MULTIPLE VIEW HUMAN ACTIVITY RECOGNITION

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Selen Pehlivan
July, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Pınar Duygulu(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. David Forsyth(Co-advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Aydın Alatan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Dr. Selim Aksoy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Fatoş Yarman Vural

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Enis Çetin

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

MULTIPLE VIEW HUMAN ACTIVITY RECOGNITION

Selen Pehlivan
Ph.D. in Computer Engineering
Supervisor: Assist. Prof. Dr. Pınar Duygulu
Co-supervisor: Prof. Dr. David Forsyth
July, 2012

This thesis explores the human activity recognition problem when multiple views are available. We follow two main directions: we first present a system that performs volume matching using constructed 3D volumes from calibrated cameras, then we present a flexible system based on frame matching directly using multiple views. We examine the multiple view systems compared to single view systems, and measure the performance improvements in recognition using more views by various experiments.

Initial part of the thesis introduces compact representations for volumetric data gained through reconstruction. The video frames recorded by many cameras with significant overlap are fused by reconstruction, and the reconstructed volumes are used as substitutes of action poses. We propose new pose descriptors over these three dimensional volumes. Our first descriptor is based on the histogram of oriented cylinders in various sizes and orientations. We then propose another descriptor which is view-independent, and which does not require pose alignment. We show the importance of discriminative pose representations within simpler activity classification schemes. Activity recognition framework based on volume matching presents promising results compared to the state-of-the-art.

Volume reconstruction is one natural approach for multi camera data fusion, but there can be few cameras with overlapping views. In the second part of the thesis, we introduce an architecture that is adaptable to various number of cameras and features. The system collects and fuses activity judgments from cameras using a voting scheme. The architecture requires no camera calibration. Performance generally improves when there are more cameras and more features; training and test cameras do not need to overlap; camera drop in or drop out is

handled easily with little penalty. Experiments support the performance penalties, and advantages for using multiple views versus single view.

Keywords: Video analysis; Human activity recognition; Multiple views; Multiple cameras; Pose representation.

ÖZET

ÇOKLU GÖRÜNTÜ KULLANARAK İNSAN HAREKETİ TANIMA

Selen Pehlivan
Bilgisayar Mühendisliği, Doktora
Tez Yöneticisi: Assist. Prof. Dr. Pınar Duygulu
İkinci Tez Yöneticisi: Prof. Dr. David Forsyth
Temmuz, 2012

Bu tez insan hareketlerinin birden çok kamera görüntüsü ile tanınması üzerine yapılan çalışmaları içermektedir. Bu çalışmalarda iki farklı yöntem önerilmiştir. Birinci yöntemde kalibre edilmiş kameralardan elde edilen hacimleri eşleştiren bir sistem, ikinci yöntemde ise görüntü karelerini eşleştiren esnek bir sistem önerilmiştir. Kullandığımız iki farklı yöntemde elde ettiğimiz sonuçlar, tek kamera görüntüleri ile yapılan çalışmalarda elde edilen sonuçlarla karşılaştırılarak, farklılıkları ve performansları incelenmiştir.

Tezin ilk bölümü geri çatılım yöntemi ile elde edilen hacimsel veriler için yoğun betimleyiciler önerir. Kameralar tarafından kaydedilen görüntü kareleri geri çatılım yöntemi ile birleştirilir ve elde edilen hacimler hareket pozlarının eşleniği olarak kabul edilir. Bu çalışmalarda üç boyutlu verilerin üzerinden hızlı ve ayırt edici özelliklere sahip yeni poz betimleyicileri önerilmiştir. Bu betimleyicilerden ilki farklı doğrultuda ve boyuttaki silindirlerin histogramıdır. Önerilen bir diğer poz tanımlayıcısı ise bakış açısından bağımsızdır yani poz hizalamasına ihtiyaç duymamaktadır. Poz tanımlayıcılarının önemi hareket tanımlama kısımları sade tutulan düzeneklerde gösterilmiştir. Sunulan hacim eşlenmesine dayalı hareket tanımlama literatüre göre başarılı sonuçlar ortaya çıkarmıştır.

Birden çok kamera verisinin işlenmesi ve ayıklanmasında hacim geri çatılım metodu seçilen en doğal yöntem olmuştur. Ancak birbiriyle örtüşen mevcut görüntüler yeterli sayıda olmayabilir. Tezin ikinci bölümünde farklı sayıda kamera ve öznitelikle çalışabilen bir hareket tanıma sistemi önerilmektedir. Bu sistem kamera görüntülerindeki hareket bulgularını oylama tekniği ile bulmaktadır ve kameraların kalibre edilmesine gerek duyulmamaktadır. Sistemin performansı kamera ve öznitelik sayısıyla orantılı olarak artmaktadır. Eğitim ve

sınama için kullanılan kamera görüntülerinin örtüşmesine gerek yoktur. Sisteme herhangi bir anda bir kameranın girişi ve çıkışı kolayca çözümlenmektedir. İnsan hareketi tanımlanmasında birden çok kameranın kullanılmasının, tek kamera kullanılmasına oranla avantajları deneylerle desteklenmiştir.

Anahtar sözcükler: Video inceleme; İnsan hareketi tanıma; Çoklu bakış; Çoklu kamera sistemi; Poz betimleyici .

Acknowledgement

I think myself as a lucky graduate student having the freedom of picking up the research problems that I found interesting. I am deeply appreciated to my advisors since I always felt their support in my decisions and studies.

I would like to express my gratitude to my supervisor, Prof. Pınar Duygulu. She always encouraged me during my studies. I am grateful for her guidance in this path full of with challenges.

I had the privilege and pleasure to work with Prof. David Forsyth. He always advised me to think on interesting problems. I learned a lot from him and I will always be indebted to him during my academic life.

I would like to thank my thesis committee, Prof. Selim Aksoy and Prof. Aydın Alatan for valuable suggestions and for discussions about various aspects of my research; and also would like to thank my defense committee Prof. Fatoş Yarman Vural and Prof. Enis Çetin for reviewing my thesis.

I appreciate all former and current members of RETINA Vision and Learning Group at Bilkent, Nazlı, Gökhan, Çağlar, Derya, Fırat, Aslı, Daniya for their collaboration and friendship. I also would like to thank members of EA128, Tayfun, Hande, Buğra, Ateş and Aytek.

I am grateful to members of Computer Vision Group at UIUC for their company. I have learned a great deal from their work.

I would like to acknowledge the financial support of TÜBİTAK (Scientific and Technical Research Council of Turkey). I was supported by the graduate fellowship during my PhD. I also would like to thank for the financial support during my visit in University of Illinois at Urbana-Champaign.

My thanks to my old friends Mürüvvet, Burcu, and Elif. I always enjoyed their company during this long journey. I would like to give my special thanks

to Cenk and all the other UIUC folks, especially Özgül, Onur, Emre, Ulya, Lale, and Ayşegül.

My parents, Serpil and Mehmet, thanks for the constant love, support, and encouragement. I could have done nothing without you.

To my parents, Serpil and Mehmet.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Activity understanding is a complex as well as interesting research subject of computer vision [1, 16, 14, 58, 45]. There is a broad range of applications for systems that can recognize human activities in videos. Medical applications include methods to monitor patient activity such as tracking of progress in stroke patients or keeping demented patients secure. Safety applications include methods detecting unusual or suspicious behavior, or detecting pedestrians to avoid accidents. Private and public video collections are getting popular and users need applications with automatic annotation features, efficient search and retrieval features on large video collections (see Figure 1.1).

Activity understanding problem remains difficult, however, for important reasons. There is no canonical taxonomy of human activities. Changes in illumination direction and viewing direction cause drastic differences in what people look like. Individuals can look very different from one another, and the same activity performed by different people can vary widely on how they are perceived in appearance.

The majority of work on this subject focuses on understanding activities from videos captured by a single camera [5, 11, 4, 63]. One problem with single camera systems is that viewpoint differences can yield massive decrease in recognition performance due to weak response against occlusion (resp. self-occlusion). View

range is limited and nearly one third of the human body degree-of-freedoms can be missed resulting in partially captured human postures and misclassified activities. Multi-camera systems have emerged as a solution [15, 9, 17, 61, 35], and now are more affordable.

(a) Large Video Collections

(b) Game Industry

(c) Robot Vision

(d) Automatic Annotations

(e) Medical Applications

(f) Safety Applications

Figure 1.1: Cameras are installed in many places and activity recognition systems are essential for surveillance, monitoring and tracking in indoor and outdoor places with methods including medical applications with a particular importance on elderly people and children, safety applications detecting unusual behaviors. Advances in activity recognition problem trigger intelligent home, office applications using human computer interfaces (HCI) for game and multimedia industry, it provides good support for features related to search, retrieval and automatic annotations of large video collections in various domains (e.g. sport, music and dance videos).

Using multiple cameras presents effective solutions to the problems related to the limited view and self-occlusions. It provides wider view range compared to using single view on the objects appearing in overlapping camera regions. While one view misses the accurate body configuration or the discriminative

appearance cues, the other view supports to recover the discriminative features from an unoccluded aspect. Multiple camera systems can provide a broad range of surveillance and tracking applications, but they bring new challenges. They require new fusion techniques over the available views from multiple cameras, new descriptors over fused data for pose matching and activity recognition, and new scalable architectures for camera networks.

In this thesis, our aim is to investigate the multiple views in activity understanding. We study the human activity recognition problem from videos having multiple views. We follow two directions and propose compact representations as well as promising architectures to fuse large amount of video data recorded by multiple cameras.

1.1 Overview of the Proposed Approaches

We propose methods for two scenarios. In the first scenario, we assume that there exist a calibrated camera system with enough number of camera views to reconstruct volumes. Camera views are fused by 3D reconstruction and pose instances are represented in the form of 3D volumes. The fused 3D volume is scaled version of a real-world human pose. It results in better invariance to changes in illumination and it no longer depends on the camera orientation. The activity recognition performance radically depends on the performance of the pose descriptors of the 3D volumes. We aim to have compact representations that are fast to implement and that perform well in pose and activity recognition. Two new pose representations for volumes are introduced and the activity recognition is carried out using volumetric features.

In the second scenario, the human activity recognition problem is investigated when there exist few number of cameras (resp. views) in an uncalibrated camera setting. Motion capture systems or systems with many cameras are designed for controlled indoor environments and they are used for computing accurate joint locations and body configurations. In fact, there is no need of having many

camera views to recognize the true activity class. The discriminative appearance features for activity recognition can be obtained through views that are less than that of used in motion capture. Our aim is to introduce a recognition system for activity recognition with scalable and straightforward architectural properties that support the system to be adaptable for various number of cameras. Every pose is represented by multiple camera frames. The system fuses the individual judgments of every frame on activities, so the activity recognition is performed as a combination of individual frame judgments.

1.1.1 Multiple View Activity Recognition Using Volumetric Features

We assume that 3D volumes obtained from calibrated cameras are available. Human poses have articulated structures that are quite different than rigid body objects resulting in high number of potential configurations. In volume based systems, it is difficult to use an articulated body model reliably [9, 17], thus most studies consider activities as 3D shapes that change over time. Standard shape descriptors used in 3D shape retrieval systems aim to provide rotation invariance [3, 20, 22], however 3D poses as part of an activity sequence need descriptors that are invariant to vertical axis rotation (e.g. the action of an upside-down person should be considered different than the action of a standing person). Particularly, we need pose descriptors resulting in invariant activity representations for recognition when combined in temporal domain.

We investigate volume matching strategies where each frame is modeled as a volume. Poses are important in activity understanding [51, 8, 34]. Key poses represented with robust and discriminative descriptors are usually enough to classify the true activity label. Therefore, a strong representation in the pose level is necessary for activity recognition. There are discriminative 2D shape descriptors in the literature [5, 4, 11]. However, they can not be used in 3D domain. This requires new pose representations for volumetric poses. We focus on representation of the poses to explore to what extent a human pose can help in understanding

human activities. We therefore keep the classification part simple and observe that good features from volumetric data are important to obtain good recognition results.

Human body parts such as torso, arms, legs look like cylinders varying in size and orientation [36]. We introduce two new volume based pose representations based on this assertion with a comparable performance to more complex systems. The first representation is the one computed with the bag-of-features method using oriented cylinders. The second representation introduces a view-independent, fast and simple encoding strategy to show volumes as the layered circular features. Both approaches model activities as time varying sequences of human poses represented by strong volumetric features extracted from voxel grids.

1.1.1.1 Pose Features Using Oriented Cylinders

Our first pose representation introduced for volumetric data uses a set of cylinder like 3D kernels. These kernels change in size and orientation. They are used to search over 3D volumes to obtain a rough estimate of volumetric shapes. We expect some cylinders to fire rarely in particular body parts (e.g. salient parts) or some cylinders that are not discriminative to fire frequently. The distribution of cylinder responses results in discriminative features in the sense of a key pose. This is further used as 3D pose descriptor. We show the performance of the proposed representation for (a) pose retrieval using Nearest Neighbor (NN) and Support Vector Machine (SVM), and for (b) activity recognition using Dynamic Time Warping (DTW) and Hidden Markov Model (HMM) based classification methods. Evaluations on the IXMAS dataset support the effectiveness of the approach.

This is our first attempt towards modeling activities using multiple camera systems with known camera matrices. The primary contributions of this study are to investigate the importance of key poses in 3D domain with volumes and to

introduce a new pose representation for an activity recognition system with multiple views. In [18], a pose descriptor is defined as the spatial distribution of 2D rectangular filters over human silhouettes. However, we can not apply it over 3D volumes. Our study extends the bag-of-rectangles idea defined for silhouette images to bag-of-cylinders descriptor for voxel data. Our volumetric representation offers significant benefits over 2D representation. It is robust against occlusion and it does not get affected by the relative actor-camera orientation while a 2D silhouette based shape descriptor may suffer from ambiguous appearance due to viewpoint variations. However, the dimensionality of the search space increases in volumes.

1.1.1.2 Pose Features Using Layered Features

Pose representation should be view independent. In other words, the activity should be recognized for any orientation of the person and for arbitrary positions of the cameras. While there are available 3D shape descriptors that provide rotational invariance for shape matching and retrieval, for human activity recognition it is sufficient to consider variations around the vertical axis of the human body. Following the idea based on the importance of view-independence in representation, we present a new pose representation based on encoding of the pose shapes that are initially provided as volumes. Simplicity of the encoding scheme helps to overcome the search related problems encountered in 3D domain.

Considering body parts as cylinders, we use their projections on to horizontal planes intersecting the human body in every layer. We assume that the intersections of the body segments in a layer can be generalized as circles. The circular features in all layers (a) the number of circles, (b) the area of the outer circle, and (c) the area of the inner circle are then used to generate a pose descriptor. The proposed descriptor is not view dependent and therefore does not require pose alignment, which is difficult to do in the case of noisy data. The pose descriptors in consecutive frames are further combined. Changes in the number, area and relative position of these circles over the body and over time are encoded as the discriminative motion descriptors. The activity recognition system classifies

an activity sample by querying over the pool of motion descriptors with known activity labels in a nearest neighbor matching strategy.

The contributions of this study can be summarized as follows. First, we introduce a new layer-based pose representation using circular features. These features store discriminative information about the characteristics of an actual human pose, e.g., number of circles to encode number of body parts, bounding circle to encode the covered area of all parts and inner circle to encode the distance in between body parts. The study demonstrates that representation has various advantages. It significantly reduces the 3D data encoding. Circular model is effective for providing view invariance and it solves pose ambiguities related to the actors styles. In addition, representation does not require pose alignment.

Second, we introduce a new motion representation for describing activities based on our pose representation. Motion features are extracted from matrices constructed as a combination of pose features over all frames. The extracted motion features encode spatial-temporal neighborhood information in twofold: variations of circular features in consecutive layers of a pose and variations of circular features in consecutive poses through time.

Finally, we use our motion representation to propose a two stage recognition algorithm using well-known distance measures: L_2 norm and EMD-L_1 "cross-bin" distance [30].

Experiments show that our method based on volume matching is better in performance than other studies based on (a) training and testing on the same single camera (b) training and testing on different single camera and (c) training on volume projections and testing on single camera. Moreover, our pose representation presents comparable results to other studies of volume matching (see Table 2.1).

1.1.2 Multiple View Activity Recognition Without Reconstruction

There are practical disadvantages of the approaches that are based on matching 3D reconstructed volumes. One must find which cameras overlap, there need to be enough overlapping views to reconstruct, and one must calibrate the cameras. Figure 1.2 shows some ambiguous cases for volume reconstruction when there exist few number of cameras used in the system. The shapes of the reconstructed volumes are dependent on the camera positions and the overlapping regions. A bad camera layout can generate bad reconstructions including extra voxels. This results in many pose hypotheses that can be ambiguous for extracting discriminative pose features and for identifying pose labels.

A simpler alternative is to match frames individually, and then to collect activity votes, and to fuse activity judgments from overlapping views. This offers significant advantages in system architecture. For example, it is easy to incorporate new features; camera dropouts and camera addition can be tolerated by the system. We show that there is no performance penalty for using a straightforward weighted voting scheme. In particular, when there are enough overlapping views to produce a volumetric reconstruction, our recognition performance is comparable with that produced by volumetric reconstructions; but when views are not enough, performance degrades fairly gracefully, even in cases where test and training views do not overlap.

The main point of this study is to show that, when one has multiple views of a person, there is no reason to build a 3D reconstruction of the person to classify. Doing so creates major practical difficulties, including the need to synchronize and calibrate cameras. But straightforward data fusion methods give comparable recognition performance in the context of a radically simpler system architecture with significant advantages. Our work emphasizes the scalability to more cameras and more appearance features. It appears to be practical for large distributed camera systems.

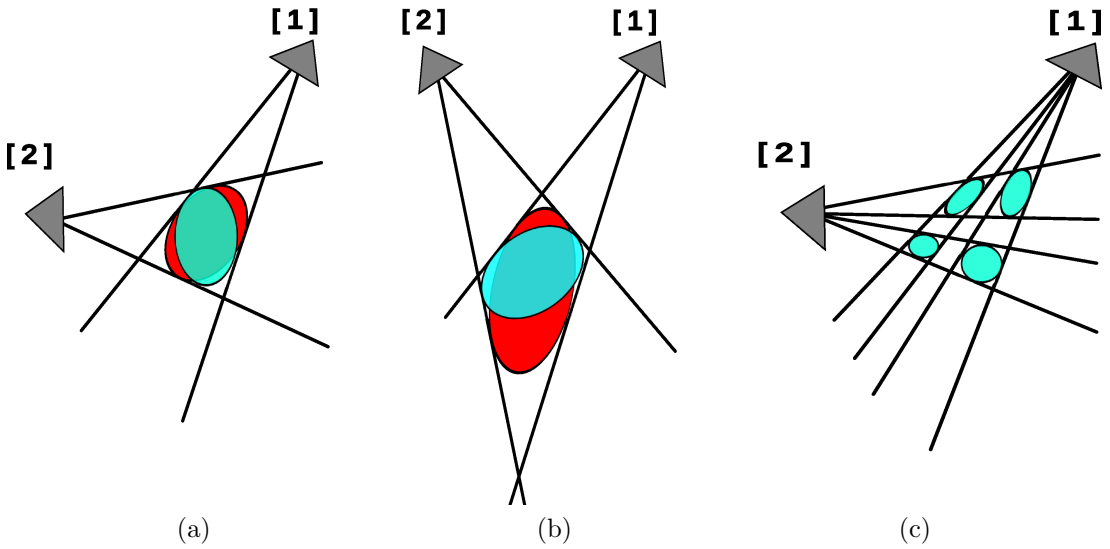


Figure 1.2: Inferring the shape of human pose using 3D reconstructed volumes may be ambiguous resulting in performance penalty in activity recognition when there are few number of cameras in the system. In example (a) and (b); fixing camera 1, and moving camera 2, the area under cones is ambiguous and respectively the volume of the object (red and blue ellipses represent two of many possible body hypotheses in the overlapping region). Example (c) shows a slice taken from a leg portion of a reconstructed volume of a walking human pose, but instead the volume looks like belonging to a four-legged object.

The contributions of the study can be summarized as follows. First, an architecture for activity recognition based on frame matching is proposed with straightforward and simpler appearance features. It confers to the state-of-the-art in performance.

Second, the system offers a straightforward data fusion technique between cameras and low level appearance features so that performance improves when new cameras and new features get available to the system. We achieve the state-of-the-art performance with many architectural advantages using the proposed system.

Third, the architecture allows training and test camera sets to differ. It is

shown that view transfer is at near state-of-the-art. This also supports camera dropouts and drop ins in real-time.

Experiments support that multiple views in activity recognition systems have significant benefits and outperform the single-view recognition systems. When the same single camera is used both for training and testing, proposed system performs at the state-of-the-art . When two cameras are used, the performance is best compared with the group of studies of types 4 and 5 in Table 2.1. The comparable performance with respect to volume reconstruction is justified by the tremendous advantages.

1.2 Organization of the Thesis

In this chapter, we introduce our proposed approaches for multiple camera human activity recognition and present a brief outline. We introduce two main scenarios and we summarize the motivations behind them with contributions as part of the thesis.

In Chapter 2, previous studies related to activity recognition are reviewed. Particularly, we present studies related to 2D image features, we review related works that investigate view invariance, and we talk about the state-of-the dataset that is used in our study and the related works experimenting on this dataset.

In Chapter 3, our study that is briefly explained in Section 1.1.1.1 is introduced. We present a new volumetric pose representation using oriented cylinders in detail.

In Chapter 4, we present a new view-independent pose representation based on circular features that is briefly explained in Section 1.1.1.2. Experiments are performed and discussed to show the robustness of proposed descriptor for pose retrieval and activity recognition.

In Chapter 5, the activity recognition architecture proposed as second scenario is introduced. The multi view activity recognition system is presented with

extensive experimental evaluation.

Lastly, in Chapter 6, the conclusion of this thesis is given with discussions on potential ideas and future plans.

Chapter 2

Related Work

The activity recognition literature is rich. There are several detailed surveys of the topic [1, 16, 14, 58, 45]. We confine our review to covering the main trends in features used, in methods that recognize activities from viewpoints that are not in the training set, the dataset used in our experiments and the related studies experimenting on this dataset.

2.1 Image Features

Low level image features can be purely spatial, or spatio-temporal. Because there are some strongly diagnostic curves on the outline, it is possible to construct spatial features without segmenting the body (for example, this is usual in pedestrian detection [10]). An alternative is to extract interest points that may lie on the body (e.g. [41]). In activity recognition, it is quite usual to extract silhouettes by background subtraction (e.g. [4, 57, 18]). Pure spatial features can be constructed from silhouettes by the usual process of breaking the domain into blocks, and aggregating within those blocks (e.g. [57, 18]). Doing so makes the feature robust to small changes in segmentation, shifts in the location of the bounding window, and so on.

Because many activities involve quite large body motions on particular limbs, the location of motions in an image can provide revealing features. Efros *et al.* [11] show that averaged rectified optical flow features yield good matches in low resolution images. Long term trajectories reduce the noisy optical flows against background motion, and get spatial-temporal structure of action [38, 37]. Laptev and Pérez show that local patterns of flow and gray level in the spatial-temporal pyramid are distinctive for some actions [27].

