


2012

## Study Of Human Activity In Video Data With An Emphasis On View-invariance

Nazim Ashraf  
*University of Central Florida*

 Part of the [Computer Sciences Commons](#), and the [Engineering Commons](#)  
Find similar works at: <https://stars.library.ucf.edu/etd>  
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact [STARS@ucf.edu](mailto:STARS@ucf.edu).

---

### STARS Citation

Ashraf, Nazim, "Study Of Human Activity In Video Data With An Emphasis On View-invariance" (2012).  
*Electronic Theses and Dissertations, 2004-2019*. 2180.  
<https://stars.library.ucf.edu/etd/2180>

# STUDY OF HUMAN ACTIVITY IN VIDEO DATA WITH AN EMPHASIS ON VIEW-INVARIANCE

by

NAZIM ASHRAF

B.Sc. Lahore University of Management Sciences, 2005

M.S. University of Central Florida, 2007

A dissertation submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy  
in the Department of Electrical Engineering and Computer Science  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Summer Term  
2012

Major Professor: Hassan Foroosh

© 2012 NAZIM ASHRAF

## ABSTRACT

The perception and understanding of human motion and action is an important area of research in computer vision that plays a crucial role in various applications such as surveillance, HCI, ergonomics, etc. In this thesis, we focus on the recognition of actions in the case of varying viewpoints and different and unknown camera intrinsic parameters. The challenges to be addressed include perspective distortions, differences in viewpoints, anthropometric variations, and the large degrees of freedom of articulated bodies. In addition, we are interested in methods that require little or no training. The current solutions to action recognition usually assume that there is a huge dataset of actions available so that a classifier can be trained. However, this means that in order to define a new action, the user has to record a number of videos from different viewpoints with varying camera intrinsic parameters and then retrain the classifier, which is not very practical from a development point of view. We propose algorithms that overcome these challenges and require just a few instances of the action from any viewpoint with any intrinsic camera parameters. Our first algorithm is based on the rank constraint on the family of planar homographies associated with triplets of body points. We represent action as a sequence of poses, and decompose the pose into triplets. Therefore, the pose transition is broken down into a set of movement of body point planes. In this way, we transform the non-rigid motion of the body points into a rigid motion of body point

planes. We use the fact that the family of homographies associated with two identical poses would have rank 4 to gauge similarity of the pose between two subjects, observed by different perspective cameras and from different viewpoints. This method requires only one instance of the action. We then show that it is possible to extend the concept of triplets to line segments. In particular, we establish that if we look at the movement of line segments instead of triplets, we have more redundancy in data thus leading to better results. We demonstrate this concept on “fundamental ratios.” We decompose a human body pose into line segments instead of triplets and look at set of movement of line segments. This method needs only three instances of the action. If a larger dataset is available, we can also apply weighting on line segments for better accuracy. The last method is based on the concept of “Projective Depth”. Given a plane, we can find the relative depth of a point relative to the given plane. We propose three different ways of using “projective depth:” (i) Triplets - the three points of a triplet along with the epipole defines the plane and the movement of points relative to these body planes can be used to recognize actions; (ii) Ground plane - if we are able to extract the ground plane, we can find the “projective depth” of the body points with respect to it. Therefore, the problem of action recognition would translate to curve matching; and (iii) Mirror person - We can use the mirror view of the person to extract mirror symmetric planes. This method also needs only one instance of the action. Extensive experiments are reported on testing view invariance, robustness to noisy localization and occlusions of body points, and action recognition. The experimental results are very promising and demonstrate the efficiency of our proposed invariants.

*To my parents for their love and encouragement.*

~

*To Arman for always being there.*

## ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Hassan Foroosh for his guidance and encouragement.

I thankfully acknowledge Yuping Shen, Adeel Bhutta, and Chuan Sun for their ideas and advice during our discussions. I am also grateful to Dr. Marshall Tappen, Dr. Charles E. Hughes, and Dr. J Michael Moshell for serving on my committee.

I am thankful to my family for their continued support and encouragement, without which this would not have been possible. I had a very memorable time in Orlando thanks to all my friends.

# TABLE OF CONTENTS

|  |    |
|--|----|
| LIST OF FIGURES . . . . .                                      | xi |
| LIST OF TABLES . . . . .                                       | xv |
| CHAPTER 1: INTRODUCTION . . . . .                              | 1  |
| 1.1 Background . . . . .                                       | 2  |
| 1.2 Projective Invariants . . . . .                            | 6  |
| 1.3 Organization of the Dissertation . . . . .                 | 7  |
| CHAPTER 2: ACTION RECOGNITION USING RANK CONSTRAINT . . . . .  | 8  |
| 2.1 Representation of Human Action . . . . .                   | 8  |
| 2.1.1 Matching Poses . . . . .                                 | 8  |
| 2.1.1.1 Homographies Induced by Body-Point Triplets. . . . .   | 9  |
| 2.1.2 Action Recognition . . . . .                             | 12 |
| 2.2 Experimental Results . . . . .                             | 12 |
| 2.2.1 Results on MoCap Data . . . . .                          | 12 |
| 2.2.1.1 Testing View-invariance and Noise Resilience . . . . . | 13 |
| 2.2.1.2 Testing Action Recognition . . . . .                   | 14 |



|  |  |    |
|--|--|----|
| 2.2.2  | Results on Real Data . . . . .                                   | 16 |
| CHAPTER 3: IMPROVING ACTION RECOGNITION USING MOTION OF LINE |  |    |
|  | SEGMENTS AND WEIGHTING . . . . .                                 | 17 |
| 3.1  | Action Recognition Using Fundamental Ratios . . . . .            | 20 |
| 3.1.1  | Representation of Pose . . . . .                                 | 20 |
| 3.1.2  | Pose Transitions . . . . .                                       | 21 |
| 3.1.3  | Matching Pose Transition . . . . .                               | 21 |
| 3.1.3.1  | Fundamental matrix induced by a moving line segment . . . . .    | 22 |
| 3.1.3.2  | Algorithm for Matching Pose Transitions . . . . .                | 26 |
| 3.1.4  | Sequence Alignment . . . . .                                     | 27 |
| 3.1.5  | Action Recognition . . . . .                                     | 28 |
| 3.2  | Weighting-based Human Action Recognition . . . . .               | 29 |
| 3.2.1  | Weights on line segments versus Weights on Body Points . . . . . | 34 |
| 3.2.2  | Automatic Adjustment of Weights . . . . .                        | 35 |
| 3.3  | Experimental Results and Discussion . . . . .                    | 37 |
| 3.3.1  | Analysis based on motion capture data . . . . .                  | 37 |
| 3.3.1.1  | Testing View Invariance . . . . .                                | 38 |
| 3.3.1.2  | Testing Robustness to Noise . . . . .                            | 40 |
| 3.3.1.3  | Performance in Action Recognition . . . . .                      | 40 |

|  |  |    |
|--|--|----|
| 3.3.2  | Results on real data . . . . .                         | 43 |
| 3.3.2.1  | UCF-CIL Dataset . . . . .                              | 43 |
| 3.3.2.2  | IXMAS data set . . . . .                               | 45 |
| 3.3.2.3  | Testing Occlusion . . . . .                            | 45 |
| 3.3.3  | How soon can we recognize the action? . . . . .        | 49 |
| CHAPTER 4: ACTION RECOGNITION USING PROJECTIVE DEPTH . . . . |  | 51 |
| 4.1  | Projective Depth . . . . .                             | 51 |
| 4.2  | Using Projective Depth . . . . .                       | 54 |
| 4.2.1  | Using Triplets . . . . .                               | 54 |
| 4.2.2  | Ground Plane . . . . .                                 | 56 |
| 4.2.2.1  | Estimating ground plane homography . . . . .           | 56 |
| 4.2.2.2  | Action Alignment . . . . .                             | 57 |
| 4.2.2.3  | Degeneracy . . . . .                                   | 58 |
| 4.2.3  | Planes in time . . . . .                               | 58 |
| 4.2.4  | Using Mirror Symmetry . . . . .                        | 59 |
| 4.2.4.1  | Using Mirror-view symmetry in Pose Recognition . . . . | 61 |
| 4.3  | Action Recognition Using Projective Depth . . . . .    | 63 |
| 4.3.1  | Experimental Results and Discussion . . . . .          | 64 |
| 4.3.1.1  | Results on MoCap Data . . . . .                        | 66 |

|   |   |    |
|---|---|----|
| 4.3.1.2   | Results on Real Data . . . . .          | 66 |
| CHAPTER 5: CONCLUSION AND FUTURE WORK . . . . . |   | 69 |
| 5.1   | Computational Complexity . . . . .      | 72 |
| 5.1.1   | Rank-4 constraint . . . . .             | 72 |
| 5.1.2   | Fundamental Ratios constraint . . . . . | 73 |
| 5.1.3   | Projective Depth Invariant . . . . .    | 73 |
| 5.1.3.1   | Using Ground Plane . . . . .            | 73 |
| 5.1.3.2   | Using Triplets . . . . .                | 73 |
| 5.1.3.3   | Using Mirror Person . . . . .           | 73 |
| 5.2   | Significance of this work . . . . .     | 74 |
| 5.3   | Future Work . . . . .                   | 76 |
| LIST OF REFERENCES . . . . .                    |   | 78 |

## LIST OF FIGURES

|            |  |    |
|------------|--|----|
| Figure 2.1 | Error surfaces for each noise levels for same and different noise levels.<br><br>The corresponding grid plots show the confusion between the same and different pose transitions. We see that there was no confusion for $\sigma = 0$ , and some confusion for $\sigma = 2$ , and $\sigma = 4$ . . . . . | 14 |
| Figure 2.2 | Comparing Performance: (a) and (b) Plots of matching scores of same and different pose transitions with increasing Gaussian noise for our likelihood function and the Sampson error, respectively. Plot (c) Confusion margin in (a) and (b). . . . .   | 14 |
| Figure 2.3 | (a) Distribution of two cameras: camera 1 is fixed (red point); camera 2 is distributed on a sphere around the subject. (b) Distribution of cameras used to evaluate view-invariance and camera parameter changes. . . . .   | 15 |

Figure 3.1 Roles of line segments in action recognition: (a) - (f) are the plots of dissimilarity scores of some line segments across frames in the walk-walk and walk-run alignments. As can be observed, line segments 1, 21 and 90 have similar error scores in both cases, which essentially means the motion of these line segments is similar in walking and running. But line segments 55, 94 and 116 have high error scores in  $\mathbf{M}'_e$  and low error scores in  $\mathbf{M}_e$ , which means that the motion of these line segments in a running sequence is different from their motion in a walking sequence. Therefore, these line segments reflect the variation in actions of walking and running and are much more useful for distinguishing between walking and running actions. . . . 31

Figure 3.2 Roles of different line segments in action recognition. We selected four sequences  $G0$ ,  $G1$ ,  $G2$ , and  $G3$  of golf-swing action, and align  $G1$ ,  $G2$ , and  $G3$  to  $G0$  using the alignment method described in Section 2, and then build error score matrix  $\mathbf{M}_e^1$ ,  $\mathbf{M}_e^2$ ,  $\mathbf{M}_e^3$  correspondingly as in above experiments. As can be observed, the dissimilarity scores of some line segments, such as line segments 53 is very consistent across individuals. Some other line segments such as line segments 6 and 50 have various error score patterns across individuals, that is, these line segments represent the variations of individuals performing the same action. . . . . 32

|            |   |    |
|------------|---|----|
| Figure 3.3 | Left: Our body model. Right: Experiment on view-invariance. Two different pose transitions $P_1 \rightarrow P_2$ and $P_3 \rightarrow P_4$ from a golf swing action are used. . . . .   | 38 |
| Figure 3.4 | Analysis of view invariance: (a) Camera 1 is marked in red, and all positions of camera 2 are marked in blue and green. (b) Errors for same and different pose transitions when camera 2 is located at viewpoints colored as green in (a). (c) Errors of same and different pose transitions when camera 2 is located at viewpoints colored as blue in (a). (d) General camera motion: Camera 1 is marked as red, and camera 2 is distributed on a sphere. (e) Error surface of same pose transitions for all distributions of camera 2 in (d). (f) Error surface of different pose transitions for all distribution of camera 2 in (d). (g) The regions of confusion for (d) marked in black (see text). | 38 |
| Figure 3.5 | Robustness to noise: $I_1$ and $I_2$ are the images in camera 1, and $I_3, I_4, I_5$ and $I_6$ are the images in camera 2. Same and different actions are distinguished unambiguously for $\sigma < 4$ . . . . .  | 39 |
| Figure 3.6 | A set of 56 sequences in 8 categories (actions) used to test the proposed method. Ballet fouettes: (1)-(4); ballet spin: (5)-(16); push-up: (17)-(22); golf swing: (23)-(30); one-handed tennis backhand stroke: (31)-(34); two-handed tennis backhand stroke: (35)-(42); tennis forehand stroke: (43)-(46); tennis serve: (47)-(56). . . . .   | 50 |

|            |   |    |
|------------|---|----|
| Figure 4.1 | A point $\mathbf{x}$ in one image is transferred via the plane $\pi$ to a matching point $\mathbf{x}'$ in the second image. . . . .   | 52 |
| Figure 4.2 | These figures explain the significance of the characteristic vector. As soon as the person moves one of his arms, there is notable change in the characteristic vector for the points that moved. . . . . | 52 |
| Figure 4.3 | The depth of left shoulder and its mirror view would be equidistant from the plane consisting of left hand, and right hand, and their mirror views. . . . .   | 63 |

## LIST OF TABLES

|           |   |    |
|-----------|---|----|
| Table 2.1 | Our method: Overall accuracy about 87%. . . . .   | 15 |
| Table 2.2 | Recognition rate for IXMAS data. . . . .  | 16 |
| Table 3.1 | Confusion matrix before applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 85.60%. . . . .   | 42 |
| Table 3.2 | Confusion matrix after applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 92.40%, which is an improvement of 6.8% compared to the non-weighted case. . . . .   | 42 |
| Table 3.3 | Confusion matrix before applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 95.83%. The actions are denoted by numbers: 1 - ballet fouette, 2 - ballet spin, 3 - pushup, 4 - golf swing, 5 - one handed tennis backhand, 6 - two handed tennis backhand, 7 - tennis forehand, 8 - tennis serve. . . . . | 44 |



|            |   |    |
|------------|---|----|
| Table 3.4  | Confusion matrix after applying weighting: The overall recognition rate is 100%, which is an improvement of 4.17% compared to the nonweighted case. The actions are the same as in Table 3.3. . . .   | 44 |
| Table 3.5  | Recognition rates in % on IXMAS dataset . . . . .   | 46 |
| Table 3.6  | Confusion matrix for IXMAS dataset before applying weighting. The actions are denoted by numbers: 1 = Check Watch, 2 = Cross Arms, 3 = Scratch Head, 4 = Sit Down, 5 = Get up, 8 = Wave, 9 = Punch, 10 = Kick, 11 = Point, and 12 = Pick Up . . . . . | 47 |
| Table 3.7  | Confusion matrix for IXMAS dataset after applying weighting: The overall recognition rate is 92.1%, which is an improvement of 4.8% compared to the nonweighted case. The actions are the same as in Table 3.6. . . . .                               | 47 |
| Table 3.8  | Confusion matrix when head and two shoulder points are occluded. The actions are the same as in Table 3.6. . . . .  | 48 |
| Table 3.9  | Confusion matrix when the right side of the body is occluded including the right shoulder, arm, hand, and knee point. . . . .   | 48 |
| Table 3.10 | Confusion matrix when the left side of the body is occluded including the left shoulder, arm, hand, and knee point. . . . .   | 48 |
| Table 3.11 | Confusion matrix when the lower body is occluded including the two knee and feet points. . . . .  | 48 |

|            |  |    |
|------------|--|----|
| Table 3.12 | This table shows how soon we can recognize an action for IXMAS dataset. . . . .                  | 49 |
| Table 4.1  | Using ground plane: Overall accuracy about 95%. . . . .  | 65 |
| Table 4.2  | Using triplets: Overall accuracy about 90% . . . . .   | 65 |
| Table 4.3  | Using mirror symmetric planes: Overall accuracy about 96% . . .                                  | 65 |
| Table 4.4  | Recognition rate for IXMAS data using ground plane. Overall accuracy: 81.4% . . . . .            | 67 |
| Table 4.5  | Recognition rate for IXMAS data using triplets. Overall accuracy: 87.3% . . . . .                | 67 |
| Table 4.6  | Recognition rate for IXMAS data using mirror symmetric planes. Overall accuracy: 90.5% . . . . . | 67 |

## CHAPTER 1: INTRODUCTION

The perception and understanding of human motion and action is an important area of research in computer vision that plays a crucial role in various applications such as surveillance, human computer interaction, ergonomics, kinesiology, video communication, animation etc. All these applications have deep impact on a number of aspects in our daily lives. For instance, surveillance systems have become a necessity for public safety in high risk areas such as airports, train stations, banks, etc. In human computer interaction, the basic idea is that the machine be able to recognize the gestures made by the human user and respond appropriately. In recent years, we have seen a boom in gaming industries in coming up with new camera equipped gaming consoles such as Microsoft Kinect. These have become immensely popular owing to more realistic interactive effects and users having to use their whole body. In the case of kinesiology, human joints are tracked for use in medical diagnostics and analysing performance. With regards to multimedia retrieval and animation, large motion capture datasets have become commonplace owing to their importance in realistic animation of human motion and it has become increasingly important to develop methods for an animator to search for similar motions from a given dataset.

Analysing human action can be divided into a set of problems including human detection, tracking of body parts / joints, and finally action recognition. In this thesis, we focus mainly on action recognition. Since the image sequence is acquired from a

camera, we lose the depth information and it is projectively distorted. Therefore, the same object can appear very different from another view-point. This is the focus of this thesis: the recognition of actions in the case of varying viewpoints and different and unknown camera intrinsic parameters.

## 1.1 Background

The problem has been the subject of extensive studies in the past, summarized in excellent surveys such as [18, 36, 37, 60, 46]. Action can be regarded as a collection of 4D space-time data observed by a perspective video camera. Due to image projection, the 3D Euclidean information is lost and projectively distorted, which makes action recognition rather challenging, especially for varying viewpoints and different camera parameters. Another source of challenge is the irregularities of human actions due to a variety of factors such as age, gender, circumstances, etc. The timeline of action is another important issue in action recognition. The execution rates of the same action in different videos may vary for different actors or due to different camera frame rates. Therefore, the mapping between same actions in different videos is usually highly non-linear.

To tackle these issues, often simplifying assumptions are made by researchers on one or more of the following aspects: (1) camera model, such as scaled orthographic [51] or calibrated perspective camera [65]; (2) camera pose, i.e. little or no viewpoint variations; (3) anatomy, such as isometry [39], coplanarity of a subset of body points [39], etc. However, in practical applications such as surveillance, actions may be viewed from different

angles by different perspective cameras. Therefore, a reliable action recognition system has to be invariant to the camera parameters or viewpoint changes. View-invariance is, thus, of great importance in action recognition, and has received relatively more attention in recent literature.

One approach to tackle view-invariant action recognition has been based on using multiple cameras: Campbell et al. [10] use stereo images to recover a 3D Euclidean model of the human subject, and extract view invariance for 3D gesture recognition; Weinland et al. [65] use multiple calibrated and background-subtracted cameras, and they obtain a visual hull for each pose from multi-view silhouettes, and stack them as a motion history volume, based on which Fourier descriptors are computed to represent actions. Ahmad et al. [2] build HMMs on optical flow and human body shape features from multiple views, and feed a test video sequence to all learned HMMs. These methods require the setup of multiple cameras, which is quite expensive and restricted in many situations such as online video broadcast or monocular surveillance.

A second line of research is based on a single camera and is motivated by the idea of exploiting the invariants associated with a given camera model, e.g. affine, or projective. For instance, Rao et al. [42] assume an affine camera model, and use dynamic instant, i.e. the maxima in the space-time curvature of the hand trajectory, to characterize hand actions. The limit with this representation is that dynamic instants may not always exist or may not be always preserved from 3D to 2D due to perspective effects. Moreover the affine camera model is restrictive in most practical scenarios. A more recent work reported by Parameswaran et al. [39] relaxes the restrictions on the camera model.

