implementation Details

In our experiment, in Section 6.3.2, $x$ $y$ coordinate, RGB value and first order derivative as in Equation 6.2, have been used to compute the region covariance matrices by applying Equation 6.3, which is derived from [98].

$$F(x, y) = \left[ \|x - x_0\| \ y \ R \ G \ B \ \|I_x\| \ I_y \right]^T \quad (6.2)$$

where $x_0$ is the $x$ coordinate of the center of bounding box, $I_x$ and $I_y$ are the first order derivative in $x$ and $y$ directions.

$$C_{R(x',y',x'',y'')} = \frac{1}{n-1} \left[ Q_{x'',y''} + Q_{(x'-1),(y'-1)} - Q_{(x''-1),y'} - Q_{x',(y''-1)} ight. \hspace{10mm} \left. - \frac{1}{n} \left( P_{x'',y''} + P_{(x'-1),(y'-1)} - P_{(x''-1),y'} - P_{x',(y''-1)} \right) \right] \hspace{10mm} (6.3)$$

where $P$ denotes the $W \times H \times d$ tensor of the first order integral images and $Q$ is the $W \times H \times d \times d$ tensor of the second order integral images, as described in [98].
6.3 Appearance representation using region covariance

6.3.1 Proposed framework

The re-identification of pedestrians comprises of two major challenges: suspect tracking in single view and the association across different views. The person of interest will be selected by a human CCTV operator in one view and some state-of-art tracking algorithms will track the object throughout the view. During the tracking our algorithm will collect suspect’s distinguishable information to associate suspect in the other views. However, in order to effectively apply the association operation, a candidate filtering employing dynamic information will greatly improve the association performance in terms of accuracy and computation cost.

6.3.1.1 Spatial-temporal estimation for re-identification

Based on both appearance and temporal information gathered in the initial viewpoint, a Bayesian probabilistic based object arrival estimation framework, inspired by [101], using region covariance matrices will be proposed.

In case of only one object being present in the surveilled area, the data association problem becomes trivial. The statistical information for objects traveling between different views will stay the same. Therefore, it is a good choice to automatically learn our framework in such circumstances.

The framework to compute the probability of associating a candidate in a target view with an object of interest in the initial view is given in Equation 6.4.

\[
\max_{d_i} p(d_i|C_{tgt}) = \max_{d_i} \frac{p(C_{tgt}|d_i)p(d_i)}{p(C_{tgt})} \tag{6.4}
\]

where, \(d_i\) denotes the \(i\)th detected person, similar to Eq. 6.1, \(p(C_{tgt}|d_i)\) equals to Equation 6.5 for region covariance based appearance comparison, \(p(d_i)\) is following a gaussian distribution
Figure 6.1: Manually selected regions across different views. Red box, stands for person object region. Blue box, stands for random background regions.

\[ p(d_i) = g(t_{ini} + 10s; 0.6s) \]. The \( p(C_{tgt}) \) represents the topology of the camera network, which could be given or learnt within one object only scenarios mentioned before.

\[ p(C_{tgt}|d_i) = \frac{1}{\sqrt{\sum \ln^2 \lambda(C_{tgt}, C_{d_i})}} \]  \hspace{1cm} (6.5)

6.3.2 Experiments

6.3.2.1 Cross-view Validation

In order to emphasis the capability of region covariance for cross view association, our feature has been tested on IXMAX dataset. In order to evaluate the performance of the algorithmically derived cross-view association, the IXMAX dataset was manually labeled to generate a ground truth. The employed labels comprised both the objects’ visual extent (bounding boxes; examples shown in Figure 6.1.) as well as identifications.

Flip Invariant Region Covariance Comparison The comparison between original and flip invariant region covariance, as mentioned in Section 6.2.1, is shown in Table 6.2. It is readily apparent that the flip invariant modification indeed improve the dissimilarity score, as defined in Eq.6.1, for the same person from different viewpoints.