Polana and Nelson [44] measure the action periodicity. Bobick and Davis show a spatial record of motion using silhouette images (a motion history image) is discriminative [5]. Blank *et al.* [4] show that joining consecutive silhouettes into a volume yields discriminative features. Laptev and Lindeberg introduce spatio-temporal interest points [25]; descriptors can be computed at these points, vector quantized then pooled to produce a histogram feature. Scovanner *et al.* [53] propose spatio-temporal extension of 2-D sift features for action videos. Alternatively, Kläser *et al.* [24] introduces 3D spatial-temporal gradients.

2.2 View Invariance

Changing the view direction can result in large changes in appearance of the silhouette and motion of the person in the image. This means that training with one view direction and testing with another view can result in significant loss of performance. Rao *et al.* [49] build viewpoint invariant features from spatio-temporal curvature points of hand action trajectories. Yilmaz and Shah [64] compute a temporal fundamental matrix to account for camera motion while the action is occurring so they can match sets of point trajectories from distinct viewpoints. Parameswaran and Chellappa [42] establish correspondences between points on the model and points in the image; then compute a form of invariant time curve, then match to a particular action. The method can learn in one uncalibrated view and match in another. However, methods to build viewpoint invariant features currently require correspondence between points. Neither one

could match if corresponding points were not visible. We show that, when a second or further camera is available, we can improve recognition without any reasoning about camera-camera calibration (fundamental matrices and volume reconstruction, but equivalently point correspondences).

Instead, Junejo *et al.* [21] evaluate pairwise distances among all frames of an action and construct self-similarity matrix that follows discriminative and stable pattern for the corresponding action. Similar to our work, there is no estimation of corresponding points. On the contrary to our method, and similar to previous methods, there is no evidence that multiple camera views improve activity recognition.

An alternative is to try and reconstruct the body in 3D. Ikizler and Forsyth [19] lift 2D tracks to 3D, then reason there. While lifting incurs significant noise problems because of tracker errors, they show the strategy can be made to work; one advantage of the approach is one can train activity models with motion capture data. Activities are modeled as a combination of various HMMs trained per body part over motion capture data. This allows recognizing complex human activities and provides a more generic system for unseen and composite activities.

Weinland *et al.* [61] build a volumetric reconstruction from multiple views, then match to such reconstructions. The main difficulty with this approach is that one needs sufficient views to construct a reasonable reconstruction, and these may not be available in practice. For activity recognition, less number of cameras with broad field of view can be enough. Alternatively, one could use volumes only during the training stage [35, 59], then generate training frames from those volumes by projection into synthetic camera planes. Doing so requires training volumes to be available; in our study, we assume that they might not be.

Another group of studies fuses the image features extracted from multiple cameras. One such work is presented in [32], where bag-of-video-words approach is applied to a multi-view dataset. The method detects interest points and extracts spatial-temporal information by quantizing them. However, it is hard to infer the poses by the orderless features. Moreover, extracted features like interest

points are highly influenced by illumination affects and actors' clothing, relative to the reconstructed volumes.

An alternative is to build discriminative models of the effects of viewpoint change (transfer learning). Farhadi and Tabrizi emphasize how damaging changes of viewpoint can be [12]; for example, a baseline method gives accuracies in the 70% range when train and test viewpoint are shared, but accuracy falls to 23% when they are not. They give a method for transferring a model from one view to another using sequences obtained from both viewpoints. Alternatively, Liu *et al.* [33] introduce more discriminative bilingual-words as higher level features to support cross-view knowledge transfer. Unfortunately, the methods require highly structured datasets. A more recent paper [13] gives a more general construction; we do not use it here, because the method cannot exploit multiple cameras.

2.3 Volumetric Representations

In [17], a 3D cylindrical shape context is presented for multiple camera gesture analysis over volumetric data. For capturing the human body configuration, voxels falling into different bins of a multi layered cylindrical histogram are accumulated. Next, the temporal information of an action is modeled using Hidden Markov Model (HMM). This study does not address view independence in 3D, instead, the training set is expanded by asking actors to perform activities in different orientations.

A similar work on volumetric data is presented by Cohen and Li [9] where view-independent pose identification is provided. Reference points placed on a cylindrical shape are used to encode voxel distributions of a pose. This results in a shape representation invariant to rotation, scale and translation.

Two parallel studies based on view-invariant features over 3D representation are performed by Canton-Ferrer et al. [7] and Weinland et al. [61]. These studies

extend the motion templates of Bobick and Davis [5] to three dimensions, calling them Motion History Volumes (MHV). MHV represents the entire action as a single volumetric data, functioning as the temporal memory of the action. In [61], the authors provide view invariance for action recognition by transforming MHV into cylindrical coordinates and using Fourier analysis. Unlike [61], Canton-Ferrer et al. [7] ensure view invariance by using 3D invariant statistical moments.

One recent work proposes a 4D action feature model (4D-AFM) to build spatial-temporal information [62]. It creates a map from 3D spatial-temporal volumes (STV) [63] of individual videos to 4D action shape of ordered 3D visual hulls. Recognition is performed by matching STV of observed video with 4D-AFM.

Stressing the importance of the pose information, in recent studies action recognition is performed using particular poses. Weinland et al. [59] present a probabilistic method based on exemplars using HMM. Exemplars are volumetric key-poses extracted by reconstruction from action sequences. Actions are recognized by an exhaustive search over parameters to match the 2D projections of exemplars with 2D observations. A similar work is that of Lv and Nevatia [35], called *Action Net*. It is a graph-based approach modeling 3D poses as transitions of 2D views that are rendered from motion capture sequences. Each 2D view is represented as a shape context descriptor in each node of the graph. For recognition, the most probable path on *Action Net* returns the matched sequence with the observed action.

In [56], Souvenir and Babbs extend shape descriptor based on radon transform and generates 64 silhouettes taken from different views of a visual hull. Action recognition is performed by estimating the optimum viewpoint parameter that would result in the lowest reconstruction error.

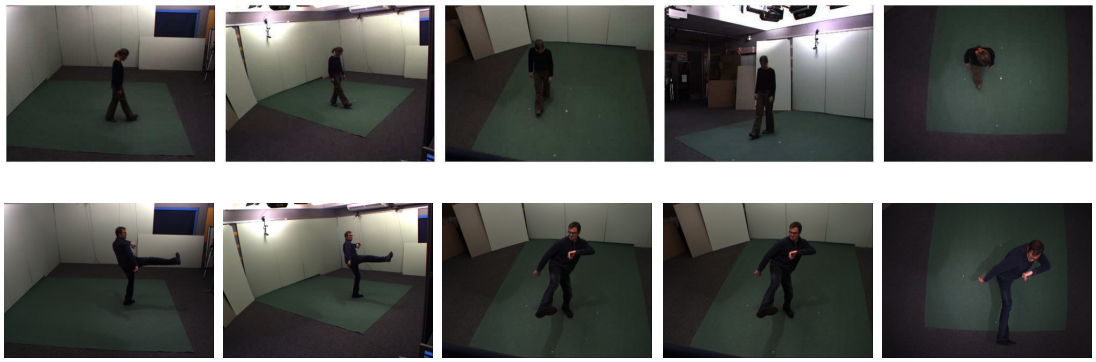


Figure 2.1: Example views from the IXMAS dataset captured by 5 different synchronized cameras [61].

2.4 Datasets

Numerous datasets are now available for activity recognition. The KTH [52] and the Weizmann [4] are the state of the art human action datasets used in the literature. The Weizmann dataset contains relatively simpler activities in fronto parallel views and static background. However, the KTH dataset has camera motion and includes actors in various clothing.

The Hollywood [26] dataset contains realistic human actions from various movies with camera motion and dynamic backgrounds. Similarly, the YouTube action dataset [31] have various challenging videos.

The IXMAS dataset [61] is a multi-camera human action dataset with different viewpoints. The cameras are fixed but the actors perform freely. We have chosen to use the IXMAS dataset because it has been quite widely used, and because it has good support for research on multiple views (each sequence is obtained from five different viewpoints). Example camera views of the IXMAS camera setting are shown in Figure 2.1.

In Table 2.1, we show recent methods applied to the IXMAS dataset, broken out by types of experiment. Some studies train and test on the same (resp different) camera. The same camera methods outperforms the different camera

Table 2.1: Results in recent studies using the IXMAS dataset. We group studies in terms of experimental strategies performed over multiple viewpoints. Studies of types 4 and 5 perform 3D reconstruction; in type 4, reconstructed volumes are used to generate training frames; in type 5, volumes are used both for training and testing. Type 5 reports the highest recognition rates on the dataset.

Type of Experiment	Method	Accuracy %
1) Train and test on the same single camera	[59]	59.57 (3 cam)
	[12]	68.80 (5 cam)
	[13]	84.60 (5 cam)
	[21]	74.00 (5 cam)
	[60]	83.90 (5 cam)
	[33]	79.86 (5 cam)
2) Train and test on different single camera	[12]	58.15 (5 cam)
	[13]	74.75 (5 cam)
	[21]	61.80 (5 cam)
	[33]	75.30 (5 cam)
3) Train on all cameras and test on single camera	[32]	73.73 (4 cam)
	[60]	83.40 (5 cam)
4) Train on projections and test on single camera	[59]	57.86 (5 cam)
	[35]	80.60 (5 cam)
	[62]	64.00 (4 cam)
5) Volume matching	[61]	93.33 (5 cam)
	[43]	90.91 (5 cam)

methods due to variations in viewpoint. The performance usually improves with low level feature setting. For example, [60] presents good results for activity recognition using local features that are robust for occlusion. Moreover, the view invariant features, e.g. [21] increases recognition. Some other group of studies rely on the training view samples that are obtained by either using multiple training camera views or projecting the reconstructed 3D volumes. These studies use a single test camera, but enriching the training samples provides good matches in appearance. The final group of studies are based on volume matching techniques using 3D volumes. They perform well among the studies in the state-of-the art.

Chapter 3

Volume Representation Using Oriented Cylinders

The natural and most common way to store human poses from multiple camera frames is to fuse views in the form of 3D volumes obtained by reconstruction [39, 23, 17, 62]. The challenging part is to find a representation which is efficient and also robust to changes in view point, or to differences in size and style of human actors while searching for poses or actions. Although 3D representations is a well studied area for 3D model matching [3, 20, 22], human poses are more challenging than any rigid body object due to articulated structure of human bodies. The high number of degree of freedoms on the human body, causing many different potential configurations, requires search algorithms specific to articulated structures of human poses.

Activities can be identified by a single key frame (resp poses), thus a discriminative pose representation results in a well performing activity recognition system. In this chapter, our objective is to introduce a compact representation for finding human body parts of 3D poses in any configuration and further 3D action sequences. We consider body parts as a set of cylinders with various orientations and sizes, and we then represent poses as distribution of these cylinders over a 3D pose. For example, it is more likely the cylinders with larger radius values to

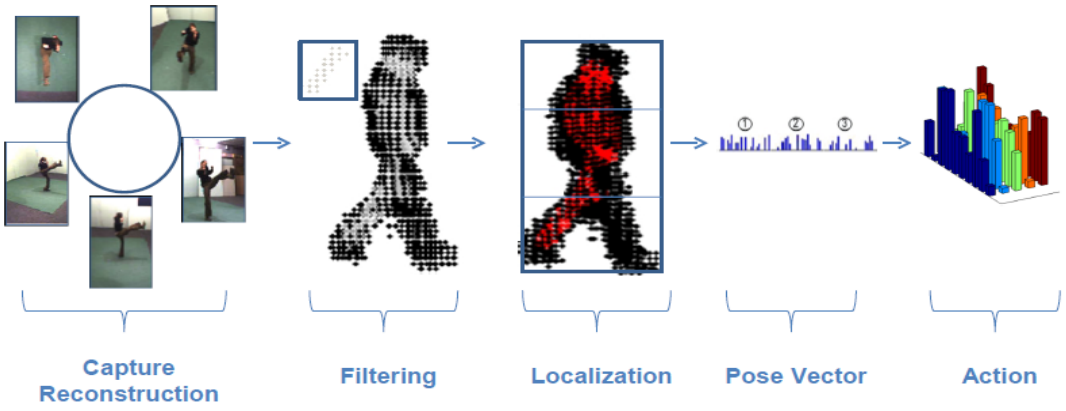


Figure 3.1: Our activity recognition system is based on the proposed pose descriptor that is designed for 3D volume matching. 3D Poses reconstructed from multiple camera views are represented as the distribution of cylinders. First, a set of 3D kernels (cylinders) that are different in orientation and size are searched over the reconstructed 3D Poses. Then, the bounding volume is extracted and then divided into parts through the vertical axis. From each sub-volume, high response volume regions with a score larger than a determined threshold are selected and a histogram of distribution of oriented cylinders is computed. The combination of these histograms constructs the feature vector to be used as a pose descriptor and the combination of pose vectors corresponds to the action matrix.

fire on human torso, longer cylinders to appear on legs or arms, or some cylinders to catch salient body parts.

The organization of the chapter is as follows. First, our proposed pose representation is introduced with details of its design in Section 3.1. Next, in Section 3.2, we present the methods used for activity recognition. Then, experimental results are presented and evaluated in Section 3.3, and the chapter is concluded with discussions in Section 3.4.

3.1 Pose Representation

A human pose is characterized by configuration of body limbs having cylinder like shapes in different sizes and orientations. This notion plays an important role while structuring our representation. We model 3D poses as a set of cylindrical kernels changing in size and orientation. Motivated by the success of representing 2D human poses as distribution of rectangles over more complicated models [18], we use the distribution of these cylinders as a compact representation for 3D human poses.

In order to find cylinder-like structures over body, we generate a set of 3D kernels and measure the correlation of these kernels with human poses that are represented in the form of volumetric data. Highly correlated regions with a cylinder are most likely corresponds to body parts in the same size and orientation. Rather than using complex models, we count the frequencies of occurrences of every cylinder. However, methods based on bag-of-feature do not have information related to the spatial location. In fact, distributions computed on local regions can exhibit more discriminative features than globally computed ones. A simple localization can help and this is provided by partitioning the bounding volume into sub-volumes. Histogram of cylinders is computed and matched separately for every sub-volume. Figure 3.1 illustrates the overall process of activity recognition procedure. In the following, the details of the proposed method are described.

3.1.1 Forming and Applying 3D Kernels

We assume that 3D poses are provided in the form of voxel grid. In order to search cylinders on the volumetric data appropriately, we construct 3D kernels in the same format. A 3D kernel is formed as a grid data consisting of voxels that are located inside a cylinder. A cylinder is defined by two parameters that are radius and height.

A set of kernels is constructed from every 3D kernel by rotating it around

its local axis α° apart (see Figure 3.2 for an example). The symmetry of kernel reduces almost one half of the search space in 3D. Therefore, the number of kernels with α° apart can be computed as follows:

$$|K(\alpha)| = 1 + (\frac{90}{\alpha} - 1)(\frac{360}{\alpha}) + (\frac{180}{\alpha}) \tag{3.1}$$

Limbs are the salient body parts that are discriminative in terms of the key pose and they occur in various sizes. Longer cylinders are more discriminative to locate a limb than shorter ones, however some limbs can be missed by just using longer cylinders due to noise factor. Therefore, we construct 3D kernels in various lengths. Similarly, limbs in different thicknesses can be detected by cylinders changing in radius. Empirically we have observed that a thick limb can be represented as a collection of thin cylinders, and therefore we choose a fix radius value for all kernels.

After building sets of kernels, we convolve every 3D 0-1 kernel with every 3D volumetric pose. This gives a similarity score between every kernel and every voxel location. The scores are scaled in the range of $[0, 1]$ and then, regions having a score more than a threshold are selected. In the experiments, we choose 0.8 as the threshold. This threshold may change depending on whether the search is over dense or sparse data. Voxels with high scores to some kernels may correspond to body parts that have the same appearance with the applied kernels. Some kernels results in high similarity score in fewer number of voxel locations than some other kernels. This information is discriminative in the sense of a pose representation.

3.1.1.1 Distribution of Cylinders

We define a pose descriptor as distribution of oriented cylinders. First, we form a set of kernels with different sizes and orientations as mentioned previously. Then, we model a representation storing the frequency of high response regions for each kernel.

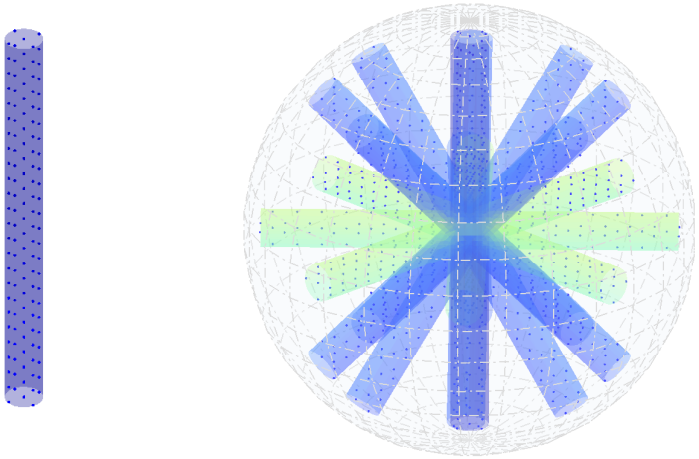


Figure 3.2: Left figure is a kernel in the form of voxel grid. Right figure is the set of kernels with various orientations around the local axis 45° apart. The number of kernels in the set is 13 for that rotation angle.

Although distribution of cylinders reveals crucial information, localization is needed for a better representation. For this purpose, we fit a bounding volume to each pose and divide the bounding volume into N equal sized sub-volumes through the vertical axis. This process corresponds to dividing the height of an actor's pose into partitions. Empirically, we have found that $N = 3$ gives the best performance. The computed histograms for each sub-volume is then combined to obtain a single histogram for a given pose. Dividing through horizontal axis is also possible, but gridding on the horizontal plane of the bounding volume conflicts with the rotation invariance property of poses around vertical axis (e.g. actors are free to perform activities in different vertical orientations).

3.2 Action Recognition

In this section, we present methods for recognizing actions using proposed pose representation. Considering actions as sequences of poses, we evaluate two methods for recognition and temporal smoothing: a) Dynamic Time Warping (DTW) and b) Hidden Markov Model (HMM).

3.2.1 Dynamic Time Warping

Action sequences performed by different actors vary in time and speed. The most common way to handle similarities among time series is to use Dynamic Time Warping (DTW) [46]. In our problem, actions are sequence of poses where each pose is modeled as a set of cylinders using proposed representation. This results in a 2-D representation per action. However, DTW for 2-D series is an NP-complete problem.

Instead, we make use of the approach [18] to find similarities between action sequences. In this approach, DTW is applied to compare 1-D series located at the same bin location of two different action representations. This is to measure the fluctuations through time at the same bin location corresponding to number of cylinders for a specific orientation and size. We sum the results of DTW for all bin comparisons and find an optimum match.

Throughout an action, some body regions in the form of voxel grid do not change. Therefore, data series at some bins of the histogram rarely change. We measure the variance at each bin to measure the amount of change through time. Then, if the variance is below a given threshold, we set the 1-D series at that bin location to zero vector.

3.2.2 Hidden Markov Model

Second method that we apply for action recognition is Hidden Markov Models (HMM). In our problem, we have a set of actions consisting of consecutive frames. A frame at time t only depends on a previous frame at time $t-1$. Thus, HMM is a good technique to model our problem. In our case, we model the problem using discrete HMMs [47].

In this approach, we first quantize all poses in training actions using k-means clustering algorithm into a set of codewords which we refer to as pose-words. Then, we construct HMM models per action class using 3 states. An unknown action sequence is assigned to a class giving the highest likelihood for its HMM model. Similar to DTW, the variance of each bin is computed to find uniform series. 1-D series with a variance below a threshold are set to zero vector.

3.3 Experimental Results

3.3.1 Dataset

We test our pose descriptor on publicly available INRIA Xmas Motion Acquisition Sequence (IXMAS) dataset [61]. We choose 5 actions performed by 12 different actors 3 times in different orientations. These actions are walk, wave, punch, kick and pick up.

Multi-view action videos are recorded by 5 cameras and videos are used to construct volumes. Volumes from multiple views are extracted using shape from silhouette technique. Results are taken using volumes in the form of 64x64x64 voxel grid.

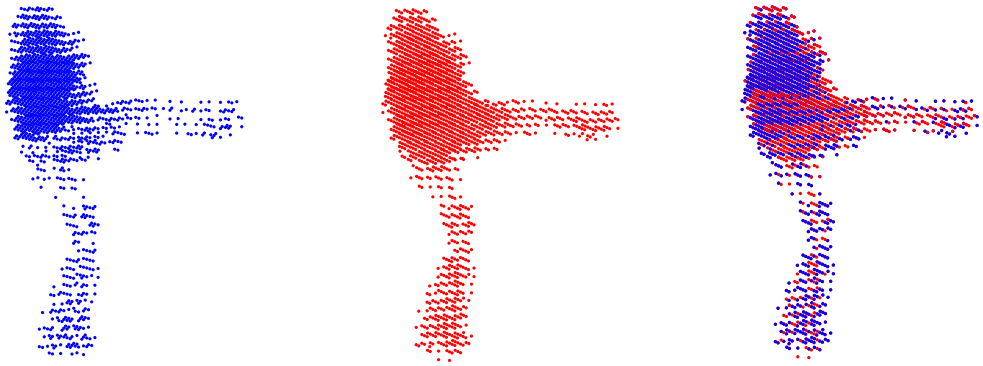


Figure 3.3: From left to right: the actual kick pose, enhanced kick pose, the difference between them. We fill the missing voxels using close morphology operator with a 3D sphere structuring element. In the third figure, we represent the actual pose and the enhanced version as overlapped to clearly represent the added voxels after closing operation.

3.3.2 Data Enhancement

Volumetric poses have defects that significantly reduce the recognition rate. Before extracting pose histograms, some techniques are used to enhance volumetric data to eliminate reconstruction defects. In our experiments, we perform morphological closing on volumetric data using sphere structural element with radius 2. The closing operator can close up internal holes corresponding to missing voxels. Data enhancement process is shown in Figure 3.3.

3.3.3 Pose Representation

During our experiments, we form the set of kernels with sizes [1x5], [1x10] and [1x20] where [n x m] means a cylinder with radius n and length m. Then, we rotate each kernel to obtain a set of oriented cylinders. These kernels are constructed with 30° apart. The number of kernels per size is 31.