They propose a quasi-view-invariant 2D approach for human action representation and recognition, which relies on the number of invariants in a given configuration of body points. Thus a set of projective invariants are extracted from the frames and used as action representation. However, in order to make the problem tractable under variable dynamics of actions they introduced heuristics, and make simplifying assumptions such as isometry about human body parts. Moreover, they require that at least five body points form a 3D plane or the limbs trace planar area during the course of an action. Ali et al. [4] introduced chaotic invariants and analyze nonlinear dynamics of human actions. Trajectories of reference joints are used as the representation of the non-linear dynamical system that is generating the action. Lv et al. [33] search for the appropriate input sequence for a given sequence.

Another promising approach is based on exploiting the multi-view geometry. Two subjects in the same exact body posture viewed by two different cameras at different viewing angles can be regarded as related by the epipolar geometry. Therefore, corresponding poses in two videos of actions are constrained by the associated fundamental matrices, providing thus a way to match poses and actions in different views. The use of fundamental matrix in view invariant action recognition is first reported by Syeda-Mahmood et al. [56] and later by Yilmaz et al. [67, 68]. They stack silhouettes of input videos into space-time objects, and extract features in different ways, which are then used to compute a matching score based on the fundamental matrices. A similar work is also presented in [19], which is based on body points instead of silhouettes.

Space-time features are essentially the primitives that are used for recognizing actions, e.g. photometric features such as the optical flow [13, 71, 59] and the local space-time features [48, 27]. These photometric features can be affected by luminance variations due to, for instance, camera zoom or pose changes, and often work better when the motion is small or incremental. On the other hand, salient geometric features such as silhouettes [7, 61, 8, 62, 67] and point sets [39, 68] are less sensitive to photometric variations, but require reliable tracking. Silhouettes are usually stacked in time as 2D [8] or 3D object [7, 67], while point sets are tracked in time to form space-time curves. Ali and Shah [5] derive a number of features from the optical flow such as gradient tensor features, divergence, etc. and apply Principal Component Analysis (PCA) to determine the dominant kinematic modes. Fathi and Mori [15] introduced a method for human action recognition based on patterns of motion by constructing mid-level motion features which are built from low-level optical flow information.

Some existing approaches are also more holistic and rely on machine learning techniques, e.g. HMM [2, 66, 16, 41, 3], SVM [48, 29, 23], Boosting [15, 28, 38] etc. As in most exemplar-based methods, they rely on the completeness of the learning data, and to achieve view-invariance are usually expensive as it would be required to learn a model from a large dataset.

Recently there has been an interest in investigating how soon an action can be recognized given its applications in human computer interfaces. Schindler and van Gool [47] present a method that can recognize action from very short sequences. Similarly Masood et al. [34] investigate reducing latency in recognizing actions.

## 1.2 Projective Invariants

The literature on projective invariants is quite rich and its history dates back to well before the invention of computer vision. In the vision community projective invariants gained popularity for object recognition in the 1990's. For instance, [30] used perspective invariants to recognize polygonal planar objects. [6, 26, 35] used affine invariants to recognize planar objects in 3D space. This discussion is excellently summarized in [11]. Formally, geometric invariants refer to the study of the invariant properties under action of a group  $G$  on an algebraic variety  $V$ . In computer vision by the very nature of the problems, in the most general case, we deal with the general linear projective group  $GL(3)$ . Given a configuration of points or of other geometric primitives (e.g. lines or planes), the number of invariants is given by the dimension of the configuration minus the dimension of the transformation group that acts upon the configuration. For instance, for a set of primitives (e.g. points) in general positions in  $\mathbf{P}^2$ , the number of invariants would be the total degrees of freedom of the configuration minus the 8 degrees of freedom of a general homography in 2D.

Invariants are often expressed by linear combinations of products of the determinants of matrices, whose columns are the homogeneous coordinates of the points in a rigid structure. This is in fact the approach Parameswaran et al. [39] used for the formulation of their framework for a given human body pose. The most commonly studied projective invariant is of course the cross-ratio of a set four collinear points (also extendable to other geometric primitives, such as pencil of lines or planes). Many invariants used for action recognition are in fact derived directly from cross ratio [39].



In Computer Vision, many invariants are derived from epipolar geometry. Epipolar geometry relates image points across different camera views. A given 3D point's image in one camera view is related to the epipolar line in the other camera view. All the epipolar lines intersect at the epipole, which is the image of the other camera center. Epipolar geometry has been used in a variety of applications such as [12, 17, 43, 9, 67, 51] because it is independent of camera internal parameters and view-point. In this thesis, we also employ epipolar geometry to derive new invariants for action recognition.

### 1.3 Organization of the Dissertation

In Chapter 2, we discuss our first method for view-invariant action recognition which is based on the rank constraint on the family of planar homographies associated with triplets of body points. We represent action as a sequence of poses and we use the fact that the family of homographies associated with two identical poses would have rank 4 to gauge similarity of the pose between two subjects, observed by different perspective cameras and from different viewpoints. Chapter 3 extends the idea of looking at the motion of triplets to that of line segments. We demonstrate this concept on fundamental ratios and show that we get better results due to more redundancy in data. We also apply weighting on the line segments to improve our results. In Chapter 4, we propose to use “projective depth” for use in action recognition. There are several ways in which we can use projective depth for action recognition and we analyze each of these options. Finally, we conclude in chapter 5, we present the computational complexity of each of the methods, and discuss the significance of this work, and future work.

## CHAPTER 2: ACTION RECOGNITION USING RANK CONSTRAINT

### 2.1 Representation of Human Action

In this work, we use the same model as [69]. We represent a human body pose  $\mathcal{P}$  by  $M$  body points:  $\mathcal{P} = \{\mathbf{m}_{i=1\dots M}\}$ . These points can be obtained by using articulated object tracking techniques such as [45]. For our experiments, we used 11 body points as show in [Figure 3.3](#). Further discussions on articulated object tracking can be found in [37, 60]. We assume that tracking has already been performed on the data, and that we have the set of labeled points for each image.

An action sequence  $\mathbf{A}$  consists of  $T$  frames:  $\{\mathcal{P}_1^A, \dots, \mathcal{P}_T^A\}$ . With this representation, comparison of two action sequences reduces to examining the similarities of the poses, as described in the following sections.

#### 2.1.1 Matching Poses

Suppose we are given two poses  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . Using point representation, a pose is characterized by a set of body points. Each triplet of non-collinear points specifies a scene plane. Therefore, a non-rigid pose can be decomposed into scene planes determined by all non-collinear triplets.

Now assume the case that  $\mathcal{P}_1$  corresponds to  $\mathcal{P}_2$ .  $\mathcal{P}_1$  and  $\mathcal{P}_2$  can then be regarded as the images of same subject viewed by two different cameras. Suppose that  $\mathcal{P}_1$  are observed by camera  $\mathbf{P}_1$  and  $\mathcal{P}_2$  by camera  $\mathbf{P}_2$ .  $\mathbf{P}_1$  and  $\mathbf{P}_2$  may have different intrinsic and extrinsic parameters. These point correspondences induce an epipolar geometry via the fundamental matrix  $\mathbf{F}$ [22]. The computation of  $\mathbf{F}$  has been well studied in the community, e.g, [32]. Note that  $\mathbf{F}$  does not correlate the entire scene, but only the body points of the subjects.

#### 2.1.1.1 Homographies Induced by Body-Point Triplets.

Let us now consider an arbitrary triplet of 3D body points,  $\Delta = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$ , which corresponds to  $\Delta_1 = \langle \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \rangle$  in  $\mathcal{P}_1$  and  $\Delta_2 = \langle \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \rangle$  in  $\mathcal{P}_2$ .  $\Delta$  determines a scene plane  $\pi_1$  in the 3D space, which induces a homography  $\mathbf{H}_1$  between  $\mathbf{P}_1$  and  $\mathbf{P}_2$ . These plane-induced homographies can be computed given four point correspondences, i.e. the image point correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{y}_i$  and the epipoles  $\mathbf{e}_1 \leftrightarrow \mathbf{e}_2$ .

A degenerate case occurs when three of the four points are collinear but we can simply discard these degenerate cases. The number of non-degenerate triplets exceeds by far the degenerate triplets, since the total number of available triplets is  $\binom{n}{3}$  for  $n$  body points.

A special case is when the epipole is at or close to infinity, all triplets then may be regarded as degenerate since the distance between three image points is negligible compared with their distances to the epipole. We solve this problem by transforming the image points in projective space, which is similar to [70]. The idea is to find the

projective transformation  $\mathbf{P}$  and  $\mathbf{P}'$  for each image, such that after transformation the epipoles and image points are finite.

As described above, each triplet in a pose induces a homography. If we have some constraint on the family of homographies induced by all the triplets in a given pose, we can exploit it for recognizing the pose and ultimately for action recognition. One such constraint can be imposed by using the following result [50]:

**Theorem 1** (Rank Constraint)

*The space of all homography matrices between two views is spanned by a 4 dimensional linear subspace of  $\mathcal{P}^8$*

The proof follows from the fact that given two views  $\mathbf{I}$  and  $\mathbf{J}$ , each plane induces a homography; and that given a homography matrix  $\mathbf{H}$  of *some plane*, defined by  $\pi^T \mathbf{X} = 0$  with  $\pi = (\mathbf{n}^T, 1)^T$ , all other homographies can be described by:

$$\lambda \mathbf{H} + \mathbf{e}' \mathbf{n}^T \tag{2.1}$$

where  $\mathbf{e}'$  is the epipole (the projection of the first camera center onto view  $\mathbf{J}$ ).

Consider the homography matrices  $\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_k$  each as a column vector stacked in a  $9 \times k$  matrix. Let  $\mathbf{H}_i = \lambda \mathbf{H} + \mathbf{v}' \mathbf{n}^T$ . The following can be easily ascertained:

$$\begin{aligned}
& \begin{bmatrix} \phantom{0} \\ \phantom{0} \\ \phantom{0} \end{bmatrix}_{9 \times k} = \begin{bmatrix} \lambda_1 \mathbf{H} \dots \lambda_k \mathbf{H} \end{bmatrix}_{9 \times k} \\
& + \begin{bmatrix} \mathbf{e}' & 0 & 0 \\ 0 & \mathbf{e}' & 0 \\ 0 & 0 & \mathbf{e}' \end{bmatrix}_{9 \times 3} \begin{bmatrix} \phantom{0} \\ \mathbf{n}_1 \dots \mathbf{n}_k \\ \phantom{0} \end{bmatrix}_{3 \times k} \\
& = \begin{bmatrix} \phantom{0} & \mathbf{e}' & 0 & 0 \\ \mathbf{H} & 0 & \mathbf{e}' & 0 \\ \phantom{0} & 0 & 0 & \mathbf{e}' \end{bmatrix}_{9 \times 4} \begin{bmatrix} \phantom{0} \\ \lambda_1 \dots \lambda_k \\ \mathbf{n}_1 \dots \mathbf{n}_k \\ \phantom{0} \end{bmatrix}_{4 \times k}
\end{aligned}$$

Therefore a 4 dimensional linear subspace of  $\mathbf{P}^8$  can be used to express any homography, thus proving theorem 1.

In practice, all the homographies obtained by all the triplets can be stacked in a  $n \times 9$  matrix,  $\mathbf{Q}$ . From the above result, if two poses are identical, then the homographies associated with all body point triplets will span a rank 4 subspace of  $\mathbf{P}^8$ . Essentially, what this implies is that from this matrix, we can obtain the  $9 \times 9$  matrix,  $\mathbf{P} = \mathbf{Q}^T \mathbf{Q}$ . We can then perform singular value decomposition on  $\mathbf{P}$ , to obtain the eigenvectors and eigenvalues of  $\mathbf{P}$ . We thus define our similarity measure as:

$$S(\mathbf{P}) = 1 - \sum_{i=5, \dots, 9} \bar{a}_i \quad (2.2)$$

where  $\bar{a}_i = a_i / \sum_{i=1, \dots, 5} a_i$ , where  $a_i$  for  $i = 1, \dots, 9$  represent the eigenvalues of  $\mathbf{P}$  in descending order.  $S(\mathbf{P})$  is maximal for similar pose transitions, and is invariant to camera calibration matrix and viewpoint variations.

### 2.1.2 Action Recognition

Previously, we discussed how we can measure similarity between two poses. For action recognition, we want to match two sequences  $A = \{I_{1\dots n}\}$  and  $B = \{J_{1\dots m}\}$ ; in other words, we need the optimal mapping  $\psi : A \rightarrow B$  such that the cumulative similarity score  $\sum_{i=1}^n S(i, \psi(i))$  is maximized, where  $S(\cdot)$  is the similarity of two poses as defined above. This can be solved by dynamic programming, which has proved effective in sequence alignment (its application in action recognition can also be found in [40, 69]). In our formulation matching score of  $A$  and  $B$  can be defined by  $\mathcal{S}(A, B) = \max_{\psi} \sum_{i=1}^n S(i, \psi(i))$ . In practice, we need a reference sequence for each known action; we maintain an action database of  $K$  actions,  $DB = \{J_t^1\}, \{J_t^2\}, \dots, \{J_t^K\}$ . Given a test sequence  $\{I_t\}$ , we match  $\{I_t\}$  against each reference sequence in  $DB$ , and classify  $\{I_t\}$  as the action of best-match, say  $\{J_t^k\}$ , if  $\mathcal{S}(\{I_t\}, \{J_t^k\})$  is above a threshold  $T$ . Our solution is invariant to camera intrinsic parameters and viewpoint because we use the view-invariant distance in equation 2.2.

## 2.2 Experimental Results

In this section we present results on both semi-synthetic data and real data.

### 2.2.1 Results on MoCap Data

To test our approach on semi-synthetic data, we used the CMU Motion Capture database (MoCap - <http://mocap.cs.cmu.edu/>), which contains sequences of various real

human actions in 3D. We used synthetic cameras to generate the images of the 3D body points.

#### 2.2.1.1 Testing View-invariance and Noise Resilience

We selected two poses  $P_{1,2}$  from KICK-BALL sequence and a pose  $Q_1$  from the GOLF-SWING sequence. Two synthesized cameras were used to observe the 3D poses; the first camera has focal length  $f_1 = 1000$  and looks at the origin of the world coordinate from a fixed location (marked by red color in Figure 2.3 (a)); camera 2 is obtained by rotating camera 1 around  $x$  and  $y$  axes of the world coordinates in increments of  $10^\circ$ , and changing the focal length randomly in the range of  $1000 \pm 300$ . Figure 2.3 (a) shows all locations of camera 2 as blue points. Camera 1 observes  $P_{1,2}$  as  $I_{1,2}$  and camera 2 observes  $P_{1,2}$  and  $Q_1$  as  $J_{1,2}^k$ ,  $k = 1, 2$ . We then added Gaussian noise to the image points, with  $\sigma$  increasing in steps of 0.25 from 0 to 7. Two score functions  $S(k)$ ,  $k = 1, 2$  were computed. 100 independent trials were repeated for each noise level and the mean and the standard deviation of both error functions were calculated. The error surfaces and confusion areas with  $\sigma = 0, 2, 4$  are shown in Figure 2.1 (a)-(c). We observe that same and different pose transitions can be identified up until  $\sigma = 5.5$ , which amounts up to possibly 16.5 pixel errors.

We compared our results with the baseline method [22, 67]. These plots are shown in Figure 2.2. To compare the results in Figure 2.2 (a) and (b), we computed *confusion margin* for each method [69]. The curves for both methods are plotted in Figure 2.2 (c).

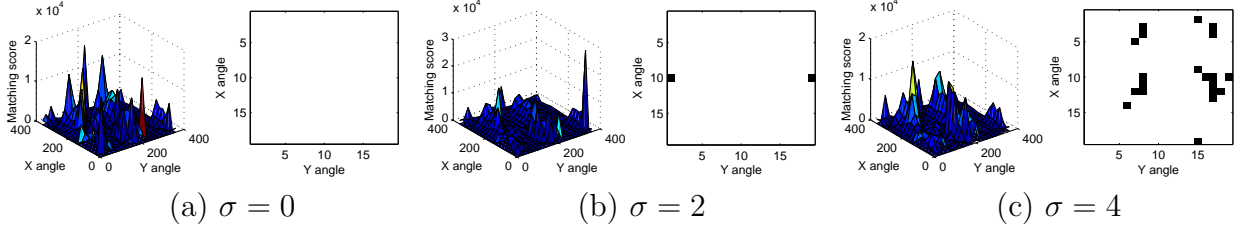


Figure 2.1: Error surfaces for each noise levels for same and different noise levels. The corresponding grid plots show the confusion between the same and different pose transitions. We see that there was no confusion for  $\sigma = 0$ , and some confusion for  $\sigma = 2$ , and  $\sigma = 4$ .

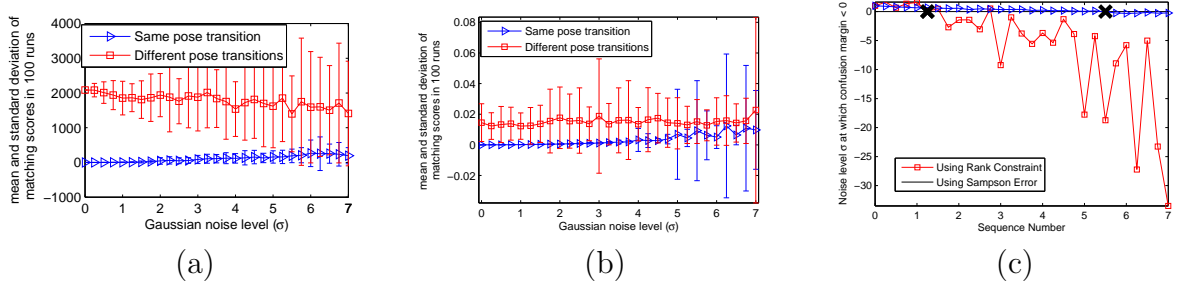


Figure 2.2: Comparing Performance: (a) and (b) Plots of matching scores of same and different pose transitions with increasing Gaussian noise for our likelihood function and the Sampson error, respectively. Plot (c) Confusion margin in (a) and (b).

### 2.2.1.2 Testing Action Recognition

We selected 4 actions from CMU’s MoCap data set consisting of “jump,” “golf-swing,” “run,” and “climb.” Each action is performed by 3 actors, and each instance of 3D action is observed by 17 cameras. The distribution of the cameras is shown in [Figure 2.3 \(b\)](#). As shown, the first camera was placed on  $(x_0, 0, 0)$ , looking at the origin of the world coordinate system, while the remaining 16 cameras were generated by rotating around the  $y$ -axis by  $\beta$  and around the  $x$ -axis by  $\alpha$ , where  $\beta = i\frac{\pi}{4}, i = 0, \dots, 7$  and  $\alpha = j\frac{\pi}{4}, j = 0, 1, 2$ . The focal lengths were also changed randomly in the range  $1000 \pm 300$ . We then added Gaussian noise with  $\sigma = 3$  to the image points.



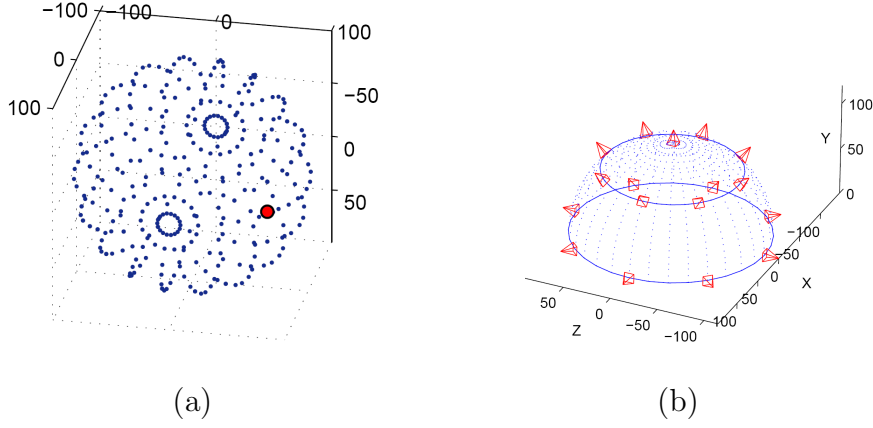


Figure 2.3: (a) Distribution of two cameras: camera 1 is fixed (red point); camera 2 is distributed on a sphere around the subject. (b) Distribution of cameras used to evaluate view-invariance and camera parameter changes.

