2016

Multi-Modality Human Action Recognition

Yu Zhu

Multi-Modality Human Action Recognition

Yu Zhu

Dissertation submitted to the Statler College of Engineering and Mineral Resources at West Virginia University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy in

Computer Science

Guodong Guo, Ph.D. Committee Chairperson
Donald A. Adjeroh, Ph.D.
Hany H. Ammar, Ph.D.
Xin Li, Ph.D.
Hong-Jian Lai, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2016

Keywords: Action Recognition, Infrared Spectrum, Near Infrared Spectrum, RGB-D, Depth Data, Data Fusion

ABSTRACT

Multi-Modality Human Action Recognition

Yu ZHU

Human action recognition is very useful in many applications in various areas, e.g. video surveillance, HCI (Human computer interaction), video retrieval, gaming and security. Recently, human action recognition becomes an active research topic in computer vision and pattern recognition. A number of action recognition approaches have been proposed. However, most of the approaches are designed on the RGB images sequences, where the action data was collected by RGB/intensity camera. Thus the recognition performance is usually related to various occlusion, background, and lighting conditions of the image sequences. If more information can be provided along with the image sequences, more data sources other than the RGB video can be utilized, human actions could be better represented and recognized by the designed computer vision system.

In this dissertation, the multi-modality human action recognition is studied. On one hand, we introduce the study of multi-spectral action recognition, which involves the information from different spectrum beyond visible, e.g. infrared and near infrared. Action recognition in individual spectra is explored and new methods are proposed. Then the cross-spectral action recognition is also investigated and novel approaches are proposed in our work. On the other hand, since the depth imaging technology has made a significant progress recently, where depth information can be captured simultaneously with the RGB videos. The depth-based human action recognition is also investigated. I first propose a method combining different type of depth data to recognize human actions. Then a thorough evaluation is conducted on spatiotemporal interest point (STIP) based features for depth-based action recognition. Finally, I advocate the study of fusing different features for depth-based action analysis. Moreover, human depression recognition is studied by combining facial appearance model as well as facial dynamic model.

Acknowledgements

It is never easy to finish the Ph.D. study and write this dissertation. It would not have been possible to finish without the help of so many people in many ways.

I would like to express my great gratitude to my supervisor Dr. Guodong Guo, for his guidance and support not only for this dissertation but throughout the time of my whole Ph.D. study. Thinking back the time 6 yeas ago, when I firstly came to USA pursuing my graduate degree in computer science, it is Dr. Guodong Guo, whose expertise, understanding, generous guidance, suggestions, valuable comments and support made it possible for me to work on such a exciting topic and complete my Ph.D. study. It was a great pleasure working with him.

I am highly grateful to the colleagues Dr. Qin Wu, Alice Lai, Chao Zhang, Yan Li, Dr. Guowang Mu, for their kind help and assistance to complete my research projects during these years.

I am thankful to all my committee members Dr. Guodong Guo, Dr. Donald A. Adjeroh, Dr. Hany H. Ammar, Dr. Xin Li and Dr.Hong-Jian Lai. Thank you all for the suggestions for my research and encouraging me with your kind words when I picked the topic for my research.

I would like to express my gratitude to all my teachers and friends who directly or indirectly helped me to complete this dissertation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human action recognition aims at automatically recognizing ongoing actions performed by humans, from unknown videos or still images. Human action recognition has many important potential applications in various areas, e.g. video surveillance, HCI (Human computer interaction), video retrieval, gaming and security. Recently, human action recognition becomes an active research topic in computer vision and pattern recognition. Although automatic human action recognition is an important technique involved in many real-world applications, and many methods have been proposed, it is still a very challenging problem. Action recognition involves with so many challenging tasks and fields including signal processing, machine learning and photography, how to recognize human actions effectively still remains an open problem.

Among various problems related to human action recognition, such as gesture recognition [3], facial expression recognition[[4], and movement behavior recognition [5], in this dissertation, we firstly put our focus on the full-body actions, which often consists different motions and required considerations of head, hand, body and feet actions. Then we explore a special action problem: depression recognition. Depression recognition can be considered as a facial action analysis problem which is related to facial movement, facial expression, and upper body movements. Among different categorizations of action recognition problem, one way of classifying different action recognition problems is the different level of the human actions. We adopt the hierarchy structure proposed in Moeslund et al. [6], where they have divided human actions into: action primitive, action and activity. An action primitive is an atomic movement such as 'head turn around' and 'leg up'; and an action is considered consists of action primitives and describe a more complex movement, e.g. 'walking', and 'running'; activity contain a number of actions, give a more complex high level meaning of human movement that is performed. For

example, 'pickup', and 'writing on board'. We focus on actions that mainly performed by single subject, and full body movement, which excludes the gesture recognition.

In the recent decades, there are a number of research works on action recognition, and a number of approaches have been proposed to solve the human action recognition on video/images sequences [7] [8] [6] [9] [10] [11]. However, most of the approaches are focusing on the RGB images sequences, which can be viewed as given a sequence of images or videos, design a system that can automatically recognize what action is being performed. The data was collected by a single camera and the video sequence is the regular RGB image sequence. In such scenario, the system performance is highly related to the different occlusion, background, viewpoint, and even light conditions of the image sequences. These variations are commonly occurred in real life. Therefore, if more information can be provided along with the image sequences, more data modalities other than the single RGB video data can be utilized, one can better represent and recognize human actions than only utilizing the single model (RGB videos). For example, multi-view action recognition [12], used a number of cameras to collect human actions from different views. Several algorithms are also designed to use the information from different views to improve the performance of recognition system. Motion capture data [6], which is commonly used for animation and video games, where sparse movement information is extracted from the markers on human body by the optical system. Thus the temporal information can be provided more precisely through the positions of markers which can represent human skeleton in the videos. More recently, action recognition on RGB-D data is popular. RGB-D data collection, e.g. using the Kinect sensor [13], can provide the depth information other than the only RGB image sequences, and from the depth a human skeleton tracking algorithm is proposed to extract the skeleton joints positions [13]. Thus for the Kinect RGB-D data, three modalities (RGB, depth, and skeleton joints) can be utilized for the study of action recognition tasks.

Although a great progress has been made in human action recognition [7], [14], there are limitations in current practice. For instance, current action recognition studies are mainly in the visible spectrum, which constrains the application to the daytime or with sufficient illumination, since humans and human actions cannot be captured in the dark or evening using the visible spectrum. In application scenarios such as visual surveillance and HCI under weak illumination or in the dark, the visible light based action recognition cannot function any more. To deal with the limitations of visible spectrum and advance human action recognition to a new level, we introduce multi-spectra action recognition. Multi-spectra action recognition is a multi-modality action recognition problem, which uses the information from different spectrum, combines the different modalities, e.g. visible and infrared, and designs the algorithm to better utilizing the

properties of each modality for action recognition. In our work, not only action recognition can be performed on visible spectrum, but also the investigation of infrared/near infrared action recognition is meaningful. To the best of our knowledge, human action recognition in infrared and near infrared has not been well studied yet. Therefore in our work, we aim to explore novel methods dealing with this new problem and improving the action recognition performance in individual spectra, e.g. infrared and near infrared. Because different modalities contain different information and with different properties, which can always complement each other for human action recognition, so we also put our focus on how to combine the different modalities and design the system to improve the final recognition performance.

Recently, human action recognition using the depth data captured by the emerging RGB-D sensors has shown a great potential in action analysis, compared to the traditional color video-based approaches. Several features and/or algorithms have been proposed for depth-based action recognition. To have a better understanding of depth videos for action recognition, we advocate the study of fusing different features for depth-based action analysis. Although data fusion has shown great success in many areas, such as multimedia analysis and biometrics, it has not been well studied yet on whether the fusion is helpful or not for depth-based action recognition, or how to do the fusion properly. In this work, we study different fusion schemes comprehensively, using diverse features for action characterization in depth videos.

Moreover, we have studied a new problem of facial action analysis for estimating human depression diagnosis scores given the video clips. This study is related facial movement analysis, such as facial expression recognition, emotion recognition, which is also a special category in human action recognition. Recent study investigators have focused on the more challenging problem of analyzing facial action unit in psychopathology assessment. Researchers have used automatic facial action analysis to help the diagnosis of depression disorder. However, the performance of the accurately predicting depression disorder is still not very good. Therefore, we study to utilizing the deep learning methods, to automatically predict the depression values given the facial action videos. This is the first time that deep learning approach are proposed to the depression recognition problem, to the best of our knowledge. We propose a two stream deep learning network with joint tuning layers, and experimental results shown significant improvement compare to the previous approaches on two large databases.

This dissertation is organized as follows: A literature review of the human action recognition is firstly conducted in Chapter II, we introduce our new multi-spectral action database in this chapter. Then, Chapter III, Chapter IV, and Chapter V, are the second part of the dissertation, which is mainly focusing on multi-spectral human action

recognition. In Chapter III, a method based on adaptive SVM is proposed to deal with the visible to infrared action recognition problem. In Chapter IV, a new heterogeneous approach is proposed by utilizing correlation mapping and information theory for the newly defined action problem: heterogeneous action recognition. In Chapter V, a new method based on the histogram of sparse codes is proposed for action recognition in infrared spectra. Next, in Chapter VI, and Chapter VII is the third part of the dissertation, where the RGB-D action recognition is studied and new approaches are presented. In chapter VI, an evaluations based on spatiotemporal interest point features on several depth-based action databases are presented. In chapter VII, we study the data fusion methods for depth-based action recognition. Moreover, in Chapter VIII, we further the new action recognition problem, visual based facial depression recognition, where deep learning method is proposed dealing with the facial action unit in order to predict the depression value of the subject. Finally, summaries and conclusions are drawn in chapter IX.

Chapter 2

Literature review

2.1 Abstract

Recently, human action recognition becomes a very active and important topic in computer vision. However, most of the works that have been done to recognize human actions are based on visible spectrum, e.g. RGB videos. In this chapter, we first take a brief literature review of the research on human action recognition, then we propose a new problem for human action recognition called multi-modality human action recognition. We have collected a new action database which captured human actions in three different spectra, i.e. visible, near infrared, and infrared. This database not only provides a benchmark for new methods on action recognition beyond visible, but also intriguers other interesting research problems on the multi-modality human action recognition.

2.2 Introduction

Human action recognition, which has a wide range of applications, attracts great research attentions in computer vision and pattern recognition. Human action recognition is to recognize actions performed by humans from unknown videos or still images. Human action recognition is very useful in many applications in various areas, e.g. video surveillance, HCI (Human computer interaction), video retrieval and game playing. Although many methods have been proposed, it is still a very challenging problem. Involved with so many challenging tasks and fields including signal processing, machine learning and photography, how to recognize human actions effectively still remains an open problem.