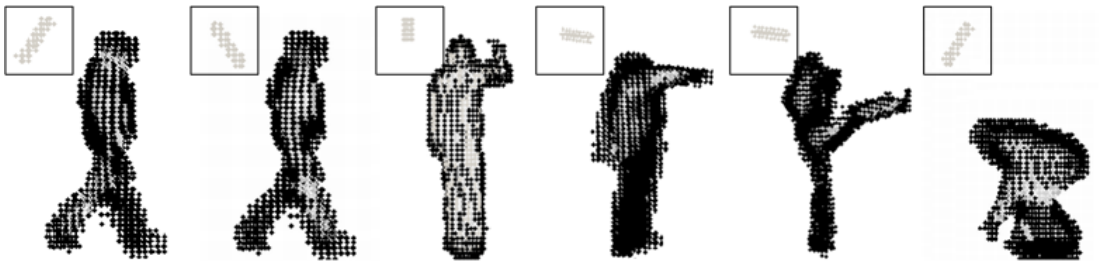


Figure 3.4: Kernel search results for five classes of 3D key poses: From left to right, poses are walk, wave, punch, kick and pick up. Figure represents poses from an arbitrary view. Gray level voxels are the high response regions for the corresponding kernel that is also drawn on the top left corner of each pose. Lower response regions below 0.8 are not counted. For walking pose, we represent search responses for two kernels in order to show different responses on the same pose. Through the experiments, we construct kernels in various sizes. For example, as shown in the third example a small size kernel is successful in finding upper arm of a wave pose. Note that, while using various length kernels, the radius size is fixed to 1 giving 3 voxels width.

After forming kernels, each one of them is searched over pose volume resulting in scores as kernel responses (see Figure 3.4) . Then, we divide voxel grid into 3 sub-volumes. The number of high response voxels to these kernels at each sub-volume are stored in a histogram of oriented cylinders. After forming histograms, we concatenate them to be used for pose inference. Please note that, each histogram is normalized prior to concatenation into a single pose vector.

3.3.4 Pose Recognition Results

There are many configurations of body parts that reveal different body poses. Among various kinds of poses, some are more discriminative known as key poses. In the following, we measure the performance of the proposed pose representation to classify key poses. We use two methods to evaluate the descriptor performance.

Table 3.1: NN-based classification results: 3 kernel sets with sizes [1x5], [1x10] and [1x20] are constructed with 30° apart. 1x5 means that a voxel grid inside a cylinder with radius 1 and length 5. The fist column lists the names of the selected key poses. The third column gives the performances when all kernels are used. The other columns gives the stand alone performances of each kernel with different sizes.

Poses	Accuracy all	Accuracy [1x5]	Accuracy [1x10]	Accuracy [1x20]
walk	96.97	93.94	96.97	96.97
wave	90.91	90.91	93.94	81.82
punch	63.64	48.48	69.70	72.73
kick	87.88	84.85	84.85	75.76
pick	96.97	90.91	96.97	93.94

The simplest method that we use is the nearest neighbor (NN) based classification. We use the Euclidean distance to find the best matched pose. The stand alone and complete performances of various sized kernels are shown in Table 3.1.

Nearest neighbor based classification requires a search over the whole dataset. Another method is multi-class Support Vector Machine (SVM) classification. SVM classifiers are formed using RBF kernel and trained using 3 actors. We evaluate the performances for 5 action classes. The results are presented in Table 3.2.

The results show that NN-based method is better than SVM based method. While taking more computation time, since all the examples in the data set are searched it is more likely to find a closer example with the NN-based method. It is likely that by increasing the number of training examples SVM-based method will be comparable to NN-based method, but the results are still acceptable with a very few examples .

Table 3.2: SVM-based classification results: 3 kernel sets with sizes [1x5], [1x10] and [1x20] are constructed with 30° apart.

Poses	Accuracy all	Accuracy [1x5]	Accuracy [1x10]	Accuracy [1x20]
walk	100	87.50	91.67	95.83
wave	91.67	54.17	95.83	79.17
punch	66.67	66.67	66.67	45.83
kick	100	100	100	95.83
pick	95.83	95.83	95.83	95.83

3.3.5 Action Recognition Results

The proposed pose descriptor is also evaluated for action recognition. We select the same set of action classes and test the same kernel configurations used during pose recognition experiments. We evaluate two methods to classify actions. First, we perform action matching by DTW. DTW gives highest performance for the combination of [1x5] and [1x10] kernels. The confusion matrix can be found in Figure 3.5.

The second method used for action recognition is HMM. Actions performed by 3 actors in 3 different orientations are used for training and the remaining dataset is used for testing. We quantize training actions using k-means clustering algorithm into 80 pose-words and construct HMM models per action class using 3 states. The recognition performances over 5 classes are shown in Table 3.3.

In our experiments, DTW gives highest recognition rate for 5 classes of actions using kernels with size [1x5] and [1x10]. HMM gives the highest accuracy using all kernels. On the other hand, DTW-based action classification requires one-to-one comparison in order to find most similar action. However, HMM only requires a trained model with a set of action samples. Therefore, it has a lower running time.

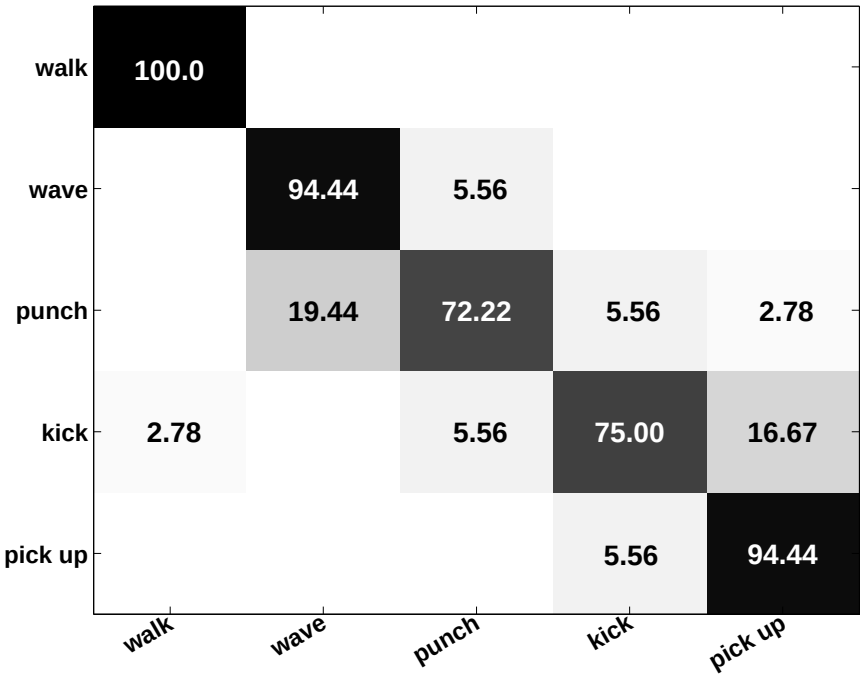


Figure 3.5: DTW-based classification results: experiments are done over 5 classes that are performed by 12 actors in 3 different viewpoint. This is the confusion matrix when [1x5] and [1x10] kernels are used.

3.4 Conclusion and Discussion

In this study, we investigate the importance of compact pose representations in a activity recognition framework and we propose a new representation in the form of distribution of oriented cylinders over volumes. The representation is for 3D human poses from multiple cameras with enough overlapping views for reconstruction.

The proposed descriptor is based on searching cylinders. It detects body parts that can change their orientations during the activity. Therefore, it is suitable for highly salient activity poses with discriminative changes in configuration. During experiments, we select 5 activity classes which have discriminative key poses.

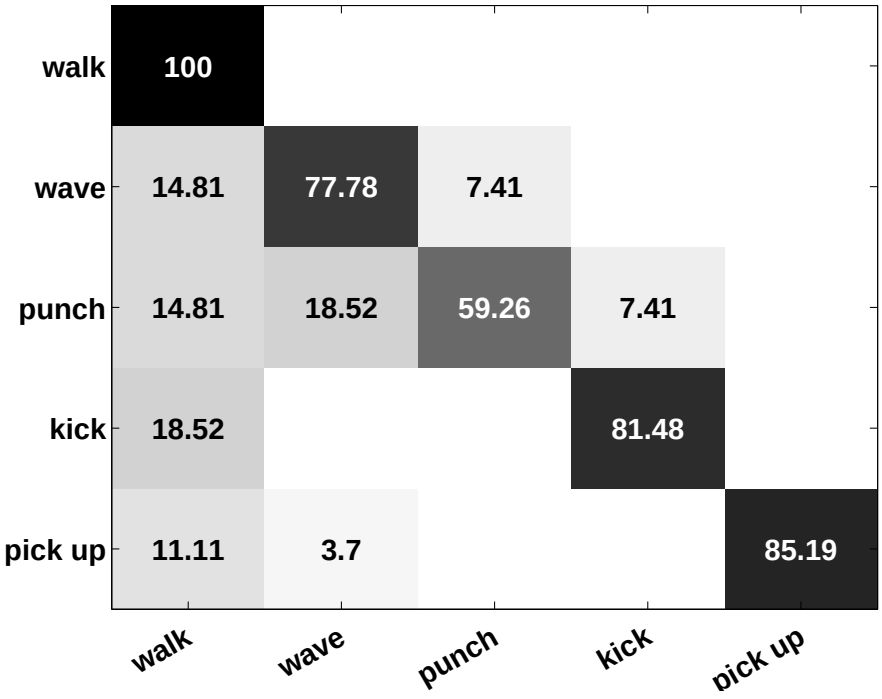


Figure 3.6: HMM-based classification results: experiments are done over 5 classes that are performed by 12 actors in 3 different viewpoint. 3 actors are used to train HMM models and remaining used for testing. This is the recognition performance when all kernel types are used.

The volumetric data used for experiments is sparse and it has defects after reconstruction. This results in low scores on some body limbs during search. The recognition results will be better when we use dense and high resolution data.

Another important point is about scaling factor. Activities are performed by different actors, thus volumetric poses vary in size. In this study, we do not scale the volumetric poses. They are used as in the original dataset. Small cylinders fire in everywhere on the pose volume. The distribution will be high for all small kernel types in all scales of a pose. Histogram normalization solves this problem without needing a volume scaling.

We observe that, the cylinders with different lengths return more discriminative results than cylinders with different radius sizes. As a result we preserve the

Table 3.3: HMM and DTW classification results respectively: experiments are done over 5 classes that are performed by 12 actors in 3 different viewpoint. The first one is the recognition rates when all kernel types are used. The second one is the recognition rates when [1x5] and [1x10] kernels are used.

Poses	DTW all	DTW [1x5][1x10]	HMM all	HMM [1x5][1x10]
walk	97.22	100	100	100
wave	94.44	94.44	77.78	62.96
punch	69.44	72.22	59.26	70.37
kick	66.67	75	81.48	62.96
pick	91.67	94.44	85.19	74.07

radius of cylinders as small as possible and form kernels with various lengths.

In this study, we do not apply any strategy to provide pose alignment. In fact, an action can be performed in any orientation by an actor. This shifts the values of histogram bins that are holding scores of kernels build by vertical axis rotation. However, we still obtain high pose retrieval results both by using NN-based classification and SVM-based classification. This shows that the training samples in different orientations can handle the changes in viewpoint. In the future, PCA-based alignment can also be applied to volumetric data prior to histogram extraction.

Chapter 4

View-indepedent Volume Representation Using Circular Features

Poses are important for understanding human activities and the strength of the pose representation affects the overall performance of the action recognition system. Based on this idea, we present a new view-independent representation for human poses. Assuming that the data is initially provided in the form of volumetric data, the volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers are then used to generate a pose descriptor. The pose descriptors of all frames in an action sequence are further combined to generate corresponding motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier. Experiments performed on the benchmark IXMAS multi-view dataset demonstrate that the performance of our method is comparable to the other methods in the literature.

The organization of the chapter is as follows. We present the view-independent pose representation in Section 4.1, the motion representation for actions using

proposed pose features in Section 4.2 and our method for action recognition in Section 4.3. Experimental results on a benchmark dataset are provided in Section 4.4 and results are compared with other studies in Section 4.5. Finally, we summarize our main contributions and we present future plans in Section 4.6.

4.1 Pose Representation

In this study, actions are considered as 3D poses that change over time. We emphasize the importance of pose information since in some cases a key pose can reveal enough information for classifying actions [51, 8, 34]. In the next section, we introduce motion features as changes in consecutive poses, necessary for learning action dynamics.

Volumetric data reveals important cues for understanding the shape of a human pose, but representing an action based on volumetric data is costly and can easily be affected by noise. Our proposed pose representation is effective in keeping the important characteristics of the poses while being efficient with the encodings used. The representation is independent of the view and translation and does not require the poses to be aligned or the actions to be performed in specified orientations and positions.

In the following, we first describe our method to encode the volumetric data, then we present the features used for representing the pose information.

4.1.1 Encoding the Volumetric Data Using Layers of Circles

Human body parts have cylindrical structures, and therefore the intersection of a pose with a horizontal plane can be modeled as a set of ellipses that can be further simplified by circles. We base our approach on this observation, and describe 3D poses using a set of features extracted from circles fitted to projections

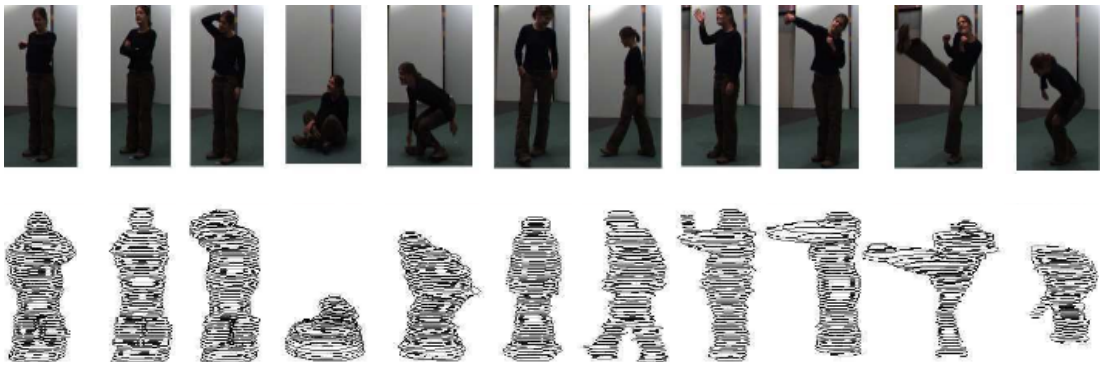


Figure 4.1: Key poses from some actions categories: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick and pick up. The 3D volumetric poses are divided into into horizontal layers. The intersection of body segments with each layer are then coded with enclosing circles.

of the volumetric data on horizontal planes. As shown in Figure 4.1, circular representation preserves the pose information while significantly reducing the data to be processed. In Section 4.4, we compare the performances when ellipses are used for representation rather than circles.

We assume that 3D poses are provided as visual hulls in the form of voxel grids constructed by multi-camera acquisition. For our representation, a voxel grid is divided into horizontal layers perpendicular to the vertical axis and turns into a collection of layers $l = 1, \ldots, n$ where n is the number of layers on the vertical axis. For instance, for a 64x64x64 voxel grid we obtain 64 layers.

Initially, each layer contains a set of pixels that is the projections of voxels on it. Since visual hull reconstruction may result in noise and defects that should be eliminated prior to feature extraction, we first evaluate each layer as a binary image consisting of pixels. Then we apply morphological closing using a disk structural element with a radius of 2 to close up internal holes and reduce noise in the data. We find the connected components (CCs) in each layer by looking at the 8-neighbors. Finally, a minimum bounding circle is fitted to each CC. Figure 4.2 illustrates the entire process on a sample pose.

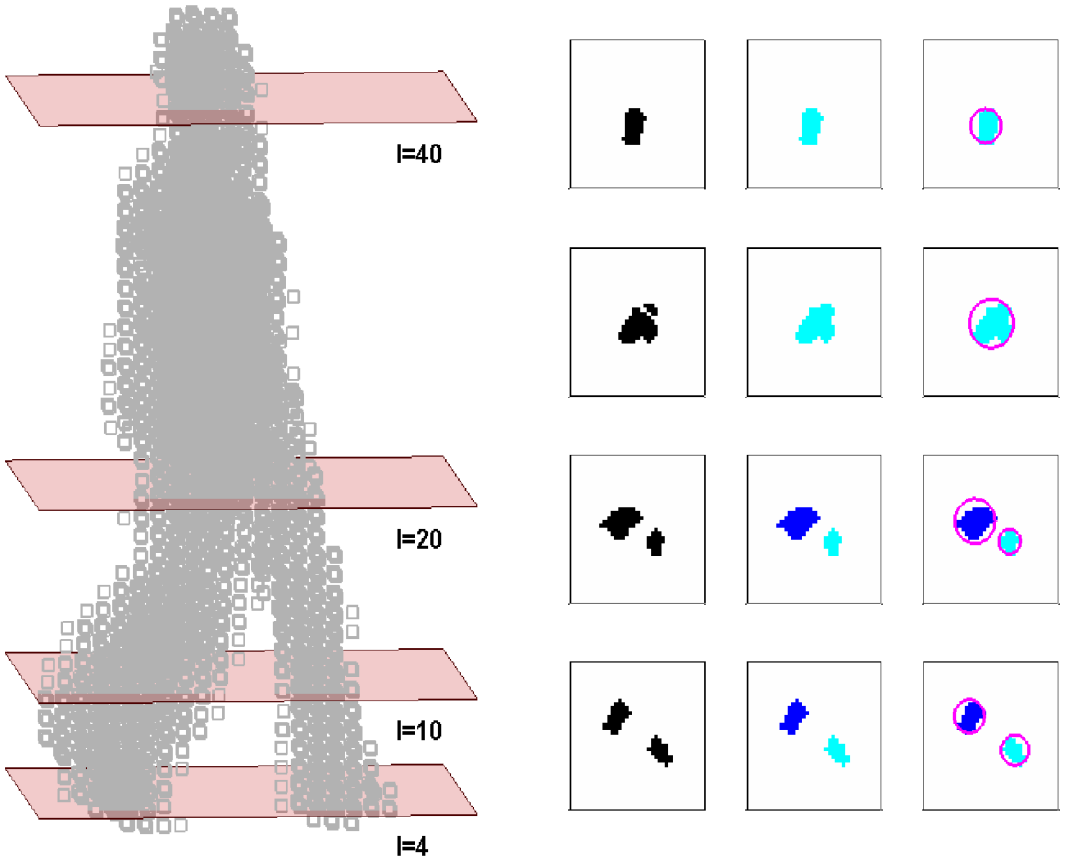


Figure 4.2: Representation of volumetric data as a collection of circles in each layer. From left to right: voxel grid of a walking pose and four sample layers (at l=4, l=10, l=20, l=40), image in each layer instance, connected components (CCs) over the enhanced image, and fitted circles.

4.1.2 Circle-Based Pose Representation

The proposed circular representation provides important local cues about the body configuration: the number of body parts passing through that layer, how much they spread over that layer and how far they are from each other. We utilize the following circular features to model these cues (see Figure 4.3). The proposed features allow a simple and easy way to provide a view-independent representation.

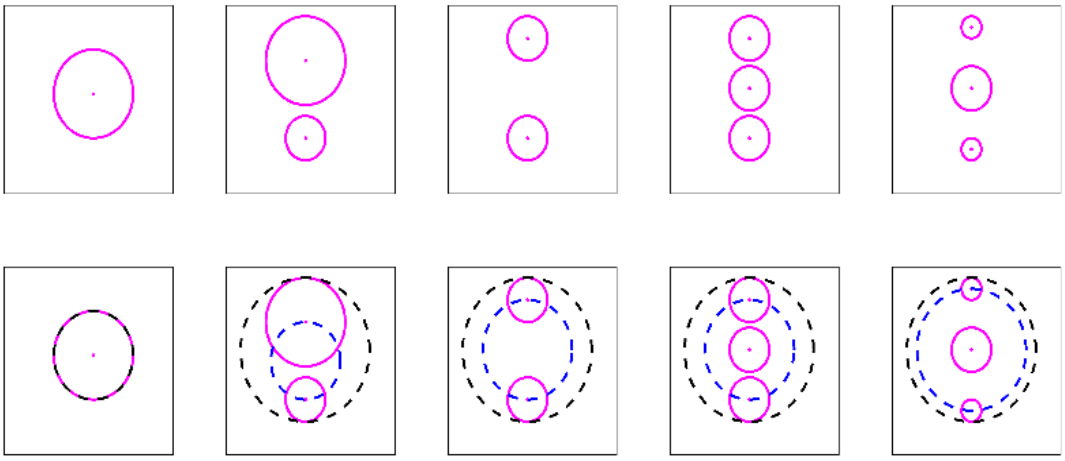


Figure 4.3: Proposed layer based circular features computed for example cases (best viewed in color): Given the fitted circles in the top row, we show proposed circular features in the bottom row. For all examples, the number of fitted circles in each layer corresponds to the number of circles feature. The outer circle (black) is the minimum bounding circle, which includes all circles, and the inner circle (blue) is the circle bounding the centers of all the circles. In the first example, there is only one fitted circle, therefore the area of the outer circle is equal to the fitted circle, and there is no inner circle. In the second and third examples, the number of circles and the areas of the outer circles are the same. However, the areas of the inner circles are different because of the distance between the fitted circles. In the fourth and fifth examples, we present cases that include three body parts (number of circles). If we compare the third and fourth cases, we observe that the areas of the outer and inner circles are the same, while the number of circles are different.

4.1.2.1 Number of circles

The articulated structure of the body creates a different number of intersections in each layer. For example, the layer corresponding to the head is likely to have a single circle, and a layer corresponding to the legs is likely to have two circles. Therefore, as our first feature, we examine the number of circles to find the number of body parts intersecting with a layer.

In this representation, a pose is described by a vector $\mathbf{c}$,

$$\mathbf{c} = [c_1, \ldots, c_n]^T \tag{4.1}$$

where each c_l, $1 \leq l \leq n$, is the number of circles extracted from layer l.

4.1.2.2 Area of outer circle

The maximum area covering all body parts passing through a layer is another important local cue to understand the body configuration in this layer, even in the case of noise. For example, the maximum area at the level of the legs can provide information such as whether the legs are open or closed.

To include all circles in that layer, the maximum area that covers all body parts in a layer is found by fitting a minimum bounding circle. We refer to this minimum bounding circle as the *outer circle*.

In this representation, a pose is described by a vector $\mathbf{o}$,

$$\mathbf{o} = [o_1, \ldots, o_n]^T \tag{4.2}$$

where each o_l, $1 \leq l \leq n$, is the area of the outer circle in layer l.

4.1.2.3 Area of inner circle

The third feature encodes spaces in a pose volume and is represented by the relative distances between body parts. For this purpose, an inner circle bounding the centers of all circles is found.

In this representation, a pose is described by a vector $\mathbf{i}$,

$$\mathbf{i} = [i_1, \ldots, i_n]^T \tag{4.3}$$

where each i_l, $1 \leq l \leq n$, is the area of the inner circle at layer l.

Note that rather than using the area of outer and inner circles, the radii could be used. The choice for area is made in order to amplify the differences. It is also empirically observed that area is better than radius.

For all feature vectors $\mathbf{c}$, $\mathbf{o}$, and $\mathbf{i}$, encoding starts in layer $l = 1$ corresponding to the bottom of the voxel grid, and a value of zero is stored for any layer with

no pixels. This kind of encoding retains the order of the extracted features and the body location with respect to the floor (bottom of voxel grid).