Table 2.1: Our method: Overall accuracy about 87%.

| Ground-truth | Recognized as |     |       |            |
|--------------|---------------|-----|-------|------------|
|              | Jump          | Run | Climb | Golf Swing |
| Jump         | 46            |     | 3     | 2          |
| Run          | 2             | 45  | 2     | 2          |
| Climb        | 2             | 2   | 46    | 1          |
| Golf Swing   |               | 5   | 6     | 40         |

Our dataset contains 204 video sequences, 4 of which are taken out to act as the reference dataset from viewpoint 1. Each sequence was matched against all actions in the database and classified as the one with highest score. For each sequence matching, 10 random initialization are tested. The classification results are shown in [Table 2.1](#). The overall classification accuracy for our method is 86.7%.

Table 2.2: Recognition rate for IXMAS data.

| Action     | Check Watch | Scratch Head | Cross Arms | Sit down | Stand up |
|------------|-------------|--------------|------------|----------|----------|
| Accuracy % | 94          | 90           | 92         | 88       | 93       |

### 2.2.2 Results on Real Data

We evaluated our method on IXMAS data set [65]. This data set contains a number of actions performed by 11 actors. Each actor performs the action 3 times and 5 camera views of each action have been provided. We tested our method on 5 actions consisting of “watch time,” “cross arms,” “scratch head,” “sit down,” “stand up.” The classification results are shown in [Table 2.2](#). The average recognition rate is 91.4%, which is comparable to MHV [65] given that we do not use multiple images and rely only on one view.

## CHAPTER 3: IMPROVING ACTION RECOGNITION USING MOTION OF LINE SEGMENTS AND WEIGHTING

In this chapter, we propose that instead of looking at the motion of triplets, we can improve performance by looking at the motion of line segments. We demonstrate this by extending the concept of *fundamental ratios*, and explore the importance of different body parts in action recognition.

A moving plane observed by a fixed camera induces a fundamental matrix  $\mathbf{F}$  between two frames, where the ratios among the elements in the upper left  $2 \times 2$  submatrix are herein referred to as the *fundamental ratios*. We show that *fundamental ratios* are invariant to camera internal parameters and orientation, and hence can be used to identify similar motions of line segments from varying viewpoints. By representing the human body as a set of points, we decompose a body posture into a set of line segments. The similarity between two actions is therefore measured by the motion of line segments and hence by their associated *fundamental ratios*. We further investigate to what extent a body part plays a role in recognition of different actions and propose a generic method of assigning weights to different body points. Experiments are performed on three categories of data: the controlled CMU MoCap dataset, the partially controlled IXMAS data, and the more challenging uncontrolled UCF-CIL dataset collected on the internet. Extensive experiments are reported on testing (i) view-invariance, (ii) robustness to noisy localization of body points, (iii) effect of assigning different weights to different body points,

(iv) effect of partial occlusion on recognition accuracy, and (v) determining how soon our method recognizes an action correctly from the starting point of the query video. This work is an extension of [52], which introduced the concept of *fundamental ratios* that are invariant to rigid transformations of camera, and were applied to action recognition. We make the following main extensions: (i) Instead of looking at fundamental ratios induced by triplets of points, we look at fundamental ratios induced by line segments. (ii) It has been long argued in the applied perception community [49] that humans focus only on the most significant aspects of an event or action for recognition, and do not give equal importance to every observed data point. We propose a new generic method of learning how to assign different weights to different body points in order to improve the recognition accuracy by using a similar focusing strategy as humans; (iii) We study how this focusing strategy can be used in practice when there is partial but significant occlusion; (iv) We investigate how soon after the query video starts our method is capable of recognizing the action - an important issue never investigated by others in the literature; and (v) our experiments are more extensive than [52] and include larger set of data with various levels of difficulty.

**Proposition 1** *Given two cameras  $\mathbf{P}_i \sim \mathbf{K}_i[\mathbf{R}_i|\mathbf{t}_i]$ ,  $\mathbf{P}_j \sim \mathbf{K}_j[\mathbf{R}_j|\mathbf{t}_j]$  with zero skew and unit aspect ratio, denote the relative translation and rotation from  $\mathbf{P}_i$  to  $\mathbf{P}_j$  as  $\mathbf{t}$  and  $\mathbf{R}$  respectively, then the upper  $2 \times 2$  submatrix of the fundamental matrix between two views is of the form*

$$\mathbf{F}^{2 \times 2} \sim \begin{bmatrix} \epsilon_{1st} \mathbf{t}^s \mathbf{r}_1^t & \epsilon_{1st} \mathbf{t}^s \mathbf{r}_2^t \\ \epsilon_{2st} \mathbf{t}^s \mathbf{r}_1^t & \epsilon_{2st} \mathbf{t}^s \mathbf{r}_2^t \end{bmatrix}, \quad (3.1)$$

where  $\mathbf{r}_k$  is the  $k$ -th column of  $\mathbf{R}$ , the superscripts  $\mathbf{s}, \mathbf{t} = 1, \dots, 3$  indicate the element in the vector, and  $\epsilon_{rst}$ ,  $\mathbf{r} = 1, 2$  is a permutation tensor<sup>1</sup>.

**Remark 1** *The ratios among elements of  $\mathbf{F}^{2 \times 2}$  are invariant to camera calibration matrices  $\mathbf{K}_i$  and  $\mathbf{K}_j$ .*

The upper  $2 \times 2$  sub-matrices  $\mathbf{F}^{2 \times 2}$  for two moving cameras can be used to measure the similarity of camera motions. That is, if two cameras perform the same motion (same relative translation and rotation during the motion), and  $\mathbf{F}_1$  and  $\mathbf{F}_2$  are the fundamental matrices between any pair of corresponding frames, then  $\mathbf{F}_1^{2 \times 2} \sim \mathbf{F}_2^{2 \times 2}$ . This also holds for the dual problem when the two cameras are fixed, but the scene objects in both cameras perform the same motion. A special case of this problem is when the scene objects are planar surfaces, which is discussed below.

**Proposition 2** *Suppose two fixed cameras are looking at two moving planar surfaces, respectively. Let  $\mathbf{F}_1$  and  $\mathbf{F}_2$  be the two fundamental matrices induced by the two moving planar surfaces. If the motion of the two planar surfaces is similar (differ at most by a similarity transformation), then*

$$\mathbf{F}_1^{2 \times 2} \sim \mathbf{F}_2^{2 \times 2} \quad (3.2)$$

where the projective equality, denoted by  $\sim$ , is invariant to camera orientation.

Here similar motion implies that plane normals undergo same motion up to a similarity transformation. The projective nature of the view-invariant equation in (3.2)

---

<sup>1</sup>The use of tensor notation is explained in details in [22], p563.

implies that the elements in the sub-matrices on the both sides of (3.2) are equal up to an arbitrary non-zero scale factor, and hence only the ratios among them matter. We call these ratios the *fundamental ratios*, and as propositions 1 and 2 state, these *fundamental ratios* are invariant to camera intrinsic parameters and viewpoints. To eliminate the scale factor, we can normalize both sides using  $\hat{\mathbf{F}}_i = |\mathbf{F}_i^{2 \times 2}| / \|\mathbf{F}_i^{2 \times 2}\|_F, i = 1, 2$ , where  $|\cdot|$  refers to absolute value operator and  $\|\cdot\|_F$  stands for the Frobenius norm. We then have

$$\hat{\mathbf{F}}_1 = \hat{\mathbf{F}}_2 \quad (3.3)$$

In practice,  $\hat{\mathbf{F}}_1$  and  $\hat{\mathbf{F}}_2$  may not be exactly equal due to noise, computational errors or subjects' different ways of performing same actions. We, therefore, define the following function to measure the residual error:

$$\mathcal{E}(\hat{\mathbf{F}}_1, \hat{\mathbf{F}}_2) = \|\hat{\mathbf{F}}_1 - \hat{\mathbf{F}}_2\|_F. \quad (3.4)$$

### 3.1 Action Recognition Using Fundamental Ratios

#### 3.1.1 Representation of Pose

Using a set of body points for representing human pose has been used frequently in action recognition primarily because a human body can be modeled as an articulate object, and secondly, body points capture sufficient information to achieve the task of action recognition [19, 24, 39, 68]. Other representations of pose include subject silhouette [7, 8, 56], optical flow [13, 59, 71], and local space time features [27, 48].

### 3.1.2 Pose Transitions

We are given a video sequence  $\{I_t\}$  and a database of reference sequences corresponding to  $K$  different known actions,  $DB = \{J_t^1\}, \{J_t^2\}, \dots, \{J_t^K\}$ , where  $I_t$  and  $J_t^k$  are labeled body points in frame  $t$ . Our goal is to identify the sequence  $\{J_t^k\}$  from  $DB$  such that the subject in  $\{I_t\}$  performs the closest action to that observed in  $\{J_t^k\}$ .

Existing methods for action recognition such as [8, 67] consider an action as a whole, which usually requires known start and end frames and is limited when action execution rate varies. Some other approaches such as [19] regard an action as a sequence of individual poses, and rely on pose-to-pose similarity measures. Since an action consists of spatio-temporal data, the temporal information plays a crucial role in recognizing action, which is ignored in a pose-to-pose approach. We thus propose using *pose transition*. One can thus compare actions by comparing their pose transitions.

### 3.1.3 Matching Pose Transition

The structure of a human can be divided into lines of body points using 2 body points. The problem of comparing articulated motions of human body thus transforms to comparing rigid motions of body line segments. According to proposition 2, the motion of a plane induces a fundamental matrix, which can be identified by its associated *fundamental ratios*. If two pose transitions are identical, their corresponding body point segments would induce the same *fundamental ratios*, which provide a measure for matching two pose transitions.

### 3.1.3.1 Fundamental matrix induced by a moving line segment

Assume that we are given an observed pose transition  $I_i \rightarrow I_j$  from sequence  $\{I_t\}$ , and  $J_m^k \rightarrow J_n^k$  from sequence  $\{J_t^k\}$  from an action dataset containing  $k$  actions.

When  $I_i \rightarrow I_j$  corresponds to  $J_m^1 \rightarrow J_n^1$ , and  $J_m^2 \rightarrow J_n^2$  one can regard them as observations of the same 3D pose transition by three different cameras  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ , and  $\mathbf{P}_3$ , respectively.

There are two instances of epipolar geometry associated with this scenario:

1. The mapping between the image pair  $\langle I_i, I_j \rangle$  and the image pairs  $\langle J_m^1, J_n^1 \rangle$ ,  $\langle J_m^2, J_n^2 \rangle$  is determined by the fundamental matrices  $\mathbf{F}_{12}$  and  $\mathbf{F}_{13}$  [22] related to  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ , and  $\mathbf{P}_3$ . Also, the mapping between image pair  $\langle J_m^1, J_n^1 \rangle$  and  $\langle J_m^2, J_n^2 \rangle$  is determined by the fundamental matrices  $\mathbf{F}_{23}$ . The projection of the camera center of  $\mathbf{P}_2$  in  $I_i$  or  $I_j$  is given by the epipole  $\mathbf{e}_{21}$ , which is found as the right null vector of  $\mathbf{F}_{12}$ . Similarly the image of the camera center of  $\mathbf{P}_1$  in  $J_m^1$  or  $J_n^1$  is the epipole  $\mathbf{e}_{12}$  given by the right null vector of  $\mathbf{F}_{12}^T$ . Similarly, the projection of the camera center of  $\mathbf{P}_3$  in  $I_i$  or  $I_j$  is given by the epipole  $\mathbf{e}_{31}$ , which is found as the right null vector of  $\mathbf{F}_{13}$ . Similarly the image of the camera center of  $\mathbf{P}_1$  in  $J_m^1$  or  $J_n^1$  is the epipole  $\mathbf{e}_{13}$  given by the right null vector of  $\mathbf{F}_{13}^T$ . Similarly the image of the camera center of  $\mathbf{P}_3$  in  $J_m^1$  or  $J_n^1$  is the epipole  $\mathbf{e}_{32}$  given by the right null vector of  $\mathbf{F}_{23}^T$ . Note that  $\mathbf{e}_{31}$  and  $\mathbf{e}_{32}$  are corresponding points in  $I_i$  or  $I_j$  and  $J_m^1$  or  $J_n^1$ , respectively. This fact would be used later on.

2. The other instance of epipolar geometry is between transitioned poses of a line segments of body points in two frames of the same camera, i.e. the fundamental matrix induced by a moving body line segment, which we denote as  $\mathcal{F}$ . We call this fundamental



matrix the *inter-pose fundamental matrix*, as it is induced by the transition of body point poses viewed by a stationary camera.

Let  $\mathbf{L}$  be a line of 3D points, whose motion lead to different image projections on  $I_i, I_j, J_m^1, J_n^1, J_m^2$  and  $J_n^2$  as  $\mathbf{L}_i, \mathbf{L}_j, \mathbf{L}_m^1, \mathbf{L}_n^1, \mathbf{L}_m^2$  and  $\mathbf{L}_n^2$ , respectively:

$$\mathbf{L}_i = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle, \mathbf{L}_j = \langle \mathbf{x}'_1, \mathbf{x}'_2 \rangle,$$

$$\mathbf{L}_m^1 = \langle \mathbf{y}_1, \mathbf{y}_2 \rangle, \mathbf{L}_n^1 = \langle \mathbf{y}'_1, \mathbf{y}'_2 \rangle.$$

$$\mathbf{L}_m^2 = \langle \mathbf{z}_1, \mathbf{z}_2 \rangle, \mathbf{L}_n^2 = \langle \mathbf{z}'_1, \mathbf{z}'_2 \rangle.$$

$\mathbf{L}_i$  and  $\mathbf{L}_j$  can be regarded as projections of a stationary 3D line  $\langle \mathbf{X}_1, \mathbf{X}_2 \rangle$  on two virtual cameras  $\mathbf{P}'_i$  and  $\mathbf{P}'_j$ . Assume that the epipoles in  $\mathbf{P}'_i$  and  $\mathbf{P}'_j$  are known and let us denote these as  $\mathbf{e}'_i = (\alpha_1, \beta_1, -1)^T$  and  $\mathbf{e}'_j = (\alpha'_1, \beta'_1, -1)^T$ , and  $\mathbf{e}'_m = (\alpha_2, \beta_2, -1)^T$  and  $\mathbf{e}'_n = (\alpha'_2, \beta'_2, -1)^T$ .

We can use the epipoles as parameters for the fundamental matrices induced by  $\mathbf{L}_i$  and  $\mathbf{L}_j$  and  $\mathbf{L}_m^1, \mathbf{L}_n^1$  [21]:

$$\mathcal{F}_1 = \begin{bmatrix} a_1 & b_1 & \alpha_1 a_1 + \beta_1 b_1 \\ c_1 & d_1 & \alpha_1 c_1 + \beta_1 d_1 \\ \alpha'_1 a_1 + \beta_1 c_1 & \alpha'_1 b_1 + \beta'_1 d_1 & \alpha_1 \alpha'_1 a_1 + \alpha'_1 \beta_1 b_1 + \\ & & \beta'_1 \alpha_1 c_1 + \beta_1 \beta'_1 d_1 \end{bmatrix} \quad (3.5)$$

$$\mathcal{F}_2 = \begin{bmatrix} a_2 & b_2 & \alpha_2 a_2 + \beta_2 b_2 \\ c_2 & d_2 & \alpha_2 c_2 + \beta_2 d_2 \\ \alpha'_2 a_2 + \beta_2 c_2 & \alpha'_2 b_2 + \beta'_2 d_2 & \alpha_2 \alpha'_2 a_2 + \alpha'_2 \beta_2 b_2 + \\ & & \beta'_2 \alpha_2 c_2 + \beta_2 \beta'_2 d_2 \end{bmatrix} \quad (3.6)$$

To solve for the 4 parameters, we have the following equations:

$$\mathbf{x}_1^T \mathcal{F}_1 \mathbf{x}_1 = 0 \quad (3.7)$$

$$\mathbf{x}_2^T \mathcal{F}_1 \mathbf{x}_2 = 0 \quad (3.8)$$

Similarly,  $\mathcal{F}_2$  induced by  $\mathbf{L}_m^1$  and  $\mathbf{L}_n^1$  can be computed from:

$$\mathbf{y}_1^T \mathcal{F}_2 \mathbf{y}_1 = 0 \quad (3.9)$$

$$\mathbf{y}_2^T \mathcal{F}_2 \mathbf{y}_2 = 0 \quad (3.10)$$

However, as we can see, this is an underdetermined system. Since we have more examples of an action in our dataset, we can use them. Given  $m^{th}$  example of the same action, we can denote  $\mathbf{e}_{\mathbf{m}1}^T$  and  $\mathbf{e}_{\mathbf{m}2}^T$  as the projection of the  $m^{th}$  camera center on the first and second camera center, respectively. Hence we have:

$$\mathbf{e}_{\mathbf{m}1}^T \mathcal{F}_1 \mathbf{e}_{\mathbf{m}1} = 0 \quad (3.11)$$

$$\mathbf{e}_{\mathbf{m}2}^T \mathcal{F}_2 \mathbf{e}_{\mathbf{m}2} = 0 \quad (3.12)$$

With  $m > 1$ , we have an overdetermined system, which can be easily solved by re-arranging the above equations in the form of  $Ax = 0$  and solving for the right null space of  $A$  to solve for the ratios.

The difficulty with Eq. 3.5 and 3.6 is that the epipoles  $\mathbf{e}'_i$ ,  $\mathbf{e}'_j$ ,  $\mathbf{e}'_m$  and  $\mathbf{e}'_n$  are unknown. Fortunately, however, the epipoles can be closely approximated as described below.

**Proposition 3** *If the exterior orientation of  $\mathbf{P}_1$  is related to that of  $\mathbf{P}_2$  by a translation, or by a rotation around an axis that lies on the axis planes of  $\mathbf{P}_1$ , then under the assumption:*

$$\mathbf{e}'_i = \mathbf{e}'_j = \mathbf{e}_1, \quad \mathbf{e}'_m = \mathbf{e}'_n = \mathbf{e}_2, \quad (3.13)$$

*we have:*

$$\mathcal{E}(\hat{\mathcal{F}}_1, \hat{\mathcal{F}}_2) = 0. \quad (3.14)$$

Under more general motion, the equalities in (3.13) become only approximate. However, we shall see in section 3.3 that this approximation is inconsequential in action recognition for a wide range of practical rotation angles. As described shortly, using equation (3.4) and the fundamental matrices  $\mathcal{F}_1$  and  $\mathcal{F}_2$  computed for every non-degenerate line segment, we can define a similarity measure for matching pose transitions  $I_i \rightarrow I_j$  and  $J_m^k \rightarrow J_n^k$ .

**Degenerate Configurations:** If the  $m^{th}$  camera projection is collinear with the 2 points in the line-segment, the problem becomes ill-conditioned. We can either ignore this camera center in favor of other camera centers (when  $m > 1$ ) or we can simply ignore the line-segment altogether. This does not produce any difficulty in practice, since with 11 body point representation used in this research, we obtain 55 possible line segments, the vast majority of which are in practice non-degenerate.

A special case is when the epipole is close to or at infinity, for which all line-segments would degenerate. We solve this problem by transforming the image points in projective space in a manner similar to Zhang et al. [70]. The idea is to find a pair of projective transformations  $\mathbf{Q}$  and  $\mathbf{Q}'$ , such that after transformation the epipoles and transformed image points are not at infinity. Note that these transformations do not affect the projective equality in Proposition 2.