Among various problems related to human action recognition, such as gesture recognition [3], facial expression recognition [4], and movement behavior recognition [5], in this work,

we put our focus on the full-body actions, which often consists different motions and required considerations of facial, hand, body and feet actions. Another way of classifying different action recognition problems is the different level of the human actions. We adopt the hierarchy proposed in Moeslund et al. [6] they have divided human actions into: action primitive, action and activity. An action primitive is an atomic movement such as 'head turn around' and 'leg up'; and an action is considered consists of action primitives and describe a more complex movement, e.g. 'walking', and 'running'; activity contain a number of actions, give a more complex high level meaning of human movement that is performed. For example, 'pickup', and 'writing on board'. We focus on actions that mainly performed by single subject, and full body movement, which excludes the gesture recognition.

In the recent decades, there are a number of works on action recognition, and a bunch of methods have been proposed to solve the human action recognition on video/images sequences [7] [8] [6] [9] [10] [11]. However, most of the approaches are focused on the RGB images sequences, which can be viewed as given a sequence of images or videos, design a system that can automatically recognize what action is being performed. The data was collected by a single camera and the video sequence is the regular RGB image sequence. In such way, the system performance is highly related to the different occlusion, background, viewpoint, and even light condition of the image sequences, which is commonly occurred in real life. Thus, if more information can be provided along with the image sequences, more data models other than the single RGB video data can be utilized, so that one can better represent and recognize human actions than only use the single model (RGB videos). For example, multi-view action recognition [12], use a number of cameras to collect human actions from different views. And algorithms are designed to use the information from different views to improve the performance of recognition system. Motion capture data [6], which is commonly used for animation and video games, where sparse movement information is extracted from the markers on human body by the optical system. Thus the temporal information can be provided more precisely through the positions of markers which can represent human skeleton in the videos. And more recently, action recognition on 3D data is popular. 3D data collection, e.g. using the Kinect sensor [13], can provide the depth information other than the only RGB image sequences, and from the depth a human skeleton tracking algorithm is proposed to extract the skeleton joints positions [13]. Thus for the Kinect 3D data, three modalities (RGB, depth, and skeleton joints) can be utilized for action recognition tasks.

Although a great progress has been made in human action recognition [7], [14], there are limitations in current practice. For instance, current action recognition studies are mainly in the visible spectrum, which constrains the application to the daytime or with

sufficient illumination, since humans and human actions cannot be captured in the dark or evening using the visible spectrum. In application scenarios such as visual surveillance and HCI under weak illumination or in the dark, the visible light based action recognition cannot function any more. To deal with the limitations of visible spectrum and advance human action recognition to a new level, we introduce multi-spectra action recognition. Multi-spectra action recognition is a multi-modality action recognition problem, which uses the information from different spectrum, combines the different modalities, e.g. visible and infrared, and designs the algorithm to better utilizing the properties of each modality for action recognition. Because different modalities contain different information and with different properties, which can always complement each other for human action recognition, so we put our focus on how to combine the different modalities and design the system to improve the final recognition performance.

2.3 Literature review

In the literature, several survey papers have been proposed recently for the area of human action recognition and human motion analysis [7] [8] [6] [9] [10] [11] [14]. In this section, we first present a general overview of different methods on action recognition. Secondly, several commonly used human action datasets are presented. Then we reviewed several spatiotemporal approaches. Finally, some recent work on depth based action recognition is presented.

There are several surveys for action recognition and human motion analysis. For human action recognition, in the early survey [15] by Bobick, a taxonomy of movement recognition, activity recognition and action recognition is used. Latter in Aggrawal and Cai's [16] work, three categories: body structure analysis, tracking and recognition are used. Gavrila [8] uses a taxonomy of 3D approaches, 3D approaches and recognition. Moeslund and Granum [17] use a taxonomy based on subsequent phases in the pose estimation. In the survey by Turaga el al., they focus on the higher-level recognition of human activity [9]. In Liang Wang's work [11], they uses similar taxonomy like [16], detection, tracking, and understanding for human action analysis. In Ronald Poppe's survey [7], action recognition methods are divided into model-based and model free categories, where model-based is generative and model-free discriminative approaches, respectively. More recently, Daniel Weinland in [10] reviewed vision based methods in three parts, action representation, segmentation and recognition. There are also some surveys focus on different problems in human motion analysis and action recognition, e.g. in [3], they review the gesture methods, in [6] a survey is conducted for the motion

TABLE 2.1: Statistics of different human action datasets.

Datasets	# Actions	# Subjects	# S./A.	# Videos
KTH [19]	6	25	25	2391
Weizmann [20]	10	9	9	90
UCF Sports Action [21]	9	N/A	N/A	200
Hollywood2 Actions [22]	12	N/A	N/A	2517
IXMAS [12]	13	12	12	36
MSRAction3D [23]	20	10	10	576
UCF50 [24]	50	N/A	N/A	6618
HMDB51 [25]	51	N/A	N/A	6849
Sports-1M [26]	487	N/A	N/A	1M

capture systems, and recently in [18], the reviews the works on human motion analysis on depth imagery.

Following Turaga el al.'s work [9], a generic action recognition system can be viewed as proceeding from sequences of images to a higher-level interpretation in a series of steps. The major steps involved are: (1) Input the video or images sequences. (2) Extract low level feature (3) Action description from low-level features (4) Semantic interpretation or action classification.

2.3.1 Representative Action Databases

There are many public available human action datasets for action recognition and analysis, which allows for the comparison of different approaches and gives more comprehensive insight of different methods. In this subsection, we introduce and describe several most widely used datasets for action recognition. Table 1 shows some statistics of these datasets.

2.3.1.1 KTH human action dataset

The KTH human action dataset [19] has six actions collected in this dataset. There are totally 25 different subjects performed each action in four different scenarios. The six actions are: walking, jogging, running, boxing, hand waving and hand clapping. Four different scenarios are: outdoors, indoors, outdoors with zooming, and outdoors with different clothing.

2.3.1.2 Weizmann human action dataset

The Weizmann human action dataset [20] recorded 10 different actions (walk, run, jump, gallop, sideways, bend, one-hand wave, two-hands wave, jump in place, jump jack and skip). Each action is performed by 10 subjects. In this dataset, the foreground silhouettes are also provided.

2.3.1.3 UCF Sports Action dataset

In the UCF sports action dataset collected by [21], 9 actions are collected and there are totally 150 video sequences. And the bounding boxes of the human are also provided. The 9 actions are: diving, golf swinging, kicking, weightlifting, horseback riding, and running, skating, swinging a baseball bat and walking.

2.3.1.4 Hollywood 2 human action dataset

There are 12 classes of actions and 10 classes of scenes in over 3669 video clips in the Hollywood2 human action dataset [22]. The video clips are collected from 69 movies and the aim is to provide human actions in realistic and challenging settings. There is a huge variety of performance of the actions, and also there are occlusions, camera movements, and background changes make this dataset more challenging.

2.3.1.5 IXMAS multi-view dataset

IXMAX multi-view dataset [12] is for view-invariant human action recognition. There are 5 cameras installed to acquire the actions from 5 different views (one top view and 4 side views). There are 13 daily-live action performed each 3 time in this dataset. The number of subjects is 11 and they can choose freely positions and orientations when performing actions. Also they provide the silhouettes with the images sequences.

2.3.1.6 MSRAction3D dataset

The Action3D dataset [23] is collected by a depth sensor, which contains 20 action types and 10 subjects. Each subject performed each action 2 or 3 times. The 20 action types are: high arm wave, horizontal arm wave, hammer, hand catch, forward punch , high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up & throw. Those actions were chosen to cover various movements of arms, legs, torso and their

combinations. There are totally 567 depth sequences in this dataset. And this dataset also has a skeleton sequence file for each depth sequence.

2.3.1.7 UCF 50 dataset

UCF50 action dataset [24] is collected from YouTube videos for action recognition with 50 different action categories. UCF50 dataset is an extension of UCF11 which has 11 action categories. Totally there are 6618 video clips. The dataset is challenging with camera motion, object appearance and pose, scale, viewpoint, and illumination conditions, etc. The 50 action categories collected in UCF50 dataset are: Baseball Pitch, Basketball Shooting, Bench Press, Biking, Biking, Billiards Shot,Breaststroke, Clean and Jerk, Diving, Drumming, Fencing, Golf Swing, Playing Guitar, High Jump, Horse Race, Horse Riding, Hula Hoop, Javelin Throw, Juggling Balls, Jump Rope, Jumping Jack, Kayaking, Lunges, Military Parade, Mixing Batter, Nun chucks, Playing Piano, Pizza Tossing, Pole Vault, Pommel Horse, Pull Ups, Punch, Push Ups, Rock Climbing Indoor, Rope Climbing, Rowing, Salsa Spins, Skate Boarding, Skiing, Skijet, Soccer Juggling, Swing, Playing Tabla, TaiChi, Tennis Swing, Trampoline Jumping, Playing Violin, Volleyball Spiking, Walking with a dog, and Yo Yo.

2.3.1.8 HMDB51 dataset

HMDB51 dataset [25] is collected mostly from movies and a small portion from other public databases like YouTube and Google Videos. The total number of video clips in this dataset is 6849, which are divided into 51 actions classes. Those action categories contain different action type, such as facial actions, body movements, body movements with object interaction or human interaction. The 51 actions are: smile, laugh, chew, talk, smoke, eat, drink, cartwheel, clap hands, climb, climb stairs, dive, fall on the floor, backhand flip, handstand, jump, pull up, push up, run, sit down, sit up, somersault, stand up, turn, walk, wave, brush hair, catch, draw sword, dribble, golf, hit something, kick ball, pick, pour, push something, ride bike, ride horse, shoot ball, shoot bow, shoot gun, swing baseball bat, sword exercise, throw, fencing, hug, kick someone, kiss, punch, shake hands, and sword fight.

2.3.1.9 Sports-1M dataset

More recently, a more larger databases for action analysis is proposed in [26] called Sports-1M dataset. This dataset contains 1 million videos belonging to 487 different action classes. There are about 1000-3000 videos per class and approximately 5% of

the video are annotated with more than one class. The classes are generally becomes fine-grained by the leaf level from internal nodes, such as Aquatic Sports, Team Sports, Winter, Sports, Ball Sports, Combat Sports, and Sports with Animals.

2.3.2 Methods: Feature extraction and description

In the research of action recognition, it is arguably that the feature extraction and description are the most important steps. Feature extraction and description aims to extract a certain type of pattern, from the input data, which is used to represent the characteristics of the data and can be used for the classification task. We follow the [6]'s work, usually the features extracted from the data can be divided into two categories: global representations and local representations. The global representations encode the video sequence as a whole. Specifically, human subject in the video sequence are first localized by detection or tracking, then the region of interest is encoded as a whole as a descriptor. Local representations describe the video sequence as a set of patches. Firstly the spatiotemporal interest points are detected in the video. Then the local patches around these interest points are considered for the feature description. Finally, the features from all the local patches are used representing the human actions. We briefly discuss the representative works on the global representation and give more details on the local representations.

2.3.2.1 Global representations

In Bobick and Davis's work [27], they extract silhouettes from a single view and aggregate differences between subsequent frames of an action sequence. This results in a binary motion energy image (MEI) which indicates the motion occurs.