4.1.3 Discussion on the Proposed Pose Representation

The proposed circular features have several advantages. First, these features store discriminative shape information about the characteristics of an actual human pose. In most cases, poses are identified by the maximum extension of the body parts corresponding to the silhouette contours in the 2D scenarios. However, this information is usually lost in single-camera systems depending on relative actor orientation. When we have volumetric data, we can easily approximate this information by the bounding circles. Moreover, fitting free form circles helps us to solve pose ambiguities related to the actors' style. It is robust to handle changes that can occur during the performance of the same pose by different actors. More features per layer can be introduced with invariance properties.

Second, our pose representation significantly reduces the 3D data encoding while increasing the efficiency and preserving the effectiveness of the system. A circle can be represented with only two parameters, i.e. center point and radius, and the average number of circles per layer is approximately 2. This significant reduction in the number of voxels is further improved with the introduction of the vectors $\mathbf{c}$, $\mathbf{o}$, and $\mathbf{i}$. In our representation, all layers are in free form and there is not single reference axis. Each circle are evaluated individually. Doing so, features handle variations in action category. For instance, volumetric shape of the performed action highly effected by actor style. This is mostly true for actions like punch, point or kick. Each actor performs in an individual style and sometimes central body axis orients differently. By fitting free form circles independent from a fix bounding body cylinder, we somehow try to solve such problems. Otherwise, we would need more samples. Although the encoding is lossy, it is enough to identify poses and further actions when combined with the temporal variations.

Most importantly, the proposed features are robust for viewing changes. In

a multi-camera system, an actor should be free to perform an action in any orientation and position. The system should thus be able to identify similar poses even in the case of viewpoint variations, that is, in the case of rotating a pose through the vertical axis. As the proposed circular features are extracted in each layer perpendicular to the vertical axis. Any change in the orientation or position of the pose will result in the translation of the extracted circles but will not affect their areas (also radius), counts, or relative positions. Therefore, the introduced three feature vectors **c**, **o**, and **i** hold pose information independently from the view and the robust to small changes in the poses. Figure 4.4 shows an example in which the descriptors are very similar while the action is performed by two different actors and from two different viewpoints.

4.2 Motion Representation

In the previous section, we present our representation to encode a 3D pose in a single action frame using feature vectors **c**, **o**, and **i**. In the following, we use the proposed pose vectors to encode human actions as motion matrices formed by concatenating pose descriptors in all action frames. Then we introduce additional motion features to measure variations in the body and temporal domains.

4.2.1 Motion Matrices

Let $\mathbf{p^t} = [p_1^t, \ldots, p_n^t]^T$ be the pose vector at frame t, where n is the number of layers, and p_l^t is any representation of the fitted circles in layer l. That is, $\mathbf{p^t}$ is one of the vectors **c**, **o**, or **i**. We define a matrix $\mathbf{P}$ as a collection of vectors $\mathbf{p^t}$, $\mathbf{P} = [\mathbf{p^1}, \ldots, \mathbf{p^m}]$, where m is the number of frames in an action period. Matrix $\mathbf{P}$ holds all poses during an action and can be visualized as an image of size $n \times m$, where n is the number of rows, and m is the number of columns (see Figure 4.5).

A generic motion descriptor should be scale invariant in order to be independent from actor, gender etc. In this study, rather than normalizing and scaling at the pose level, we apply them to motion matrices containing consecutive frames.

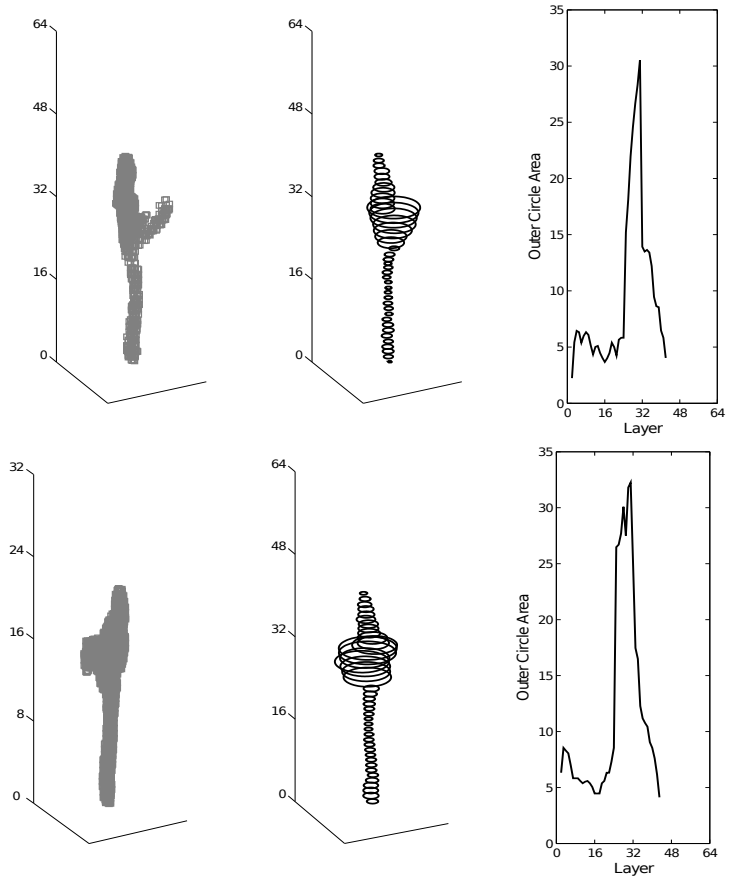


Figure 4.4: Evaluation of features in terms of view invariance on two kick poses performed by two different actors. The left column is the volumetric data for a pose from the kick action (64x64x64 voxel grid). The middle column is the representation of the pose as a collection of bounding circles. The right column is the plot of the area of the outer circle's feature vector showing the bounding circle's area per layer. Note that the feature vectors of the same pose from different viewpoints are very similar.

In this way, we obtain a better understanding of body variation in terms of width and height through time.

Let $\mathbf{P}$ be any of the motion matrices described above with entries p_l^t, $l = 1, \ldots, n$ and $t = 1, \ldots, m$. First we obtain a normalized matrix entry pn_l^t as follows:

$$pn_l^t = \frac{p_l^t - min(\mathbf{P})}{max(\mathbf{P}) - min(\mathbf{P})} \tag{4.4}$$

where $min(\mathbf{P})$ and $max(\mathbf{P})$ are minimum and maximum values of matrix $\mathbf{P}$ respectively.

Then, we resize the $\mathbf{P}$ matrix by bilinear interpolation to an average size trained over samples. Resizing eliminates frame count and layer count differences among matrices, while preserving motion patterns.

Since we work on voxel data, the valuable information will be the voxel occupancy in a layer and its variation in amount and direction over time (corresponding to the velocity of the parts). Using matrix $\mathbf{P}$, a specific action can be characterized with the changes in the circles over the body or over time. A single column stores information about the amount of change in the shape of the pose (body) at a given time t. Similarly, a single row stores information about the amount of change through the entire action period (time) for a given layer l.

In order to extract this information, we evaluate the differences on the features of the circles in consecutive layers and in consecutive frames. We apply a vertical differentiation operator G_b, as an approximation to vertical derivatives, over matrix $\mathbf{P}$ to find the change over the body, and refer to the resulting matrix as $\mathbf{P_b}$. Similarly, we convolve matrix $\mathbf{P}$ with a horizontal differentiation operator, G_t, to find the change over time, and refer to the resulting matrix as $\mathbf{P}_t$.

The two new matrices $\mathbf{P}_b$ and $\mathbf{P}_t$, together with the original matrix $\mathbf{P}$, are then used as the motion descriptors that encode variations of the circles over the body and time to be used in recognizing actions.

Remember that, $\mathbf{P}$ can be created using any of the pose descriptors $\mathbf{c}$, $\mathbf{o}$, or $\mathbf{i}$. Therefore, a motion descriptor consists of the following set of matrices, which we refer to as *Motion Set*:

$$\mathcal{M} = \{\mathbf{C}, \mathbf{C}_t, \mathbf{C}_b, \mathbf{O}, \mathbf{O}_t, \mathbf{O}_b, \mathbf{I}, \mathbf{I}_t, \mathbf{I}_b\} \tag{4.5}$$

where $\mathbf{C} = [\mathbf{c^1}, \ldots, \mathbf{c^m}]$ is the matrix generated using the number of circles feature, $\mathbf{O} = [\mathbf{o^1}, \ldots, \mathbf{o^m}]$ is the matrix generated using the area of the outer circle feature, $\mathbf{I} = [\mathbf{i^1}, \ldots, \mathbf{i^m}]$ is the matrix generated using the area of inner circle

feature, and the others are the matrices obtained by applying vertical and horizontal differentiation operators over these matrices. The summary of the motion descriptors is given in Table 4.1.

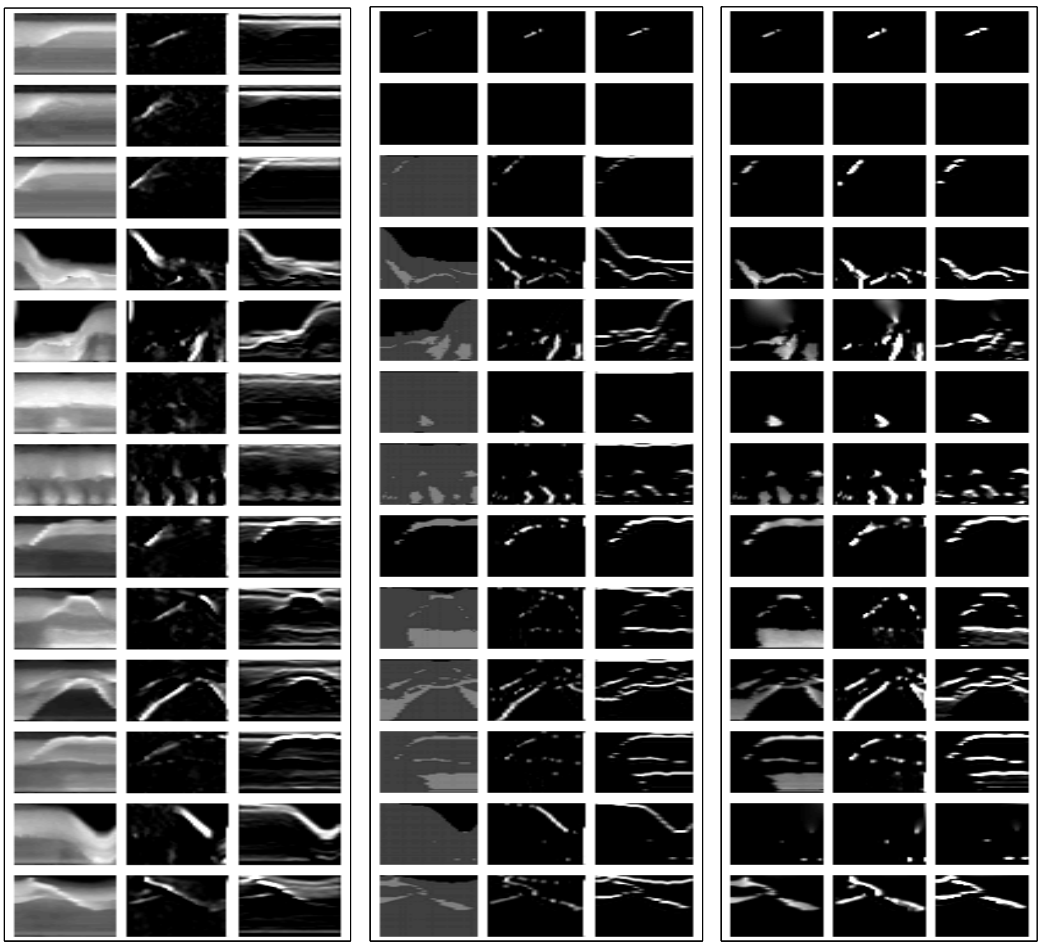


Figure 4.5: *Motion Set* for 13 actions. From top to bottom: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up and throw. From left to right: $\mathbf{O}$, $\mathbf{O}_t$, $\mathbf{O}_b$, $\mathbf{C}$, $\mathbf{C}_t$, $\mathbf{C}_b$, $\mathbf{I}$, $\mathbf{I}_t$, $\mathbf{I}_b$.

Table 4.1: *Motion Set*: the motion descriptor as a set of motion matrices

Number of Circles		
$\mathbf{C} = [\mathbf{c^1} \dots \mathbf{c^m}]$	$\mathbf{C}_t = G_t \circ \mathbf{C}$	variation of circle count in a fixed layer through time
	$\mathbf{C}_b = G_b \circ \mathbf{C}$	variation of circle count through body
Area of Outer Circle		
$\mathbf{O} = [\mathbf{o^1} \dots \mathbf{o^m}]$	$\mathbf{O}_t = G_t \circ \mathbf{O}$	variation of outer circle in a fixed layer through time
	$\mathbf{O}_b = G_b \circ \mathbf{O}$	variation of outer circle through body
Area of Inner Circle		
$\mathbf{I} = [\mathbf{i^1} \dots \mathbf{i^m}]$	$\mathbf{I}_t = G_t \circ \mathbf{I}$	variation of inner circle in a fixed layer through time
	$\mathbf{I}_b = G_b \circ \mathbf{I}$	variation of inner circle through body

4.2.2 Implementation Details for Motion Matrices

We perform some operations over matrix $\mathbf{P}$ for pruning and noise removal prior to the normalization and formation of new matrices, $\mathbf{P}_b$ and $\mathbf{P}_t$.

After pose vector concatenation, matrix $\mathbf{P}$ is in the size of $64 \times m$, where 64 is the height of the voxel grid and m is the action period. However, some layers (rows) are not occupied by the body for the entire action. Therefore, first operation is to clean up the motion matrices from rows that are not used through out the action period. After pruning, we obtain a $n \times m$ matrix, where n depends on the maximum height of the actor over all action frames. Similarly, action periods may vary from one actor to another, resulting in a different number of frames.

The provided volumetric data is obtained by shape from the silhouette technique. However, extracted silhouettes have some defects, affecting the volumes and our circular representation. After pruning, we perform interpolation over motion matrices $\mathbf{P}$ to enhance the circular representation and to fill gaps corresponding to missing circles. In some layers, missing voxels result in smaller circle sizes. We apply a spatio-temporal smoothing over the motion matrices to tackle this problem. If a circle in a layer has a smaller value than the average of its 4-neighbors then it is replaced with the average value.

4.3 Action Recognition

With the introduction of the motion descriptors in the previous section, action recognition is turned into a problem of matching the *Motion Set* ($\mathcal{M}$) of the actions.

We present a two-level action classification scheme to improve the classification accuracy. In the first level, actions are classified in terms of global motion information by a simple approach that divides actions into two groups: upper-body actions such as punch or wave and full-body actions such as walk or kick. Then, in the second level, actions are classified into basic classes based on motion matrix similarities measured by Earth Mover's distance (EMD) or Euclidean distance.

4.3.1 Full-Body vs. Upper-Body Classification

A simple way to classify actions is to evaluate global motion information such as detecting whether the motion occurs in the upper or lower part of the human body. We observe that lower-body motions cause changes in the entire body configuration. We propose an action classification method that decides whether a test instance is a member of the full or upper body *Motion Sets*.

We represent each action with a feature that reveals the amount of motion in

each layer during an action period. For this purpose, we calculate v_l, the variation of the outer circle area, for all layers of $n \times m$ matrix $\mathbf{O}$, where n is the number of body layers, m is the length of the action period. Then, we train a binary Support Vector Machine (SVM) classifier using the defined feature vector, $\mathbf{v} = v_1 \dots v_n$, to learn upper-body and lower-body actions. SVM classifier is formed using RBF kernel.

As will be discussed later, the proposed two-level classification method increases the recognition performance. Moreover, the total running time decreases a considerable amount, parallel to the descending pairs to be compared.

4.3.2 Matching Actions

After the first-level classification as an upper-body or full-body action, we perform the second-level classification. We calculate a *Motion Set*, $\mathcal{M}$, for each action and use it for comparison with other actions. For this purpose, we use the nearest neighbor classification.

Let us assume $\mathcal{M}^i$ and $\mathcal{M}^j$ are *Motion Sets* containing motion matrices for two action instances. Let $\mathbf{P}^i$ be one $n \times m$ matrix from set $\mathcal{M}^i$, where n is the layer count and m is the time. During the comparison of the two actions, we compute a global dissimilarity $D(\mathbf{P}^i, \mathbf{P}^j)$. We use the same methodology to compare all the matrices in the *Motion Sets*. Note that if a motion belongs to the upper-body set, we perform a comparison only for the upper body layers, $\frac{n}{2} \dots n$.

4.3.3 Distance Metrics

There are many distance metrics, such as L_p norms, Earth Mover's Distance [50], Diffusion Distance[29] and χ^2. L_p norms and χ^2 are affected by small changes in bin locations, whereas, EMD and Diffusion Distance alleviate shift effects in bin locations and in some cases outperform other distance metrics [30]. Diffusion distance compares histogram based local descriptors. Modeling the histogram

differences as temperature field, it performs the diffusion process on that field [29].

We test two different distance measures. One is well-known L_2 norm and the other one is the shift invariant EMD. Even when we resize motion matrices to the same size, the starting frames of action sequences are always different. Since the starting point can change, we thought we need a shift invariant metric to solve the problem. Although EMD is robust for shift effects, we find that L_2 gives better performance than EMD in some cases. This is most probably due to sift invariance of EMD in the spatial (body) domain.

Classical EMD algorithms have high $O(N^3 \log N)$ computational complexity. In this study, we apply EMD-L_1, which is a $O(N^2)$ EMD algorithm based on L_1 ground distance [30]. Furthermore, EMD-L_1 algorithm does not require normalization to the unit mass over the arrays. This increase its feasibility over our motion matrices, which are only scale normalized.

4.4 Experimental Results

4.4.1 Dataset

We test our method on the publicly available IXMAS dataset [61], which is a benchmark dataset of various actions taken with multiple cameras. There are 13 actions in this dataset: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up and throw. Each action is performed three times with free orientation and position by 12 different actors. Multi-view videos are recorded by five synchronized and calibrated cameras (see Figure 2.1).

Since our focus is on pose and motion representation, we do not deal with reconstruction and use the available volumetric data in the form of a 64x64x64 voxel grid. For motion segmentation, we use the segmentations provided by Weinland et al. [61].

Table 4.2: Minimum, maximum and average sizes of all primitive actions after pruning.

	Layer Count (n)	Frame Count (m)
min	41	13
max	58	99
avg	48	40

4.4.2 Evaluation of Pose Representation

In the first part of the experiments, we evaluate the performance of proposed pose representation. For this purpose, we construct a key-pose collection obtained from the IXMAS action dataset. Among the dataset actions, sit down, stand up and pick up have identical key-poses, but in different order. Similarly, punch, point and throw actions can be grouped as identical in terms of key-poses. We construct a pose collection of nine categories consisting of 324 key-poses in total; key-poses are selected from every action video of the dataset (see Table 4.3).

We measure pose retrieval performance by a simple nearest neighbor classifier using L_2 norm. EMD distance is not used for pose matching, since shift invariance is not expected in body domain. We use $\mathbf{o}$ and $\mathbf{o_b}$ poses vectors, with variation through the body, as our feature. Experiments show that the $\mathbf{o_b}$ feature outperforms $\mathbf{o}$. Accuracies and class performances for the $\mathbf{o_b}$ feature are shown in Table 4.3. The average pose recognition accuracy is 87.66% when $\mathbf{o_b}$ is circle area.

While computing the $\mathbf{o}$ and $\mathbf{o_b}$ features, we follow a similar procedure as for motion matrices. First we obtain a pruned circular representation to remove unoccupied layers at the top and bottom of the visual hulls, then normalize it with the maximum value. Then, we resize all pose vectors to a fixed size of 48 using bilinear interpolation. Finally, we compute the $\mathbf{o_b}$ feature by applying a $[-1\ 0\ 1]$ filter over $\mathbf{o}$.

In addition to circle fitting, we perform experiments for a representation based

Table 4.3: Leave-one-out nearest neighbor classification results for pose representation evaluated with $\mathbf{o_b}$. Vectors are compared using the L_2 norm.

	Acc (%)	check watch	cross arms	scratch head	sit	walk	wave	point punch	kick	pick
C_{area}	87.66	80.56	88.89	94.44	100.00	100.00	77.78	80.56	77.78	88.89
E_{area}	86.2	86.11	100.00	77.78	100.00	97.22	66.67	63.89	83.33	100.00
E_{major}	79.94	61.11	63.89	77.78	100.00	100.00	75.00	83.33	72.22	86.11

on ellipses. We compute $\mathbf{o}$ vectors by ellipses rather than circles. For ellipses, we define three features that can be used instead of the circle area: ellipse area, ellipse major axis and ellipse minor axis. The results are illustrated in Figure 4.6.

Similarly, we test the performance of the circle radius for pose retrieval. Although performances of all features are very close to each other, circle area outperforms others. Moreover, using circle area rather than circle radius provides a slightly better performance. Here, the scale factor reveals the differences among circles.

In addition to fitting ellipses to the bounding area, $\mathbf{o}$, we also perform experiments for fitting ellipses to all CCs. Although ellipses are the correct representation for body projection over layers, sometimes they cannot identify the true orientation of the body part for some cases especially if we are working on small size blobs. We experiment on a set of walking poses having two CCs at lower layers. When we fit ellipses to CCs, Each can lead an arbitrary orientation (major axis orientation), irrelevant from body part orientation in the layer. Then we think the best features to be used for ellipses is the area or major axis for computing $\mathbf{o}$. But in this case, circles are better in performance. We also conducted some more experiments to measure if we can provide temporal variation using ellipse rotations. During experiments we try to measure rotation angle of

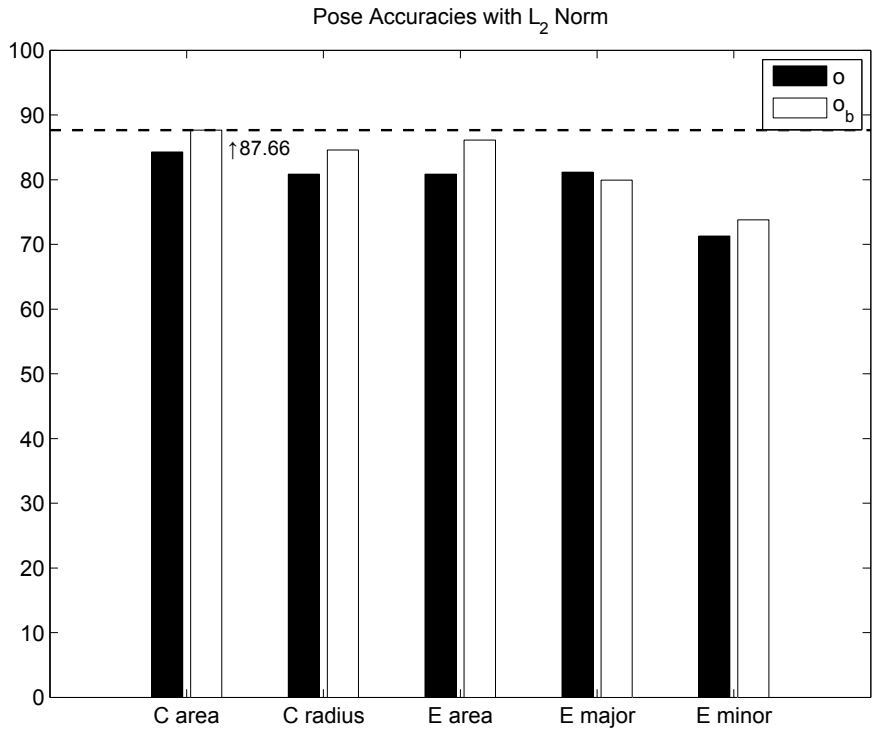


Figure 4.6: Leave-one-out classification results for pose samples. Poses are classified using the L_2 norm with features $\mathbf{o}$ and $\mathbf{o_b}$. Results show performances of the $\mathbf{o}$ feature when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis* and *Ellipse Minor Axis*, respectively.

ellipse major axis. However this requires alignment. Moreover, we looked for the relative angular variations. But this is quite noisy due to CC problems explained above.