### 3.1.3.2 Algorithm for Matching Pose Transitions

The algorithm for matching two pose transitions  $I_i \rightarrow I_j$  and  $J_m^k \rightarrow J_n^k$  is as follows:

1. Compute  $\mathbf{F}, \mathbf{e}_1, \mathbf{e}_2$  between image pair  $\langle I_i, I_j \rangle$  and  $\langle J_m^k, J_n^k \rangle$  using the method proposed in [20].
2. For each non-degenerate line segment  $\mathbf{L}_\ell$  that projects onto  $\mathbf{L}_i, \mathbf{L}_j, \mathbf{L}_m^k$  and  $\mathbf{L}_n^k$  in  $I_i, I_j, J_m^k$  and  $J_n^k$ , respectively, compute  $\hat{\mathcal{F}}_1, \hat{\mathcal{F}}_2$  as described above, and compute  $e_\ell = \mathcal{E}(\hat{\mathcal{F}}_1, \hat{\mathcal{F}}_2)$  from equation (3.4).
3. Compute the average error over all non-degenerate line segments using

$$E(I_i \rightarrow I_j, J_m^k \rightarrow J_n^k) = \frac{1}{L} \sum_{\ell=1 \dots L} e_\ell, \quad (3.15)$$

where  $L$  is the total number of non-degenerate line segments.

4. If  $E(I_i \rightarrow I_j, J_m^k \rightarrow J_n^k) < E_0$ , where  $E_0$  is some threshold, then the two pose transitions are matched. Otherwise, the two pose transitions are classified as mismatched.

### 3.1.4 Sequence Alignment

We represent an action  $A = \{I_{1,\dots,n}\}$  as a sequence of pose transitions,  $\mathcal{P}(A, r) = \{I_{1 \rightarrow r}, \dots, I_{(r-1) \rightarrow r}, I_{r \rightarrow (r+1)}, \dots, I_{r \rightarrow n}\}$ <sup>2</sup>, where  $I_r$  is an arbitrarily selected reference pose. If two sequences  $A = \{I_{1\dots n}\}$  and  $B = \{J_{1\dots m}\}$  contain the same action, then there exists an alignment between  $\mathcal{P}(A, r_1)$  and  $\mathcal{P}(B, r_2)$ , where  $I_{r_1}$  and  $J_{r_2}$  are two corresponding poses. To align the two sequences of pose transitions, we used dynamic programming. Therefore, our method to match two action sequences  $A$  and  $B$  can be described as follows:

1. Initialization: select a pose transition  $I_{i_0} \rightarrow I_{i_1}$  from  $A$  so that two poses are distinguishable. Then find its best matched pose transition  $J_{j_0} \rightarrow J_{j_1}$  in  $B$ , by checking all pose transitions in the sequence as described in section 3.1.3.
2. For all  $i = 1 \dots n, j = 1 \dots m$ , compute

$$\mathbf{S}_{i,j} = \begin{cases} \tau - E(I_{i_0} \rightarrow I_i, J_{j_0} \rightarrow J_j) & i \neq i_0, j \neq j_0 \\ \tau - E(I_{i_0} \rightarrow I_{i_1}, J_{j_0} \rightarrow J_{j_1}) & i = i_0, j = j_0 \\ 0 & \text{otherwise} \end{cases}$$

where  $\tau$  is a threshold, e.g.,  $\tau = 0.3$ .  $\mathbf{S}$  is the matching score matrix of  $\{I_{1,\dots,n}\}$  and  $\{J_{1,\dots,m}\}$ .

---

<sup>2</sup>For brevity of notation, we denote pose transition  $I_i \rightarrow I_j$  as  $I_{i \rightarrow j}$ .

3. Initialize the  $n \times m$  accumulated score matrix  $\mathbf{M}$  as

$$\mathbf{M}_{i,j} = \begin{cases} \mathbf{S}_{i,j} & i = 1 \text{ or } j = 1 \\ 0 & \text{otherwise} \end{cases}$$

4. Update matrix  $\mathbf{M}$  from top to bottom, left to right ( $i, j \geq 2$ ), using

$$\mathbf{M}_{i,j} = \mathbf{S}_{i,j} + \max\{\mathbf{M}_{i,j-1}, \mathbf{M}_{i-1,j}, \mathbf{M}_{i-1,j-1}\}.$$

5. Find  $(i^*, j^*)$  such that

$$(i^*, j^*) = \arg \max_{i,j} \mathbf{M}_{i,j}.$$

Then back trace  $\mathbf{M}$  from  $(i^*, j^*)$ , and record the path  $P$  until it reaches a non-positive element.

The matching score of sequences  $A$  and  $B$  is then defined as  $\mathcal{S}(A, B) = \mathbf{M}_{i^*, j^*}$ .

The back-traced path  $P$  provides an alignment between two video sequences. Note that this may not be a one-to-one mapping, since there may exist horizontal or vertical lines in the path, which means that a frame may have multiple candidate matches in the other video. In addition, due to noise and computational error, different selections of  $I_{i_0} \rightarrow I_{i_1}$  may lead to different valid alignment results.

### 3.1.5 Action Recognition

To solve the action recognition problem, we need a reference sequence (a sequence of 2D poses) for each known action, and maintain an action database of  $K$  actions,  $DB = \{J_t^1\}, \{J_t^2\}, \dots, \{J_t^K\}$ . To classify a given test sequence  $\{I_t\}$ , we match  $\{I_t\}$

against each reference sequence in  $DB$ , and classify  $\{I_t\}$  as the action of best-match, say  $\{J_t^k\}$ , if  $\mathcal{S}(\{I_t\}, \{J_t^k\})$  is above a threshold  $T$ . Due to the use of view-invariant *fundamental ratios* vector, our solution is invariant to camera intrinsic parameters and viewpoint changes, when the approximation of epipoles is valid. One major feature of the proposed method is that there is no training involved and we can recognize an action from a single example. This is experimentally verified in section 3.3.

### 3.2 Weighting-based Human Action Recognition

In the previous section, we saw how *fundamental ratios* can be used for action recognition. However, we assumed that all bodily joints have equal share in determining the action. This goes against common logic. For instance, in tennis, the feet movement will not be very discriminative of the action, whereas the upper body movement would be critical. There is evidence in applied perception literature [49] supporting the intuitive notion that different body parts have different contributions in determining the action.

With the line segment representation of human body pose, a similar assertion can be made on body line segments. Some line segments are more critical to recognizing action. Therefore, it would be reasonable to assume that by assigning appropriate weights to the similarity errors of body point line segments, the performance of pose and action recognition could be improved.

To test our idea, we selected two different sequences of walking action  $WA = \{I_{1...l}\}$  and  $WB = \{J_{1...m}\}$ , and a sequence of running action  $R = \{K_{1...n}\}$ . We aligned sequence  $WB$  and  $R$  to  $WA$ , using the alignment method described in section 3.1.4, and

obtained the corresponding alignment/mapping  $\psi : WA \rightarrow WB$  and  $\psi' : WA \rightarrow R$ . As discussed in section 3.1.3, the similarity of two poses is based on error scores of all body-point line segments motion. For each pair of matched poses  $\langle I_i, J_{\psi(i)} \rangle$ , we stacked the error scores of all line segments as a vector  $\mathbf{V}_e(i)$ :

$$\mathbf{V}_e(i) = \begin{bmatrix} E(\mathbf{L}_1) \\ E(\mathbf{L}_2) \\ \vdots \\ E(\mathbf{L}_T) \end{bmatrix}, \quad (3.16)$$

We then built an error score matrix  $\mathbf{M}_e$  for alignment  $\psi_{WA \rightarrow WB}$ :

$$\mathbf{M}_e = \begin{bmatrix} \mathbf{V}_e(1) & \mathbf{V}_e(2) & \dots & \mathbf{V}_e(l) \end{bmatrix}. \quad (3.17)$$

where each row  $i$  of  $\mathbf{M}_e$  indicates the dissimilarity scores of line segment  $i$  across the sequence, and the expected value of each column  $j$  of  $\mathbf{M}_e$  is the dissimilarity score of pose  $I_j$  and  $J_{\psi_{WA \rightarrow WB}(j)}$ . Similarly we built an error score matrix  $\mathbf{M}'_e$  for alignment  $\psi_{WA \rightarrow R}$ .

To analyze the role of a line segment  $i$  in differentiating between walking and running, we can compare the  $i$ -th row of  $\mathbf{M}_e$  and  $\mathbf{M}'_e$ , as shown in Figure 3.1 (a) - (f). We found that some line segments such as line segments 1, 2 and 11 have similar error scores in both cases, which means the motion of these line segments are similar in walking and running. Other line segments 19, 46 and 49 have high error scores in  $\mathbf{M}'_e$  and low error scores in  $\mathbf{M}_e$ . This means that the motion of these line segments in a running sequence is different from their motion in a walking sequence. Line segments 55, 94 and 116 reflect the variation in actions of walking and running, thus are more informative than



line segments 1, 21 and 90 for the task of differentiating between walking and running actions.

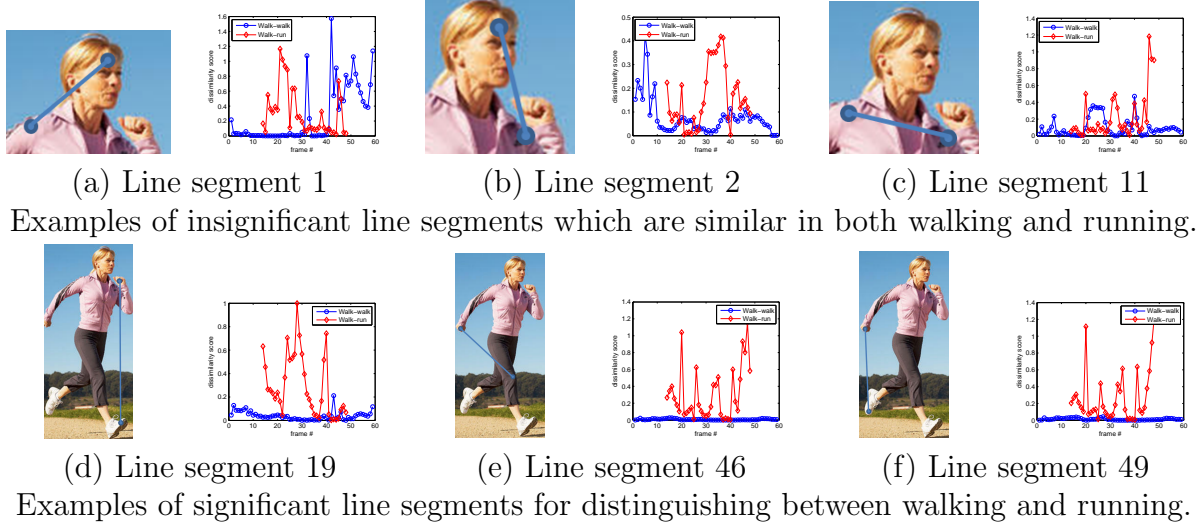


Figure 3.1: Roles of line segments in action recognition: (a) - (f) are the plots of dissimilarity scores of some line segments across frames in the walk-walk and walk-run alignments. As can be observed, line segments 1, 21 and 90 have similar error scores in both cases, which essentially means the motion of these line segments is similar in walking and running. But line segments 55, 94 and 116 have high error scores in  $\mathbf{M}'_e$  and low error scores in  $\mathbf{M}_e$ , which means that the motion of these line segments in a running sequence is different from their motion in a walking sequence. Therefore, these line segments reflect the variation in actions of walking and running and are much more useful for distinguishing between walking and running actions.

We analyzed sequences of different individuals performing the same action in order to gauge the relative importance of line segments in recognizing them as the same action. We selected four sequences  $\mathbf{G}0$ ,  $\mathbf{G}1$ ,  $\mathbf{G}2$ , and  $\mathbf{G}3$  of golf-swing action, and aligned  $\mathbf{G}1$ ,  $\mathbf{G}2$ , and  $\mathbf{G}3$  to  $\mathbf{G}0$  using the alignment method described in section 3.1.4, and then built error score matrices  $\mathbf{M}_e^1$ ,  $\mathbf{M}_e^2$ ,  $\mathbf{M}_e^3$  as described above. From the illustrations of  $\mathbf{M}_e^1$ ,  $\mathbf{M}_e^2$ ,  $\mathbf{M}_e^3$  in Figure 3.2 (a), (b) and (c), the dissimilarity scores of some line segments, such as line segments 53 (see Figure 3.2 (f)) , is very consistent across individuals. Some other line segments such as line segments 6 (Figure 3.2 (d)) and 50 (Figure 3.2 (e)) have

various error score patterns across individuals, that is, these line segments represent the variations in individuals performing the same action.

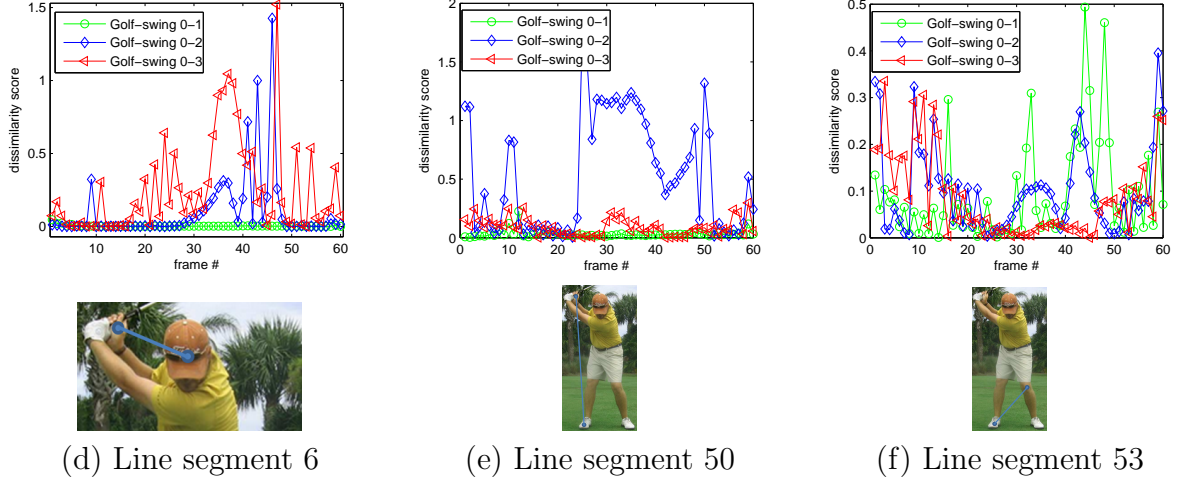


Figure 3.2: Roles of different line segments in action recognition. We selected four sequences  $G0$ ,  $G1$ ,  $G2$ , and  $G3$  of golf-swing action, and align  $G1$ ,  $G2$ , and  $G3$  to  $G0$  using the alignment method described in Section 2, and then build error score matrix  $\mathbf{M}_e^1$ ,  $\mathbf{M}_e^2$ ,  $\mathbf{M}_e^3$  correspondingly as in above experiments. As can be observed, the dissimilarity scores of some line segments, such as line segments 53 is very consistent across individuals. Some other line segments such as line segments 6 and 50 have various error score patterns across individuals, that is, these line segments represent the variations of individuals performing the same action.

**Definition 1** We call a line segment a significant line segments of an action  $R$  if it is able to differentiate between  $R$  and other actions. Line segments which are unable to distinguish between  $R$  and other actions are referred to as trivial line segments of action  $A$ .

A typical significant line segments should be able to convey the variations between actions while tolerating the variations of the same action performed by different individuals. Line segments 19, 46 and 49 are significant line segments for walking action, while line segment 53 is a significant line segment for the golf-swing action.

Therefore, we should place more emphasis on the significant line segments while reducing the negative impact of trivial line segments. This means that we should be assigning appropriate weights to the body-point line segments. In our approach to action recognition, this can be achieved by assigning appropriate weights to the similarity errors of body point line segments in equation (3.15). That is, equation (3.15) can be rewritten as:

$$E(I_i \rightarrow I_j, J_m^k \rightarrow J_n^k) = \sum_{\ell=1 \dots L} \omega_\ell e_\ell, \quad (3.18)$$

where  $L$  is the total number of non-degenerate line segments and  $\omega_1 + \omega_2 + \dots + \omega_L = 1$ .

But how do we determine the optimal set of weights  $\omega_i$  for different actions. We need an automatic assignment of weight values for a robust and efficient action recognition system. To achieve this, we use a fixed size dataset of training sequences to learn weight values. Our method works as follows: suppose we are given a training dataset  $\mathcal{T}$  which consists of  $K \times J$  action sequences for  $J$  different actions, performed by  $K$  different individuals. Let  $\omega_\ell$  be the weight value of body joint with label  $\ell$  ( $\ell = 1 \dots L$ ) for a given action. We need to find the optimal weights  $\omega_\ell$  that maximize the similarity error between sequences of different actions and minimize those of same actions. Since the size of the dataset and the alignments of sequences are fixed, this turns out to be an optimization problem over  $\omega_\ell$ . So we need to define a good objective function  $f(\omega_1, \dots, \omega_L)$  for this purpose, and use optimization.

### 3.2.1 Weights on line segments versus Weights on Body Points

Given a human body model of  $n$  points, we have at most  $\binom{n}{2}$  line segments, and need to solve a  $\binom{n}{2}$  dimensional optimization problem for weight assignment. Using a human body model of 11 points, this yields an extremely high dimensional ( $\binom{11}{2} = 55$  dimensions) problem. We also know that the body point line segments are not independent of each other. In fact, adjacent line segments are correlated by their common body point, and the importance of a line segments is also determined by the importance of its two body points. Therefore, instead of using  $\binom{n}{2}$  variables for weights of  $\binom{n}{2}$  line segments, we assign  $n$  weights  $\omega_{1\dots n}$  to the body points  $P_{1\dots n}$ , where:

$$\omega_1 + \omega_2 + \dots + \omega_n = 1. \quad (3.19)$$

The weight of a line segments  $\mathbf{L} = \langle P_i, P_j \rangle$  can then be computed as:

$$\lambda_{\mathbf{L}} = \frac{\omega_i + \omega_j}{n} \quad (3.20)$$

Note that the definition of  $\lambda$  in (3.20) ensures that  $\lambda_1 + \lambda_2 + \dots + \lambda_T = 1$ . Using (3.20), equation (3.18) is rewritten as:

$$\mathcal{E}(I_1 \rightarrow I_2, J_i \rightarrow J_j) = \frac{1}{n} \text{Median}_{1 \leq i < j \leq n}((\omega_i + \omega_j) \cdot E(\mathbf{L}_{i,j})), \quad (3.21)$$

By introducing weights  $\{\omega_{1\dots n}\}$  to body points, we reduce the high dimensional optimization problem to a lower dimensional, and more tractable problem.

### 3.2.2 Automatic Adjustment of Weights

Given two sequences  $A = \{I_{1\dots N}\}$ ,  $B = \{J_{1\dots M}\}$ , and the known alignment  $\psi : A \rightarrow B$ , the similarity of  $A$  and  $B$  is:

$$\mathcal{S}(A, B) = \sum_{l=1}^N S(l, \psi(l)) = N\tau - \quad (3.22)$$

$$N \sum_{l=1}^N \mathcal{E}(I_{l \rightarrow r_1}, J_{\psi(l) \rightarrow r_2}), \quad (3.23)$$

where  $r_1$  and  $r_2$  are computed reference poses, and  $\tau$  is a threshold, which we set as suggested in [53, 54]. Therefore, the approximate similarity score of  $A$  and  $B$  is:

$$\begin{aligned} \bar{\mathcal{S}}(A, B) = N\tau - \frac{1}{N} \sum_{l=1}^N \sum_{1 \leq i < j \leq n} \\ (\omega_i + \omega_j) \cdot E^{l, \psi(l)}(\mathbf{L}_{i,j}). \end{aligned} \quad (3.24)$$

Considering that  $N$ ,  $\tau$ ,  $n$  and  $E^{l, \psi(l)}(\mathbf{L}_{i,j})$  are constants given the alignment  $\psi$ , equation (3.24) can be further rewritten into a simpler form:

$$\bar{\mathcal{S}}(A, B) = a_0 - \sum_{i=1}^{n-1} a_i \cdot \omega_i, \quad (3.25)$$

where  $\{a_i\}$  are constants computed from (3.24).