Also latter a motion history image (MHI) is constructed where the pixel intensities are a recency function of the silhouette motion. In the matching phase, they adopt the Hu moments [28] which are known to yield shape discrimination in a translation and scale invariant manner.

Blank et al. [20] have proposed an estimation method for motion flows from a 3-D space-time volume to recognize human actions. They first stack silhouettes over a video sequence to form a space-time volume. Then a Poisson equation is used to derive local space-time saliency and orientation features. Each local feature gives a local match score during the matching step. By aggregating these scores over all the local patches, the overall correlation between the templates is computed. Their system can recognize various types of human actions.

In the work proposed by Shechtman and Irani [29], motions from 3-D space-time volumes are used to recognize human actions. They have computed a 3-D space-time video template, and utilized the correlation to measure the similarity between an observed video volume and the template volumes.

Ke et al. [30] modeled human activities using segmented spatiotemporal volumes. They proposed to apply a hierarchical meanshift to cluster similarly colored voxels, and obtains several segmented volumes. The motivation is to represent the actor volume segments automatically, and measure their similarity to the action model.

Rodriguez et al. [21] have analysis space-time volumes by synthesizing filters: The adopted the maximum average correlation height (MACH) filters, to solve the action recognition problem. A synthesized filter is generated for each action class, and action classification is performed by applying the synthesized MACH filter and analyzing the response on the new action sequence.

Recent work has shown that dense sampling can help improve the action recognition result over sparse interest points. Wang et al. [31] in their work proposed dense trajectories and motion boundary descriptors for action recognition. Their idea is to obtain the feature trajectories by tracking the points in video through a dense grid in each frame. So that the quality of trajectories is increased. Specifically, the features points are firstly sampled on a grid for each spatial scale. Then tracking is applied frames by frame in a dens optical flow field. The trajectory shape is then represented by relative point coordinates. Finally, three different descriptors are computed along the trajectory cells.

2.3.2.2 Local representations

Space-time trajectories. Trajectory-based approaches interpret an action as a set of space-time trajectories. A subject is generally represented as a set of points corresponding to the skeleton joints positions. And the joints positions are recoded as the human performing the action, constructing a 3D or 4D representation of the action. In the literature there are several approaches used the space-time trajectories to represent and recognize actions. In Sheikh et al. [32]'s work, they proposed to represent human actions by utilizing a set of 13 joint, and the trajectories are computed in a 4-dimensional space. A projection is used to obtain normalized trajectories of an action, so that the view-invariant similarity between two sets of trajectories can be measured. In Yilmaz and Shah [33]'s work, a methodology to compare action videos obtained from moving cameras is proposed, they used the multi-view geometry between two actions, and the set of joint trajectories in the 4 dimensional space are used for the action recognition.

Spatiotemporal features. Among various methods on action recognition for RGB video sequences, STIPs (Spatiotemporal Interest Points) features have shown promising performance on many action datasets [14]. In this section, we summarize the commonly used spatial-temporal interest point detectors as well as the feature descriptors.

Interest points detectors. Harris3D detector was proposed on [34]. It locates the spatial-temporal volumes where have large variations along not only space but also temporal directions in video sequence. A spatial temporal second-moment matrix is used to model video sequence f,

$$\mu = g\left(\cdot; \sigma_i^2, \tau_i^2\right) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix},$$

where $g(\cdot)$ is a Gaussian weight function and L is the convolution of f with a spatiotemporal Gaussian kernel. The interest point locations are determined by computing the local maxima of the response function:

$H = \det(\mu) - k \cdot trace^3(\mu)$.

Cuboids [5] detector computes the interest points location by the local maxima of the response function R. R is defined as: $R = (I * g * hev)^2 + (I * g * hod)^2$, where g is the 2D Gaussian smoothing kernel, hev and hod are a quadrature pair of 1D Gabor filter, which are defined as $hev = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $hev = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$.

Willems et al. [35] proposed Hessian detector which measures the strength of each interest point using a Hessian matrix. The response function is defined as $S = |det(H)|$ where H is a Hessian matrix,

$\begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$,which is similar to the Harris3D detector.

2.3.2.3 Local feature descriptors

Given a set of interest point locations, various feature descriptors can be used to represent the local space-time content. Given the spatial scale σ and temporal scale τ at each interest point location, a local volume is then extracted and employed to create the feature vector.

Klaser et al. extends the HOG to 3D and proposed the HOG3D descriptor in [36]. It computes the histogram of 3D gradient orientations. They used integral videos to

efficiently compute the gradients and combined both shape and motion information at the same time.

HOG/HOF descriptor is proposed by Laptev et al. [37], they use the combination of the histogram of gradient (HOG) and histogram of optical flow (HOF) accumulated from the local volume, as the feature vector.

Cuboids descriptor is proposed by [5] along with the Cuboids detector. For each given STIP point $(x; y; t; \sigma; \tau)$ a feature descriptor is computed as a 3D patch centered at $(x; y; t)$. The gradient at each spatiotemporal location is computed in each cuboid and the histogram is used as the feature vector. Also PCA can be applied to reduce the dimensionality.

The extended SURF (ESURF) descriptor [35], is proposed with the Hessian detector, which is an extension of SURF [38]. For each local volume, the feature vector is computed by storing the sum of uniformly sampled responses of Haar-waveletes along three directions.

2.3.3 Action classification

When an action is represented as feature vectors for a single frame or a video sequence, human action recognition becomes a classification problem. In many cases, feature vector are high-dimensional, to reduce the computational complexity, dimensionality reduction can be applied before classification. PCA [39] is common linear dimensionality reduction approach. For classification stage, k-Nearest Neighbor classifier [40] use the distance between the feature vector of testing set and those in the training set. The test label is determined by the most common label among the k closest training samples of the testing sample. Support Vector Machines (SVMs), is considered as a discriminative classifier which focus on separating data rather than modeling them [41]. SVM learn a hyper-plane in the feature space that described by a number support vectors. SVM has been widely used for action recognition, especially with local representation, such as histogram of bag-of-words. When the feature is represented according to certain moments in time, the action recognition can be viewed as temporal state-space models. Dynamic time warping (DTW) [42], Hidden Markov Model (HMM) [43] and Conditional random fields (CRF) [44] are the representative approaches dealing with these problems.

2.3.4 Deep learning method

With the success that deep learning approaches have shown promising performance on many areas in computer vision and pattern recognition, very recently, deep learning

methods are introduced and employed to human action analysis. In [26] , deep convolutional neural network is proposed to classify large action data which contains 1 million video clips from 487 action classes. In their work, they proposed to utilize the AlexNet and modified the architecture to handle the temporal information within the action videos. Besides, they also proposed a multi-resolution architecture with a two stream manner. Experimental result on the Sport-1M dataset shown their deep learning approach obtained promising performance.

2.4 A New Multi-Spectra Action Databases

We collected a new human action dataset for the multi-model human action recognition, called Multi Spectra Human Action (MSA) dataset. The dataset contains two different modalities, visible light and thermal infrared. Our dataset contains 1800 video clips, 30 different actions performed by 30 different persons and sampled in 2 different spectrums. In this experiment, whole dataset is sub-sampled the first 4 seconds of each video clip, with original resolution (frame width x height: 320×240) and frame rate (30 fps). The 30 actions collected in our dataset are: boxing, checkingwatch, drinking, exercisejumping, fixinghair, handclapping, handwaving, helpsignaling, horizontalstretching, kicking, knockingondoor, marching, movingbox, openingdoor, organizingtable, pickingup, reading, running, sittingstanding, squating, telephoning, typing, usingmicrowave, usingremotecontrol, verticaljumping, walking, wipingboard, wipingtable, writing, and writingonboard. Figure showed some example images of the actions in this dataset. Figure **??** shows some example images of this action dataset.

Human action recognition has been widely explored using RGB videos or images. Among various approaches, space-time based methods such as space-time volumes, spatiotemporal features and trajectories are popular. In this section, we evaluate the performance of various spatiotemporal features on our multi-model dataset, to provide a better understanding that how the spatiotemporal features perform on the data other than visible light.

2.5 Research Problems

Our MSA database is unique. Many research problems can be studied and explored on the MSA database, which may not be possible on some previous action databases. To promote and advance the research on human behavior analysis both broader (multiple modalities) and deeper (with a large number of actions), we captured the MSA database

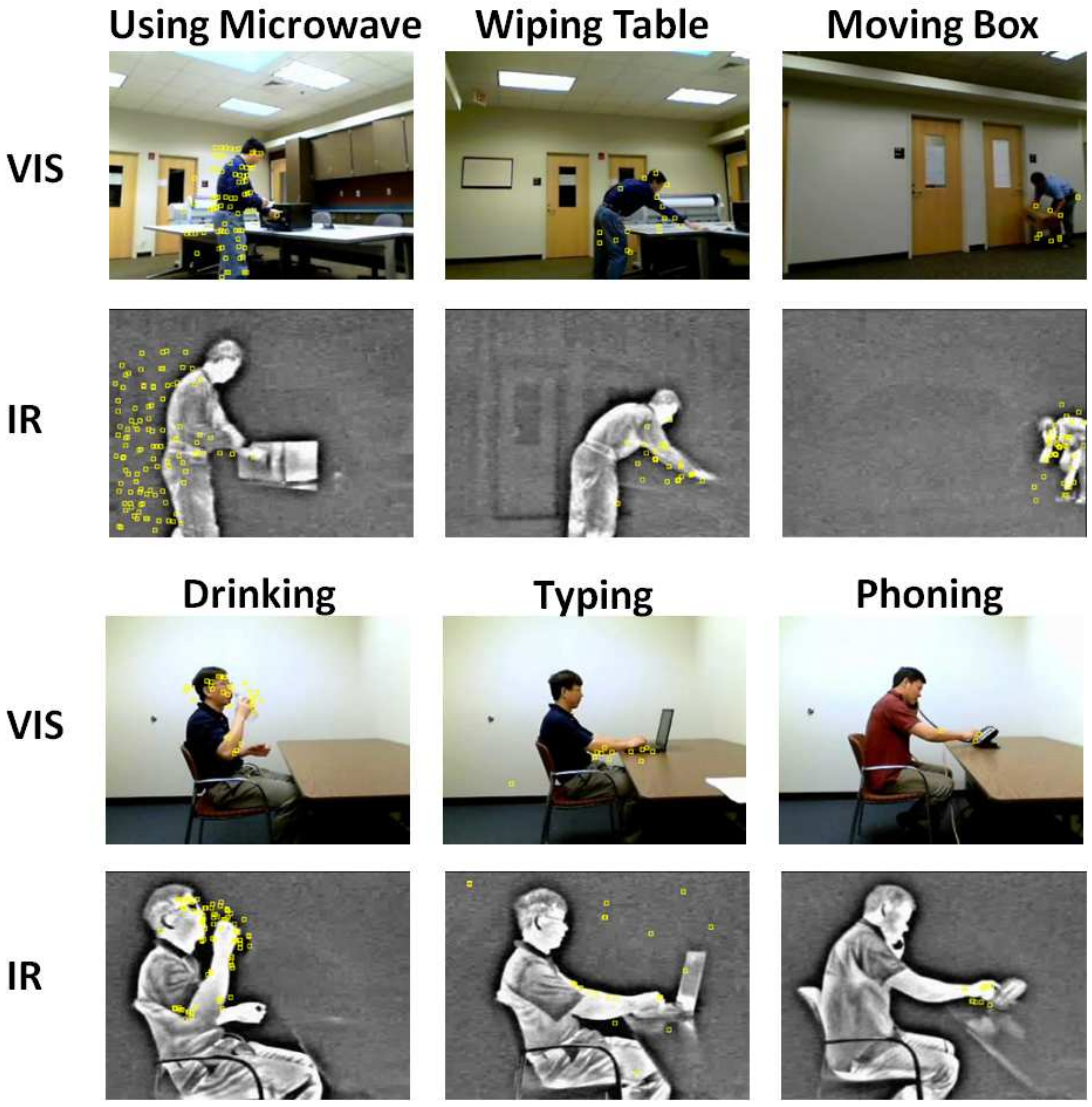

FIGURE 2.1: Examples of the actions in the dataset, and the interest point locations detected are also showed (yellow dots). From the images one can see that the STIPs detector can detect interest point in the infrared images but the locations are very different from visible spectrum.

and provide it to researchers for further study. The variety of spatiotemporal features will be provided too. Our benchmark studies of single spectrum and cross-spectral action recognitions build a basis, and may inspire new research explorations. To make it explicit, we propose several potential research directions that are worthy of investigation based on the MSA database.