4.4.3 Evaluation of Motion Representation and Action Recognition

Each action is represented with a *Motion Set*, $\mathcal{M}$, where motion descriptors $\mathbf{C}$, $\mathbf{O}$, $\mathbf{I}$ are obtained by concatenation of pose descriptors. After enhancement, all $n \times m$ main motion matrices $\mathbf{C}$, $\mathbf{O}$, and $\mathbf{I}$ are resized to a fixed size matrix by bilinear interpolation. For the IXMAS dataset, considering the average, minimum and

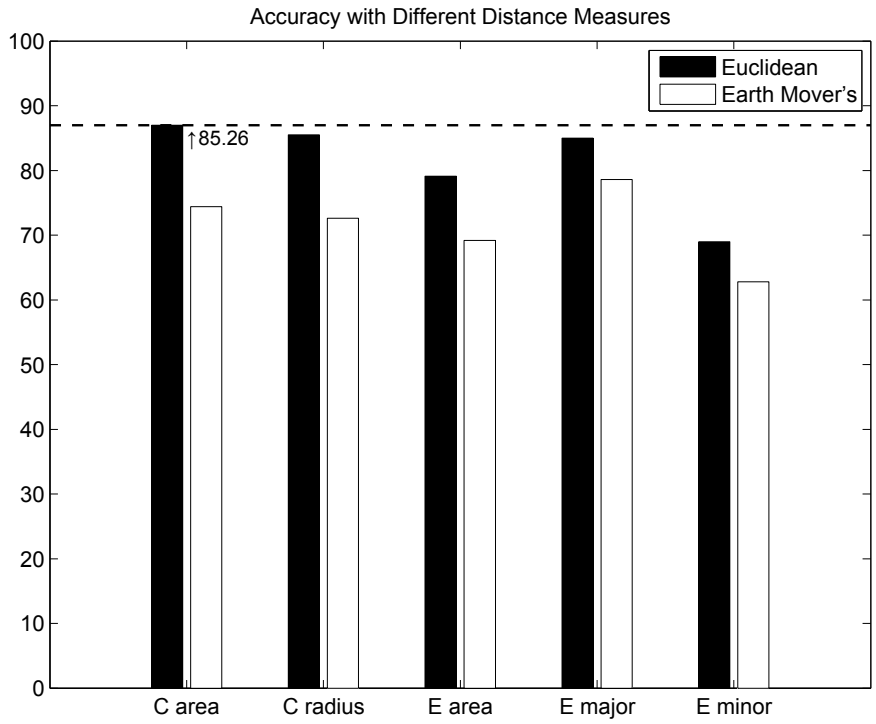


Figure 4.7: Leave-one-out classification results for action samples. Actions are classified with the **O** feature using L_2 and EMD. We show performances of the **O** feature when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis* and *Ellipse Minor Axis*, respectively.

maximum layer and frame sizes for actions (shown in Table 4.2), matrices are scaled into a fixed 48×40 size matrix. Other values are also experimented with, but only slight changes are observed.

Other matrices in $\mathcal{M}$, i.e. $\mathbf{C}_b$, $\mathbf{C}_t$, $\mathbf{O}_b$, $\mathbf{O}_t$, $\mathbf{I}_b$, and $\mathbf{I}_t$, are obtained by applying the *Sobel operator* on matrices **C**, **O**, **I** to measure the circular variations through the body and time.

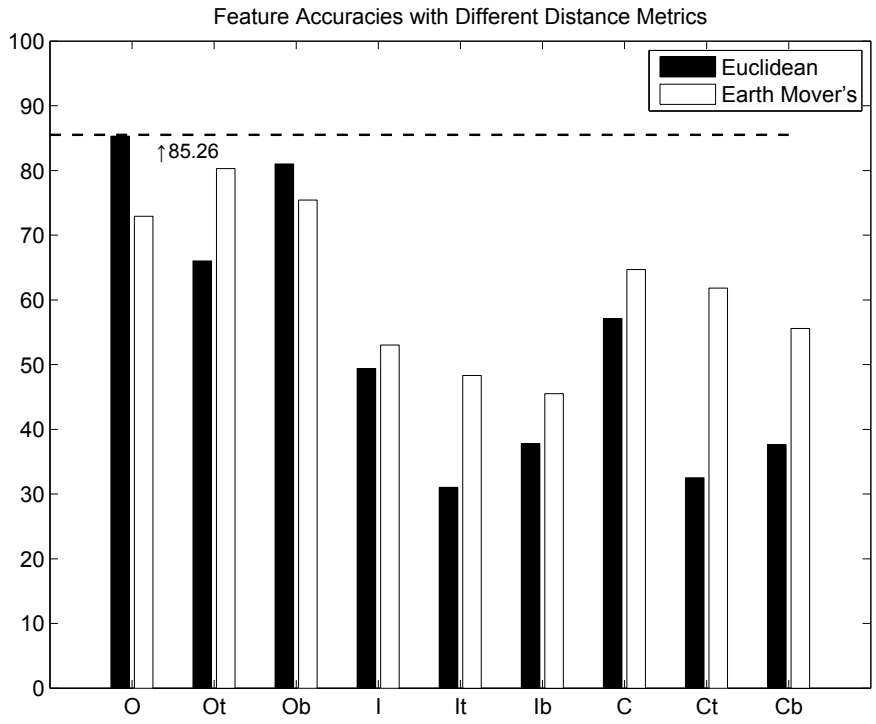


Figure 4.8: Leave-one-out classification results for action samples. Actions are classified with various features using L_2 and EMD. Results denote that **O** outperforms others with Euclidean Distance and $\mathbf{O_t}$ has the highest accuracy for Earth Mover's Distance.

4.4.3.1 Two-level classification rates

We perform a two-level action classification. In the first level, a simple binary SVM classifier is trained for full-body vs. upper-body classification. On the IXMAS dataset, we labeled check watch, cross arms, scratch head, wave, punch, point, and throw actions as upper-body actions, and sit down, get up, turn around, walk, kick and pick up as full-body actions. We selected three actors for each action for training, and use the rest of the actors for testing. We obtain a classification accuracy of 99.22%, which is almost perfect.

In the second level, actions are compared within their corresponding upper-body or full-body sets using the nearest neighbor method with different distance metrics, L_2 norm and EMD-L_1. We use the leave-one-out strategy. All samples

belonging to an actor are excluded, to be used as test samples, and the samples remaining are used as training samples. The accuracy results are obtained by averaging over all the actors.

4.4.3.2 Performance of stand-alone motion representation

In the first part of the experiments, we evaluate stand-alone recognition performances for each motion representation (see Figure 4.8). The results indicate that the accuracy of matrix $\mathbf{O}_t$ is higher than the accuracies of all other motion matrices when EMD is used as the distance measure. In EMD experiments, motion matrices $\mathbf{O}_t$ and $\mathbf{O}_b$ outperform the original matrix $\mathbf{O}$. However, this is not true for matrices $\mathbf{C}$ and $\mathbf{I}$. For Euclidean case, $\mathbf{O}$ outperforms the rest, with the highest 85.26% accuracy. But in this case, $\mathbf{O}_t$ has a lower performance than $\mathbf{O}$ and $\mathbf{O}_b$. Detailed accuracies are given in Table 4.4.

Table 4.4: Stand-alone recognition performances of each matrix in *Motion Set* for 13 actions and 12 actors. The motion matrices are scaled to 48×40. Results denote that $\mathbf{O}$ outperforms others with Euclidean Distance and $\mathbf{O_t}$ is the best one for Earth Mover's Distance.

Acc (%)	O	O_t	O_b	I	I_t	I_b	C	C_t	C_b
Euclidean	85.26	66.00	81.00	49.40	31.00	37.80	57.10	32.50	37.60
Earth Mover's	72.90	80.34	75.40	53.00	48.30	45.50	64.70	61.80	55.60

Similar to pose experiments, we also evaluate the behavior of the most discriminative feature in the cases of different structures fitted to the voxels. For this purpose, we show performances of the $\mathbf{O}$ feature in Figure 4.7 when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis* and *Ellipse Minor Axis*, respectively, with different distance metrics.

4.4.3.3 Performance of combined motion representation

In the second part of the experiments, we evaluate the recognition performance as a mixture of motion matrices to utilize different motion features at the same time. For this purpose, we combine two motion matrices from *Motion Set* using a linear weighting scheme. Stand-alone evaluation indicates that matrices $\mathbf{O}$ and $\mathbf{O}_t$ have the best performances for Euclidean and EMD, respectively. Therefore we pick them as the best features and add other features as pairwise combinations. We calculate the weighted sum of distances for two action sequences i and j as:

$$D(i,j) = \alpha\, D(\mathbf{O}^i, \mathbf{O}^j) + (1-\alpha)\, D(\mathbf{P}^i, \mathbf{P}^j) \tag{4.6}$$

where $\mathbf{P} \in \mathcal{M}$ and $\mathbf{P} \neq \mathbf{O}$ ($\mathbf{P} \neq \mathbf{O_t}$ for EMD).

As shown in Figure 4.9, $\mathbf{O}_t$ and $\mathbf{O}_b$ give the best performances with $\alpha = 0.8$, resulting in an accuracy **85.90%** for EMD measurement. On the other hand, $\mathbf{O}$ and $\mathbf{I}$ pair gives the highest accuracy of **86.97%** with Euclidean Distance and $\alpha = 0.9$ value.

To further incorporate all features, we select one feature matrix per feature type that has the highest accuracy among three. We pick either $\mathbf{P}$, $\mathbf{P}_t$ or $\mathbf{P}_b$ from each type. Based on the stand-alone performances, one matrix outperform others in the same feature type. This time, we calculate the weighted sum of distances with *alpha* and *beta* parameters. For Euclidean distance, we report a maximum accuracy of 88.63% with weights 0.7, 0.1 and 0.2 for the $\mathbf{O}$, $\mathbf{C}$ and $\mathbf{I}$ combination, respectively. For the EMD experiment, we report 85.26% with same weights for $\mathbf{O_t}$, $\mathbf{C}$ and $\mathbf{I}$, respectively.

Figure 4.10 shows the confusion matrices for the best configurations. In the experiments, we observed no important difference in the performance with the addition of new features, and for simplicity we prefer to combine only three features.

The recognitions of three actions, namely, wave, point and throw, have lower performances with respect to other action categories. The main problem the with

wave action is confusion with the scratch head action. In many wave sequences, even in the volumetric data, the periodicity of the wave action is rarely or never recognized. Some actors just move hand, while others move arms. From this observation, a low recognition rate is to be expected.

Similarly, the point and punch actions are confused with each other. Both actions contain similar poses, but a difference between punch and point is in the duration of the actions. In motion patterns, we observe high peak for punch actions. This peak can be a high value occurring in the same layer but with a short duration, or it can be an increase through the z axis (corresponding y axis in the motion matrix). After scaling, the distinctive pattern is preserved and peak variations among these two actions can be clearly observed. But again, depending on the actor' style, the duration is not consistent; some actors perform a punch action very slowly, and it looks like the point action. In such cases, it is likely that, the scaling of the motion matrices results in the similar representation of these two action categories.

Another low recognition performance is observed for the throw action. The dataset contains variants of the throw action, performed by different actors as throw from above and throw from below. Even in this situation, we achieve 80.56% accuracy for the throw action, with fewer training samples in the same category. Aside from this, throw is mainly confused with punch action, because of similar peak patterns related to actor style.

4.4.3.4 Single-level classification rates

In this part of the experiments, we evaluate the system's performance on single-level classification (i.e. classifying a full set of actions without classifying them into full-body vs. upper-body actions). As in the two-level case, $\mathbf{O}$ and $\mathbf{O}_t$ outperform others when we compare their stand-alone performances. Similarly, we compute the distances among action samples by first selecting one feature from each feature type, and then combining the weighted sum of three distances. For the Euclidean experiment, the $\mathbf{O}$ feature again gives the highest accuracy

82.69% when combined with **I** using weights 0.8, 0.2. For EMD, we report an accuracy of 82.05% by combining **O** and **I** with weights 0.7, 0.3. Please note that both experiments are done by combining three feature types. However, **C** does not increase the results. The confusion matrices of the best reported results are given in Figure 4.11.

We observe that adding a two-level classification improves the overall accuracy from 82.69% to 88.69%. Moreover, if execution time is crucial, this method decreases the running time by eliminating half of the comparisons between pairs for matching.

4.5 Comparison with Other Studies

The IXMAS dataset is introduced in [61], and used in several other studies. In this section, we compare our results with the other studies in the literature that use the same dataset. Remember that we use the full dataset, with 13 actions and 12 actors, and obtain a **88.63%** recognition performance.

In some studies [61, 59, 62], not the full set, but a subset consisting of 11 actions and 10 actors is used. In order to compare our method with those studies, we also test our method over 11 actions and 10 actors. With a similar evaluation of different configurations as in the full case, we obtain a 98.7% performance for the upper-body vs. full-body classification, and for action recognition in a two-level classification scheme the best reported performance, **90.91%**, is obtained for the **O**, **I** and **C** combination with weights 0.7, 0.1, 0.2, respectively using Euclidean Distance. Similarly, we report a 90.30% recognition accuracy for the EMD experiment, for an $\mathbf{O}_t$, **I** and **C** combination with weights 0.7, 0.2, 0.1 respectively. The confusion matrix for 11 actions, and 10 actors is shown in Figure 4.12.

Although a direct comparison is difficult, we arrange studies in terms of performances on the multi-view IXMAS dataset in Table 4.5. As can be seen from the results, our performance is comparable to the best result in the literature

Table 4.5: Comparison of our method with others tested on the IXMAS dataset.

Method	Accuracy (over 11 actions)	Accuracy (over 13 actions)
Weinland et al. [61]	93.33%	-
Our method	**90.91**%	**88.63**%
Liu et al. [32]	-	82.8%
Weinland et al. [59]	81.27%	-
Lv et al. [35]	-	80.6%
Yan et al. [62]	78.0%	-

and superior to the other studies in terms of the recognition rate. This is very promising considering the efficiency of the proposed encoding scheme.

Weinland et al. [61] report an accuracy of 93.33% over 11 actions. However, in their later work [59], they obtain 81.27% accuracy on the same dataset. This shows that 3D-based recognition is more robust than 2D to 3D matching. Lv et al.[35] use 2D matching. Note that, they test on the whole dataset, however they further add standing action, and divide throw action into two action categories as throw above and throw below. Liu and Shah[32] report the second-highest score over 13 actions. Their approach is based on multiple images features rather than 3D representations. However, it proposes an orderless feature set, which is difficult to use for pose estimation.

4.6 Conclusion and Discussion

The experimental results show that the descriptor is discriminative for pose and activity recognition. Good performances are achieved with the proposed volume matching approaches. This shows when we have a calibrated camera system, volumes perform well.

With simpler encoding scheme, it performs better than our previous pose descriptor based on bag-of-features. This also shows that view-independent features

are more robust.

Although we study on volumetric data in the form of a voxel grid, the idea of circular features can be easily adapted to other 3D representations, such as surface points. The proposed descriptor can also achieve a good performance for shape matching and retrieval of human-body-like shapes. Moreover, the same approach can be used to fit a skeleton to a 3D shape by connecting the centers of circles in each layer.

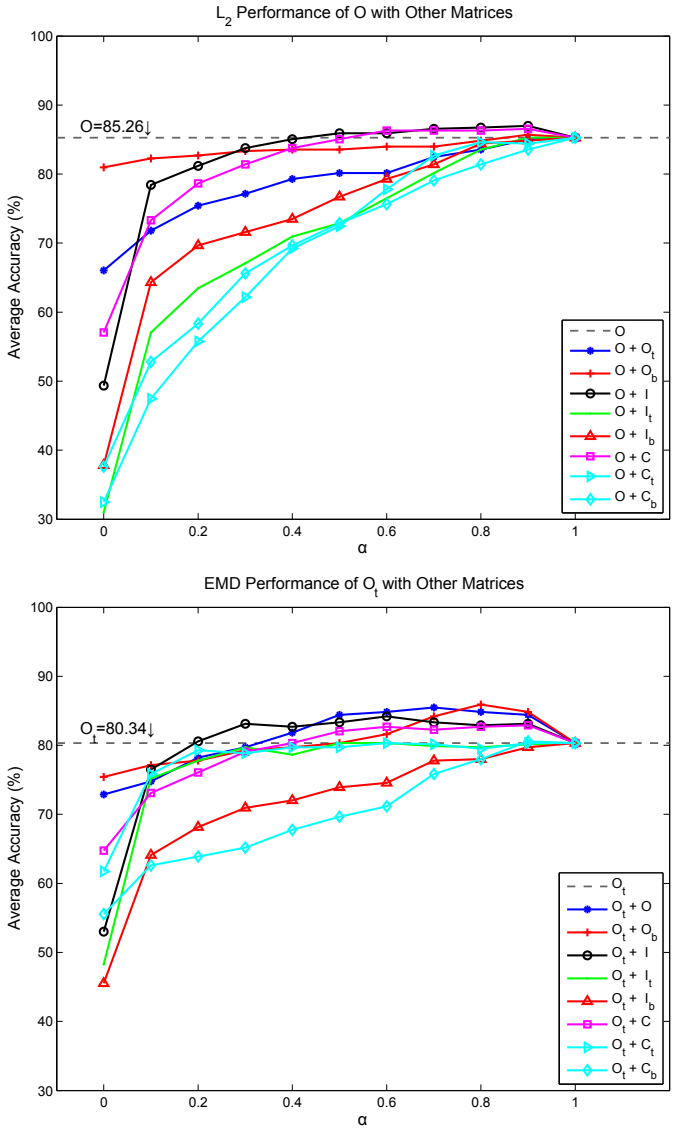


Figure 4.9: α search for two-level classification: the combination of $\mathbf{O}$ matrix with others using L_2 and the combination $\mathbf{O}_t$ matrix using EMD. $\mathbf{O}$ and $\mathbf{I}$ pair gives the highest accuracy (**86.97%**) with Euclidean Distance and $\alpha = 0.9$ value. $\mathbf{O}_t$ and $\mathbf{O}_b$ give the best performance, with $\alpha = 0.8$, resulting in an accuracy of **85.90%** for EMD measurement.

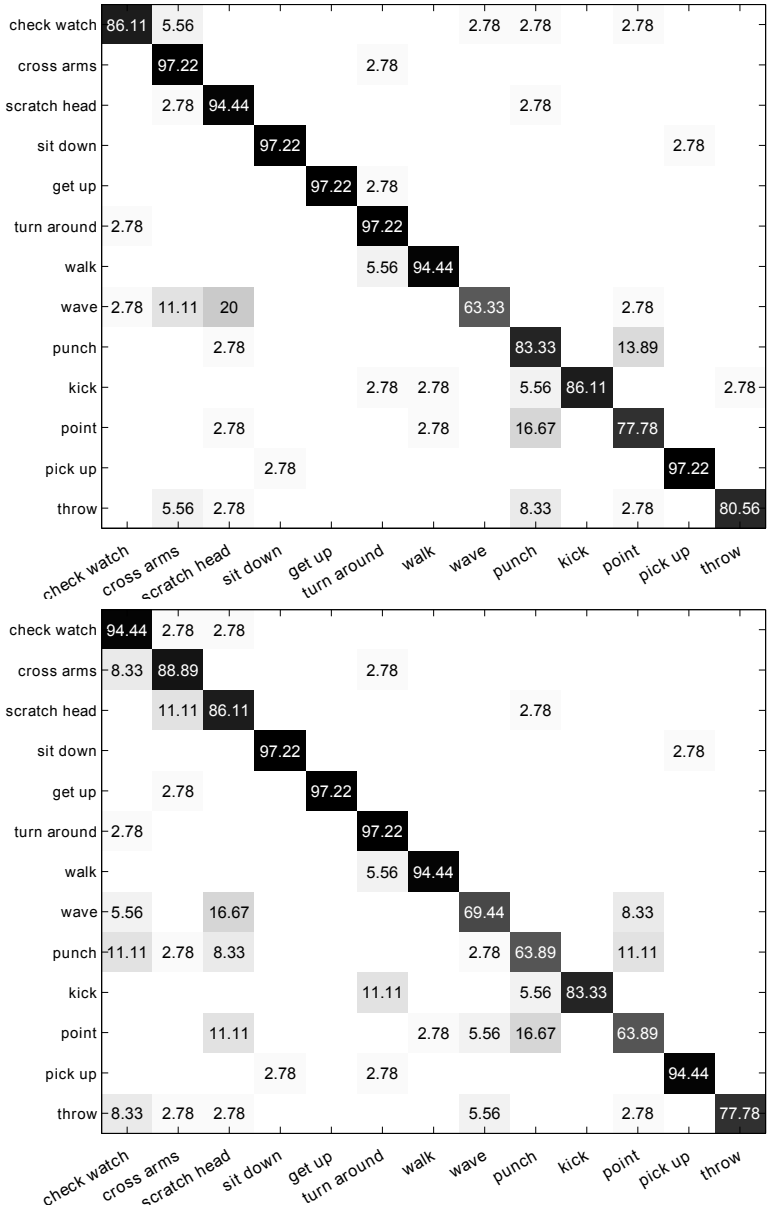


Figure 4.10: Confusion matrices for two-level classification on 13 actions and 12 actors. The top confusion matrix is for Euclidean Distance, with an accuracy 88.63%, while the bottom one is the result of the EMD, with an accuracy of 85.26%. Results are computed by the sum of weighted distances over three feature matrices. Matrix size is 48×40.