A good objective function would give a higher weighting to significant line segments while trivial line segments would be assigned lower weights. Suppose we have a training dataset  $\mathcal{T}$  which consists of  $K \times J$  action sequences for  $J$  different actions, each of which with  $K$  pre-aligned sequences performed by various individuals.  $\mathcal{T}_k^j$  is the  $k$ -th sequence in the group of action  $j$ , and  $\mathcal{R}^j$  is the reference sequence of action  $j$ . To find the optimal weight assignment for action  $j$ , we define the objective function as:

$$f^j(\omega_1, \omega_2, \dots, \omega_{n-1}) = \mathcal{Q}_1 + \alpha \mathcal{Q}_2 - \beta \mathcal{Q}_3, \quad (3.26)$$

where  $\alpha$  and  $\beta$  are non-negative constants and

$$\mathcal{Q}_1 = \frac{1}{K} \sum_{k=1}^K \bar{\mathcal{S}}(\mathcal{R}^j, \mathcal{T}_k^j), \quad (3.27)$$

$$\mathcal{Q}_2 = \frac{1}{K} \sum_{k=1}^K \bar{\mathcal{S}}(\mathcal{R}^j, \mathcal{T}_k^j)^2 - \mathcal{Q}_1^2, \quad (3.28)$$

$$\mathcal{Q}_3 = \frac{1}{K(J-1)} \sum_{1 \leq i \leq J, i \neq j} \sum_{k=1}^K \bar{\mathcal{S}}(\mathcal{R}^j, \mathcal{T}_k^i). \quad (3.29)$$

The optimal weights for action  $j$  are then computed using:

$$\langle \omega_1, \dots, \omega_{n-1} \rangle = \underset{\omega_1, \omega_2, \dots, \omega_{n-1}}{\operatorname{argmax}} f^j(\omega_1, \dots, \omega_{n-1}, \alpha, \beta). \quad (3.30)$$

In this objective function, we use  $\mathcal{T}_1^j$  as the reference sequence for action  $j$ , and the term  $\mathcal{Q}_1$  and  $\mathcal{Q}_2$  are the mean and variance of similarity scores between  $\mathcal{T}_1^j$  and other sequences in the same action.  $\mathcal{Q}_3$  is the mean of similarity scores between  $\mathcal{T}_1^j$  and all sequences in other different actions. Hence  $f^j(\omega_1, \omega_2, \dots, \omega_{n-1})$  achieves high similarity scores for all sequences of same action  $j$ , and low similarity scores for sequences of different actions. The second term  $\mathcal{Q}_2$  may be interpreted as a regularization term to ensure the consistency of sequences in the same group.

Since  $\mathcal{Q}_1$  and  $\mathcal{Q}_3$  are linear functions, and  $\mathcal{Q}_2$  is quadratic polynomial, our objective function  $f^j(\omega_1, \omega_2, \dots, \omega_{n-1})$  is quadratic polynomial function, and the optimization problem becomes a quadratic programming (QP) problem. There are a number of methods for solving the QP problem, including interior point, active set, conjugate gradient, etc. In our problem, we adopted the conjugate gradient method, with the initial weight values set to  $\langle \frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n} \rangle$ .

**Degenerate line segments:** As before, degenerate line segments are ignored. As explained earlier, with 11 body points, we obtain a total of 55 possible triplets, the vast majority of which are in practice non-degenerate.

### 3.3 Experimental Results and Discussion

We first examine our method on semi-synthetic data. In particular, we first demonstrate that our method is resilient to viewpoint changes and noise. We then present our results for action recognition and demonstrate that weighting considerably improves our results. We then present our results on two sets of real video data: the IXMAS multiple view data set [65], and our own data set consisting of a total of 56 video sequences of 8 actions (available at <http://cil.cs.ucf.edu/actionrecognition.html>).

#### 3.3.1 Analysis based on motion capture data

We generated our data based on the CMU Motion Capture Database, which consists of 3D motion data for a large number of human actions. We generated the semi-synthetic data by projecting 3D points onto images through synthesized cameras. In other words, our test data consist of video sequences of true persons, but the cameras are synthetic, resulting in semi-synthetic data to which various levels of noise were added. Instead of using all body points provided in CMU’s database, we employed a body model that consists of only eleven points, including head, shoulders, elbows, hands, knees and feet (see [Figure 3.3](#)).

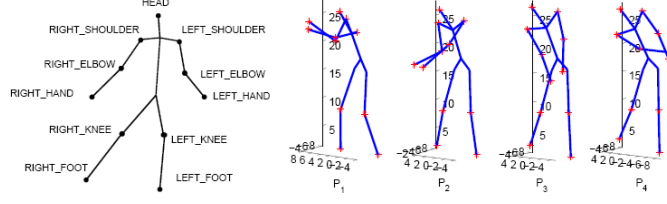


Figure 3.3: Left: Our body model. Right: Experiment on view-invariance. Two different pose transitions  $P_1 \rightarrow P_2$  and  $P_3 \rightarrow P_4$  from a golf swing action are used.

### 3.3.1.1 Testing View Invariance

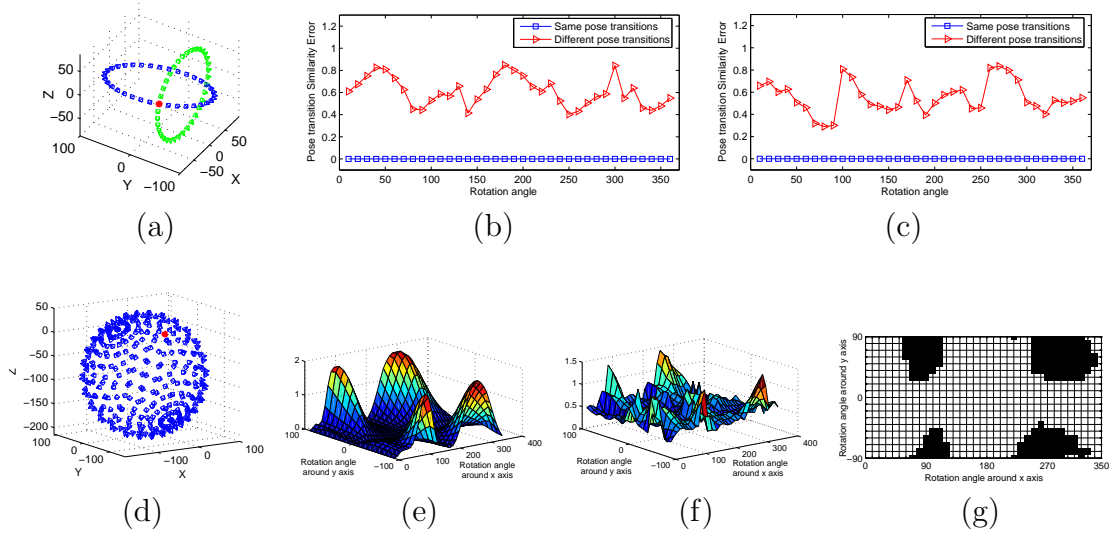


Figure 3.4: Analysis of view invariance: (a) Camera 1 is marked in red, and all positions of camera 2 are marked in blue and green. (b) Errors for same and different pose transitions when camera 2 is located at viewpoints colored as green in (a). (c) Errors of same and different pose transitions when camera 2 is located at viewpoints colored as blue in (a). (d) General camera motion: Camera 1 is marked as red, and camera 2 is distributed on a sphere. (e) Error surface of same pose transitions for all distributions of camera 2 in (d). (f) Error surface of different pose transitions for all distribution of camera 2 in (d). (g) The regions of confusion for (d) marked in black (see text).

We selected four different poses  $P_1, P_2, P_3, P_4$  from a golf swinging sequence (see [Figure 3.3](#)). We then generated two cameras as shown in [Figure 3.4](#) (a): camera 1 was placed at an arbitrary viewpoint (marked by red color), with focal length  $f_1 = 1000$ ; camera 2 was obtained by rotating camera 1 around an axis on  $x$ - $z$  plane of camera 1



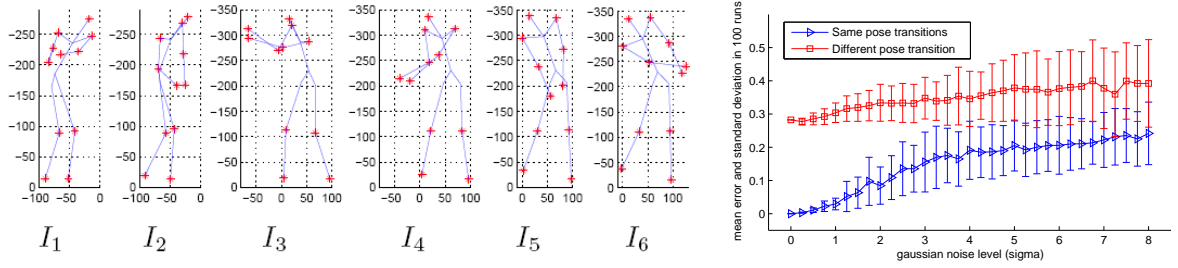


Figure 3.5: Robustness to noise:  $I_1$  and  $I_2$  are the images in camera 1, and  $I_3, I_4, I_5$  and  $I_6$  are the images in camera 2. Same and different actions are distinguished unambiguously for  $\sigma < 4$

(colored as green), and a second axis on  $y$ - $z$  plane of camera 1 (colored as blue), and changing focal length as  $f_2 = 1200$ . Let  $I_1$  and  $I_2$  be the images of poses  $P_1$  and  $P_2$  on camera 1 and  $I_3, I_4, I_5$  and  $I_6$  the images of poses  $P_1, P_2, P_3$  and  $P_4$  on camera 2, respectively. Two sets of pose similarity errors were computed at all camera positions shown in Figure 3.4 (a):  $E(I_1 \rightarrow I_2, I_3 \rightarrow I_4)$  and  $E(I_1 \rightarrow I_2, I_5 \rightarrow I_6)$ . The results are plotted in Figure 3.4 (b) and (c), which show that, when two cameras are observing the same pose transitions, the error is zero regardless of their different viewpoints, confirming proposition 3.

Similarly, we fixed camera 1 and moved camera 2 on a sphere as shown in Figure 3.4 (d). The errors  $E(I_1 \rightarrow I_2, I_3 \rightarrow I_4)$  and  $E(I_1 \rightarrow I_2, I_5 \rightarrow I_6)$  are shown in Figure 3.4 (e) and (f). Under this more general camera motion, the pose similarity score of corresponding poses is not always zero, since the epipoles in equations (3.5) and (3.6) are approximated. However, this approximation is inconsequential in most situations, because the error surface of different pose transitions is in general above that of corresponding pose transitions. Figure 3.4 (h) shows the regions (black colored) where approximation is invalid. These regions correspond to the situation that the angles be-

tween camera orientations around 90 degrees, which usually implies severe self-occlusion and lack of corresponding points in practice. The experiments on real data in section 3.3.2 also show the validity of this approximation under practical camera viewing angles.

### 3.3.1.2 Testing Robustness to Noise

Without loss of generality, we used the four poses in Figure 3.3 to analyze the robustness of our method to noise. Two cameras with different focal lengths and view-points were examined. As shown in Figure 3.5,  $I_1$  and  $I_2$  are the images of poses  $P_1$  and  $P_2$  on camera 1 and  $I_3, I_4, I_5$  and  $I_6$  are the images of  $P_1, P_2, P_3$  and  $P_4$  on camera 2. We then added Gaussian noise to the image points, with  $\sigma$  increasing from 0 to 8 pixels. The errors  $E(I_1 \rightarrow I_2, I_3 \rightarrow I_4)$  and  $E(I_1 \rightarrow I_2, I_5 \rightarrow I_6)$  were computed. For each noise level, the experiment was repeated for 100 independent trials, and the mean and standard deviation of both errors were calculated (see Figure 3.5). As shown in the results, the two cases are distinguished unambiguously until  $\sigma$  increases to 4.0, i.e., up to possibly 12 pixels. Note that the image sizes of the subject were about  $200 \times 300$ , which implies that our method performs remarkably well under high noise.

### 3.3.1.3 Performance in Action Recognition

We selected 5 classes of actions from CMU’s MoCap dataset: walk, jump, golf swing, run, and climb. Each action class is performed by 3 actors, and each instance of 3D action is observed by 17 cameras, as shown in Figure 2.3. The focal lengths were changed randomly in the range of  $1000 \pm 300$ .

Our dataset consists of totally 255 video sequences, from which we generated a reference action Database (DB) of 5 video sequences, i.e. one video sequence for each action class. The rest of the dataset was used as test data, and each sequence was matched against all actions in the DB and classified as the one with the highest score. For each sequence matching, 10 random initializations were tested and the best score was used. Classification results without weighting are summarized in [Table 3.1](#). The overall recognition rate is 85.60%.

For weighting, we build a MoCap training dataset which consists of total of  $2 \times 17 \times 5 = 170$  sequences for 5 actions (walk, jump, golf swing, run, and climb): each action is performed by 2 subjects, and each instance of action is observed by 17 cameras at different random locations. We use the same set of reference sequences for the 5 actions as the unweighted case, and align the sequences in the training set against the reference sequences. To obtain optimal weighting for each action  $j$ , we first aligned all sequences against the reference sequence  $\mathcal{R}^j$ , and stored the similarity scores of line segments for each pair of matched poses. The objective function  $f^j(\omega_1, \omega_2, \dots, \omega_{10})$  is then built based on equation (3.26), and the computed similarity scores of line segments in the alignments.  $f^j(\cdot)$  is a 10-dimensional function, and the weights  $\omega_i$  are constrained by

$$\begin{cases} 0 \leq \omega_i \leq 1, i = 1 \dots 10, \\ \sum_{i=1}^{10} \omega_i \leq 1. \end{cases} \quad (3.31)$$

The optimal weights  $\langle \omega_1, \omega_2, \dots, \omega_{10} \rangle$  are then searched to maximize  $f^j(\cdot)$ , with the initialization at  $\langle \frac{1}{11}, \frac{1}{11}, \dots, \frac{1}{11} \rangle$ . The conjugate gradient method is then applied to solve this optimization problem. After performing the above steps for all the actions, we

Table 3.1: Confusion matrix before applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 85.60%.

| Ground-truth | Recognized as |      |            |     |       |
|--------------|---------------|------|------------|-----|-------|
|              | Walk          | Jump | Golf Swing | Run | Climb |
| Walk         | 42            | 2    | 1          | 3   | 2     |
| Jump         | 2             | 46   |            | 1   | 1     |
| Golf Swing   | 1             | 1    | 45         | 2   | 1     |
| Run          | 4             | 3    |            | 41  | 2     |
| Climb        | 4             | 3    | 1          | 2   | 40    |

Table 3.2: Confusion matrix after applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 92.40%, which is an improvement of 6.8% compared to the nonweighted case.

| Ground-truth | Recognized as |      |            |     |       |
|--------------|---------------|------|------------|-----|-------|
|              | Walk          | Jump | Golf Swing | Run | Climb |
| Walk         | 45            | 1    | 1          | 2   | 1     |
| Jump         | 2             | 47   |            | 1   |       |
| Golf Swing   |               | 1    | 47         | 1   | 1     |
| Run          | 2             | 1    |            | 46  | 1     |
| Climb        | 1             | 1    |            | 2   | 46    |

obtained a set of weights  $\mathcal{W}^j$  for each action  $j$  in our database. Classification results are summarized in [Table 3.2](#). The overall recognition rate is 92.4%, which is an improvement of 6.8% compared to the unweighted case.

### 3.3.2 Results on real data

#### 3.3.2.1 UCF-CIL Dataset

The UCF-CIL dataset consists of video sequences of 8 classes of actions collected on the internet (see [Figure 3.6](#)): ballet fouette, ballet spin, push-up exercise, golf swing, one-handed tennis backhand stroke, two-handed tennis backhand stroke, tennis forehand stroke, and tennis serve. Each action is performed by different subjects, and the videos are taken by different unknown cameras from various viewpoints. In addition, videos in the same class of action may have different starting and ending points, thus may be only partially overlapped. The execution speeds also vary in the sequences of each action. Self-occlusion also exists in many of the sequences, e.g., golf, tennis, etc.

We built an action database DB by selecting one sequence for each action; the rest were used as test data, and were matched against all actions in the DB. The action was recognized as the one with the highest matching score for each sequence. The confusion matrix is shown in [Table 3.3](#), which indicates an overall 95.83% classification accuracy for real data. As shown by these results, our method provides a successful recognition of various actions by different subjects, regardless of camera intrinsic parameters and viewpoints.

Table 3.3: Confusion matrix before applying weighting: Large values on the diagonal entries indicate accuracy. The overall recognition rate is 95.83%. The actions are denoted by numbers: 1 - ballet fouette, 2 - ballet spin, 3 - pushup, 4 - golf swing, 5 - one handed tennis backhand, 6 - two handed tennis backhand, 7 - tennis forehand, 8 - tennis serve.

| Ground-true<br>actions | Recognized as action |    |    |    |    |    |    |    |
|------------------------|----------------------|----|----|----|----|----|----|----|
|                        | #1                   | #2 | #3 | #4 | #5 | #6 | #7 | #8 |
| #1                     | 3                    |    |    |    |    |    |    |    |
| #2                     | 1                    | 10 |    |    |    |    |    |    |
| #3                     |                      |    | 5  |    |    |    |    |    |
| #4                     |                      |    |    | 7  |    |    |    |    |
| #5                     |                      |    |    |    | 3  |    |    |    |
| #6                     |                      |    |    |    | 1  | 6  |    |    |
| #7                     |                      |    |    |    |    |    | 3  |    |
| #8                     |                      |    |    |    |    |    |    | 9  |

Table 3.4: Confusion matrix after applying weighting: The overall recognition rate is 100%, which is an improvement of 4.17% compared to the nonweighted case. The actions are the same as in [Table 3.3](#).

| Ground-true<br>actions | Recognized as action |    |    |    |    |    |    |    |
|------------------------|----------------------|----|----|----|----|----|----|----|
|                        | #1                   | #2 | #3 | #4 | #5 | #6 | #7 | #8 |
| #1                     | 3                    |    |    |    |    |    |    |    |
| #2                     |                      | 11 |    |    |    |    |    |    |
| #3                     |                      |    | 5  |    |    |    |    |    |
| #4                     |                      |    |    | 7  |    |    |    |    |
| #5                     |                      |    |    |    | 3  |    |    |    |
| #6                     |                      |    |    |    |    | 7  |    |    |
| #7                     |                      |    |    |    |    |    | 3  |    |
| #8                     |                      |    |    |    |    |    |    | 9  |

We test each sequence using the take-one-out strategy. With weighting, the classification results are summarized in [Table 3.4](#). The overall recognition rate is 100%, which is an improvement of 4.17% compared to the nonweighted case.

### 3.3.2.2 IXMAS data set

We also evaluated our method on IXMAS data set [65], which has 5 different views of 13 different actions, each performed 3 times by 11 different actors. We tested on actions,  $\{1, 2, 3, 4, 5, 8, 9, 10, 11, 12\}$ . Similar to [65], we applied our method on all actors except for “Pao” and “Srikumar,” and used “Andreas 1” under “cam1” as the reference for all actions similar to [54]. The rest of the sequences were used to test our method. The recognition results are shown in Table 3.6 for non-weighted case. The average recognition rate is 87.3%. For weighting, we tested each sequence by randomly generating a reference dataset of  $2 \times 5 \times 10 = 100$  sequences for 10 actions performed by 2 people observed from 5 different viewpoints. The results are shown in Table 3.7. The average recognition rate is 92.1%, which boosts 4.8% over the non-weighted case. In addition, we compare our method to others in Table 3.5. As can be seen, our method improves on each camera view.