2.5.1 Statistical Analysis for Action Recognition

Since the MSA database has a relatively large number of actions and thirty people per action, it can be used to do some statistical analysis for action recognition. For

instance, one can use the MSA database to study: How does the number of training examples (persons) influence the recognition performance? Can a method perform action recognition with a small number of learning examples? How many examples are needed to learn an action? How to determine a good number of training examples for a specific algorithm? Which methods perform better for a small number of actions and which methods perform better for a large number of actions? Is it possible to share "common" features among different actions?

2.5.2 Thermal Infrared Action Recognition

The MSA database can be used for action recognition in the thermal IR spectrum. In our evaluation, the recognition performance in IR is comparable to the visible light. But the recognition accuracy is still not high when all 30 actions are used. One can develop new methods for action recognition in IR and evaluate on the MSA database.

2.5.3 Cross-Spectral Action Recognition

Based on our benchmark study in Section IV, the cross spectral action recognition has extremely low performance. It is demanding to explore advanced methods to improve the performance of cross-spectral action recognition.

2.5.4 Multispectral Data Fusion

All three spectra are available in the MSA database. It is possible to use the MSA database to study multispectral data fusion for action recognition. For example, it is interesting to investigate if the action recognition performance can be improved when all spectra are fused together, and how much the improvement could be. Data fusion is usually a useful scheme to improve decision making, but it is not clear yet on how useful the multispectral data fusion could be in human action recognition.

2.5.5 Human Object Interaction

In the MSA database, there are about half of the actions containing human object interactions. For instance, "moving box" or "opening door" has the human interacting with different objects. While most previous databases have actions performed by the humans only, such as walking, jogging, waving, boxing, etc. studying human object interaction may be helpful to improve the action recognition performance, and also improve the human detection and object detection accuracies [8], [20].

2.5.6 Cross-Domain Action Recognition

Our MSA database has a large number of actions. It may serve as a kind of "dictionary" of typical actions. Consequently, those typical actions can be learned on the MSA, and then applied to other scenarios for either action recognition or detection [45]. For example, apply to action analysis in videos from the Internet [6] or movies [11]. Since the actions in our MSA database are captured in a laboratory scenario, which is different from the acquisition conditions for those actions in the Internet or movies, we call it cross-domain action recognition.

First a brief literature review of human action recognition in video sequences is presented. Then a novel multispectral database is introduced to human action analysis. Comparing to the previous action databases, the database is broader (multiple modalities: VIS and IR), and larger (30 actions). It can serve as a common database to evaluate action recognition methods and promote new research. Finally, we have presented some interesting research directions that can be explored based on the multi-spectra action database. Our work will inspire new research efforts and advance for action recognition. In the next chapters, we focus on developing more powerful methods recognize human actions from the infrared spectrum and explore the cross-spectral action recognition problem.

Chapter 3

A Study on Visible to Infrared Action Recognition

3.1 Abstract

Human action recognition is important in image and video processing with many applications. With the development of sensor technology, different cameras can be used for action acquisition, e.g., infrared cameras. Is it possible to adapt the visible light action recognizers to a new modality or domain? In this chapter, we study the feasibility to adapt the action recognizer learned from visible light spectrum to infrared. A preliminary result is obtained on a large database based on an adaptive learning method, demonstrating the potential to perform cross-spectral action recognition.

3.2 Introduction

Human action recognition is important for image and video processing and understanding [7] [46]. Action recognition has many applications, e.g., video surveillance, video retrieval, and human-computer interaction (HCI) [7] [46]. As a pattern recognition problem, a typical approach to action recognition contains two major steps: feature extraction and classification [7]. In feature extraction, there are a number of methods to obtain space-time representation of the action videos. Among various space-time features, the spatiotemporal interest point (STIP) based methods usually perform well for visible light action recognition [14]. The STIP features can reduce the redundancy of raw video data significantly to derive a concise representation. In developing action classifiers, the support vector machines (SVM) [47] are usually adopted because of the good performance compared with other classifiers.

Significant progress has been made recently in action recognition [46], however, the state-of-the-art approaches to action recognition are mainly in the visible light spectrum. On the other hand, with the sensor technology and hardware development, human actions can be acquired by other sensors different from the visible light cameras. For example, the thermal infrared cameras can be used to capture human actions. Actually the infrared spectrum has been used for face recognition [48], but seldom for action recognition [49]. A nice property of the infrared spectrum is that it can capture humans in the dark, which is very useful for night surveillance or HCI under dim light.

It could be interesting to study the difference between the actions captured by different sensors. More importantly, if the actions captured in the visible light can be "transferred" to other domains or modalities for recognition, it will have great impact in practice, since a number of existing action videos in the visible light spectrum could be utilized to learn action recognizers and then work on a different domain. As a result, there may be no need to collect a large training set using the new sensor. Instead, the adapted classifiers can still be used for action recognition with the new sensor. It could be too costly to manually collect a large number of action examples and build separate recognizers for every sensor modality or domain. Actually, transfer learning has become an active research topic in machine learning [50], but it has seldom been exploited for action recognition with different modalities.

In this chapter, we study the problem of action recognition from visible to infrared. Our goal is to understand the feasibility of using the training examples in visible light spectrum to help action recognition in infrared. We study empirically the difference between the extracted features from the same actions but in two different spectra, and then develop methods to build relations between actions in the two different spectra.

In the following, we introduce our methods for visible to infrared action recognition, which is based on classifier adaptation. Then we briefly describe the spatiotemporal features for action representation in both spectra in Section 3.4. The experiments are conducted on a large database, and finally, we draw conclusions.

3.3 Adapting the SVM to Infrared Action Recognition

To study the feasibility of executing action recognition from visible to infrared, we investigate the method based on learning the difference between the classifiers in the two different spectra using A-SVM. Based on this exploitation, we can understand how well it can be performed for the novel task: visible to infrared action recognition.

Suppose we have a training set of features extracted from visible light action videos, and the corresponding labels, denoted by $\mathcal{D}^V = \{(\mathbf{x}_k^V, y_k^V)\}_{k=1}^m$. We first train a standard SVM for the visible light action classification with the following optimization problem [47],

$$\min_{\mathbf{w}} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{m} \xi_k \tag{3.1}$$

$$s.t. \;\; \xi_k \geq 0, \;\; y_k^V \mathbf{w}^T \phi(\mathbf{x}_k^V) \geq 1 - \xi_k$$

where ξ_k are the slack variables to deal with non-separable examples in training, and $\sum_k \xi_k$ measures the total classification error. T is the transpose, and $\phi(\mathbf{x})$ is the feature mapping to a high dimensional space with the kernel function $K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$. The weight vector $\mathbf{w}$ is the solution of the SVM to determine the decision boundary between two classes. C is a parameter to balance the two items in the objective function.

The above SVM classification is for a two-class problem. For our multi-class action recognition, we use pair-wise comparisons to combine the binary classification results to get the final decision.

After learning the visible light action classifier $f^V(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$ based on the optimization given in (3.1), now we consider how to adapt the classifier $f^V(\mathbf{x})$ to deal with infrared action recognition.

To adapt the visible light action classifier to an infrared action classifier, $f(\mathbf{x})$, we need to have a small number of labeled action examples in the infrared spectrum, $(\mathbf{x}_i, y_i) \in \mathcal{D}^{IR}$, for $i = 1, 2, \cdots, n$. The adaption of the SVM classifier $f^V(\mathbf{x})$ to $f(\mathbf{x})$ is based on learning a delta function [51], which can model the difference between the two classification functions, i.e.,

$$f(\mathbf{x}) = f^V(\mathbf{x}) + \delta f(\mathbf{x}) = f^V(\mathbf{x}) + \mathbf{w}^T \phi(\mathbf{x}) \tag{3.2}$$

where $\mathbf{w}$ are to be estimated to determine the delta function. Now, the objective function to optimize for the adaptive SVM (or A-SVM) [51] becomes

$$\min_{\mathbf{w}} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i \tag{3.3}$$

$$s.t. \;\; \xi_i \geq 0, \;\; y_i f^V(\mathbf{x}_i) + y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i$$

Note that the objective function is similar to the standard SVM, but the weight vector $\mathbf{w}$ has a different meaning, because it determines the function $\delta f(\mathbf{x})$ rather than $f(\mathbf{x})$.

Therefore, the objective function in Eq. (3.3) seeks a decision boundary that is close to the boundary of the visible action classifier in the feature space, and also separates the action features extracted from the infrared spectrum. The parameter C balances the two items, i.e., the visible action classifier and the training examples in infrared. When C is larger, the influence of the visible action classifiers will be smaller.

The objective function in Eq. (3.3) can be written as the Lagrangian function,

$$\begin{aligned} L_P = \tfrac{1}{2} \parallel \mathbf{w} \parallel^2 + C \textstyle\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} r_i \xi_i \\ - \textstyle\sum_{i=1}^{n} \alpha_i \left(y_i f^V(\mathbf{x}_i) + y_i \mathbf{w}^T \phi(\mathbf{x}_i) - (1 - \xi_i) \right) \end{aligned} \tag{3.4}$$

where $\alpha_i \geq 0$, $r_i \geq 0$ are Lagrange multipliers. To minimize the function L_P, one can compute the derivative with respect to $\mathbf{w}$ and ξ and set the derivatives to zero. Then the results are

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i) \\ \alpha_i &= C - r_i \end{aligned} \tag{3.5}$$

The Lagrange dual objective function of Eq. (3.4) is given by

$$L_D = \sum_{i=1}^{n} (1 - \lambda_i) \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3.6}$$

where $\lambda_i = y_i f^V(\mathbf{x}_i)$. The solutions α can be computed by maximizing L_D under the constraints $0 \leq \alpha_i \leq C$. It can be solved by a quadratic programming (QP) problem solver [47]. Given the solution $\hat{\alpha}$, the new decision function can be obtained,

$$f(\mathbf{x}) = f^V(\mathbf{x}) + \sum_{i=1}^{n} \hat{\alpha}_i y_i K(\mathbf{x}, \mathbf{x}_i) \tag{3.7}$$

where $(\mathbf{x}_i, y_i) \in \mathcal{D}^{IR}$. The adapted classifier $f(\mathbf{x})$ can be considered as augmented from the visible light action classifier $f^V(\mathbf{x})$ with support vectors from the subset of infrared actions.