The association difficulties in terms of dissimilarity score between different cameras is agreed to the recognition difficulties as observed in [102]. The top view (cam4 in Figure 6.1) is always the hardest and most different view comparing to all the other views.
Chapter 6. *Multi-camera Human Tracking*

### Figure 6.2: Region Covariance Comparison across different cameras. It could be noticed that if the angle between two cameras are relatively small, the dissimilarity score is also small, vice versa. The dissimilarity score is improved for flip invariant modification comparing to the original region covariance definition.

<table>
<thead>
<tr>
<th>Camera0</th>
<th>Object</th>
<th>Random1</th>
<th>Random2</th>
<th>Random3</th>
<th>Random4</th>
<th>Random5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cam0</td>
<td>0.0000</td>
<td>3.3609</td>
<td>4.2299</td>
<td>3.5711</td>
<td>5.7794</td>
<td>3.1420</td>
</tr>
<tr>
<td>cam1</td>
<td>1.6651</td>
<td>4.1097</td>
<td>4.6089</td>
<td>3.8254</td>
<td>6.3832</td>
<td>3.9948</td>
</tr>
<tr>
<td>cam2</td>
<td>2.5637</td>
<td>3.3296</td>
<td>2.5511</td>
<td>4.2476</td>
<td>4.1075</td>
<td>3.9635</td>
</tr>
<tr>
<td>cam3</td>
<td>1.9698</td>
<td>3.7446</td>
<td>5.3856</td>
<td>3.7523</td>
<td>6.8131</td>
<td>3.1656</td>
</tr>
<tr>
<td>cam4</td>
<td>3.4690</td>
<td>4.2409</td>
<td>3.6679</td>
<td>4.5668</td>
<td>5.5380</td>
<td>4.8219</td>
</tr>
</tbody>
</table>

**Object vs. Random Background**  Even though, applying flip invariant modification could improve the dissimilarity score for the same target across different viewpoints, it is not sufficient to illustrate whether we would find the object out of the whole image.

**Experiment Design**  In order to compare the dissimilarity between object and background, we manually selected five random background regions from each camera, shown in Figure 6.1 by blue rectangles. Results for all cameras using the proposed flip invariant modification are shown in Table 6.3-6.7.

### 6.3.2.2 Real Transportation Security Administration (TSA) Dataset Validation

In this section, the region covariance feature is tested with surveillance footage recorded in the facilities of a major airport located in the United States of America. The major challenges
of such a scenario are large field of view camera setup, realistic illumination variances across cameras, and most importantly heavily crowded scenes. The distance distribution, which is the basis for the probability model introduced in Section 6.3.1 has been estimated between initial and target camera. In order to explore the effective feature combinations, different color schemes, RGB, HSV and LAB are tested in two circumstances: same person within

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Camera1} & \text{Object} & \text{Random1} & \text{Random2} & \text{Random3} & \text{Random4} \\
\hline
\text{cam0} & 1.6651 & 4.4799 & 4.6749 & 2.4344 & 5.4825 \\
\text{cam1} & 0.0000 & 3.8220 & 5.2181 & 2.6532 & 6.0418 \\
\text{cam2} & 2.6699 & 5.4360 & 3.4298 & 4.4559 & 4.4355 \\
\text{cam3} & 2.5093 & 3.6255 & 5.7669 & 1.9405 & 6.5169 \\
\text{cam4} & 2.9122 & 4.7021 & 4.2724 & 4.1536 & 3.0977 \\
\hline
\end{array}
\]

**Figure 6.4:** Templates from all cameras compare to all regions from camera 1. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the second row of Figure 6.2.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Camera2} & \text{Object} & \text{Random1} & \text{Random2} & \text{Random3} & \text{Random4} \\
\hline
\text{cam0} & 2.5637 & 3.7577 & 5.6624 & 3.5454 & 4.1316 \\
\text{cam1} & 2.6699 & 4.4134 & 6.2195 & 4.4559 & 4.4355 \\
\text{cam2} & 0.0000 & 3.9734 & 3.9089 & 3.4968 & 2.3574 \\
\text{cam3} & 3.7776 & 4.6379 & 6.6780 & 4.5407 & 5.2462 \\
\text{cam4} & 2.1932 & 4.7517 & 5.2853 & 4.7536 & 3.4534 \\
\hline
\end{array}
\]

**Figure 6.5:** Templates from all cameras compare to all regions from camera 2. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the third row of Figure 6.2.