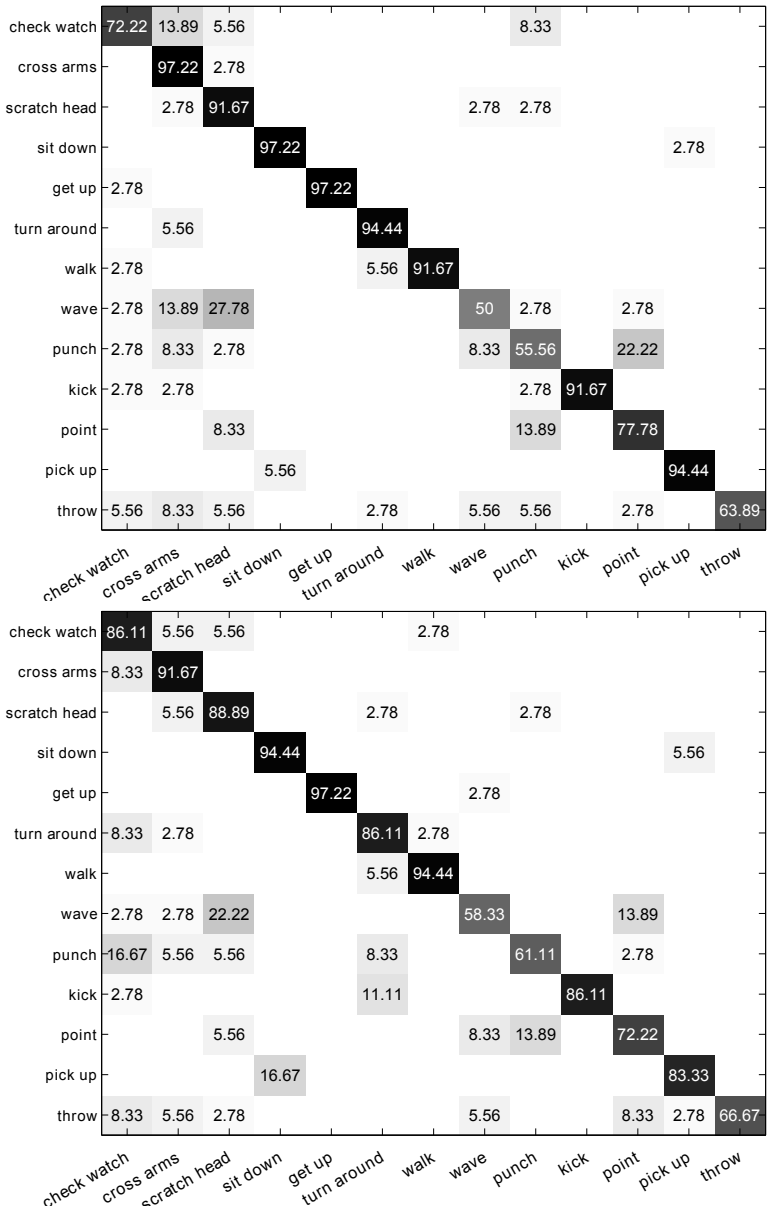


Figure 4.11: Confusion matrices for single-level classification on 13 actions and 12 actors. The top confusion matrix is for Euclidean computation giving 82.69% accuracy by combining **O** and **I** with weights 0.8, 0.2. The bottom confusion matrix is the EMD computation reported 82.05% accuracy by combining **O** and **I** with weights 0.7, 0.3. Results are computed by the sum of weighted distances over three feature matrices. Matrix size is 48×40.

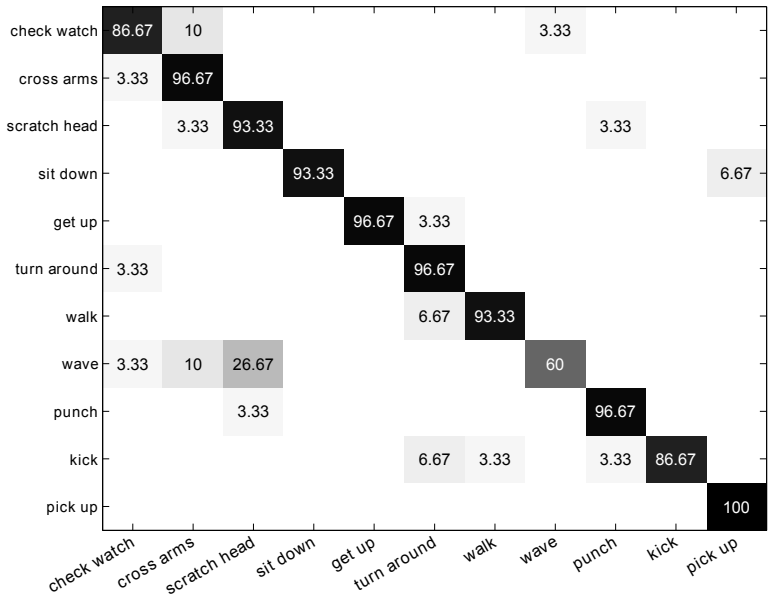


Figure 4.12: Confusion matrix for two-level classification on 11 actions and 10 actors, with an accuracy of 90.91%. This is the result of **O**, **I** and **C** combination weights 0.7, 0.1, 0.2. Matrix size is 48×40.

Chapter 5

Multiple View Activity Recognition Without Reconstruction

Generally, we expect that having multiple views makes recognizing human activity easier. There is support for this view in the literature (for example, see Section 2). However, these results tend not to take into account various desirable engineering features for distributed multi-camera systems. In such systems, we may not be able to get accurate geometric calibrations of the cameras with respect to one another (for example, if the cameras are dropped into a terrain). Cameras might drop in or out at any time, and we need a simple architecture that can opportunistically exploit whatever data is available. We will not be able to set cameras at fixed locations with respect to the moving people, meaning that training data might be obtained from different view directions than test data.

In this chapter, we describe an architecture to label activities using multiple views. Figure 5.1 shows the main structure of our architecture. We assume that there are one or more cameras, and that each camera can compute one or more blocks of features representing each frame. Breaking features into blocks allows us to insert new sets of features without disrupting the overall architecture. In

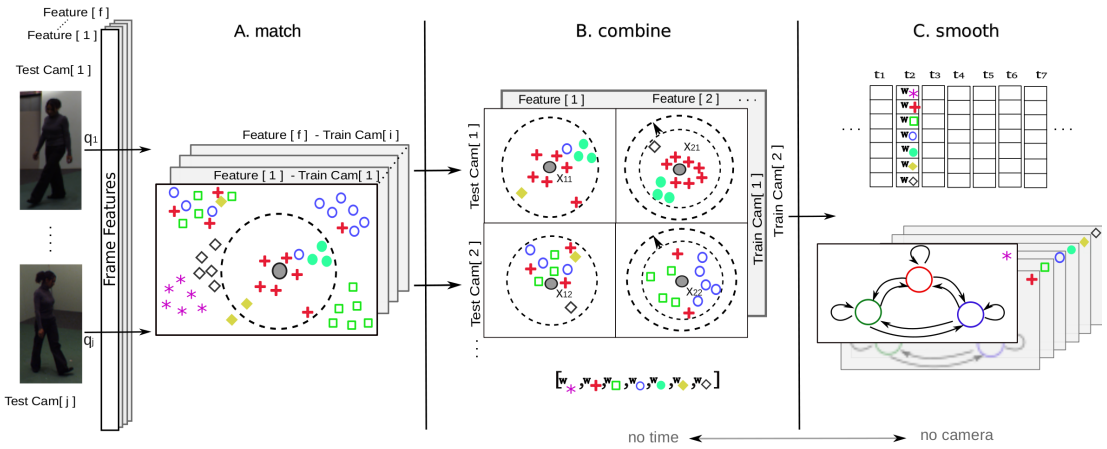


Figure 5.1: Our architecture is designed for cases where one or more cameras observe a moving person. Each camera can report one or more blocks of features. There are three core steps. First, each block of features for each camera finds a set of plausible matches in a training dataset. Second, these matches are combined to produce a set of confidence estimates for each possible label, using an approach where the most certain match dominates; these confidence estimates hide the number of cameras and the types of the features from the next step. Finally, a temporal smoothing step estimates the action label.

the first step, each block of features for each frame of each camera is used for a nearest neighbor query, independent of all other cameras, frames or blocks.

In the second step, the resulting matches are combined with a weighting scheme. Because we do not know the viewing direction of any camera with respect to the body, some frames (or feature blocks) might be ambiguous. We do expect that having a second view should disambiguate some frames, so it makes sense to combine matches over cameras. However, close matches are very likely to be right. This suggests using a scheme that (a) will allow several weakly confident matches that share a label to support one another and (b) allow strongly confident matches to dominate (see Figure 5.2). This stage reports a distribution of similarity weights over labels, but conceals the number of cameras or of features uses to obtain it, so that later decision stages can abstract away these details (Section 5.2.1). Finally, we use temporal smoothing, to estimate the action in a short sequence (Section 5.2.2).

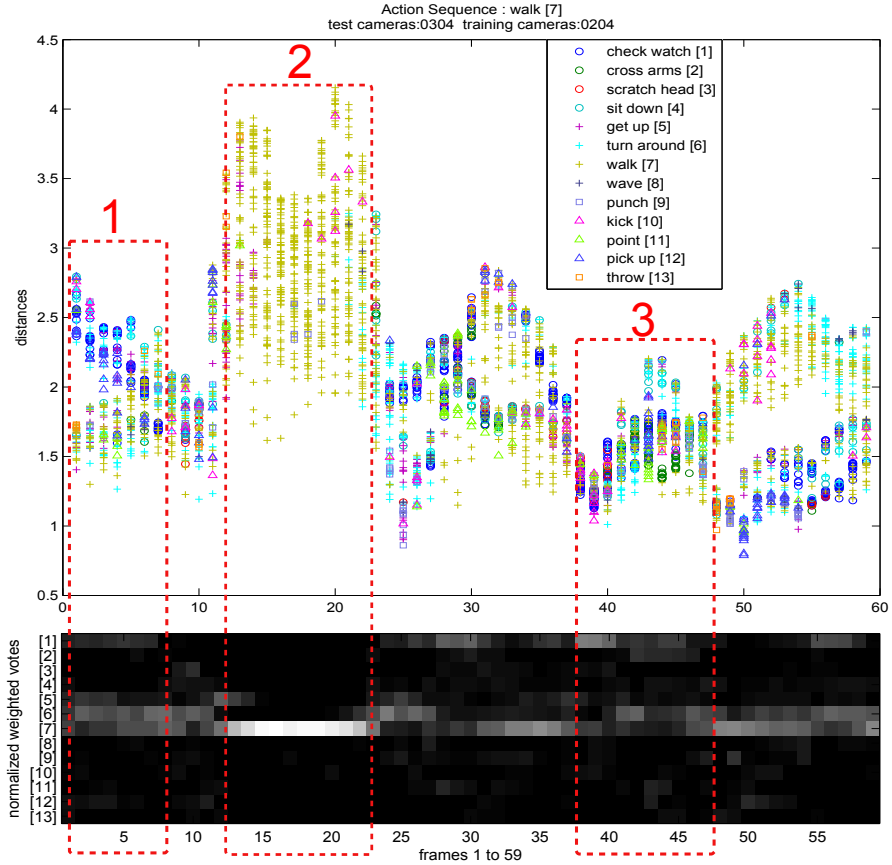


Figure 5.2: Individual frames in sequences can be highly ambiguous, but sequences tend to contain unambiguous frames. Top shows, for each frame in a walk sequence, the distances from those frames to labelled frames. For some frames (e.g. the block labelled 3), many different labels lie nearby; this may be caused by the fact that standing or rest poses occur in many sequences. Other blocks of frames are quite unambiguous (e.g. 2). Some cameras might be certain while others are not (block 1; the two traces are from different cameras). This means that evidence should be pooled over cameras and over sequences in ways that favor more confident views. Bottom shows activity weights evaluated using our method; notice that blocks that are unambiguous have low entropy in the weights.

Our architecture requires no camera calibration. Engineering in new sets of features is easy. When a set of features in a camera is confident, it dominates the labeling process for that frame. Similarly, the frames in a sequence that are confident dominate the decision for a sequence. Our experiments demonstrate that our method performs at the state of the art. We show results for several types of features. It is straightforward to incorporate new cameras or new features into our method. Performance generally improves when there are more cameras and more features. Our method is robust to differences in view direction; training and test cameras do not need to overlap. Discriminative views can be exploited opportunistically. Performance degrades when the test and training data do not share viewing directions. Camera drop in or drop out is handled easily with little penalty.

The organization of the chapter is as follows. We present the low level image features that are used for frame matching in Section 5.1, the proposed fusion method based on combination of activity judgments in Section 5.2. Experimental results for recognition performance in various camera cases are provided in Section 5.3. Finally, the chapter is summarized in Section 5.4.

5.1 Image Features

Each camera frame can be represented by many different types of features. In this section, we define our choice of features, but many others can be used too as our architecture is able to work with blocks of features at one time. While selecting features, we want feature construction to be simple yet robust. For practical reasons, it should not require camera calibration, or point correspondence. Selected features should be able to work with minor segmentation errors, and ideally they should not be affected significantly by change in viewpoint. Particularly, we expect them to be discriminative enough for matching.

We use three types of features: a coarse silhouette shape feature, a fine silhouette shape feature; and a motion feature. Two of them are computed from image

silhouettes. Silhouettes are simple and fairly robust to understand entire body shape while easing variations among people and background [5]. The third one is the motion feature based on optical flows among consecutive frames. Unlike silhouette based shape features, motion features are quite informative for matching individual frames of some activities.

All three types of features are computed over the same region of interest. This region corresponds to the bounding box around human silhouette, which is extracted by background subtraction. Cropped region is resized as having the maximum of width or height 120 while keeping aspect ratio, and placed inside a 120×120 image. For all feature types, the final feature vector is normalized by l_2 norm.

5.1.1 Coarse Silhouette Feature

Fine gridding structures both in vertical and horizontal axis reveals the true spatiality, but it may also decrease the sensitivity against changes in viewpoint due to overfitting. Our question is "How much spatiality in feature computation is necessary to cope with viewpoint variations?". In particular, we have a pool of frames just being in the perspective of train camera set. When any frame in some other perspective is queried, the chance of accurate matching severely decreases. Therefore, a coarser encoding may behave better in some cases.

The rough shape of the silhouette can be discriminative. For example, in many walking frames the arms are away from the body, but this does not happen in standing frames. For the coarse feature, we compute the contour extracted from a 120×120 silhouette image. The feature is constructed from the distance between the outermost contour points at each scan-line. The silhouette is divided into n horizontal layers, and the maximum, minimum and average distance within each layer are stacked to form the feature. For computation, we use a simple scan line algorithm on contour images. Figure 5.3 shows maximum line values.

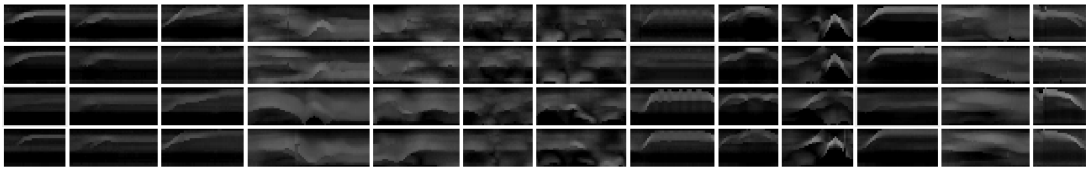


Figure 5.3: Example coarse form line features (silhouettes are divided into 20 layers). Frames correspond to column vectors and videos are shown as aggregated vectors. Action videos are performed by the same actor from the IXMAS dataset. From left to right, actions are in the following order: check watch, cross arm, scratch head, sit down, stand up, get up, turn around, walk, wave, punch, kick, point, pick up, throw. From top to bottom, each row corresponds to views from four cameras of the IXMAS dataset.

5.1.2 Fine Silhouette Feature

The second feature is an encoding of pixel occupancy over the 120×120 silhouette image, using two common local gridding structures (radial and square respectively).

According to first structure, silhouette data is first divided into $n \times n$ cells, then each cell is divided into m radial bins [57]. The number of set pixels are counted at each bin, and the feature vector is constructed by aggregating and then normalizing by l2. Second structure divides the silhoutte image into $n \times n$ square cells [28].

5.1.3 Motion Feature

Motion features are quite informative for some activities (sit down versus get up). Any standard optical flow implementation can be used for this purpose. We apply the Kanade-Lucas-Tomasi Feature Tracker [55] to each pair of frames. Flow vectors inside the region of interest are encoded as two different channels corresponding to x and y axis motions [11]. Each channel is formed into 120×120 image and divided into square grids. Then, these are aggregated into a single feature vector.

5.2 Label Fusion

In our system, we fuse information at three levels to produce three kinds of label. First, different blocks of features in a single camera must produce the camera-specific label for a frame. Second, the camera specific labels for a frame must produce the frame label. And finally, the frame labels must produce the sequence label.

5.2.1 Combining Cameras and Features

We label each block of features from each camera frame with k-nearest-neighbors (kNN) [6, 48]. kNN is simple but it outperforms most other classifiers. The main drawback is the computational complexity, especially over huge action dataset. There are many powerful and fast algorithms [54, 40, 2]. We use the E^2LSH implementation of Locality Sensitive Hashing (LSH) [2]. It is based on multiple hash tables drawn from a family that decrease the potential search failures. Using this algorithm, we find exact nearest neighbors and compute pairwise distances between feature vectors by Euclidean distance.

We want to compute a weighted vote vector for a test frame recorded in test camera $j \in \mathcal{C}'$. Votes are collected for each action class from nearest neighbors recorded in training cameras $i \in \mathcal{C}$. Distances in between the query frame and its kNN varies from one frame to another. Thus, first we normalize all absolute Euclidean distances using l_1 norm. Then, we transform distances using Gaussian radial basis function (RBF). Each match has an action label, denoted by a. Its distance to query frame is used as the weighted vote for this label, so the most similar(closest) matches tend to vote with the highest values. Let we recover a set of training frames $\hat{X}_f^{ij}$ for test frame x_f^j computed in block of feature $f \in F$. First, we normalize all absolute distances between query frame and $\hat{X}_f^{ij}$ as

$$dist(x_f^j, \hat{x}_{f,r}^{ij}) = \frac{\parallel x_f^j - \hat{x}_{f,r}^{ij} \parallel_2}{\sum_{r \in \hat{X}_f^{ij}} \parallel x_f^j - \hat{x}_{f,r}^{ij} \parallel_2} \tag{5.1}$$

where r is $\hat{x}_{f,r}^{ij} \in \hat{X}_f^{ij}$.

We now construct a set of weighted votes for that query frame and that block of feature per action class with label $k \in \{1...n\}$.

$$w_f^{ij}(k) = \sum_{a \in A_k} e^{\frac{-dist(x_f^j, \hat{x}_{f,a}^{ij})^2}{2\sigma^2}} \tag{5.2}$$

where a is $\hat{x}_{f,a}^{ij} \in A_k$ and $A_k \subset \hat{X}_f^{ij}$ with action label k. Query frame is more likely to have the same action label with the closest matches. Vote vector, w_f^{ij}, is a n dimensional feature vector. We normalize w_f^{ij} vector by l_1 norm.

$$w_f^{ij} = \frac{w_f^{ij}}{\parallel w_f^{ij} \parallel_1} \tag{5.3}$$

Multiple vote vectors are computed for every training and test camera pairs and blocks of features. Now we must combine these votes from different blocks of features, and over viewing cameras. We know that while a single frame can be discriminative, many others may be ambiguous. Also, we know that descriptors of all cameras and features may agree or disagree on action label. We would like the final combination of all possible cameras and block descriptors, tend towards the most confident one. Also, we would like to do this in a straightforward manner for all features and all camera combinations. We do so by forming a weighted, normalized sum

$$w(k) = \frac{1}{\mid \mathcal{F} \mid + \mid \mathcal{C} \mid + \mid \mathcal{C}' \mid} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{C}'} w_f^{ij}(k) \tag{5.4}$$

where weight vectors are combined over test cameras, training cameras and low

level image features. This conceals the multiple information of camera views and low level appearance features from activity recognition stage. Combined vector represents the votes on the frame label, and it is a high level feature.

This process ensures that strongly confident blocks of features will have a major effect on the final vector w, as will strongly confident viewing directions. Architecture fuses what we have with little fass, and copes with dropouts of cameras and features without changing trained model. We now have, for each test frame, a multi-view weight vector where the most certain class label should be dominant. Weighted combination of various features over cameras can be computed over features F. We do so by minimizing least square error for weight vectors learned using training cameras (Section 5.3.3).

Combining features over computed match set is feasible, since system does not have to compute pool of features for every new shape feature. This allows the system to distribute jobs among cameras. Each camera can work with different blocks of features and the final matches will be combined in a central pool. So, cameras do not have to aware of each other's processes. The only thing to be learned is in β combination case.

5.2.2 From Frame Labels to Sequence Labels

We now have a set of weights associating each frame in a sequence with a label. If the frame is unambiguous, then this weight vector will have one large value; if the frame is ambiguous, then several values might be quite large. We cannot treat these weights as probabilities, but can use them as features for a classifier. Naïve Bayes (NB) is one natural approach, but assumes that frame-frame dynamics are not particularly significant in classifying a sequence. The traditional alternative is to learn an Hidden Markov Model (HMM) for each action, then choose the action with the highest likelihood on the data.

5.2.2.1 Naïve Bayes

A Naïve Bayes model assumes that the weight vector associated with each frame is emitted independently, conditioned on action class. This ignores the temporal structure of activities, but is well adapted to sequences where some frames are ambiguous and others are highly distinctive (which seems to be the case for most activity datasets). Assuming per-frame weight vectors w are independent conditioned on action, the class posterior probability is:

$$P(k|w) \propto \pi(k) \prod_{k=1}^{n} P(w(k)|k) \tag{5.5}$$

where $\pi(k)$ is the class prior. We assume that the per-frame weight vectors are emitted with a Gaussian conditional distribution, and that the covariance is diagonal and the same per weight vector. We then have

$$\log P(k|seq) \propto -\frac{1}{2\sigma^2} \sum_{m \in seq} [(w^m - \mu_k)^2] + \log \pi(k) + L \tag{5.6}$$

L is the constant term. Now assume the class priors are uniform; then we obtain the activity label for the sequence as:

$$\arg\max_k - \sum_{m \in seq} (w^m - \mu_k)^2 \tag{5.7}$$

Parameter μ_k is a mean weight vector of size n computed for every action class using training samples. μ_k and w vectors are normalized vectors having positive values summed to one. Here, normalized vectors, transformed distances and mislabeling of shared poses guaranty robustness. They do not have a major impact on the classification results.

5.2.2.2 Hidden Markov Model

An alternative which encodes temporal structure and has a long history in activity recognition is the continuous HMM [47]. HMM provides temporal smoothing over video frames. We model each action category as three state HMM with four gaussian mixtures. The parameters of the HMM are set according to the best rate in the range of 2 to 6 over a small subset of activity dataset. 3 state HMM is perform well and it does not much differ from that of 2 or 4. The highest likelihood HMM model for each test sequence defines the action label for that sequence.

5.3 Experimental Results

Dataset: To test our approach, we need a dataset consisting of videos in multiple viewpoints performed by actors in free orientations. We use the publicly available INRIA Xmas Motion Acquisition Sequence (IXMAS) dataset [61] (This has been widely used by many others [35, 59, 12, 32, 21]). It contains 13 actions (N_k) performed 3 times (N_o) in different orientations by 12 actors (N_p). The action sequences are recorded in 5 cameras (N_c). [12] reports 10% average cross camera accuracy if one trains with camera 5, and shows how different its view when just appearance features are used without any special setup. We follow it and exclude camera 5, as its overhead view strongly differs from the others. Figure 5.4 gives summary information for this dataset.