3.3.1 Correlation between Visible and Infrared Actions

The canonical correlation analysis (CCA) is a standard method to learn the correlation between two modalities. Here we verify if the CCA method can be used for our visible to infrared action recognition and compared with the A-SVM method represented by Eq. (3.7). CCA is to describe the linear relation between two multidimensional variables as

the problem of finding basis vectors for each set such that the projections of the two variables on their respective basis vectors are maximally correlated [52] [53].

Let p-dimensional x and q-dimensional y denote the two sets of real-valued zero-mean random variables (i.e., $x \in R^p$ and $y \in R^q$). Let $p \times N$ matrix X be the data matrix of the first set, and $q \times N$ matrix Y be the data matrix of the second set. The CCA method computes two projection vectors, $w_x \in R^p$ and $w_y \in R^q$, such that the correlation coefficient

$$\rho = \frac{w_x^T X Y^T w_y}{\sqrt{(w_x^T X X^T w_x)(w_y^T Y Y^T w_y)}} \tag{3.8}$$

is maximized [52] [53]. Since ρ is invariant to the scaling of w_x and w_y, CCA can be formulated equivalently as

$$\max_{w_x, w_y} w_x^T X Y^T w_y, \tag{3.9}$$

subject to $w_x^T X X^T w_x = 1$, and $w_y^T Y Y^T w_y = 1$.

It can be shown [53] that w_x can be obtained by solving the following generalized eigenvalue problem,

$$X Y^T (Y Y^T)^{-1} Y X^T w_x = \lambda X X^T w_x, \tag{3.10}$$

where λ is the eigenvalue corresponds to the eigenvector w_x. It has also been shown [53] that multiple projection vectors under certain orthonormality constraints consist of the top l eigenvectors of the generalized eigenvalue problem in (3.10). The corresponding w_y can be found [53] using $w_y = \frac{(Y Y^T)^{-1} Y X^T w_x}{\sqrt{\lambda}}$.

In our study, X represents the data matrix, and Y represents the label space. After the dimension of data X is reduced, we use a least square fitting to build the relation between the dimension reduced feature and label Y. Then the prediction of Y for the test data is based on the least square fitting result. This simple least square fitting method can work well, and is also applied to other CCA extensions, which will be introduced in this section later.

The CCA method has shown its success in some image processing problems, e.g., image annotation [53], action classification [54], and face recognition [55][56]. However, it is unknown whether the CCA method can be used for cross-spectral action recognition or not. Here we exploit the CCA method to measure the correlations between visible and infrared actions, and compare the CCA based method with the A-SVM method in our task of visible to infrared action recognition.

3.4 Spatiotemporal Features

In action recognition, the space-time features are usually extracted rather than using raw videos [7]. The learning methods introduced in Section 3.3 are based on extracted features. In visible light action recognition, it is very popular to use the space-time interest point (STIP) based representations [14]. Among various STIP features, the cuboid detector and descriptor proposed by Dollár et al. [5] can work well for visible light action representation. In our study, we use the cuboid detector and descriptor for both visible light and infrared actions, and we measure how different the extracted features will be from the same actions using the same method.

The cuboid detector [5] computes a response function $R = (I * g * h_{even})^2 + (I * g * h_{odd})^2$, where I is the image frame, g is a 2D spatial Gaussian function for smoothing, and h_{even} and h_{odd} are a quadrature pair of 1D Gabor filter for temporal filtering. Interest points are detected by the local maxima of the response function R. Given the detected locations, the cuboid descriptor can characterize the local space-time content. The detector and descriptor can be combined together to characterize the actions locally in each video. The cuboid descriptor [5] is based on the local spatiotemporal response R with given space-time parameters. The gradients computed at each pixel within the space-time patch can be concatenated into a vector to describe the local content. To reduce the dimensionality, the principal component analysis (PCA) can be used on the descriptions.

After feature detection and description, we perform a clustering of the spatiotemporal features into a limited number of clusters, e.g., 250, for the training action videos in each spectrum. These cluster centers are used as the "keywords" for actions [14] in each spectrum. Histogram features are then extracted for each action video based on counting the number of keywords appeared in each action video. The SVM is used as the classifier for action recognition.

3.5 Experiments

To study the visible to infrared action recognition experimentally, a large database is used. We extract features and investigate different methods for action recognition.

3.5.1 Database

To study the new problem called cross-spectral action recognition, we use a large database with two spectra, the visible light and thermal infrared. Action videos are captured by

FIGURE 3.1: Some examples in our action database. The two rows are the visible and infrared actions: running, drinking and kicking.

two cameras, the Logitech Quickcam Pro 5000 and the Thermal-Eye 300D, for the two spectra. The two cameras are close to each other so that each action can appear in both cameras' field of view (FOV). The database has 30 actions of 30 people captured by the two cameras. In total, there are 1,800 (=30×30×2) videos in the database. The number of action categories 30 is large, even compared to some existing visible action databases. For instance, the popular KTH database [19] contains six actions, and the Weizmann database [20] has 10 actions only. In the database, there are 30 actions: fixinghair, handclapping, horizontalstretching, marching, squating, usingmicrowave, usingremotecontrol, wipingtable, boxing, kicking, movingbox, pickingup, running, openingdoor, organizingtable, walking, exercisejumping, wipingboard, knockingondoor, verticaljumping, drinking, handwaving, helpsignaling, reading, telephoning, typing, writing, sittingstanding, writingonboard, and checkingwatch. The captured videos have different lengths, each has about 21 seconds in average. Some example pairs are shown in Fig. 3.1. We plan to make the database available to other researchers in order to advance the field of research on action recognition.

3.5.2 Action Recognition Results

Because of the randomness of selecting subjects and the K-means clustering, the experiments are conducted 10 times to obtain the final accuracies by taking average. Five individuals were randomly selected with all the 30 actions for learning, and the remaining 25 persons for testing.

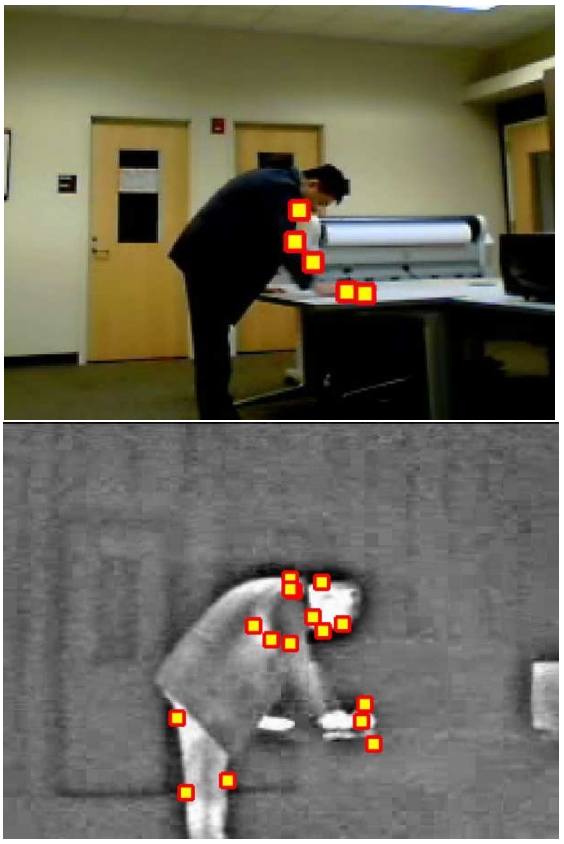

FIGURE 3.2: Spatiotemporal interest points detected in the same actions but from two spectra - Left: visible light; Right: infrared.

First, we study the difference between the features extracted from actions performed by the same persons and captured at the same time but from two different spectra: visible light and infrared. The feature extraction methods are the same, based on the cuboid detector and descriptor. We found that the detected interest point locations are different in the two spectra, as one example shown in Fig. 3.2, where the STIP locations are displayed. We use the actions in the visible spectrum to train the action classifier, and then apply to the infrared actions for testing. The accuracy is 5.2%, which is very low, although it is higher than a random guess (about 3.3% accuracy). For comparison, a bar graph of the accuracy is shown (as the baseline result) in Fig. 3.4. This result demonstrates that the spatiotemporal features extracted from the two spectra are very different, although the motions are almost the same performed by the same individuals. The difference in spatial appearance influences action feature extraction. Thus a direct matching between the two spectra cannot get a satisfactory result.

Second, we exploit the CCA method to measure the correlations between visible and infrared actions, and compare the CCA based method with the A-SVM method in our task of visible to infrared action recognition. The canonical correlation analysis (CCA) [52][53] is a standard method to learn the correlation between two modalities. The CCA

TABLE 3.1: Accuracies for different number of training samples. (±) means The standard deviation of accuracy.

	5 Subjects	10 Subjects	20 Subjects
Baseline	5.2% (±1.5%)	5.0% (±1.1%)	4.0% (±0.8%)
CCA	27.8% (±1.8%)	31.2% (±3.1%)	47.2% (±9.2%)
A-SVM	50.4% (±1.7%)	54.6% (±1.6%)	61.2% (±1.1%)
VIS Only	53.5% (±1.6%)	57.6% (±1.8%)	65.7% (±2.8%)
IR Only	37.7% (±1.4%)	42.8% (±1.4%)	46.7% (±1.9%)

can learn two sets of bases for action features extracted from the two spectra of the same individuals. Then the two sets of bases are used to project the features in each spectrum, respectively. About 50 basis vectors are used for each set of bases. The transformed features are used to train the SVM (with RBF kernel) for action recognition. Using the transformed features of the same testing data as used for the baseline, we get an accuracy of 27.8%, which is higher than the baseline result, as shown in Fig. 3.4.

Third, we investigate the adaptive SVM method for action recognition from visible to infrared. We want to validate if the adaptive SVM method can really adapt the classifiers learned from the visible light features to the infrared. The standard SVMs are leaned in the visible light training actions, and then adapted to the infrared actions based on the learning examples in infrared. Then the test examples in the infrared spectrum are used for recognition. In A-SVM, RBF kernel is used and cross-validation scheme is applied for parameter selection. In specific, the parameters used are: $\gamma = 0.004$, $cost = 1$, $\gamma = 0.002$, $cost = 1.4$, and $\gamma = 0.004$, $cost = 0.6$ for the 5, 10 and 20 subjects group. The accuracy for 5 subjects group obtained 50.4% based on the A-SVM method, which is significantly higher than the correlation-based methods using CCA, as shown in Fig. 3.4. The confusion matrix is shown in Fig 3.3. One can see misclassification for similar actions impacted the overall recognition rate, e.g, "writing" was recognized incorrectly to the action "reading" and "typing", which actions have similar motion and ambiguous to the system. But the accuracies are improved comparing to IR only, e.g. typing 14%(IR) to 25%(Cross), reading 14%(IR) to 23%(Cross).