### 3.3.2.3 Testing Occlusion

As discussed earlier, we handle occlusions by ignoring the line segments involving the occluded points. Since there are a total of 11 points in our body model, there are a total of 55 line segments. If, let’s assume, 3 points are occluded, there are still 28 line segments. While the non-weighted method would be expected to degenerate when lesser line segments are used, weighting the line segments would still be able to differentiate between actions, which are dependent on the non-occluded points. While our previous experiments implicitly involve self-occlusion, in this section, we want to rigorously test

Table 3.5: Recognition rates in % on IXMAS dataset

| Method                    | all         | cam1        | cam2        | cam3        | cam4        | cam5        |
|---------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>fundamental ratios</i> | 87.3        | 92.0        | 89.6        | 86.6        | 82.0        | 78.0        |
| without weighting         |             |             |             |             |             |             |
| <i>fundamental ratios</i> | <b>92.1</b> | <b>94.2</b> | <b>93.5</b> | <b>94.4</b> | <b>92.6</b> | <b>82.2</b> |
| with weighting            |             |             |             |             |             |             |
| Weinland [64]             | 83.5        | 87.0        | 88.3        | 85.6        | 87.0        | 69.7        |
| Weinland [63]             | 57.9        | 65.4        | 70.0        | 54.3        | 66.0        | 33.6        |
| Reddy [44]                | 72.6        | 69.6        | 69.2        | 62.0        | 65.1        | -           |
| Tran [57]                 | 80.2        | -           | -           | -           | -           | -           |
| Junejo [25]               | 72.7        | 74.8        | 74.5        | 74.8        | 70.6        | 61.2        |
| Liu [31]                  | -           | 76.7        | 73.3        | 72.0        | 73.0        | -           |
| Farhadi [14]              | 58.1        | -           | -           | -           | -           | -           |
| Shen [54]                 | 90.2        | -           | -           | -           | -           | -           |

our method when occlusion is present. In particular, we test for these different scenarios:

(i) Upper body is occluded including the head and shoulder points. (ii) The right side of the body is occluded including the shoulder, arm, hand, and knee points. (iii) The left side of the body is occluded including the shoulder, arm, hand, and knee points. (iv) Lower body is occluded including the knee and feet points. Therefore (i) has 3 occluded points and the rest of the test cases have 4 occluded points. The results are shown in

Table 3.8, Table 3.9, Table 3.10, and Table 3.11.



Table 3.6: Confusion matrix for IXMAS dataset before applying weighting. The actions are denoted by numbers: 1 = Check Watch, 2 = Cross Arms, 3 = Scratch Head, 4 = Sit Down, 5 = Get up, 8 = Wave, 9 = Punch, 10 = Kick, 11 = Point, and 12 = Pick Up

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 87.2 | 89.6 | 85.1 | 83.1 | 89.6 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 90.4 | 89.6 | 82.1 | 91.1 | 85.3 |

Table 3.7: Confusion matrix for IXMAS dataset after applying weighting: The overall recognition rate is 92.1%, which is an improvement of 4.8% compared to the nonweighted case. The actions are the same as in [Table 3.6](#).

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 93.4 | 94.6 | 89.1 | 87.2 | 94.8 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 95.6 | 93.3 | 87.1 | 95.6 | 90.1 |

As can be seen from these results, our method is able to recognize actions even when such drastic occlusions are present. The few low percentages in the tables correspond to actions that are more or less dependent on the occluded part. For instance, “kick” action has a percentage of only 5.5% when lower body is occluded. But this action is solely based on the lower part of the body. Therefore, it is not surprising that the recognition rate is low. In general, the recognition rates are low since we are using lesser number of line segments, and more importantly, we are using lesser number of points to compute the fundamental matrix (when 4 points are occluded, we are forced to use the 7 point algorithm [21]).

Table 3.8: Confusion matrix when head and two shoulder points are occluded. The actions are the same as in [Table 3.6](#).

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 85.5 | 91.1 | 83.3 | 81.1 | 91.1 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 92.3 | 90.3 | 83.3 | 90.4 | 83.3 |

Table 3.9: Confusion matrix when the right side of the body is occluded including the right shoulder, arm, hand, and knee point.

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 83.3 | 54.5 | 5.5  | 58.8 | 61.3 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 3.3  | 10.3 | 79.1 | 5.6  | 16.1 |

Table 3.10: Confusion matrix when the left side of the body is occluded including the left shoulder, arm, hand, and knee point.

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 3.3  | 47.5 | 75.5 | 57.7 | 66.7 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 83.3 | 73.3 | 76.7 | 77.1 | 66.7 |

Table 3.11: Confusion matrix when the lower body is occluded including the two knee and feet points.

|                    |      |      |      |      |      |
|--------------------|------|------|------|------|------|
| Action             | 1    | 2    | 3    | 4    | 5    |
| Recognition rate % | 86.6 | 83.3 | 78.1 | 45.2 | 54.8 |
| Action             | 8    | 9    | 10   | 11   | 12   |
| Recognition rate % | 81.1 | 79.3 | 5.5  | 78.1 | 36.6 |

### 3.3.3 How soon can we recognize the action?

We also experimented with how soon our method is able to distinguish between different actions. This is helpful to gauge whether our method would be able to perform real-time or not. To do this, we looked at all the correctly classified sequences and the results are summarized in [Table 3.12](#). So, for instance, for action 1, on average we can detect the action after 60% of the sequence. The best case and the worst case are also provided.

Table 3.12: This table shows how soon we can recognize an action for IXMAS dataset.

| % of Sequence used: |    |    |    |    |    |    |    |    |    |    |
|---------------------|----|----|----|----|----|----|----|----|----|----|
| Action              | 1  | 2  | 3  | 4  | 5  | 8  | 9  | 10 | 11 | 12 |
| Best case           | 30 | 33 | 56 | 35 | 40 | 56 | 48 | 45 | 60 | 37 |
| Worst case          | 88 | 77 | 91 | 67 | 77 | 88 | 81 | 89 | 92 | 79 |
| Average case        | 60 | 50 | 77 | 56 | 66 | 69 | 63 | 77 | 78 | 55 |

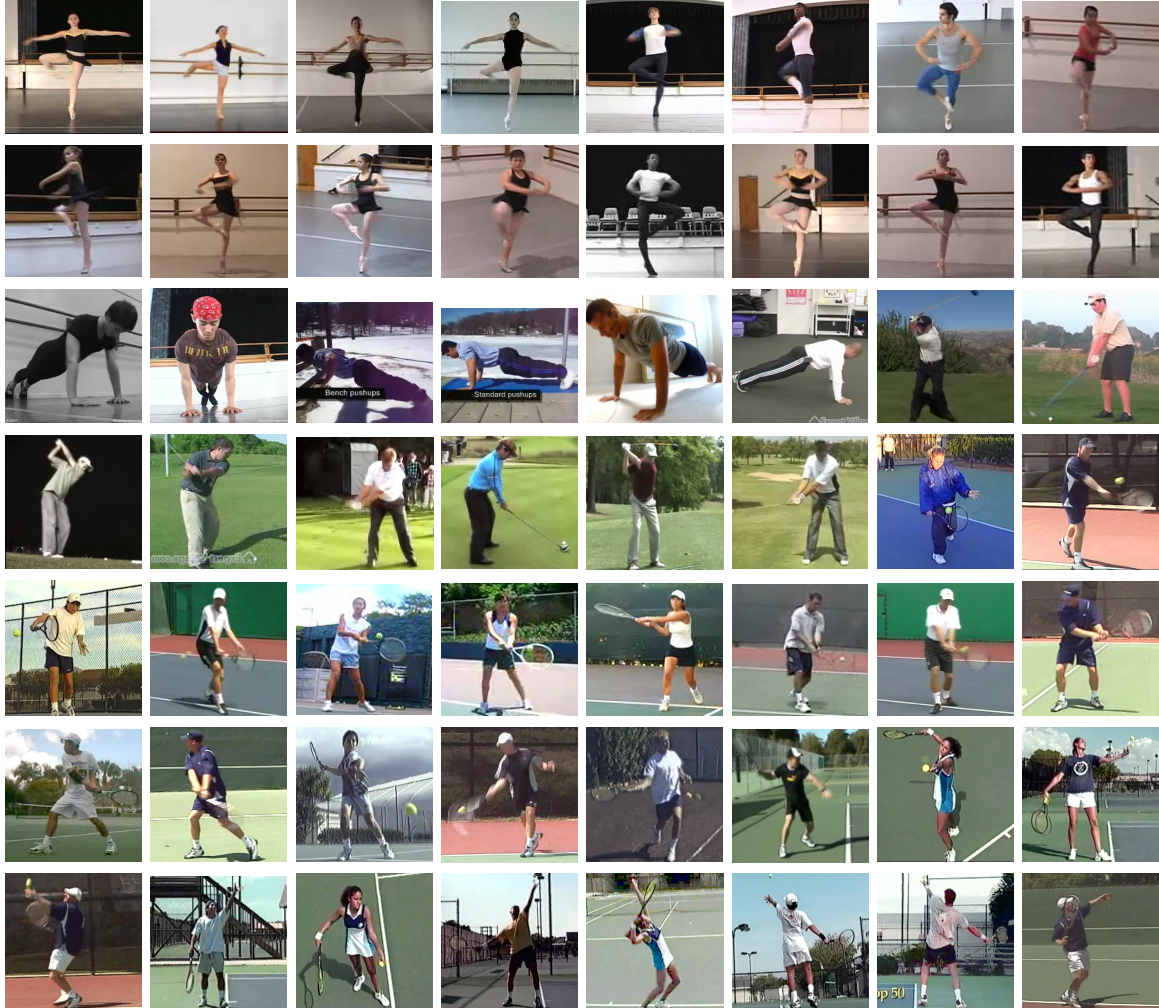


Figure 3.6: A set of 56 sequences in 8 categories (actions) used to test the proposed method. Ballet fouettes: (1)-(4); ballet spin: (5)-(16); push-up: (17)-(22); golf swing: (23)-(30); one-handed tennis backhand stroke: (31)-(34); two-handed tennis backhand stroke: (35)-(42); tennis forehand stroke: (43)-(46); tennis serve: (47)-(56).

## CHAPTER 4: ACTION RECOGNITION USING PROJECTIVE DEPTH

We propose to use the concept of the “Projective Depth” for use in action recognition. Since the image sequence is acquired from a camera, we lose the depth information. However, given a 3D plane viewed by two camera, it is possible to find the “projective depth” of a given point relative to this plane. Let us first look at the concept of “projective depth.”

### 4.1 Projective Depth

A world point  $\mathbf{X} = (\mathbf{x}^T, \rho)^T$  is imaged at  $\mathbf{x}$  in the first image and at

$$\mathbf{x}' = (1 - \rho)\mathbf{H}\mathbf{x} + \rho\mathbf{e}' \quad (4.1)$$

in the second image. This world point introduces a parallax relative to the plane as illustrated in [Figure 4.1](#). Since  $\mathbf{x}'$ ,  $\mathbf{e}'$ , and  $\mathbf{H}\mathbf{x}$  are collinear, the scalar  $\rho$  is the parallax *relative* to the plane  $\pi$ , which can be expressed as:

$$\rho = \frac{\mathbf{x}' - \mathbf{H}\mathbf{x}}{\mathbf{e}' - \mathbf{H}\mathbf{x}} \quad (4.2)$$

$\rho$  is 0 implies the point is on the plane. Otherwise the *sign* of  $\rho$  indicates which *side* of the plane  $\pi$  the point  $\mathbf{X}$  is. However, in the absence of oriented projective geometry the sign of a homogenous object, and the side of a plane have no meaning. To solve this,

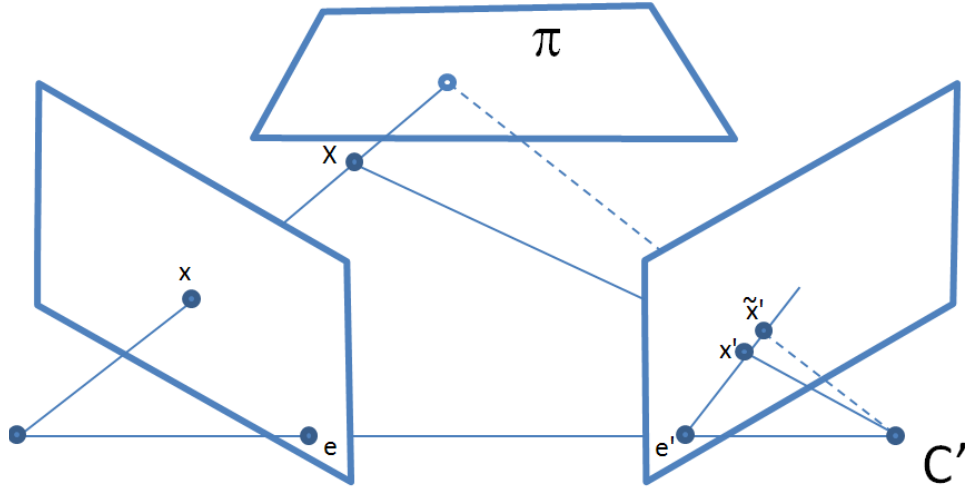


Figure 4.1: A point  $\mathbf{x}$  in one image is transferred via the plane  $\pi$  to a matching point  $\mathbf{x}'$  in the second image.

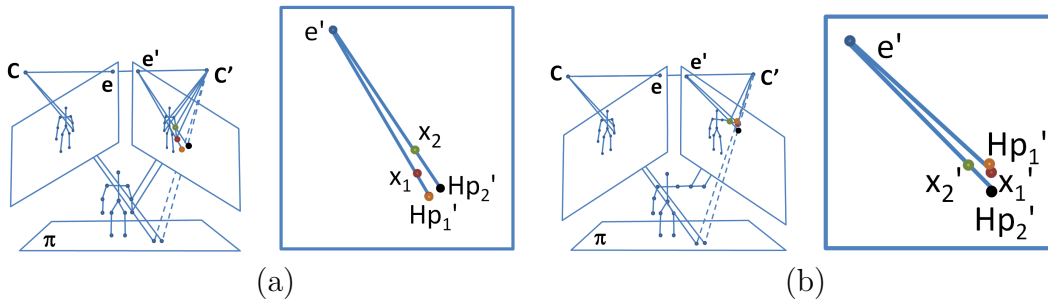


Figure 4.2: These figures explain the significance of the characteristic vector. As soon as the person moves one of his arms, there is notable change in the characteristic vector for the points that moved.

instead of using the projective depth directly, we use the scaled absolute value of the difference of depths as our invariant:

**Definition 2** (Canonical pose)

*We shall call the image points  $\mathbf{p}_{i=1,\dots,k}$  of a set of fixed points in a stationary camera  $\mathbf{P}$ , a canonical pose of the  $k$  points.*

Note that the definition does not impose constraints such as points in general position or non-coplanarity.

**Definition 3** Let  $\mathbf{m}_{i=1,\dots,k}$  be a set of image points in a camera  $\mathbf{P}_1$  that are in one-to-one correspondence with the points in the canonical pose and a homography  $\mathbf{H}_2$  that is consistent with the fundamental matrix  $\mathbf{F}$  between the set of points and the points in the canonical pose. Let also  $\mathbf{m}'_{i=1,\dots,k}$  be the images of these points after moving to new locations and a homography  $\mathbf{H}_2$  that is consistent with the fundamental matrix  $\mathbf{F}$  between the set of moved points and the points in the canonical pose we define the “characteristic vector” of the  $k$  moving points as

$$\mathbf{t} = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix} \quad (4.3)$$

where

$$b_i = \frac{(\mathbf{m}_i - \mathbf{H}_1 \mathbf{p}_i)_x}{(\mathbf{e}' - \mathbf{H}_2 \mathbf{p}_i)_x} - \frac{(\mathbf{m}'_i - \mathbf{H}_2 \mathbf{p}_i)_x}{(\mathbf{e}' - \mathbf{H}_2 \mathbf{p}_i)_x} \quad (4.4)$$

$$= \frac{(\mathbf{m}_i - \mathbf{H}_1 \mathbf{p}_i)_y}{(\mathbf{e}' - \mathbf{H}_1 \mathbf{p}_i)_y} - \frac{(\mathbf{m}'_i - \mathbf{H}_2 \mathbf{p}_i)_y}{(\mathbf{e}' - \mathbf{H}_2 \mathbf{p}_i)_y}, \quad (4.5)$$

$(\cdot)_x$  and  $(\cdot)_y$  denote the  $x$  and  $y$  coordinates of the argument vector, and  $\mathbf{e}'$  is the epipole in the second image.

**Proposition 4** (Invariance of Characteristic Vector)

Assume two sets of freely moving points that are in one-to-one correspondence with the points in the canonical pose are observed by two distinct cameras  $\mathbf{P}_1$  and  $\mathbf{P}_2$ . If the

*motion of the two sets of points differ up to similarity, then the associated characteristic vectors would differ up to scale.*

An important constraint in the definition of the characteristic vector is the consistency of  $\mathbf{H}$  with the fundamental matrix  $\mathbf{F}$ . This was established by Viéville et al. [58] as the condition that  $\mathbf{H}^T \mathbf{F}$  has to be skew symmetric. The latter implies that  $\mathbf{H}^T \mathbf{F} + \mathbf{F}^T \mathbf{H} = 0$ .

## 4.2 Using Projective Depth

A key issue is thus, how can one find a set of 4 or more point correspondences that yield a homography  $\mathbf{H}$  that satisfies this condition. This issue is of practical interest in our problem, because in practice it would be impossible to find corresponding planes between two actions performed by two different subjects at totally different locations viewed by two different cameras.

There can be multiple ways of using Projective Depth including using triplets, ground plane, and planes based on movement. Let us analyze these options:

### 4.2.1 Using Triplets

As described earlier in 2, The 3D body structure of a human can be divided into triplets of body points, each of which determines a plane in the 3D space when the points are not collinear. The problem of comparing articulated motions of human body thus transforms to comparing rigid motions of body planes (triplets). According



to proposition 4, the motion of a plane induces a fundamental matrix, which can be identified by its associated *fundamental ratios*. If two pose transitions are identical, their corresponding body point triplets have the same *fundamental ratios*, which provide a measure for matching two pose transitions.

We can divide the body points into a set of triplets. The 3 points of each triplet along with the epipole define a plane. We can use these planes to calculate the “projective depth” of every other point. To match two poses, it would be necessary to match their projective depths.

Given a body model with 11 body points, we have  $\binom{11}{3} = 165$  triplets and for every triplet, we have  $11 - 3 = 8$  projective depths. The total projective depths would equal the number of triplets times the number of projective depths for each triplet or  $165 \times 8 = 1320$  in our case. This is a lot of data to work with and would ensure noise is filtered out.

**Degenerate triplets:** A homography cannot be computed from four correspondences if three points are collinear. Even when three image points are close to collinear the problem becomes ill-conditioned. We call such triplets as degenerate, and simply ignore them in matching pose transitions. This does not produce any difficulty in practice, since with 11 body point representation used in our experiments, we obtain 165 possible triplets, the vast majority of which are in practice non-degenerate.

A special case is when the epipole is close to or at infinity, for which all triplets would degenerate. We solve this problem by transforming the image points in projective space in a manner similar to Zhang et al. [70]. The idea is to find a pair of projective

transformations  $\mathbf{Q}$  and  $\mathbf{Q}'$ , such that after transformation the epipoles and transformed image points are not at infinity. Note that these transformations do not affect the projective equality in Proposition 4.

#### 4.2.2 Ground Plane

We can use the ground plane to estimate the depth of each body point. Then we would have exactly 11 projective depths corresponding to a 11 point body model per frame. With the exception of the foot points, the ground plane is always relatively far away from the body points and hence, we can be sure that the projective depths are large enough to be meaningful. The problem of action recognition would then translate to matching curves, as we would have 11 curves corresponding to each action. However, in this case, we have a much smaller set to work with (Only 11 projective depths per frame).

##### 4.2.2.1 Estimating ground plane homography

Let  $\mathbf{m}_1$  and  $\mathbf{m}_2$  be two arbitrary points in a camera  $\mathbf{P}_1$  and in correspondence with  $\mathbf{p}_1$  and  $\mathbf{p}_2$  in the canonical pose. Let also  $\mathbf{m}_3$  be any arbitrary point in  $\mathbf{P}_1$ . Then the corresponding point  $\mathbf{p}_3$  in the canonical camera must satisfy the epipolar constraint  $\mathbf{p}_3^T \mathbf{F} \mathbf{m}_3 = 0$ . This provides a one parameter family of solutions in the form of  $\mathbf{p}_3(\alpha)$  for  $\mathbf{p}_3$ . Taking the epipoles as the fourth corresponding points, defines a one-parameter family of homographies  $\mathbf{H}(\alpha)$  that map the four points  $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$  and  $\mathbf{e}$  to  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$  and  $\mathbf{e}'$ . The optimal parameter  $\alpha$  that would impose the consistency condition is then

found using

$$\begin{aligned} \alpha^* &\sim \arg \min \text{trace}(\mathbf{H}(\alpha)^T \mathbf{F} \mathbf{F}^T \mathbf{H}(\alpha)) \\ \text{subject to } &\|\mathbf{H}^T(\alpha^*) \mathbf{F} + \mathbf{F}^T \mathbf{H}^T(\alpha^*)\| \text{ is minimized} \end{aligned} \quad (4.6)$$

This is a constraint minimization of a polynomial cost function, for which there is a closed form solution.