Training and testing: Experiments are carried out by leave-one-actor-out cross validation. For a given training $\mathcal{C}$ and test $\mathcal{C}'$ camera sets, experiments are done N_p times and the results are averaged as being the final accuracy. The number of cameras $N_{\mathcal{C}}$ and $N_{\mathcal{C}'}$ associated with training camera set $\mathcal{C}$ and testing set $\mathcal{C}'$ may vary; and sets are disjoint if $\mathcal{C} \cap \mathcal{C}' \equiv \emptyset$.

In each leave one actor out experiment, all video samples of an actor p_i,

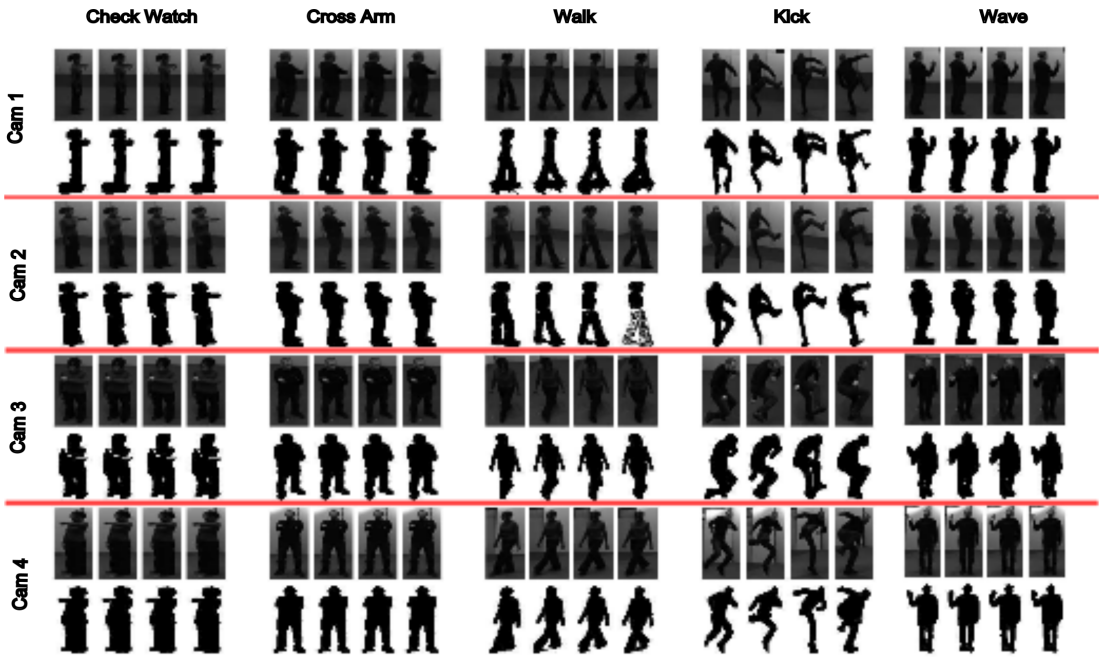


Figure 5.4: Example frames of some action classes from the IXMAS Dataset. Every actor performs every action in 3 different orientations. Cameras 1 and 2 are placed close and almost have side views. Camera 3 is installed higher than others, therefore cross camera recognition using camera 3 is hard due to great variation in viewpoint. Camera 4 is the most controlled one having a reasonable height and side view. In cropped images, it is shown that true body pose gets lost in some views depending on the camera and actor orientation. Also, background substraction results with some defects (median filter is applied over binary images prior to feature extraction.)

$i = \{1, .., N_p\}$, recorded in $\mathcal{C}'$ are taken from the dataset and used for testing. The rest of the actors $\{p_j\}_{j \neq i}$ whose videos are recorded in cameras of $\mathcal{C}$ is used as training samples to learn activity models.

The dataset has $N_p \times N_o \times N_c = 180$ videos per action. Assume we choose two cameras for training and testing, $N_{\mathcal{C}} = 2$ and $N_{\mathcal{C}'} = 2$. Every action model is learned using $(N_p - 1) \times N_o \times N_{\mathcal{C}} = 66$ videos. First, every frame of $N_o \times N_{\mathcal{C}} = 6$ training videos associated with p_j, where $j \neq i$, is queried over pools. These pools contain frames of $(N_p - 2) \times N_o \times N_{\mathcal{C}} = 60$ videos of $\{p_k\}_{k \neq \{i,j\}}$ recorded in training cameras $\mathcal{C}$. Then, closest matches are retrieved and combined over cameras and features(see Section 5.2.1). Finally, training videos in the form of

Table 5.1: Average single camera recognition rates ($\sigma = 0.05$, $knn = 20$). One-shared camera experiment corresponds to using the same camera, while no-shared one means using different cameras. For all features, using the same camera gives the highest accuracy.

	NB (%)		HMM (%)	
	one-shared	no-shared	one-shared	no-shared
line	61.49	44.53	59.94	44.68
radial	68.54	48.24	67.25	47.69
optical flow	82.10	55.47	80.50	55.63
"thin" combination	79.43	57.46	78.10	57.75
β combination	83.28	61.06	79.54	58.97

aggregated vote vectors are used to train the corresponding action classifier (see Section 5.2.2).

During testing, we follow the same order as in training. This time, $N_o \times N_{\mathcal{C}'} = 6$ videos of p_i are used for testing. Frames are queried over the pools that contain video frames of $\{p_k\}_{k \neq \{i\}}$ recorded in cameras of $\mathcal{C}$. Please note that pools and query samples are totally disjoint in terms of actor, and also in term of camera views if $\mathcal{C} \cap \mathcal{C}' \equiv \emptyset$ during experiments.

Pooling and memoization: Frame matching is the most time consuming operation. Through experiments, match sets per different actor and camera combinations are overlapped and recomputed multiple times through experiments. We think constructing multiple smaller pools with little data variation is always better then a single large pool.

We make use of memoization and multiple pools. We construct pools for every camera and feature pair containing all videos of an actor(including left-right reflections of frames, except line feature). This results in $N_p \times N_{\mathcal{C}}$ pools for each feature type. For each query sequence, kNN sets from different pools are stored in sorted order. Then, we combine them according to camera, and feature of interest using merge sort. This decreases the time to construct pool tables, decreases retrieval time and avoid redundant computations. By this mean, expansion of training pools with new actors, actions and cameras are possible.

5.3.1 Single Camera Recognition Rates

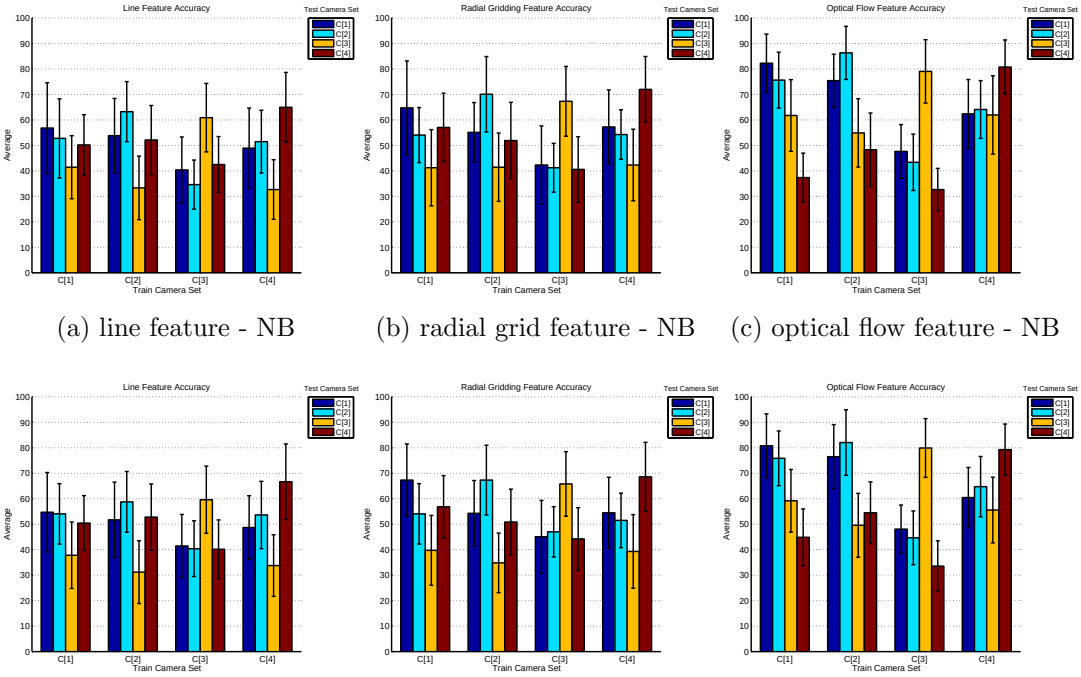
In the first experimental setup, we use one camera for training and testing. There are two distinct conditions here: the test view appears in the training set (the one-shared camera case), or the test view does not appear in the training set (the no-shared camera case, where we expect reduced recognition rates).

We perform experiments for radial grid, line and optical flow features separately. Experiments are done with many different parameter configurations for each feature. The best results are achieved with the following parameters: radial grid feature with 5×5 square and 18 radial cells; line with 20 horizontal slices; and optical flow with 8×8 gridding. The number of closest matches used are 20, and σ value used to compute transform distances in (Eq. 5.2) is 0.05.

We have experimented with both the square grid feature and the radial grid feature, but find the radial grid to be slightly more accurate. Since they encode the same information, we report only experiments on the radial grid feature. Optical flow feature significantly outperforms others with more than 10% accuracy in one-shared camera case. If we compare three features, line feature has lower recognition than others in one-shared camera. However, we will see later that it helps to improve recognition when combined with other features. We double the set of training frames by inserting left-right reflections; this increases recognition performance.

Figure 5.5 shows activity classification results in detail for individual features in single view. When the training set has frames from the same viewpoint as the test set, accuracy is high. Combined features results in 83.28% accuracy, comparable with the state of the art systems (Table 2.1). Notice that recognition rates fall off 21% (β combination) when test and training sets do not share viewpoints; the fall off is consistent with those reported in Table 2.1. We conclude NB and HMM are reasonable choices of classifiers, and we have reasonable features. Table 5.3 compares classifiers results, for different feature cases, combinations and shared-view cases. The IXMAS dataset is not highly dynamic with few periodic actions (walk and wave). In this case, HMM with 3 states work well, and 2 state

or 4 state models work almost as well. A similar discussion holds when NB and HMM results are compared. NB with no temporal information may outperform HMM on the same dataset. This is due to the nature of the dataset. HMM can behave better than NB in another dataset.



(a) line feature - NB (b) radial grid feature - NB (c) optical flow feature - NB

(d) line feature - HMM (e) radial grid feature - HMM (f) optical flow feature - HMM

Figure 5.5: Single camera recognition rates of NB classifier for each individual feature type (similar discussion holds for the HMM classifier). Parameters σ and knn are chosen according to best rate ($\sigma = 0.05$, $knn = 20$). For no-shared camera combination, features (a) and (b) result in closer accuracies when the viewpoints are similar (eg cameras 2 and 4), and the motion feature (c) gets severely affected from view point variation. This is because silhouette features keep the rough shape information across similar views, but motion feature can be dissimilar across views per frame basic. In general, significant view variation is crucial for no-shared camera case (eg the combination with camera 3 performs the worst). For the one-shared camera combination, the motion feature (c) is quite robust and is around 80%.

Table 5.2: Average two camera recognition rates for each feature and camera combination ($\sigma = 0.05$, $knn = 20$). In two-shared camera experiments, the same set of camera pair is used for both training and testing. In one-shared experiments, just one camera is the same, and no-shared one means that camera pairs are disjoint.

	NB (%)			HMM (%)		
	two-shared	one-shared	no-shared	two-shared	one-shared	no-shared
line	71.08	65.79	59.26	69.12	65.16	60.15
radial	78.70	71.43	63.78	77.39	70.63	63.07
optical flow	86.82	80.09	69.73	79.81	75.81	68.34
"thin" combination	86.86	81.96	74.50	82.23	77.63	71.51
β combination	88.07	83.17	75.64	83.48	78.98	72.86

5.3.2 Two Camera Recognition Rates

When there are two test and two training cameras, three cases are important: two views are shared; one view is shared, and no views are shared. Table 5.2 summarizes the average recognition rates for NB and HMM classifiers for various view and feature cases.

As one would expect from the single camera results, when test and training sets share viewpoints performance increases. However, notice that having a second view tends to increase recognition rates quite strongly, even if there is no sharing of views between test and training sets (15% in this case). We conjecture that this is because there is a good chance that one view is unambiguous and correct, even when views are not shared. Figure 5.6 shows detailed recognition rates for different sets of test and training camera. When the same set of cameras are used during training and testing, recognition accuracy is 88.07% in β combination-NB setting. When one camera is shared, accuracy falls to 83.17%. This is comparable to one-shared single camera recognition rates. If camera set is disjoint, β combination results in 75.64% which is 14% higher than the no-shared single camera case.

The two-train, two-test case is best compared to systems of types 4 and 5 in Table 2.1, because in this case one could come up with training (resp. test) volumes, though we do not use as many views to build the volumes as those systems do. Notice that the most favorable cases for our system are those where

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	70.73	60.26	64.53	61.32	66.45	54.7
Test C[1 3]	62.82	69.44	62.18	68.59	57.69	67.09
Test C[1 4]	65.81	64.1	70.94	63.25	70.73	65.81
Test C[2 3]	63.25	67.95	61.11	70.94	64.1	66.88
Test C[2 4]	70.73	64.53	76.07	67.31	76.71	67.09
Test C[3 4]	54.27	66.67	64.32	63.03	61.75	67.74

(a) line feature - NB

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	75.64	65.6	69.44	71.79	72.65	62.18
Test C[1 3]	68.8	79.7	69.87	76.07	64.96	74.57
Test C[1 4]	70.94	69.23	77.14	66.45	76.07	70.3
Test C[2 3]	69.02	75.64	64.1	80.13	70.51	73.08
Test C[2 4]	75.85	66.67	77.35	73.08	83.12	70.94
Test C[3 4]	58.33	70.73	67.74	69.02	66.03	76.5

(b) radial grid feature - NB

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	87.82	74.79	83.76	76.5	82.05	65.6
Test C[1 3]	83.55	88.89	80.56	85.47	76.92	79.91
Test C[1 4]	79.49	68.16	86.75	69.02	86.32	74.36
Test C[2 3]	86.54	85.9	81.62	85.9	81.62	80.13
Test C[2 4]	80.34	59.19	88.25	72.44	89.32	75.64
Test C[3 4]	66.03	76.92	82.48	76.5	80.56	82.26

(c) optical flow feature - NB

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	63.46	61.54	64.74	62.82	65.81	57.05
Test C[1 3]	57.69	66.88	58.55	66.67	56.84	65.17
Test C[1 4]	64.53	64.74	67.74	64.74	71.37	63.89
Test C[2 3]	62.61	69.02	60.26	72.01	62.39	67.31
Test C[2 4]	68.59	66.45	73.93	68.38	77.14	67.31
Test C[3 4]	55.56	68.16	61.97	67.52	59.19	67.52

(d) line feature - HMM

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	76.28	66.24	67.52	69.23	72.44	62.82
Test C[1 3]	65.6	75	68.16	75.21	62.61	73.5
Test C[1 4]	70.51	68.38	75.21	67.95	76.71	74.36
Test C[2 3]	68.16	73.5	60.26	80.77	70.3	73.29
Test C[2 4]	72.44	68.8	74.79	71.37	81.2	71.79
Test C[3 4]	55.98	69.87	65.6	69.87	66.24	75.85

(e) radial grid feature - HMM

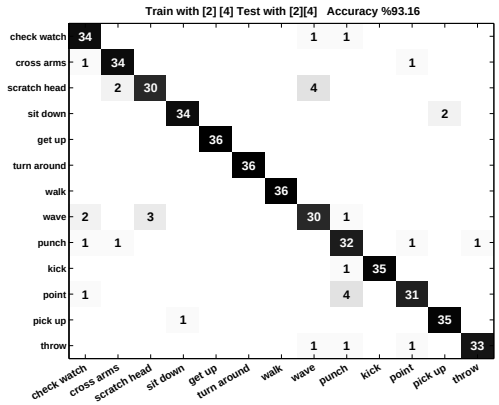
	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	79.49	73.5	75.21	73.5	77.56	66.24
Test C[1 3]	76.5	82.91	74.36	77.99	72.01	79.06
Test C[1 4]	76.5	70.3	79.27	69.87	76.92	68.59
Test C[2 3]	76.07	82.26	75	78.42	74.79	78.42
Test C[2 4]	78.63	63.25	80.56	70.09	81.2	72.22
Test C[3 4]	63.68	72.65	72.65	72.22	70.51	77.56

(f) optical flow feature - HMM

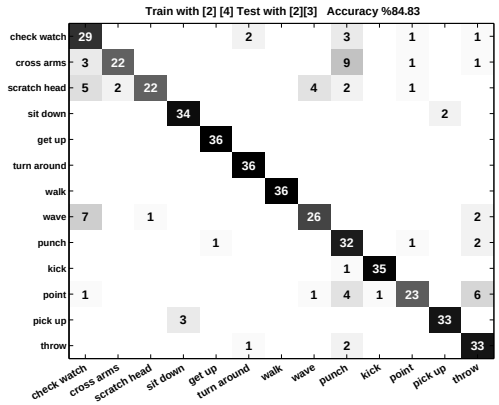
Figure 5.6: Two camera recognition rates of NB classifier for various camera combination($\sigma = 0.05$, $knn = 20$). On every accuracy matrix, diagonal entries denote results when all training and test cameras are shared. Off-diagonal(colored blue) entries show disjoint camera sets. (a), (b) and (c) show individual feature performances. The maximum recognition over all experiments is computed by training and test camera pair [2 4]. (Similar discussion holds for the HMM classifier results)

test and training views are shared (the diagonals in Figure 5.6). The recognition rates here compare well with those of types 4 and 5 in Table 2.1, with some slight fall-off most likely due to the relatively small number of views. We are reporting the worst case for multi-view combination, where there are only two views. In particular, notice that for all sets of views and a reasonable feature combination, this worst case is very well behaved (Figure 5.10). Figure 5.7 shows example confusion matrices for experiments on camera pairs [2 4] on no-shared, one-shared and two-shared cameras cases. The comparable performance with respect to volume reconstruction is justified by the tremendous advantage of not needing to build these volume features, and so being able to cope with drop-out, cameras not calibrated to one another, and so on.

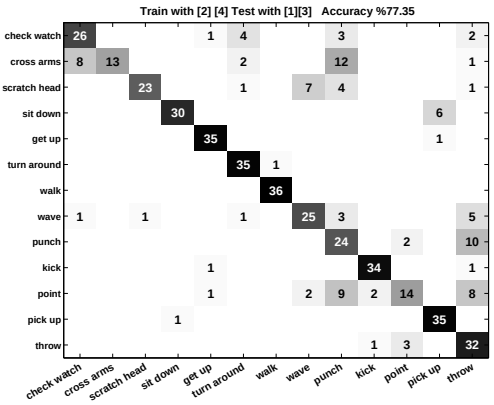
There is some evidence of a link between recognition rates and reconstruction here. Cameras 2 and 4, the best behaving pair (93.16% recognition rate in



(a) β combination-NB:two-shared



(b) β combination-NB:one-shared

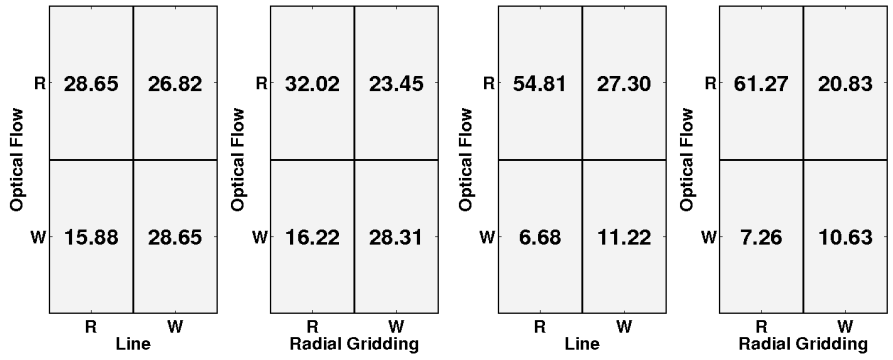


(c) β combination-NB:no-shared

Figure 5.7: Confusion matrices over activity classes for some individual experiments from Figure 5.10b. When camera pair [2 4] is used for training, (a) shows two-shared cameras case (testing with camera pair [2 4]), (b) shows one-shared camera case (testing with camera pair [2 3]), (c) shows no-shared camera case(testing with camera pair [1 3])

Figure 5.10b), are close to perpendicular, and so would reconstruct quite good volumes. This is most likely because it is hard to have two cameras that would (a) reconstruct a volume well and (b) are both ambiguous; understanding this phenomenon would be valuable.

Figure 5.12 shows the performance of HMM for various test camera cases when the system is trained using cameras [2 4]. This is a video sequence of 13 activities performed by an actor. We apply a sliding window including 23 frames on the video sequence and each window is recognized with the highest likelihood activity label. The most confusing sequence is cameras [1 3] due to non-shared



(a) Same Cameras (b) Different Cameras

Figure 5.8: Correlation matrices among features when trained with one camera using NB. Matrix entries show the number of test samples that are classified as correct or incorrect when different low level image features are used. (a) shows if test and training cameras are same. (b) shows if cameras are different(eg train with camera 1 and test with camera 2).

camera condition. When there is a single camera shared among training and test pairs, the recognition increases. Two-shared (cameras [2 4]) performs the best.

5.3.3 The Effects of Combining Image Features

Simple experiments suggest that different sets of features make errors that are somewhat uncorrelated, suggest that it is worthwhile to combine features. In particular, for single camera experiments in no-shared camera case, we obtain on average P(radial is right | optical flow is wrong)= 0.40, and P(line is right | optical flow is wrong)= 0.37 (eg 37% of the positive instances are labelled true by radial feature while they are labelled false by optical flow feature). Figure 5.8 shows the correlation matrices on average between optical flow feature and other features for no-shared(different) and one-shared(same) camera combinations.

We have two schemes for feature combination (Section 5.2.1); summing votes, or summing weighted votes. As Tables 5.3 and 5.2 indicate, weighting votes increases accuracy. Summing votes is a simple method that directly gets the average of feature votes. The weighted voting procedure learns weight matrix β using training samples.