In order to get a better comparison, experiments with different number of training samples were conducted. The training data from visible and infrared are balanced, which means same number of training samples from both spectra are used. The experimental results are shown in Table 3.1. From the results one can see that, the recognition rate increases when more samples are used for training. Also the A-SVM method performs consistently well when different number of training and testing sample are used.

In summary, the spatiotemporal features extracted from the two spectra are quite different, therefore the direct matching cannot work for VIS to IR action recognition. The

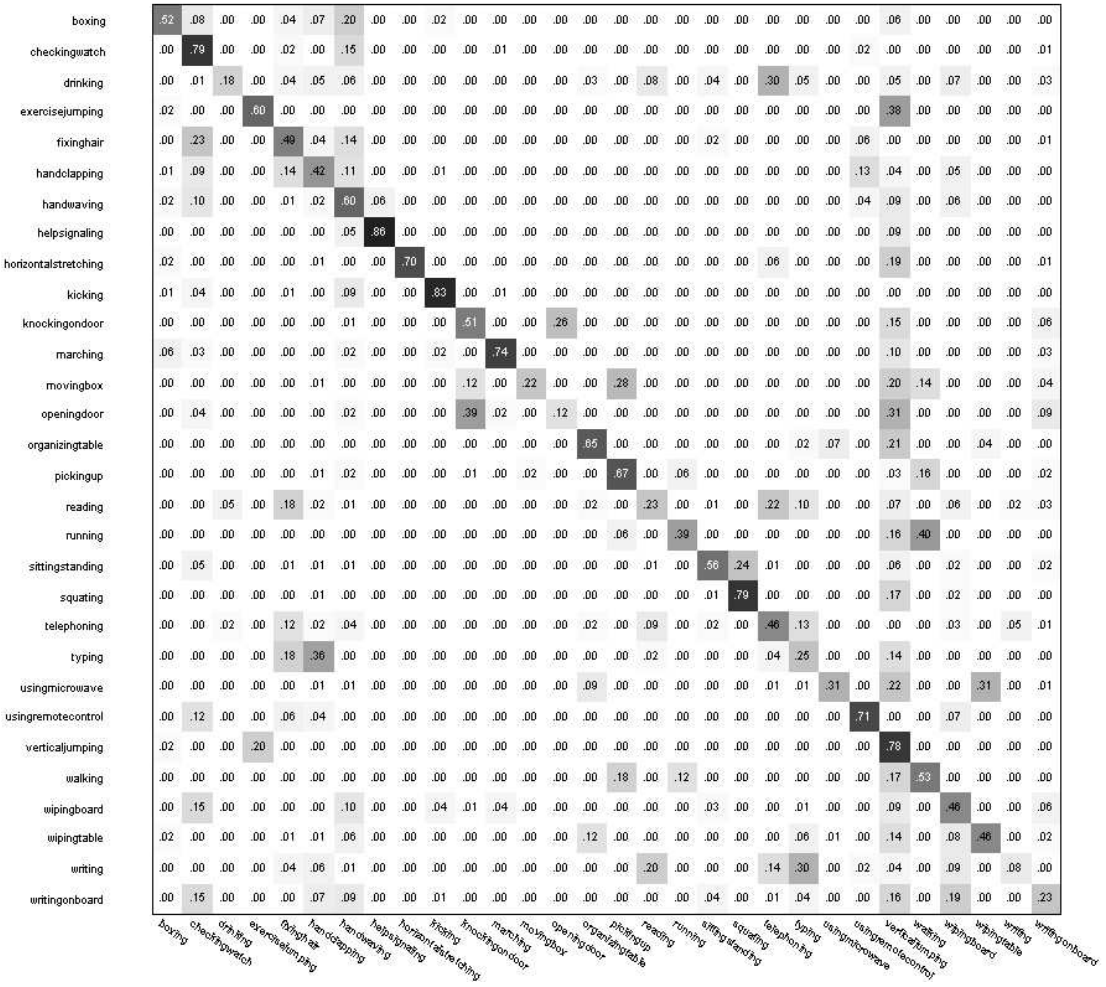

FIGURE 3.3: The confusion matrix shows the recognition result from visible to infrared, when using the A-SVM method and 5 subjects are used for training.

correlation learning via CCA can improve the accuracy over the baseline, but its performance is still significantly lower than the A-SVM. The higher accuracy of A-SVM shows that the classifiers learned from the visible actions can be adjusted to recognize actions in infrared spectrum. On the other hand, the accuracy is still not very high. Although the cross-spectral recognition accuracy by the A-SVM is higher than the pure IR (37.7%), but is lower than the pure visible action recognition (53.5%) in our experiment. Note that our problem is different from the traditional action recognition [14]. There are 30 actions in our study, and we use a small number of training examples under the transfer learning framework, while there are more training examples and less actions in traditional action recognition [14].

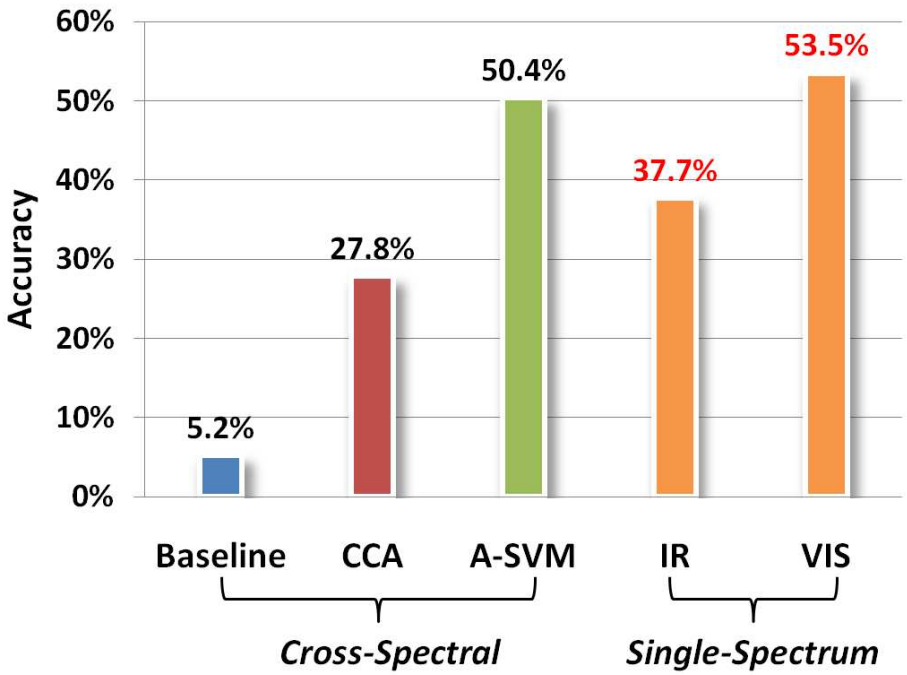

FIGURE 3.4: The action recognition results. The baseline result is based on a direct matching (concatenating VIS and IR features in training). The CCA method can learn the correlation between VIS and IR and improves the result, while the A-SVM is significantly better than the CCA. Single-Spectrum used the same 5 subjects training, 25 subjects testing only from one spectrum.

3.6 Conclusions

We have studied the problem of visible to infrared action recognition. The study is performed on a large database with 30 actions of 1,800 videos in two spectra, which makes our study statistically meaningful. We have shown that the spatiotemporal features extracted from the two spectra are quite different, thus a direct matching cannot perform well. A correlation based approach with the CCA can improve the accuracy over the direct matching, while the adaptive SVM method can perform significantly better. Therefore, our preliminary study demonstrates the potential that the action classifier can be learned from the visible light actions and adapted to the infrared for action recognition. In future research, we will explore other adaptation methods to further improve the performance.

Chapter 4

Heterogeneous Action Recognition: From Visible to Thermal Infrared

4.1 Abstract

Human action recognition is a very active research topic in computer vision and pattern recognition. A number of methods have been proposed for action recognition based on visible light imagery. However, visible light cameras cannot capture human subjects or their motion under dark illumination conditions. The performance of action recognition may also degrade due to complex background or illumination variations. In contrast, thermal infrared (IR) offers a different source of information. It has advantages over the visible light, since the IR is insensitive to illumination changes, and can work in either day or night. We propose to study heterogeneous action recognition (HAR) from visible to thermal infrared imagery. The aim is to learn the spectral relations that can maximize the mutual information between the two spectra. Two schemes are investigated, one is based on spectral correlation mapping, and the other is based on learning the transitions on manifold to represent the connections between the two spectra. And discriminative learning technique is used to further improve the correlated features. In HAR, our aim is to utilize the visible light action patterns to help improve the infrared action recognition. Comprehensive experiments are conducted on a multi-spectral action database and promising results are obtained, showing the feasibility of HAR.

FIGURE 4.1: Example images of visible (top) and infrared (bottom) of different actions. The actions shown are: hand clapping, wiping board, opening door, and using microwave.

4.2 Introduction

In recent years, human action recognition has become a very active research topic in computer vision and pattern recognition. By recognizing ongoing actions performed by human subjects from videos or images, applications in various domains can be developed with practical usage, e.g., advanced user interfaces, video surveillance, gaming, and security [57]. A number of approaches have been proposed for action recognition [57] [9] [46]. Various human action databases have also been collected and used to validate action recognition performance [58] [46].

However, current action recognition focuses mainly on the visible light or the RGB/intensity imagery. The visible light cameras may not work in dark illumination, resulting in degradation of the action recognition performance. Also, the developed methods for visible action recognition might not be optimal for the case that the illumination varies or the background is noisy. The fact that thermal infrared cameras can capture human subjects in poor light or dark conditions, can overcome the drawbacks of the light-sensitive visible cameras. Thermal infrared cameras can sense temperature emissions from human, which is an intrinsic property and independent of illumination conditions. Thus thermal infrared imagery offers a different source of information, which can work on either day or night. In several areas of computer vision, infrared imagery has been studied and shown promising results, e.g., face recognition [48] [59] [60], facial expression recognition [61], human detection and tracking [62], and human gait analysis [49], [63]. In this paper, we propose to study the heterogeneous human action recognition problem. Motivated by the heterogeneous face recognition problems [64] [65] [66], in particular, we study the scenario that is recognizing the human actions in the infrared spectrum, by utilizing the actions from the visible light. The proposed visible to infrared action

recognition has many practical applications, e.g., surveillance and night vision systems. But the problem has not been well studied, to the best of our knowledge. In our opinion, the exploration of heterogeneous human action recognition, could enrich the research on action recognition and provide a new perspective for human action analysis.