#### 4.2.2.2 Action Alignment

For a given test sequence, we first calculate the characteristic vectors with respect to a canonical pose of a human subject over all frames in the sequence. This basically yields a time series of characteristic vectors. The canonical pose may be for instance a person simply standing right up. If we regard the characteristic vector  $\mathbf{t}$  as a random, scaled vector, then given a set  $\mathcal{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_M\}$  of  $M$  different series of reference characteristic vectors corresponding to  $M$  different actions, our goal is to find  $\mathbf{r}_m$  that best matches the test sequence. For the time being assume that the test sequence and all reference sequences are of the same length of  $K$  and are aligned. Assuming a normal distribution of noise and errors with variance  $\sigma^2$ , the probability of the characteristic vector of an unknown action  $\mathbf{t}$  to match  $\mathbf{r}_m$  is given by:

$$p(\bar{\mathbf{t}}|\bar{\mathbf{r}}_m) \sim \exp\left(-\frac{\|\bar{\mathbf{t}} - \bar{\mathbf{r}}_m\|^2}{2\sigma^2}\right) \quad (4.7)$$

where  $\bar{\mathbf{t}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}$  and  $\bar{\mathbf{r}}_m = \frac{\mathbf{r}_m}{\|\mathbf{r}_m\|}$ .

Assuming conditional independency over time, we can solve the problem by minimizing the following log-likelihood function:

$$m^* \sim \arg \min_{m=1,\dots,M} \sum_{K=1,\dots,K} \|\bar{\mathbf{t}}^k - \bar{\mathbf{r}}_m^k\|^2 \quad (4.8)$$

where  $m^*$  is the estimated optimal index for the matched sequence in the database. In practice one may attempt to improve upon this formulation by constraining the fact that the motion of a given point in time must be smooth. However, as seen in the experimental section this maximum likelihood solution is sufficient for providing good results.

#### 4.2.2.3 Degeneracy

We consider a component with a value of zero as a degenerate case. In fact, a value of zero would indicate that the point moves inside a plane parallel to the plane of eigenvectors of the matrix  $\mathbf{H}$ , or is motionless. Although, this may happen in practice, it is highly unlikely that in an action all points remain motionless or all have a coplanar motion parallel to the plane of eigenvectors of  $\mathbf{H}$ .

#### 4.2.3 Planes in time

Another option is to use the planes in time. As the person moves in time, we have more points to use. However, this leads to a an extreme amount of data since we are effectively choosing 3 points from the number of body points times the number of frames. Assuming the number of body points is 11 and the length of the video is 60, then this amounts to a total of  $\binom{60 \times 11}{3} = 47698420$ , which is huge and is not practical to work with. For this reason, we did not pursue this course of research.

#### 4.2.4 Using Mirror Symmetry

This work builds on the work of [1], which analyzed the idea of 3D reconstruction from a single perspective view of a mirror symmetric scene. The work demonstrated that the mirror view is equivalent to the observing the same scene with two cameras. Let's first quickly review their work since we are going to build on that. In particular, we would be looking into Lemma 1:

**Lemma 1** *The image of a scene that is symmetric with respect to an unknown plane, formed by an arbitrary projective camera, is identical to the image of the scene formed by the (virtual) projective camera symmetric of the first one with respect to the scene's 3-D (unknown) symmetry plane.*

Assume we have an image of a symmetric shape. We can place the origin  $\mathbf{O}$  of the world on the symmetry plane. Let  $\mathbf{X}$  denote a world point represented by the vector  $[x \ y \ z \ 1]^T$  and let  $\mathbf{x}$  denote the corresponding homogenous 3-vector  $[U \ V \ W]$ . Let the camera be defined by the  $3 \times 4$  matrix  $\mathbf{P} = \mathbf{M}[I \mid -\tilde{\mathbf{C}}]$ , where  $\mathbf{M} = \mathbf{K}\mathbf{R}$  where  $\mathbf{K}$  is the  $3 \times 3$  calibration matrix, and  $\mathbf{R}$  is the  $3 \times 3$  rotation matrix from the world coordinate system to the camera coordinate system; and  $\tilde{\mathbf{C}}$  is the inhomogenous  $3 \times 1$  vector of the camera center coordinates in the world coordinate system. A world point  $\mathbf{X}$  is mapped to the  $\mathbf{x}$  by this relation:

$$\mathbf{x} = \mathbf{P}\mathbf{X} = \mathbf{M}[I \mid -\tilde{\mathbf{C}}]\mathbf{X} \quad (4.9)$$

The world point symmetric to  $\mathbf{X}$  with respect to the symmetry plane is  $\bar{\mathbf{X}} = \mathbf{Z}\mathbf{X}$ ,

where:

$$\mathbf{Z} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.10)$$

and we note:

$$\tilde{\mathbf{Z}} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

The image point  $\bar{\mathbf{x}}$  of the world point  $\bar{\mathbf{X}}$  seen by the camera at center  $\mathbf{C}$  is:

$$\bar{\mathbf{x}} = \mathbf{M}[I| - \tilde{\mathbf{C}}]\mathbf{Z}\mathbf{X} \quad (4.12)$$

Now consider a virtual camera, which is symmetric to camera at center  $\mathbf{C}$  with respect to the object's symmetric plane. Hence its center would be  $\bar{\mathbf{C}} = \mathbf{Z}\mathbf{C}$ , and it would project a world point  $\mathbf{X}$  according to this relation:

$$\mathbf{x}' = \bar{\mathbf{M}}[I| - \tilde{\bar{\mathbf{C}}}]\mathbf{X} \quad (4.13)$$

where  $\bar{\mathbf{M}} = \mathbf{M}\tilde{\mathbf{Z}}$ . Substituting symmetric elements by their expression:

$$\mathbf{x}' = \mathbf{M}\tilde{\mathbf{Z}}[I| - \tilde{\bar{\mathbf{C}}}]\mathbf{X} = \mathbf{M}[I| - \tilde{\mathbf{C}}]\mathbf{Z}\mathbf{X} = \bar{\mathbf{x}} \quad (4.14)$$

Similarly:

$$\bar{\mathbf{x}}' = \mathbf{M}\tilde{\mathbf{Z}}[I| - \tilde{\mathbf{C}}]\mathbf{Z}\mathbf{X} = \mathbf{M}[I| - \tilde{\mathbf{C}}]\mathbf{Z}\mathbf{Z}\mathbf{X} = \mathbf{x} \quad (4.15)$$

This means that the image of a pair of symmetric points viewed by a real camera is equivalent to the case of a virtual camera viewing the same symmetric points being reversed in the real and virtual view.

#### 4.2.4.1 *Using Mirror-view symmetry in Pose Recognition*

Our goal can be stated as follows: Given a 3D pose viewed by two cameras  $C_1$  and  $C_2$ , we want to extract planes from the scene to estimate the projective depths of body points relative to the plane. This information can then be used for pose-recognition and extended to action recognition.

Applying mirror view symmetry would relate  $C_1$  and its mirror view, and  $C_2$  and its mirror view only. Furthermore, this would assume that the action is symmetric, which is not the case with most of the actions.

But recall that we already know the epipolar geometry between  $C_1$  and  $C_2$  and therefore, we can use this information. Furthermore, we are interested in a mirror view of the person, which can be used to extract co-planar points.

Let us refer to an example to illustrate this concept: Consider the case of an asymmetric pose and the hand points  $\mathbf{X}_{lefthand}$  and  $\mathbf{X}_{righthand}$  are viewed by the two cameras  $C_1$  and  $C_2$  and the corresponding image points are:  $\mathbf{x}_{lefthand}$ , and  $\mathbf{x}_{righthand}$  and  $\mathbf{x}'_{lefthand}$ , and  $\mathbf{x}'_{righthand}$ , respectively. Let us first consider camera  $C_1$ : If the pose is symmetric, in the mirror view,  $\bar{\mathbf{x}}_{lefthand} = \mathbf{x}_{righthand}$  and  $\bar{\mathbf{x}}_{righthand} = \mathbf{x}_{lefthand}$ . However, if the pose is not symmetric, this would not be true. But we can think of the virtual body point that would have been there had it been a symmetric pose:  $\bar{\mathbf{x}}_{lefthand} = \mathbf{Z}'_1 \mathbf{x}_{righthand}$ ,

where

$$\mathbf{Z}'_1 = \begin{bmatrix} -1 & 0 & t_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

where  $t_1$  corresponds to some translation. Similarly,  $\bar{\mathbf{x}}_{righthand} = \mathbf{Z}'_1 \mathbf{x}_{lefthand}$ . We can think of camera  $C_2$ , where  $\bar{\mathbf{x}}'_{lefthand} = \mathbf{Z}'_2 \mathbf{x}'_{righthand}$  and  $\bar{\mathbf{x}}'_{righthand} = \mathbf{Z}'_2 \mathbf{x}'_{lefthand}$ , where

$$\mathbf{Z}'_2 = \begin{bmatrix} -1 & 0 & t_2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.17)$$

So we have two unknowns  $t_1$  and  $t_2$ , which are the unknown translations. Now consider another 3 point, let's say the left shoulder point,  $\mathbf{X}_{leftshoulder}$ , which is viewed in camera  $C_1$  and  $C_2$  as  $\mathbf{x}_{leftshoulder}$  and  $\mathbf{x}'_{leftshoulder}$ , respectively.

The virtual mirror symmetric points would be  $\bar{\mathbf{x}}_{leftshoulder} = \mathbf{Z}'_1 \mathbf{x}_{leftshoulder}$  and  $\bar{\mathbf{x}}'_{leftshoulder} = \mathbf{Z}'_2 \mathbf{x}'_{leftshoulder}$ . Both  $\mathbf{x}_{leftshoulder}$  and  $\bar{\mathbf{x}}_{leftshoulder}$  would have the same 'depth' relative to the plane defined by the points,  $\mathbf{x}_{lefthand}$ ,  $\mathbf{x}_{righthand}$ ,  $\bar{\mathbf{x}}_{lefthand}$ , and  $\bar{\mathbf{x}}_{righthand}$  in camera  $C_1$  and  $\mathbf{x}'_{lefthand}$ ,  $\mathbf{x}'_{righthand}$ ,  $\bar{\mathbf{x}}'_{lefthand}$ , and  $\bar{\mathbf{x}}'_{righthand}$  in camera  $C_2$  (Refer to [Figure 4.3](#)).

Let  $\mathbf{H}$  be the homography relating the points,  $\mathbf{x}_{lefthand}$ ,  $\mathbf{x}_{righthand}$ ,  $\bar{\mathbf{x}}_{lefthand}$ , and  $\bar{\mathbf{x}}_{righthand}$  in camera  $C_1$  and  $\mathbf{x}'_{lefthand}$ ,  $\mathbf{x}'_{righthand}$ ,  $\bar{\mathbf{x}}'_{lefthand}$ , and  $\bar{\mathbf{x}}'_{righthand}$  in camera  $C_2$ , we have:



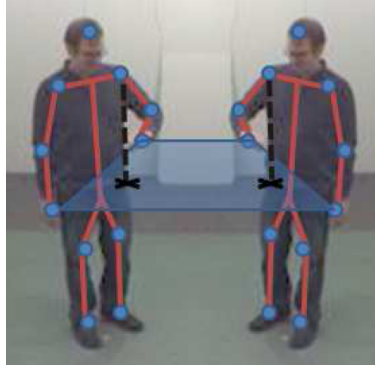


Figure 4.3: The depth of left shoulder and its mirror view would be equidistant from the plane consisting of left hand, and right hand, and their mirror views.

$$\begin{aligned}\rho_1 &= \frac{\mathbf{x}'_{leftshoulder} - \mathbf{H}\mathbf{x}_{leftshoulder}}{\mathbf{x}'_{leftshoulder} - \mathbf{e}'} = \\ \frac{\bar{\mathbf{x}}'_{leftshoulder} - \mathbf{H}\bar{\mathbf{x}}_{leftshoulder}}{\bar{\mathbf{x}}'_{leftshoulder} - \mathbf{e}'} &= \frac{\mathbf{Z}_2\mathbf{x}'_{leftshoulder} - \mathbf{H}\mathbf{Z}_1\mathbf{x}_{leftshoulder}}{\mathbf{Z}_2\mathbf{x}'_{leftshoulder} - \mathbf{e}'}\end{aligned}\quad (4.18)$$

Similarly, in the other direction we have:

$$\begin{aligned}\rho_2 &= \frac{\mathbf{x}_{leftshoulder} - \mathbf{H}^{-1}\mathbf{x}'_{leftshoulder}}{\mathbf{x}_{leftshoulder} - \mathbf{e}} = \\ \frac{\bar{\mathbf{x}}_{leftshoulder} - \mathbf{H}^{-1}\bar{\mathbf{x}}'_{leftshoulder}}{\bar{\mathbf{x}}_{leftshoulder} - \mathbf{e}} &= \frac{\mathbf{Z}_1\mathbf{x}_{leftshoulder} - \mathbf{H}^{-1}\mathbf{Z}_2\mathbf{x}'_{leftshoulder}}{\mathbf{Z}_1\mathbf{x}_{leftshoulder} - \mathbf{e}}\end{aligned}\quad (4.19)$$

So we have two equations to solve for the two unknowns,  $t_1$  and  $t_2$ . Solving these equations, we get  $t_1 = -2e_x$  and  $t_2 = -2e'_x$ .

### 4.3 Action Recognition Using Projective Depth

For action recognition, instead of estimating the depths and aligning the two sequences each time we need to test a new motion sequence, we store the depths in a volume. Thus, when we are using ground plane, we have a 3D volume of  $4 \times \text{number of body points} \times \text{number of frames} \times \text{number of frames}$ . Here the first  $4 \times \text{number of body points}$  is the

characteristic vector (we use both x and y coordinates and the characteristic vector in both dimensions), and the characteristic vector is calculated for every frame in the sequence. Similarly, when we use triplets, the volume has dimensions of  $4 \times \text{number of body points} \times \text{number of frames} \times \binom{\text{number of body points}}{3} \times \text{number of frames}$ . This is because in each frame, we get a set of  $\binom{\text{number of body points}}{3}$  planes, and we calculate the characteristic vector for each of these frames for the entire sequence. Similarly, using mirror symmetry, we have a  $4 \times \text{number of body points} \times \text{number of frames} \times \binom{\text{number of body points}}{2} \times \text{number of frames}$  dimensional volume.

The idea is that this volume is characteristic of the action. Our objective is to approximate this volume into compact vectors for use in action recognition. We use rank-1 decomposition described in [55] to generate compact representations of the volume, which is then used for action recognition. Given two motion sequences  $m_i$  and  $m_j$ , we can obtain the corresponding discriminant vectors,  $\mathbf{v}_i = \{\mathbf{D}_T^i, \mathbf{D}_F^i, \mathbf{D}_R^i\}$  and  $\mathbf{v}_j = \{\mathbf{D}_T^j, \mathbf{D}_F^j, \mathbf{D}_R^j\}$ . Then the similarity of the two motion sequences can be calculated using  $\|\mathbf{v}_i - \mathbf{v}_j\|$ .

#### 4.3.1 Experimental Results and Discussion

In this section we present results on both CMU MoCap data and IXMAS dataset [65].

Table 4.1: Using ground plane: Overall accuracy about 95%.

| Ground-truth | Recognized as |      |            |     |       |
|--------------|---------------|------|------------|-----|-------|
|              | Walk          | Jump | Golf Swing | Run | Climb |
| Walk         | 49            |      |            |     | 1     |
| Jump         | 1             | 49   |            |     |       |
| Golf Swing   |               |      | 49         |     | 1     |
| Run          | 5             |      |            | 42  | 3     |
| Climb        | 2             |      |            |     | 48    |

Table 4.2: Using triplets: Overall accuracy about 90%

| Ground-truth | Recognized as |      |            |     |       |
|--------------|---------------|------|------------|-----|-------|
|              | Walk          | Jump | Golf Swing | Run | Climb |
| Walk         | 44            |      | 1          | 3   | 2     |
| Jump         | 2             | 45   |            | 1   | 2     |
| Golf Swing   |               | 1    | 47         | 2   |       |
| Run          | 2             | 2    | 1          | 44  | 1     |
| Climb        | 3             | 0    | 1          | 2   | 44    |

Table 4.3: Using mirror symmetric planes: Overall accuracy about 96%

| Ground-truth | Recognized as |      |            |     |       |
|--------------|---------------|------|------------|-----|-------|
|              | Walk          | Jump | Golf Swing | Run | Climb |
| Walk         | 49            |      |            |     |       |
| Jump         | 1             | 49   |            |     |       |
| Golf Swing   |               |      | 49         |     | 1     |
| Run          | 2             |      |            | 45  | 3     |
| Climb        | 3             |      |            |     | 47    |

#### 4.3.1.1 Results on MoCap Data

To test Action Recognition, we used the same setup as 3.3.1.3. The first frame was chosen as the canonical pose (our method is not sensitive to the choice of the canonical pose, and using a different pose does not make a difference). We used leave one out cross validation. The classification results are shown in Table 4.1, Table 4.2, and Table 4.6. The overall classification accuracy for our method is 95%, 90%, and 96%, using ground plane, triplets, and mirror symmetry, respectively. The results are remarkably good despite the extreme viewpoint changes and variations in camera intrinsic parameters.

#### 4.3.1.2 Results on Real Data

We evaluated our method on IXMAS data set [65]. We tested our method on 10 actions consisting of “watch time,” “cross arms,” “scratch head,” “sit down,” “stand up,” “wave,” “punch,” “kick,” “point,” and “pick up.” This would correspond to testing on  $10 \times 3 \times 5 \times 10 = 1500$  different videos. We used leave one out cross validation to test our results.

In our experiments, we chose the first frame of “watch time” in “cam0” view of “Amel” as our canonical pose (our method is not sensitive to the choice of the canonical pose, therefore using a different pose of a different person from another viewpoint does not make a difference). The results are shown in Table 4.4, Table 4.5, and Table 4.6. The overall recognition rates are 81.4%, 87.3%, and 90.5% using ground plane, triplets, and mirror symmetry, respectively.

Table 4.4: Recognition rate for IXMAS data using ground plane. Overall accuracy: 81.4%

|            |             |            |              |          |          |
|------------|-------------|------------|--------------|----------|----------|
| Action     | Check Watch | Cross Arms | Scratch Head | Sit down | Stand up |
| Accuracy % | 77.8        | 84.8       | 88.6         | 88.6     | 77.8     |
| Action     | Wave        | Punch      | Kick         | Point    | Pick up  |
| Accuracy % | 67.8        | 77.8       | 88.6         | 77.8     | 84.8     |

Table 4.5: Recognition rate for IXMAS data using triplets. Overall accuracy: 87.3%

|            |             |            |              |          |          |
|------------|-------------|------------|--------------|----------|----------|
| Action     | Check Watch | Cross Arms | Scratch Head | Sit down | Stand up |
| Accuracy % | 80.4        | 87.0       | 89.1         | 87.0     | 95.7     |
| Action     | Wave        | Punch      | Kick         | Point    | Pick up  |
| Accuracy % | 80.0        | 95.5       | 95.5         | 84.1     | 79.5     |

Table 4.6: Recognition rate for IXMAS data using mirror symmetric planes. Overall accuracy: 90.5%

|            |             |            |              |          |          |
|------------|-------------|------------|--------------|----------|----------|
| Action     | Check Watch | Cross Arms | Scratch Head | Sit down | Stand up |
| Accuracy % | 84.8        | 91.3       | 91.3         | 91.3     | 100      |
| Action     | Wave        | Punch      | Kick         | Point    | Pick up  |
| Accuracy % | 77.8        | 95.5       | 100          | 84.1     | 88.6     |

Using mirror symmetry outperforms ground plane and triplets. Ground plane can be thought of as the subset of mirror symmetry since the two feet points and their mirror views give us the roughly the ground plane (unless in the first frame, the feet are not on the ground but even then that plane can be thought of as the ground plane). Hence, the lower recognition rate using ground plane with respect to using mirror symmetry is understandable. For the triplets, the accuracy seems to be low compared to mirror symmetry because the planes extracted from triplets are always very close to the body points. In fact, for useful information to be extracted, it is essential that some of the

body points move really far away from the body. Otherwise, all the projective depths map to zero. Mirror symmetry has none of these issues, and therefore, it is not surprising that mirror symmetry gives the best performance.