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
\text{Camera3} & \text{Object} & \text{Random1} & \text{Random2} & \text{Random3} & \text{Random4} \\
\hline
\text{cam0} & 1.9698 & 4.6329 & 5.1325 & 4.1477 & 4.7600 \\
\text{cam1} & 2.5093 & 4.9228 & 5.7230 & 3.4408 & 5.1539 \\
\text{cam2} & 3.7776 & 3.5410 & 3.4991 & 5.1255 & 3.0381 \\
\text{cam3} & 0.0000 & 5.4557 & 6.0650 & 3.1825 & 5.7771 \\
\text{cam4} & 4.1100 & 3.8417 & 4.7538 & 4.2392 & 3.8097 \\
\hline
\end{array}
\]

**Figure 6.6:** Templates from all cameras compare to all regions from camera 3. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the fourth row of Figure 6.2.
Chapter 6. *Multi-camera Human Tracking*

<table>
<thead>
<tr>
<th>Camera4 Object</th>
<th>Random1</th>
<th>Random2</th>
<th>Random3</th>
<th>Random4</th>
<th>Random5</th>
</tr>
</thead>
<tbody>
<tr>
<td>cam0</td>
<td>3.4690</td>
<td>6.6872</td>
<td>5.7980</td>
<td>5.6489</td>
<td>4.1493</td>
</tr>
<tr>
<td>cam1</td>
<td>2.9122</td>
<td>7.2460</td>
<td>6.0597</td>
<td>5.9102</td>
<td>4.4389</td>
</tr>
<tr>
<td>cam2</td>
<td>2.1932</td>
<td>5.1959</td>
<td>3.7647</td>
<td>3.7138</td>
<td>2.3514</td>
</tr>
<tr>
<td>cam3</td>
<td>4.1100</td>
<td>7.7483</td>
<td>6.9728</td>
<td>6.8097</td>
<td>5.1951</td>
</tr>
<tr>
<td>cam4</td>
<td>0.0000</td>
<td>6.0484</td>
<td>4.4753</td>
<td>4.4507</td>
<td>3.4155</td>
</tr>
</tbody>
</table>

**Figure 6.7:** Templates from all cameras compare to all regions from camera 4. We can notice that the object region is always the most similar region compared to the other random regions. Note, the object score column is equal to the fifth row of Figure 6.2.

**Figure 6.8:** Histogram of pairwise dissimilarity score between the *same* person within different viewpoints. Each sub figure represents a specific colormap. (a): RGB (b) HSV (c) LAB. The x axis represents the dissimilarity score, and the y axis represents number of comparison within that particular range.

Different cameras, different person within different cameras.

**Same Person Different Camera (SPDC)** Here, the appearance of the same person from different viewpoints has been compared in terms of region covariance dissimilarity score. The histogram of the dissimilarity scores for different color schemes has been shown in Figure 6.8.

**Different Person Different Camera (DPDC)** Here, the appearance of *different* persons across different viewpoints has been compared as discussed Section 6.3.2.2. The histogram of the dissimilarity scores for different color schemes is shown in Figure 6.8.
Comparing of SPDC, the dissimilarity score of DPDC is supposed to be larger than the score in SPDC. Since only the RGB-color space based features match our assumption, we have choose RGB as our final color space.