The most challenging issue in visible to infrared action recognition lies in the fact that actions are captured from different devices (visible and thermal infrared cameras), with a large disparity. Different from heterogeneous face recognition problems, e.g., in [66] and [65], where the target is to match facial images of the same subjects from different modalities (e.g., visible, NIR, or sketch), the spatial-temporal action data are more challenging because the same action may be performed by different subjects with different appearance and moving speed. To bridge the inter-modality gap, we first explored an approach by adapting the visible data to the infrared for action recognition [1], which has two key steps: firstly two SVM classifiers are learned respectively on visible and infrared data. Then a delta function is learned to model the difference between the two classifiers. In this work, rather than reducing the modality gap at the classification stage, we explore approaches in the feature level from an information theoretic perspective. The idea is to learn the relationships between the visible light and infrared, maximizing the mutual information [67] between the two different spectra of action patterns.

In order to maximize the mutual information and reduce the gap between the visible light and thermal infrared modalities for the purpose of action recognition, we explore two schemes. The first is based on learning the correlations between the two spectra of spatiotemporal action patterns. The correlated features are further fed into a discriminative learning method, such that the features that are mapped to a new space become more separable. The second scheme is to learn the action transition from visible to infrared, based on learning on a manifold. The action patterns from visible and infrared are represented by the projections on a series of subspaces along the geodesic path, so that the actions patterns from different spectra can be "connected" on the geodesic path, resulting in a new action feature representation.

The organization of the paper is as follows: We address the HAR problem from an information theoretical perspective in Section 4.3. Then we introduce the scheme of learning the spectral correlations and the discriminative mapping in Section 4.4. The scheme that learns the transitions from visible light to infrared on the manifold is presented in Section 4.5. The experiments are conducted in Section 4.7. Finally, we draw some conclusions.

4.3 Heterogeneous Action Recognition from an information theoretic perspective

In this paper, we propose to address the problem, called heterogeneous action recognition (HAR). HAR is about recognizing human actions across different modalities, e.g., visible light and infrared. The challenge of HAR is that the discrepancy of action patterns from different modalities.

In HAR, the modality gap could be reduced by making the action patterns from different modalities "closer", so that the actions from one modality can be used to help the recognition in another modality.

Specifically, denote the features extracted from the visible light and infrared actions as $F^{vis}$and F^{ir}. We first measure the modality gap between the action patterns in two spectra, in a quantitative manner. By reducing this modality gap, the new action patterns could represent the action better and is expected to be "transferred" more easily from one modality to the other. To achieve this goal, we use mutual information to measure the relationships between different modalities. Maximizing the mutual information means the reduction of the gap between two modalities.

In information theory, Mutual information (MI) [67] is a measure of the uncertainty of X with the knowledge of Y. MI is a nonnegative symmetric measure, being equal to zero if and only if X and Y are statistically independent. In other words, the statistical correlation between two random variables X and Y can be measured by the MI. In our case, the MI provides a quantitative measure of the strength of correlation between different modalities, e.g., the higher MI indicates that the two modalities are more related. In this way, by computing the MI before and after applying any learning methods, one can tell the usefulness of the methods quantitatively, for heterogeneous action recognition. That is, whether the MI is increased or not.

Entropy [67], denoted as $H(X)$, quantifying the uncertainty of distribution of X, can be defined as: $H(X) = -\Sigma_{x \in X} p(x) \log p(x)$. Formally, mutual information can be written as:

$$I\left(X; Y\right) = H\left(X\right) + H(Y) - H\left(X, Y\right), \tag{4.1}$$

where $H(X)$ is the marginal entropy of X, and $H\left(X, Y\right)$ is the joint entropy.

Because the features extracted from action data are discrete (e.g., histogram features), when compute the MI, we estimate the histogram distribution using a fixed number of bins. The probability $p(X = x)$ can be estimated by the frequency of occurrence of $X = x$, divided by the total number of bins.

In our problem, we want to lean a correlation through mapping, coding, or projection of the features between two spectra, so that the mutual information between the correlated features $\hat{F}^{vis}$, $\hat{F}^{ir}$ can be increased. The mutual information is denoted as:

$$I\left(\hat{F}^{vis};\hat{F}^{ir}\right) = H\left(\hat{F}^{vis}\right) + H\left(\hat{F}^{ir}\right) - H\left(\hat{F}^{vis},\hat{F}^{ir}\right). \tag{4.2}$$

For example, we can measure the gap between action patterns extracted from visible light and infrared using the MI, which obtained 0.41 bit (see more details in Sec. 4.7.3). It is expected that by using the proposed schemes, the MI can be increased so that the features from visible and infrared are "pulled closer". In [66] and [65], CCA [53] is derived as the solution of their optimization function, which is represented using the MI and the KL-divergence [67] for heterogeneous face recognition. Here, we propose to explore two schemes based on correlation and manifold learning, to maximize the MI for our heterogeneous action recognition. It is new to investigate these kinds of methods for heterogeneous action recognition. Our experimental results (see Section 4.7) show that, CCA is not optimal for our HAR problem. The schemes we explore here achieve higher mutual information and better recognition performance than the CCA formulation as in [66] and [65].

Given the criteria of mutual information maximization, we will present two schemes for HAR in the following sections.

4.4 Reducing the Modality Gap based on Spectral Correlation

In order to maximize the mutual information and reduce the modality gap, our first scheme is to learn correlations, between actions in two spectra. To learn the correlations, we explore a specific method, called Partial Least Squares (PLS) [68], although other methods may be used as well. Our new investigation is to examine if the PLS method can increase the MI and improve the performance for HAR. After the correlation mapping, a discriminative learning is applied to improve the recognition performance further.

4.4.1 Learning Correlation between Heterogeneous Action Patterns

The objective of learning correlation is to map the heterogeneous actions onto a common subspace, where the modality gap is expected to be minimized. The Partial Least Square (PLS) [68] method is investigated. The PLS can model relations between sets of observed

variables with latent variables. It can also reduce the dimensionality and show success in some computer vision problems such as face recognition[56], pedestrian detection[69] [70], and age estimation [71]. It is a new investigation of the PLS for heterogeneous action recognition. Our goal is to explore if the PLS technique can maximize the mutual information between two action modalities.

Given two sets of variables, e.g., action patterns in the visible and infrared spectra, let $\mathbf{V}$ denote the $(n \times N)$ zero-mean matrix of the first data set (e.g., visible features) and $\mathbf{F}$ denote the $(n \times M)$ zero-mean matrix of second data set (e.g., infrared features). PLS decomposes the two matrices in the following form:

$$\mathbf{V} = \mathbf{T}\mathbf{P}^T + \mathbf{R_v} \tag{4.3}$$

$$\mathbf{F} = \mathbf{U}\mathbf{Q}^T + \mathbf{R_f} \tag{4.4}$$

where $\mathbf{T}$ and $\mathbf{U}$ are the $(n \times p)$ matrices of latent vectors, p is the number of extracted latent vectors. Matrices $\mathbf{P}$ and $\mathbf{Q}$ having the size $(N \times p)$ and $(M \times p)$ are the matrices of loadings, while matrices $\mathbf{R_v}$ and $\mathbf{R_f}$ of size $(n \times N)$ and $(n \times M)$ represent the matrices of residual. The PLS method, which is in its classical form [72], finds the weight vector w and c, such that:

$$\begin{aligned} [cov\,(t,u)]^2 &= [cov\,(Vw, Fc)]^2 \\ &= \max_{|r|=|s|=1} [cov\,(Vr, Fs)]^2 \end{aligned} \tag{4.5}$$

where $cov(t,u) = t^T u/n$, denotes the sample covariances between the score vectors t and u. After extraction of the score vectors t and u, the matrix V and F are deflated by subtracting their rank-one approximation based on t and u. This process iteratively repeats until convergence, or achieves a desired number of extracted weight vectors.

The objective function of PLS can be rewritten as:

$$max[cov\,(t,u)]^2 = max\,var\,(t)\,[corr(t,u)]^2 var\,(u)\,, \tag{4.6}$$

where $var\,(t)$ denotes the sample variance, $corr(t,\,u)$ denotes the sample correlation, which represents the criterion of maximizing correlation and the variance in both V and F modalities. As a result, it is expected that by applying the PLS projection, the mutual information which measures the statistical correlation between visible and infrared data could be increased.

Note that in Eqn. (4.5), when F is a 1-dimensional vector representing the class label of the data set V, PLS then only learns a set of projection vectors $W = w_1, w_2, ..., w_N$, by maximizing the covariance between the data and corresponding class labels. In that

case, the PLS acts as a dimension reduction method, which is similar to PCA and LDA. In this paper we denote PLS_d the case where PLS is applied to one single modality for dimensionality reduction, where F is the set of class labels, and denote PLS_c the case where F is the action features of the visible or infrared, aiming at learning the correlations between actions in two spectra.

After learning the PLS model, the feature vectors v_i and f_i from different data sets (e.g., visible light and infrared) are projected with the learned weight vectors w and c, respectively. The $(1 \times p)$ latent vectors z_i^1 and z_i^2 can be obtained as the correlated feature vectors. In this way, we can bring the visible data along with the infrared data in the training stage, and it is expected to learn a better action classification model for infrared action recognition.

By using a correlation-based learning approach, the learned projections can make the same actions captured from different spectra "closer" to each other in the new feature space. In order to represent heterogeneous actions better using discriminative mapping and improve the separation between different actions, we explore the Lorentzian Discriminant Projection (LDP) [73] method, which is based on Lorentzian geometry and extended to the general relativity.

4.4.2 Increase the Discriminative Capability

Since the PLS method mainly focuses on the correlation between different modalities, it may not separate heterogeneous actions well. It might be a good idea to perform discriminative mapping to enhance the separation between different actions.

Several methods could be used for discriminative projection, e.g., Linear Discriminant Analysis (LDA) [74], Marginal Fisher Analysis (MFA) [75], Maximum Margin Criterion (MMC) [76], Locality Preserving Projection (LPP) [77], etc. The Lorentzian Discriminant Projection [73] method is a relatively new method, and has not been explored extensively. so it is interesting to examine the LDP method for the new problem of HAR.

The main idea of LDP is to discover the intrinsic local discriminant and global geometric structure of the data, by constructing a Lorentzian manifold and learning the metric tensor on the manifold. Compared to other methods such as the MFA, the LDP does not require many parameter settings, such as the graph weight matrix including the number of inter-class and intra-class neighbors in MFA.

Let the input data set be $S_{\mathbf{x}} = \{\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_m\}, \mathbf{x}_i \in \Re^n$, and $L_{\mathbf{x}} = \{L_1, L_2, \dots L_m\}$ the class labels, m is the number of the samples. The goal of LDP is to transform the

original high-dimensional space $\Re^n$ into a low-dimension, i.e., $S_z \subset \Re^d$, where $d \ll n$. The output of LDP is a new feature set S_z, with a projection matrix $\mathbf{U}$.