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Test C[1]	76.92	68.16	52.99	62.61
Test C[2]	72.22	80.77	51.07	63.03
Test C[3]	51.5	46.37	79.06	50.85
Test C[4]	62.39	61.54	46.79	80.98

(a) "thin" combination - NB

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Test C[1]	81.41	74.15	54.7	65.6
Test C[2]	77.14	86.11	54.49	65.17
Test C[3]	58.12	51.07	82.48	54.7
Test C[4]	62.82	64.74	50	83.12

(b) β combination - NB

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Test C[1]	73.08	69.02	54.06	63.46
Test C[2]	67.31	79.49	55.77	60.04
Test C[3]	51.28	47.01	79.7	49.79
Test C[4]	60.26	64.32	50.64	80.13

(c) "thin" combination - HMM

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Test C[1]	77.99	69.23	55.34	63.25
Test C[2]	73.29	80.56	56.2	60.9
Test C[3]	52.14	45.73	78.63	52.35
Test C[4]	61.11	63.89	54.27	80.98

(d) β combination - HMM

Figure 5.9: Single camera recognition accuracies on combined features. (a) and (c) show the "thin" combination of radial-line-optical flow features, (b) and (d) show the combination of features with parameters learned by least square error.

Parameters are learned using weight vectors from some subset of training samples. For a given training camera set C, let $W \in R^{M*t}$ be a training set of vote vectors. Vote matrices W_f are concatenated into a single matrix W; where t equals to $\mid \mathcal{F} \mid \times n$, $\mid \mathcal{F} \mid$ is the number of features and n is the number of action classes. $Y \in R^{M*n}$ is a matrix consisting of $\{0, 1\}$; and y_{lk} is 1 when the lth sample belogs to the kth action class. We learn a matrix of weights $\beta \in R^{t*n}$. Each column vector $\beta_{.k}$ is learned using $y_{.k}$ and W by minimizing least square with the following

$$\begin{aligned} &\arg\min_{\beta} \parallel Y - W\beta \parallel_2^2 \\ &s.t.\ 1^T\beta_{.k} \leq t\ , \beta_{.k} \geq 0, \end{aligned} \tag{5.8}$$

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	85.04	77.35	78.42	80.13	83.55	68.59
Test C[1 3]	81.41	88.03	83.97	86.97	76.28	82.05
Test C[1 4]	83.12	80.34	87.61	78.85	85.26	78.85
Test C[2 3]	81.84	84.62	80.56	87.61	82.05	80.13
Test C[2 4]	86.54	76.5	87.61	83.12	89.74	79.7
Test C[3 4]	66.24	81.84	78.42	83.76	76.07	83.12

(a) "thin" combination - NB

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	86.97	78.63	81.2	83.33	85.04	69.87
Test C[1 3]	83.12	88.68	83.76	87.18	77.35	84.4
Test C[1 4]	83.97	80.56	87.39	81.84	86.11	80.13
Test C[2 3]	83.97	85.68	78.63	88.89	84.83	80.34
Test C[2 4]	88.89	76.28	87.39	84.62	93.16	81.84
Test C[3 4]	69.87	82.05	77.78	84.19	77.14	83.33

(b) β combination - NB

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	81.41	72.65	73.08	77.14	74.79	65.81
Test C[1 3]	74.57	81.41	78.85	81.41	70.94	78.85
Test C[1 4]	80.56	73.5	82.05	73.5	79.7	73.5
Test C[2 3]	79.91	82.05	75.43	84.83	76.71	79.91
Test C[2 4]	83.12	76.07	81.84	80.56	82.48	77.78
Test C[3 4]	67.31	76.07	74.36	78.85	73.29	81.2

(c) "thin" combination - HMM

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1 2]	83.12	75.43	75.21	82.05	81.41	68.16
Test C[1 3]	76.92	80.98	74.57	83.97	74.79	79.27
Test C[1 4]	78.63	76.71	80.77	77.99	83.33	75.21
Test C[2 3]	78.21	81.62	73.5	86.75	80.77	76.92
Test C[2 4]	82.48	74.79	82.91	83.12	87.82	77.78
Test C[3 4]	67.95	77.78	73.93	81.2	76.07	81.41

(d) β combination - HMM

Figure 5.10: Two camera recognition accuracies on combined features. The (a) and (c) show the "thin" combination of radial-line-optical flow features. The (b) and (d) indicate the combination of features with parameters learned by least square error. On every accuracy matrix, diagonal entries denote results when all training and testing cameras are shared. Off-diagonal(colored red) entries show disjoint camera sets. The maximum recognition over all experiments is computed by training and test camera pair [2 4]. For this camera pair, the best reported result is 93.16% using β combination of features with NB classification. We experiment on 13 action classes. This is comparable to the best results using volumes: the highest reported result is 93.33% accuracy using volumes but over 11 action classes [61].

Table 5.3: Average recognition rates(%) for no-shared camera case showing the addition of a second test camera. First row shows average no-shared camera rates of Figure 5.11b when the system is trained with the corresponding camera; and second one shows average no-shared camera results of Figure 5.9b. Having a second view during testing improves performance.

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Testing with 1 Camera	66.03	63.32	53.06	61.82
Testing with 2 Cameras	78.13	72.58	59.55	70.51

5.3.4 The Effects of Camera Drop Out

In real time, the number of active cameras in a system can change due to drop out or drop in. Figure 5.11 shows the accuracies of β combination for training with two cameras and testing with one camera. Comparing Figure 5.11a with the diagonal of Figure 5.10b, we observe the average performance drop when a camera switches to offline mode. For example, for training camera pair [1 2], testing with camera 1 has 80.34%, testing with camera 2 has 83.97%, but testing with cameras [1 2] gives 86.97% accuracy. Performance drops with a camera lost, but camera dropouts are not a catastrophe.

We also perform sliding window recognition on a test video sequence to measure the effect of camera drop out. Figure 5.13 shows the performances when a test camera is switch into offline mode.

Similarly, we can compare Figure 5.11b with Figure 5.9b to evaluate the addition of a second camera in real time. When camera 1 is the only camera that is used for training, testing with camera 1 performs 81.41% while testing with camera pair [1 2] gives 85.9%. Table 5.3 summarizes the performance of camera drop in for no-shared camera case. For example, when the camera 1 is used for training, the average recognition rate over testing with camera 2 or 3 or 4 is 66.03% and the average recognition rate over testing with camera pairs [2 3] or [2 4] or [3 4] is 78.13%. This shows that even with disjoint camera sets, addition of a second camera during test time increases performance.

	Train C[1 2]	Train C[1 3]	Train C[1 4]	Train C[2 3]	Train C[2 4]	Train C[3 4]
Test C[1]	80.34	74.36	76.07	71.15	75.21	64.1
Test C[2]	83.97	68.38	72.22	79.7	82.91	66.03
Test C[3]	54.7	79.06	58.33	79.27	56.41	77.78
Test C[4]	66.67	65.38	80.98	64.96	81.84	74.79

(a) two versus one camera

	Train C[1]	Train C[2]	Train C[3]	Train C[4]
Test C[1 2]	85.9	86.54	60.26	68.38
Test C[1 3]	86.11	75	83.33	72.44
Test C[1 4]	86.75	78.42	59.19	82.48
Test C[2 3]	80.13	85.26	80.34	70.73
Test C[2 4]	83.76	89.74	59.19	83.76
Test C[3 4]	70.51	64.32	80.56	82.48

(b) one versus two cameras

Figure 5.11: NB recognition accuracies with β combined features($\sigma = 0.05$, $knn = 20$). (a) shows recognition performance of a single camera system using action models trained with two cameras. (b) shows system performance with addition of a second test camera.

5.4 Conclusion and Discussion

Our method combines the confidence estimate of multiple cameras and various appearance features. This helps to solve mislabeled frames with a more accurate activity recognition. In our experiments, we show having a second view outperforms single view systems with a considerable amount.

We have shown that, by not reconstructing in 3D, we have gained really significant advantages in system architecture without losing much in performance. Views need not be calibrated to one another; training and test cameras do not need to overlap; cameras can drop in and out with little penalty; new features can be incorporated easily; and discriminative views can be exploited opportunistically. We believe our system is appropriate for large distributed camera systems with its flexible architecture.

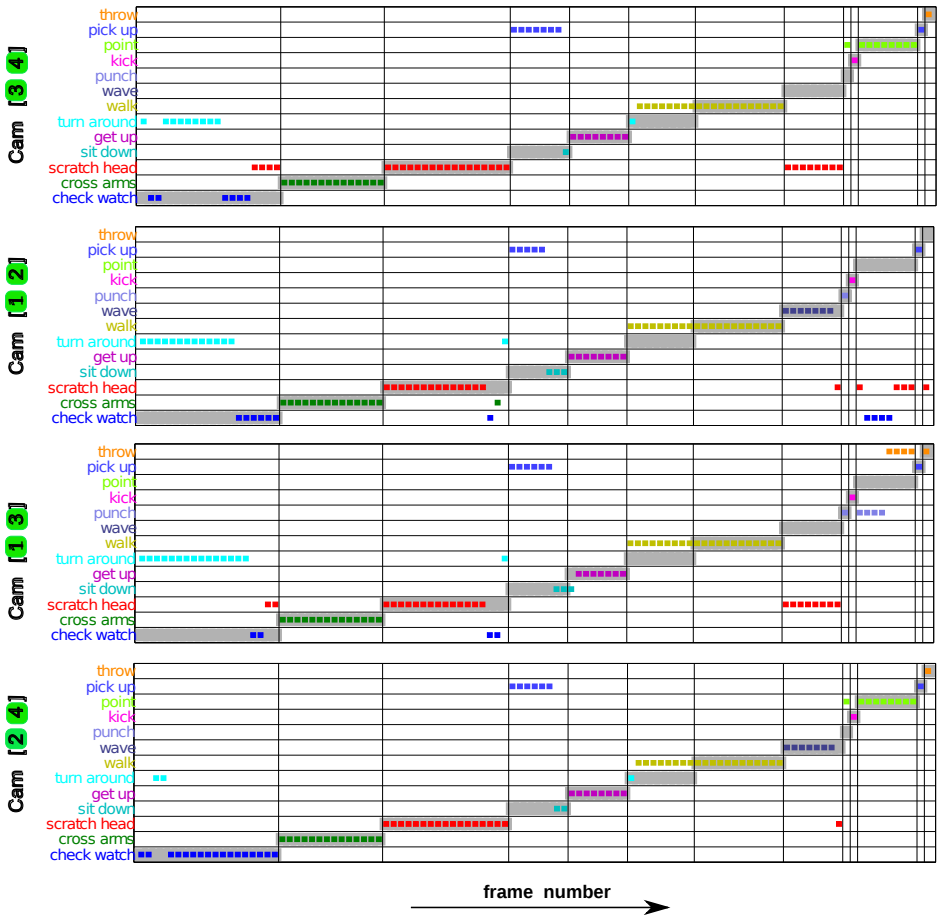


Figure 5.12: A case for *actor alba with* 1*st orientation* trained with cameras [2 4]. A window size of 23 frames moves over the action videos. Each window is given to HMMs and labeled with label of the highest likelihood one. In the first and second figures, system tests with cameras [3 4] and cameras [1 2]-*one shared* with train set [2 4]. In the third figure, system test with cameras [1 3] - *no shared* with train set [2 4]. Lastly, system tests with the same pair- *both shared*.

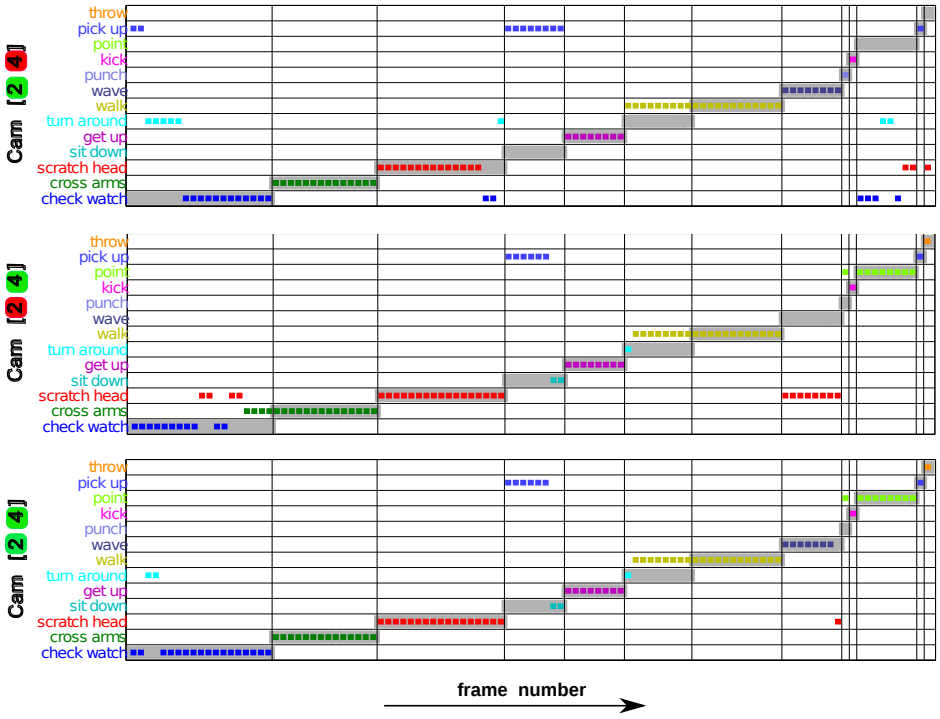


Figure 5.13: A case for *actor alba with* 1*st orientation* trained with cameras [2 4]. A window size of 23 frames moves over the action videos that corresponds to 1sec. recording. Each window is given to HMMs and labeled with label of the highest likelihood model. It is shown what happens when we switch one camera into off-line mode. In the first figure, camera [2] is online, in the second figure camera [4] is online and in the third both cameras are online.

Chapter 6

Conclusions, Discussions and Future Directions

Multiple-view camera systems have significant benefits over single-view systems. If there is more than one camera with overlapping views, the activity judgment is more accurate. If cameras do not overlap, they contribute to judgment in disjoint time intervals (e.g., while an actor is walking, the discriminative pose is captured by different cameras deployed in different positions). In this thesis, we explore two approaches in the activity recognition framework with multiple views. However, both methods can also be investigated for the recognition of other object types in videos and in images. Proposed methods match either volumes or individual silhouettes of human poses. Proposed descriptors can be applied to other articulated and rigid objects and can be queried to find nearest matches. Similarly, the proposed systems can be used for object retrieval and further to recognize their motions in videos.

6.1 Discussion on Volumes

First direction is to assume that there is a multi camera system with camera calibration information. If the camera amount and camera layout is enough for

reconstruction, volume is the most natural way of data fusion. If such system exists, there is no need to have complex models and complex recognition systems. Importance of key poses for action recognition is shown without using action dynamics. However, these studies and related shape descriptors are mostly on single-view activity recognition and we can not use them in 3D domain. Therefore, we propose two new pose descriptors and measure their performances under the action recognition framework. Here, volumes are converted to proposed pose descriptors and actions are shown as the concatenation of pose descriptors and various techniques are applied for action recognition. Our objective is to show the importance of pose descriptor to retrieve key poses and further understand actions in 3D domain.

We first model poses in 3D as the distribution of oriented cylinders. We use voxel grids constructed using silhouettes and search various size cylindrical kernels over it. Experiments show that the proposed descriptor is good for pose retrieval and action recognition. Pose alignment is one issue. We observe that when there are sufficient number of training video samples with various actors orientations, model can find a true match. Alternatively, one can use a technique like PCA-alignment. Oriented cylinders are applications of bag-of-features method using 3D kernels. It does not model the internal structure of parts. Low level pairwise features with relative location information like relative angles can be defined among 3D kernels. These features can be more discriminative when compared to the unary features and they encode the local patch alignments contrary to initial global pose alignment that is error prone to noise.

Second representation is based on the view-independent circular features extracted over horizontal layers of volumes. It goes better with encoding the relative ordering of features in consecutive layers contrary to histogramming the oriented cylinders. The proposed encoding is lossy, but it is robust to handle discriminative shape features over volumes. It is also fast to implement. We achieved state-of-the art results and again show that pose representation is important for the activity recognition performance. As the future direction, the study can be extended to various other basic features with view-invariant properties.

Activity recognition systems using volumes require special camera setups. Therefore, indoor and controlled environments allow the maintenance of such systems. When it is possible to setup such systems, 3D representations solve the issues regarding camera orientations and viewpoint variations, resulting in better activity recognition performance. We mostly investigate the importance of 3D pose descriptors while keeping activity recognition framework simple. In the future, the proposed systems can be advanced with complex activity recognition methods and higher performances can be obtained.

6.1.1 Future Work on Volumetric Features

As future work to volume based approaches proposed in this thesis, the following issues and directions can be investigated to improve recognition rates

- Pose alignment
- Pairwise features among oriented cylinders (resp circular features)
- Other view-invariant features
- Complex activity recognition methods

6.2 Discussion on Camera Architecture

The second direction focuses on a multiple camera architecture with desirable engineering features including camera setup, scalability, and activity performance. Volume reconstruction requires calibrated camera systems. However, such special setups are not easy to maintain. Individual frame judgments can be used while providing a more flexible system architecture. We propose a voting scheme incorporating label judgments from various camera frames and features. The proposed architecture reduces the system requirements and make the system maintenance easy. It does not require camera calibration, camera overlap, or hard frame synchronization, it is adaptable to any number of cameras at any time interval and

it can work in both indoor and outdoor environments. When it is thought as part of a distributed camera network, it has a low network load transferring only low level features, the activity votes. Extensive evaluation through experiments show that the system is feasible with many of these features with little performance penalty compared to state-of-the art volume matching approaches. Experiments also show having a second view outperforms single view with a considerable amount and if two camera positions have enough overlapping views for volume reconstruction, there is no need for volumes.

6.2.1 Future Work on Camera Architecture

Camera architecture is extensible in many ways, particularly for surveillance systems. It can be extended and explored for wide area indoor/outdoor applications. Some future directions can be as follows

- We assume that there exist a single person in the scene. But, there can be many people and there needs to have a method for frame correspondence among multiple views of a scene. A simple method based on shape and motion features can be used to match person regions in different views.
- Our study focuses on the recognition of simple human actions. The architecture can be tested on more complex activities. In this study, label votes are collected from frames but instead one can collect votes from body parts for complex activity labels.
- Our architecture can work with any kind of low level feature. We have used simple silhouette based appearance features and optical flow based motion feature, but it is worth to try other kind of features like part based HOG features or view invariant low level features.
- The system can be extended for distributed camera network where there exist multiple cameras with overlapping and non overlapping views. Tracking of the person in multiple cameras is also important to investigate.

6.3 Related Publications

- S. Pehlivan and P. Duygulu. 3D human pose search using oriented cylinders. In International Conference on Computer Vision Workshops, 2009.
- S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. Computer Vision and Image Understanding, 2011.
- S. Pehlivan and D. Forsyth. Multiple View Activity Recognition Without Reconstruction. PAMI, [Under Review]

Bibliography

[1] J. Aggarwal and Q. Cai. Human motion analysis: A review. In *Computer Vision and Image Understanding*, pages 428–440, 1999.

[2] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 459–468, 2006.

[3] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *Advances in Spatial Databases*, pages 207–226, 1999.

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, volume 2, pages 1395–1402, 2005.

[5] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001.

[6] L. Bottou and V. Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

[7] C. Canton-Ferrer, J. Casas, and M. Pardas. Human model and motion based 3d action recognition in multiple view scenarios. In *Proc. European Signal Processing Conf*, 2006.

[8] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. In *Workshop on Models versus Exemplars in Computer Vision*, volume 1, 2001.

[9] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *International Workshop on Analysis and Modeling of Faces and Gestures*, pages 74–81, 2003.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[11] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *International Conference on Computer Vision*, pages 726–733. IEEE, 2003.

[12] A. Farhadi and M. Tabrizi. Learning to recognize activities from the wrong view point. In *European Conference on Computer Vision*, pages 154–166, 2008.

[13] A. Farhadi, M. Tabrizi, I. Endres, and D. Forsyth. A latent model of discriminative aspect. In *International Conference on Computer Vision*, pages 948–955, 2009.

[14] D. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision 1*, 1(2-3):77–254, 2005.

[15] D. Gavrila and L. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[16] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3):334–352, 2004.

[17] K. Huang and M. Trivedi. 3d shape context based gesture analysis integrated with tracking using omni video array. In *Workshop on Vision for Human-Computer Interaction (V4HCI), in conjunction with CVPR*, pages 80–80, 2005.

[18] N. Ikizler and P. Duygulu. Histogram of oriented rectangles: A new pose descriptor for human action recognition. *Image and Vision Computing*, 27(10):1515–1526, 2009.

[19] N. İkizler and D. Forsyth. Searching for complex human activities with no visual examples. *International Journal of Computer Vision*, 80(3):337–357, 2008.

[20] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[21] I. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):172–185, 2011.

[22] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 156–164, 2003.

[23] R. Kehl and L. Gool. Markerless tracking of complex human motions from multiple views. 104:190–209, 2006.

[24] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, 2008.

[25] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[26] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[27] I. Laptev and P. Pérez. Retrieving actions in movies. In *International Conference on Computer Vision*, pages 1–8, 2007.

[28] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *International Conference on Computer Vision*, pages 444–451, 2009.

[29] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 246–253. IEEE, 2006.

[30] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

[31] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1996–2003, 2009.

[32] J. Liu and M. Shah. Learning human actions via information maximization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[33] J. Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3209–3216, 2011.

[34] G. Loy, J. Sullivan, and S. Carlsson. Pose-based clustering in action sequences. In *Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis*, pages 66–72, 2003.

[35] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[36] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 200(1140):269–294, 1978.

[37] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *European Conference on Computer Vision*, 2010.

[38] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *International Conference on Computer Vision*, 2009.

[39] I. Mikić, M. Trivedi, E. Hunter, and P. Cosman. Human body model acquisition and tracking using voxel data. *International Journal of Computer Vision*, 53(3):199–223, 2003.

[40] D. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. In *CGC 2nd Annual Fall Workshop on Computational Geometry*, 1997.

[41] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *European Conference on Computer Vision*, 2010.

[42] V. Parameswaran and R. Chellappa. Human action-recognition using mutual invariants. *Computer Vision and Image Understanding*, 98(2):294–324, 2005.

[43] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *Computer Vision and Image Understanding*, 115(2):140–151, 2011.

[44] R. Polana and R. Nelson. Detecting activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2–7, 1993.

[45] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990, 2010.

[46] L. Rabiner and B. H. Juang. *Fundamentals of speech recognition*. 1993.

[47] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[48] D. Ramanan and S. Baker.

[49] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[50] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[51] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[52] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition*, volume 3, pages 32–36, 2004.

[53] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM Multimedia*, pages 357–360, 2007.

[54] G. Shakhnarovich, T. Darrell, and P. Indyk. *Nearest-neighbor methods in learning and vision: theory and practice*. 2006.

[55] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[56] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[57] D. Tran and A. Sorokin. Human activity recognition with metric learning. *European Conference on Computer Vision*, pages 548–561, 2008.

[58] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: a survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[59] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *International Conference on Computer Vision*, pages 1–7, 2007.

[60] D. Weinland, M. Özuysal, and P. Fua. Making action recognition robust to occlusions and viewpoint changes. In *European Conference on Computer Vision*, 2010.

[61] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[62] P. Yan, S. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008.

[63] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 984–989, 2005.

[64] A. Yilmaz and M. Shah. Matching actions in presence of camera motion. *Computer Vision and Image Understanding*, 104(2):221–231, 2006.