## CHAPTER 5: CONCLUSION AND FUTURE WORK

In this dissertation we study geometric invariants in human motion and their application to view-invariant action recognition. Geometric invariants are important in computer vision because with perspective projection, it is very hard to relate objects across different views. Therefore, it is very useful if we can find geometric properties of objects, which are invariant to the intrinsic parameters of the camera and viewpoint changes. In this dissertation, we study three different geometric invariants for pose recognition, which can be extended to action recognition.

To study poses, we propose decomposing the body points into a set of triplets or line segments. This has several advantages: (i) The matching of non-rigid motion of human body points is transformed to matching the rigid motion of body point triplets or line segments; (ii) We get an highly over-determined formulation of the problem as with  $N$  body points, we have  $\binom{N}{3}$  triplets and  $\binom{N}{2}$  line segments. This allows us to achieve robustness to noise and occlusion; and (iii) Anthropometric restrictions, such as coplanarity of some body points, can be relaxed.

The first geometric invariant we propose is the Rank 4 constraint. We exploit the fact that the family of homography matrices span a 4 dimensional linear subspace of  $P^8$  and hence can be used to identify similar poses. If the poses match, then the rank of the family of homography matrices stacked as column vectors would be 4. If, however, the

poses are not similar, then the rank would be higher than 4. This observation is used to measure the similarity of two poses and is extended to action recognition using dynamic programming.

Secondly, we extend the fundamental ratios invariant. The fundamental ratios invariant is motivated by the observation that if camera calibration matrix has zero skew and unit aspect ratio, the upper left  $2 \times 2$  sub-matrix is solely dependent on the rotation and translation of the cameras, and is independent of camera internal parameters. The ratios among the elements in the upper left  $2 \times 2$  sub-matrix are referred to as the Fundamental Ratios. This was used to measure the similarity of two pose transition by estimating the fundamental matrix induced by a moving triplet of body points. In this dissertation, we ask whether it is possible to obtain the fundamental matrix induced by a moving line segment. This introduces more redundancy and is experimentally shown to perform better than point triplets.

We also present a weighting strategy to further improve our results. This is motivated by the fact that not all line segments play the same role in determining the correct action. For instance, the upper body plays a more critical role in boxing, whereas the lower body has more significance in cycling. Therefore, we want to be able to assign different weights to line segments for various actions to improve the accuracy. We present our weighting strategy and present experimental results, which demonstrate that using weighting considerably improves the overall recognition rate. This is a general scheme that can be applied to other methods as well such as Rank 4 constraint.



We also introduced the projective depth invariant which uses the projective depth relative to planes for finding the similarity between two poses. The challenge is finding planes in the scene. We propose three different strategies for extracting planes between two frames: (i) Ground Plane: We assume that the two feet points are on the ground plane and we use the fact that the homography matrix corresponding to the plane must be consistent with the epipolar geometry to estimate the ground plane homography. (ii) Triplets: We use body point triplet planes ; and (iii) Mirror Person: We present a novel method of using the mirror view of a person so that any line segment and its mirror counterpart can be used as a plane. Using the ground plane can be roughly thought of a subset of using the mirror person because the two feet points and their mirror counterparts are very closely related to the ground plane. The difference between using triplets and mirror person is easier to analyze when we consider their counterparts for 3D points. The triplets correspond to the plane formed by the triplet while the mirror person is equivalent to taking the mirror view of the person and using each line segment and its mirror view. The difference essentially lies in the planes extracted.

We present extensive experimental results, which show that our method can accurately identify human poses from video sequences when they are observed from totally different viewpoints with different camera parameters. We used semi-synthetic data to test view invariance and noise resilience. We present results on action recognition on 4 different datasets including CMU MoCap dataset, IXMAS dataset, UCF-CIL dataset, and Kinect dataset. For fundamental ratios using line segments, extensive experiments are reported on testing (i) view-invariance, (ii) robustness to noisy localization of body

points, (iii) effect of assigning different weights to different body points, (iv) effect of partial occlusion on recognition accuracy, and (v) determining how soon our method recognizes an action correctly from the starting point of the query video.

## 5.1 Computational Complexity

Let us analyze the number of computations for each of our methods:

If the number of body points are  $N$  for a given frame, and the total number of frames for two sequences are  $f_1$  and  $f_2$ .

### 5.1.1 Rank-4 constraint

Since dynamic programming is used to align the two sequences, we need to populate the  $f_1 \times f_2$  accumulated DP path matrix. To do this, all the homography matrices between the two views have to be calculated for each entry in the matrix. There are a total of at most  $\binom{N}{3}$  triplets. Therefore, the total number of calculations would be  $f_1 f_2 (c_1 + \binom{N}{3} c_2)$ , where  $c_1$  is a constant time for calculating the rank of the matrix consisting of all the homography matrices stacked as column vectors and calculating the error, and  $c_2$  is a constant time used for calculating the homography. This gives us  $\mathcal{O}(f_1 f_2 (c_1 + \binom{N}{3} c_2))$ .

A typical sequence is around 50 frames long and we use 11 body points in our experiments, therefore, the number of floating point operations can be estimated to be around  $50 \times 50 \times \binom{11}{3} = 412500$ .

### 5.1.2 Fundamental Ratios constraint

Again, dynamic programming is used to align the two sequence. The total number of calculations would be  $f_1 f_2 (c_1 + \binom{N}{3} c_2)$ , where  $c_1$  is constant time for calculating the error between all line segments and if weighting is used, applying the weights on the individual line segments, and  $c_2$  is constant time for calculating the induced fundamental matrix using the line segment. This gives us  $\mathcal{O}(f_1 f_2 \binom{N}{3})$  for this method.

Typical number of floating point operations for this method are  $50 \times 50 \times \binom{11}{2} = 137500$ .

### 5.1.3 Projective Depth Invariant

In projective depth, all the depths are stored in a volume. Let's analyze the different computations needed to obtain this volume:

#### 5.1.3.1 Using Ground Plane

Total number of calculations =  $4N f_1 f_1$ , which gives us  $\mathcal{O}(N f_1 f_2)$ . The typical number of floating point operations is  $4 \times 11 \times 50 \times 50 = 110000$ .

#### 5.1.3.2 Using Triplets

Total number of calculations =  $4N f_1 f_1 \binom{N}{3}$ , which gives us  $\mathcal{O}(N f_1 f_2 \binom{N}{3})$ . Typical number of floating point operations =  $4 \times 11 \times 50 \times 50 \times \binom{11}{3} = 18150000$ .

#### 5.1.3.3 Using Mirror Person

Total number of calculations =  $4N f_1 f_1 \binom{N}{2}$ , which gives us  $\mathcal{O}(N f_1 f_2 \binom{N}{3})$ .

This is because both the x and y-coordinate of the projective depth in both directions are used as the feature (which gives us 4) and the projective depth is calculated for every point in the body for every frame. Every frame contributes a plane for ground depth; for triplets, each frame contributes  $\binom{11}{3}$  planes; and each frame gives  $\binom{11}{2}$  planes when using mirror person. Typical number of floating point operations =  $4 \times 11 \times 50 \times 50 \times \binom{11}{2} = 6050000$ .

Please note that since estimating the depth volume is only dependent on the original sequence, we can find the depth volume for the database of actions and decompose them offline to find the characteristic vectors and store them. For a query action, we need to find its depth volume and decompose it, which gives us its characteristic vectors, and compare these to vectors from the database of actions.

## 5.2 Significance of this work

One may ask the significance of this work given that many other methods are able to give comparable or better results. We are tackling a very hard problem and there are two facets of this problem: (i) View invariance: We assume that the test action and the examples in the dataset may be from very different viewpoints; and (ii) the number of examples are very limited. Both of these factors are very important in that usually the methods in the literature, which address view-invariance, assume they have a huge set of actions from different view points so that they are able to train a classifier, which is able to give good results for different view points. But what would happen if these methods had a handful of videos? We on the other hand, are tackling a much harder

problem, which is what if only a few instances of the action were given. And therefore, in our experiments on Rank 4 and fundamental ratios, we use just one instance of each action as our database. For Rank 4 constraint, on the moCap dataset, when we test on 4 actions, we have just 4 videos in the dataset, one for each action. Similarly, when we test for 5 actions in IXMAS dataset, there are only 5 videos in our dataset, one for each action. For fundamental ratios using line segments, if we are testing on  $N$  actions, we only need  $3N$  examples in the dataset.

The end goal is that a user or animator may be able to define a new action simply by capturing a single video. In this context, our findings are that geometric invariants are indispensable in that they provide us with geometric properties of the object, which are invariant to different viewpoints and intrinsic parameters of the camera.

The idea of estimating projective depth, stacking them in a volume and decomposing them is directed more towards motion retrieval. Large motion capture datasets have become commonplace owing to their importance in realistic animation of human motion. With this development, it has become increasingly important to develop methods for an animator to search for similar motions from a given dataset. Since, the depth vector is only dependent on the sequence, we can find the volumes for the database of actions and decompose them offline and store the characteristic vectors. Hence, given a query action, we need to decompose it and compare its characteristic vector with a set of other vectors, which is very fast.

### 5.3 Future Work

Currently these methods are not real-time and the aim should be to make these methods real time for use in human computer interface (HCI) applications. Each of the methods lends itself to a parallel implementation very well. For rank-4 constraints, each homography can be calculated in parallel since they are independent of each other. Similarly, when using fundamental ratios, each fundamental ratio induced by different line segments can be estimated separately. This is true of projective depth as well. Therefore, these methods can be implemented very well using parallel programming for GPUs and this is an open problem to be explored.

Furthermore, each of the methods can potentially use less number of frames. In fact, we demonstrated this concept for fundamental ratios. The same ideas can be applied to projective depth as well. Naturally, our depth volume would be smaller if less frames are used. Reducing the latency needs to be explored for rank 4 constraint and projective depth.

It would also be interesting to apply weighting on rank 4 constraint, fundamental ratios using triplets, and projective depth. It would also be interesting to see whether the weights are substantially different depending on which constraint was used. This is an open problem, which needs to be explored in more detail.

One of the things we did not pursue in more detail is trying to make the action symmetric and this would be interesting to pursue in more detail to determine if the planes yielded by this approach are the same planes as the mirror person planes.

Another promising extension could be using the different projective depth approaches in unison. For a given frame, we have the ground planes, triplet planes, and mirror person planes, and we selected only one of these to calculate the projective depths. It would be interesting to use these different strategies together. This is motivated by the fact that one technique performs better on one set of actions, while another performs better on others. Therefore, it could be very useful if we could work out a method for using these together.

It would also be interesting to see how these methods can differentiate between different styles in a given category. For instance, different tennis players have variations in their shots and the question is whether these methods can detect these.

## LIST OF REFERENCES

- [1] Francois A., Medioni G., and Waupotitsch R. Mirror symmetry => 2-view stereo geometry. *IJCVU*, 21(2):137–143, 2003.
- [2] M. Ahmad and S.W. Lee. HMM-based Human Action Recognition Using Multiview Image Sequences. *ICPR*, 1:263–266, 2006.
- [3] Mohiuddin Ahmad and Seong-Whan Lee. Human action recognition using shape and clg-motion flow from multi-view image sequences. *Pattern Recogn.*, 41:2237–2252, July 2008.
- [4] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1 –8, oct. 2007.
- [5] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(2):288 –303, feb. 2010.
- [6] K. Arbter, WE Snyder, H. Burkhardt, and G. Hirzinger. Application of affine-invariant Fourier descriptors to recognition of 3-D objects. *IEEE Trans. PAMI*, 12(7):640–647, 1990.
- [7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. ICCV*, volume 2, pages 1395– 1402, 2005.
- [8] AF Bobick and JW Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.
- [9] Robert C. Bolles and H. Harlyn Baker. Epipolar-plane image analysis: a technique for analyzing motion sequences. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 26–36, 1987.
- [10] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, and A. Pentland. Invariant features for 3-d gesture recognition. *FG*, 0:157–162, 1996.
- [11] R.J. Campbell and P.J. Flynn. A Survey Of Free-Form Object Representation and Recognition Techniques. *COMPUTER VISION AND IMAGE UNDERSTANDING*, 81(2):166–210, 2001.
- [12] Carlo Colombo, Alberto Del Bimbo, and Federico Pernici. Metric 3d reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. *IEEE Trans. PAMI*, 27(1):99–114, 2005.



- [13] AA Efros, AC Berg, G. Mori, and J. Malik. Recognizing action at a distance. *ICCV*, pages 726–733, 2003.
- [14] Ali Farhadi and Mostafa Kamali Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV (1)’08*, pages 154–166, 2008.
- [15] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, june 2008.
- [16] X Feng and P. Perona. Human action recognition by sequence of movelet codewords. In *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 717–721, 2002.
- [17] Andrew W. Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3d model construction for turn-table sequences. In *SMILE*, pages 155–170, 1998.
- [18] DM Gavrilu. Visual analysis of human movement: A survey. *CVIU*, 73(1):82–98, 1999.
- [19] A. Gritai, Y. Sheikh, and M. Shah. On the use of anthropometry in the invariant analysis of human actions. *ICPR*, 2:923–926, 2004.
- [20] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. PAMI*, 19(6):580–593, 1997.
- [21] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [22] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [23] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, oct. 2007.
- [24] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 14:201–211, 1973.
- [25] I. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE Trans. PAMI*, 99(Preliminary), 2010.
- [26] R. Kondepudy and G. Healey. Use of invariants for recognition of three-dimensional color textures. *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, 11(11):3037–3049, 1994.
- [27] I. Laptev, S.J. Belongie, P. Perez, J. Wills, C. universitaire de Beaulieu, and UC San Diego. Periodic Motion Detection and Segmentation via Approximate Sequence Alignment. *ICCV*, 1:816–823, 2005.

- [28] I. Laptev and P. Perez. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, oct. 2007.
- [29] Ivan Laptev, Barbara Caputo, and Tony Lindeberg. Local velocityadapted motion events for spatio-temporal recognition. *CVIU*, pages 207–229, 2007.
- [30] G. Lei. Recognition of planar objects in 3-D space from single perspectiveviews using cross ratio. *Robotics and Automation, IEEE Transactions on*, 6(4):432–437, 1990.
- [31] Jingen Liu and Mubarak Shah. Learning human actions via information maximiza-tion. In *CVPR’08*, pages –1–1, 2008.
- [32] Quan-Tuan Luong and Olivier D. Faugeras. The fundamental matrix: theory, algo-rithms, and stability analysis. *IJCV*, 17, 1996.
- [33] Fengjun Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, pages 1–8, june 2007.
- [34] S.Z. Masood, C. Ellis, A. Nagaraja, M.F. Tappen, J.J. LaViola Jr., and R. Suk-thankar. Measuring and reducing observational latency when recognizing actions. In *The 6th IEEE Workshop on Human Computer Interaction: Real-Time Vision Aspects of Natural User Interfaces (HCI2011), ICCV Workshops*, 2011.
- [35] Q. Minh and W. Wageeh. Wavelet-Based Affine Invariant Representation: A Tool for Recognizing Planar Objects in 3D Space. *IEEE Trans. PAMI*, pages 846–857, 1997.
- [36] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231–268, 2001.
- [37] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.
- [38] T. Ogata, W. Christmas, J. Kittler, and S. Ishikawa. Improving human activity detection by combining multi-dimensional motion descriptors with boosting. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 295–298, 0-0 2006.
- [39] V. Parameswaran and R. Chellappa. View invariants for human action recognition. *CVPR*, 2:613–619, 2003.
- [40] Vasu Parameswaran and Rama Chellappa. View invariance for human action recog-nition. *IJCV*, 66(1):83–101, 2006.
- [41] D. Ramanan, D.A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):65–81, jan. 2007.

- [42] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *IJCV*, 50(2):203–226, 2002.
- [43] A. Rav-Acha, Y. Pritch, D. Lischinski, and S. Peleg. Dynamosaics: video mosaics with non-chronological time. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1, 2005.
- [44] Kishore K. Reddy, Jingen Liu, and Mubarak Shah. Incremental action recognition using feature-tree. In *ICCV’09*, pages 1010–1017, 2009.
- [45] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. *ICCV*, pages 612–617, 1995.
- [46] Ronald and Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.
- [47] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.
- [48] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. *ICPR*, 3:32–36, 2004.
- [49] A.C. Schütz, D.I. Brauna, and K.R. Gegenfurtner. Object recognition during foveating eye movements. *Vision Research*, 49(18):2241–2253, 2009.
- [50] Amnon Shashua and Shai Avidan. The rank 4 constraint in multiple ( $= 3$ ) view geometry. In *ECCV ’96: Proceedings of the 4th European Conference on Computer Vision- Volume II*, pages 196–206, London, UK, 1996. Springer-Verlag.
- [51] Y.S. Sheikh and M.M. Shah. Exploring the Space of a Human Action. *ICCV*, 1:144–149, 2005.
- [52] Y. Shen and H. Foroosh. View-invariant action recognition using fundamental ratios. In *Proc. of CVPR*, pages 1–6, 2008.
- [53] Y. Shen and H. Foroosh. View-invariant recognition of body pose from space-time templates. In *Proc. of CVPR*, pages 1–6, 2008.
- [54] Y. Shen and H. Foroosh. View-invariant action recognition from point triplets. *IEEE Trans. PAMI*, 31(10):1898–1905, 2009.
- [55] C. Sun, I. Junejo, and H. Foroosh. Action recognition using rank-1 approximation of joint self-similarity volume. In *ICCV*, 2011.
- [56] T. Syeda-Mahmood, A. Vasilescu, S. Sethi, I.B.M.A.R. Center, and CA San Jose. Recognizing action events from multiple viewpoints. *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 64–72, 2001.

- [57] Du Tran and Alexander Sorokin. Human activity recognition with metric learning. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *ECCV*, volume 5302 of *Lecture Notes in Computer Science*, pages 548–561. Springer Berlin / Heidelberg, 2008.
- [58] T. Vieville and Q.-T. Luong. Motion of points and lines in the uncalibrated case. Technical Report RR 2054, INRIA, Oct 1993.
- [59] L. Wang. Abnormal Walking Gait Analysis Using Silhouette-Masked Flow Histograms. *ICPR*, 3:473–476, 2006.
- [60] L. Wang, W. Hu, and T Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, 2003.
- [61] L. Wang and D. Suter. Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model. In *CVPR*, pages 1–8, 2007.
- [62] L. Wang, T. Tan, H. Ning, and W. Hu. Silhouette analysis-based gait recognition for human identification. *IEEE Trans. PAMI*, 25(12):1505–1518, 2003.
- [63] Daniel Weinland, Edmond Boyer, and Rémi Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, pages 1–7, 2007.
- [64] Daniel Weinland, Mustafa Özuysal, and Pascal Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, pages 635–648, 2010.
- [65] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2-3):249–257, 2006.
- [66] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 379 –385, jun 1992.
- [67] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. *CVPR*, 1:984–989, 2005.
- [68] A. Yilmaz and M. Shah. Matching actions in presence of camera motion. *CVIU*, 104(2-3):221–231, 2006.
- [69] Nazim Ashraf Yuping Shen and Hassan Foroosh. Action recognition based on homography constraints. *ICPR*, pages 1–4, 2008.
- [70] Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *CVIU*, 82(2):174–180, 2001.
- [71] G. Zhu, C. Xu, W. Gao, and Q. Huang. Action Recognition in Broadcast Tennis Video Using Optical Flow and Support Vector Machine. *LNCS*, 3979:89–98, 2006.