### 6.4 Conclusion and future work

In this chapter we tested the capability of region covariance feature for data association across non-overlapped cameras. A flip invariant modification for region covariance and a Bayesian probabilistic based framework to estimate the probability of spatial-temporal occupation has been proposed. Our experiments show that the modified region covariance matrix could decrease the dissimilarity score for the same object across viewpoints. The proposed modified features have been tested on real world dataset and according to our experiments the RGB is a better color map compared to HSV and LAB for cross camera data association. Due to the difficulties of this challenge itself, it would be hard to solve the whole problem at once, here we have only attempted to solve one portion of this challenge, and later, a more systematical way to estimate the probabilistic framework will be introduced by the other following researchers.
Chapter 7

Conclusion

In this thesis, we explored the computer vision problems using dynamical information. Our study mainly focuses on surveillance system-related challenges, such as activity recognition within a single view and multiple views, person detection and tag-and-track. All challenges addressed in this thesis employ underlying dynamical systems, which has rarely been addressed before. We have proposed some simple yet elegant techniques to utilize this information, such as subspace identification and massive dynamical system analysis.

In Chapter 3, a new dynamic invariance-based cell tracking and counting algorithm for very noisy biological microscopic fluorescently-labeled image sequences has been proposed. Our algorithm does not need to assume a certain distribution or dynamic model to accomplish tracking and counting. Instead, it exploits the dynamic information encoded in the initialized trajectory for each cell and predicts distribution based on the dynamic invariance information encoded using the initial trajectory. Our experiments show that the use of dynamic invariance information leads to an performance of up to 90% in terms of sensitivity.

In Chapter 4, we proposed a new time-series approach for human activity recognition that does not need to identify a model. Instead, it exploits the dynamic information encoded in the structure of Hankel matrices built from the data. We showed that trajectories corresponding to the same activity live in a subspace and that the DSA – i.e. the canonical correlations between the associated subspaces – can be used to classify the activities. Our experiments show that
using DCC to increase the separation between classes, and using Hankel matrices to capture
the temporal information both result in significant improvements of the overall classification
performance. The proposed approach was tested with the KTH database and the much more
challenging TV Interaction database, achieving an overall performance of 93.6% and 68%,
respectively, which are significantly higher than the highest reported performance using the
same experimental protocols.

In Chapter 5, activity recognition across multiple views was solved using a newly proposed
dynamics-based feature (Hankelet) for activity recognition and a simplified distance to com-
pare them. Hankelets are easily formed from very short tracklets which do not require persis-
tent tracking, and they 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
classical same-view activity recognition. 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 improved perfor-
mane by over 400% compared to using cuboids on the IXMAS database. Finally, compared
to other cross-view approaches that specifically address viewpoint changes, the use of a sub-
set of Hankelets (i.e. bi-lingual Hankelets) to compensate for self-occlusions resulted in an
average accuracy of 90.57% – a 20% increase in performance on the IXMAS dataset over the
previous leading algorithms. Due to Hankelet’s excellent performance on cross view activ-
ity recognition problem, we would propose to generalize Hankelets in other topics, such as
re-identification challenges.

In Chapter 6, a texture-based feature, region covariance, and a general Bayesian framework
have been proposed for the tag-and-track challenge. Our experiments show that region co-
variance could capture appearance across different views and the general Bayesian framework
provides acceptable performance in a challenging real-life airport scenario.
Bibliography


[35] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, “Person re-
identification by symmetry-driven accumulation of local features,” in CVPR, pp. 2360–
2367, 2010.


tracking-by-detection using a detector confidence particle filter,” in ICCV, pp. 1515–
1522, 2009.

[38] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Context tracker: Exploring support-

detection,” in IEEE Frame Rate Workshop, pp. 1–8, 1999.

[40] M. Isard and A. Blake, “Condensationconditional density propagation for visual track-


[42] B. North, A. Blake, M. Isard, and J. Rittscher, “Learning and classification of complex

[43] F. Xiong, O. I. Camps, and M. Sznaier, “Dynamic context for tracking behind occlu-


respondence between multiple cameras for people tracking,” PAMI, vol. 28, no. 4,