First let set $S_{\mathbf{z}_i}$, in which the elements share the same class label with the sample $\mathbf{z}_i$, be $S_{\mathbf{z}_i} = \left\{\mathbf{z}_i, \mathbf{z}_1^i \ldots \mathbf{z}_{m_i-1}^i\right\}$, m_i is the number of samples with the same class label as $\mathbf{z}_i$. Then the presentation of a m_i -tuple point $\mathbf{d}_{y_i}$can be defined as: $\mathbf{d}_{y_i} = \left[d\left(\mathbf{z}_i, \mathbf{z}_1^i\right), d\left(\mathbf{z}_i, \mathbf{z}_2^i\right), \ldots d\left(\mathbf{z}_i, \mathbf{z}_{m_i-1}^i\right), d\left(\mathbf{z}_i, \bar{\mathbf{z}}\right)\right]^T$ where $d\left(\mathbf{z}_i, \mathbf{z}_j\right)$ is the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$, and $\bar{\mathbf{z}} = \frac{1}{m}\Sigma_{i=1}^m \mathbf{z}_i$, is the geometric centriod of $S_{\mathbf{z}}$.

In such a way, $\mathbf{d}_{y_i}$can be viewed as points sampled from a manifold $\mathbb{L}_1^{m_i}$furnished with a Lorentzian metric tensor g_l:

$$\begin{aligned} g_l\left(\mathbf{d}_{y_i}, \mathbf{d}_{y_i}\right) &= \mathbf{d}_{y_i}^T \mathbf{G}_i^l \mathbf{d}_{y_i} \\ &= tr\left(\left(\mathbf{Z}_i \mathbf{D}_i\right) \mathbf{G}_i^l \left(\mathbf{Z}_i \mathbf{D}_i\right)^T\right) \end{aligned} \tag{4.7}$$

where $\mathbf{Z}_i = \left[\mathbf{z}_i, \mathbf{z}_1^i \ldots \mathbf{z}_{m_i-1}^i, \bar{\mathbf{z}}\right]$, $\mathbf{D}_i = \left[\mathbf{e}_{m_i}, -\mathbf{I}_{m_i \times m_i}\right]^T$, and $\mathbf{e}_{m_i}$ is an all-one column vector of length m_i, $\mathbf{I}_{m_i \times m_i}$ is an identity matrix of size $m_i \times m_i$.

The next step is to learn the Lorentzian metric matrices $\mathbf{G}_i^l$. The metric $\mathbf{G}_i^l$ consists of two parts: the positive-definite part, to measure the within-class similarity, and local geometry, denoted as $\hat{\Lambda}_i$; and the negative-definite part, to measure the global geometric structure denoted as $\hat{\lambda}_i$. It can be computed by [73]:

$$\hat{\Lambda}_i\left(p,\, q\right) = \begin{cases} \frac{\left(\hat{\mathbf{D}}_{x_i}\right)^{-1} \mathbf{e}_{m_i-1}}{\mathbf{e}^{\mathbf{T}}{}_{m_i-1} \left(\hat{\mathbf{D}}_{x_i}\right)^{-1} \mathbf{e}_{m_i-1}} & if\ p = q, \\ 0 & otherwise. \end{cases} \tag{4.8}$$

where $\hat{\mathbf{D}}_{x_i} = diag\left(d\left(\mathbf{z}_i, \mathbf{z}_1^i\right)^2, \ldots d\left(\mathbf{z}_i, \mathbf{z}_{m_i-1}^i\right)^2\right)$ and $\hat{\lambda}_i = \Sigma_{j=1}^{m_i-1} \hat{\Lambda}_i\left(j,\, j\right)$.

After computing the metric matrix $\mathbf{G}_i^l$, the projection matrix $\mathbf{U}$ can be obtained by:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{u} = \lambda\mathbf{u}, \tag{4.9}$$

where $\mathbf{L}$ is derived from the total Lorentzian metric tensor given by Eqn. (4.7). $X = [x_1, x_2 \cdots, x_m, \bar{x}]$ and $\bar{x}$ is the centriod of S_x. Data samples can thus be projected using $\mathbf{z} = \mathbf{U}^T\mathbf{x}$. After the LDP projection, we expect that the separation of different actions can be enhanced.

4.5 Reducing the Modality Gap Based on Manifold Learning

To maximize the mutual information and reduce the discrepancy between heterogeneous actions, our second scheme is to apply manifold learning techniques, which could be viewed as learning the spectral transitions between the two spectra. It learns a path on a manifold to connect actions in two different spectra (see Fig. 4.2). A bridge between the two modalities is built to reduce the gap between them. In this way, by representing the features using subspaces along the path on manifold, the two modalities is believed to be closer to each other. In the following, firstly we introduce the fundamentals and mathematical concepts of the manifold. Secondly the scheme based on manifold learning is presented.

A manifold is a topological space that is locally similar to Euclidean space, which can be thought intuitively as a smooth, curved surface embedded in higher dimensional Euclidean spaces [78]. For differentiable manifolds, the tangent space is used to define the derivatives of the curves on the manifold. The tangent space at a point is the plane tangent to the manifold at that point. The minimum length curve connecting two points on the manifold is called the geodesic, and the distance between two points is given by the length of this curve called geodesic distance.

Grassmann manifold [79], which is a special class of Riemannian manifold, is defined as quotient spaces of orthogonal group. A quotient space of a manifold can be viewed as the result of "gluing together" certain points of the manifold [80]. A point in the Grassmann manifold is a particular subset of the orthogonal matrices, and the Grassmann manifold itself is the collection of all these subsets. Two points on a Grassmann manifold is considered to be equivalent if one can be mapped into the other by an orthogonal matrix [79].

To create the "transition" between the visible light and infrared actions, we use the subspaces along the geodesic path on the Grassmann manifold. Fig. 4.2 shows the idea of this approach intuitively.

Before the learning process conducted on the manifold, one needs to construct the orthogonal linear subspaces. The principal component analysis (PCA) can be applied to obtain the subspaces.

Denote the set of feature vectors extracted from the visible light actions as $\mathbf{D}_{vis} = \{\mathbf{s}_i\} \in \mathbb{R}^D,\ i = 1 \dots N_{vis}$, and the set from infrared as $\mathbf{D}_{ir} = \{\mathbf{t}_i\} \in \mathbb{R}^D,\ i = 1 \dots N_{ir}$. The PCA is applied to $\mathbf{D}_{vis}$ and $\mathbf{D}_{ir}$, respectively, and the output bases $\mathbf{P}_{vis} \in \mathbb{R}^{D \times d}$ and

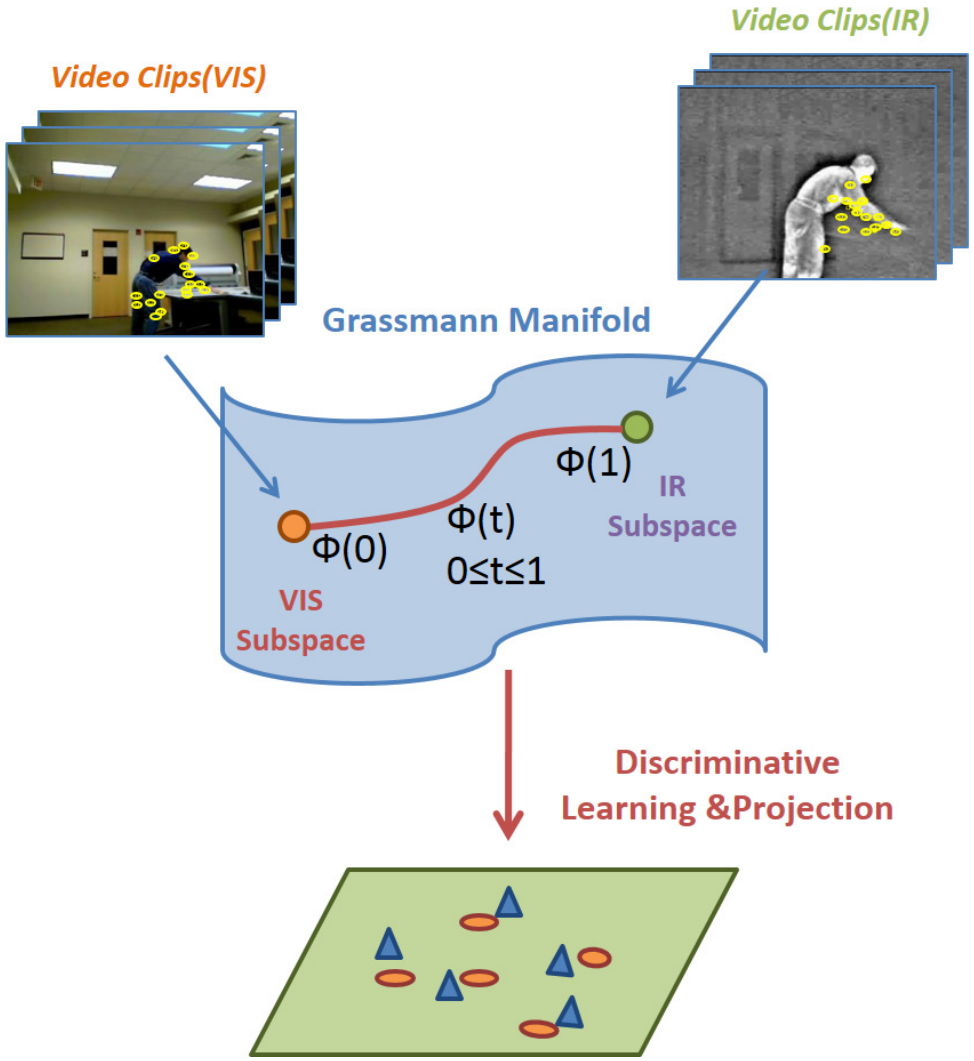

FIGURE 4.2: Learning the transitions between heterogeneous action patterns based on the Grassmann manifold.

$\mathbf{P}_{ir} \in \mathbb{R}^{D\times d}$ are considered as the subspaces, where d is the number of top eigenvectors of each covariance matrix.

The Grassmann manifold, $G(d, D)$, is formed by the collection of all the subspaces of d dimensions. The subspaces of the dataset in visible light and infrared spectra are mapped to two points on a Grassmann manifold. We want to find the transition from one point to the other, so that the visible actions can be smoothly connected to the infrared. The key idea is to compute the geodesic path between two points, and then utilize the intermediate subspaces to learn the new feature representations.

Specifically, let $\mathbf{R}_{vis} \in \mathbb{R}^{D\times(D-d)}$ be the orthogonal complement of $\mathbf{P}_{vis}$, i.e., $\mathbf{R}_{vis}^{T}\mathbf{P}_{vis} = 0$. Then a geodesic flow is constructed through the canonical metric on the Grassmann manifold, which is induced by the Frobenius norm on the tangent space [79]. The geodesic flow Φ is thus parameterized as $\Phi(i) \in G(d, D)$, $i \in [0, 1]$, with the constraints

$\Phi(0) = \mathbf{P}_{vis}$ and $\Phi(1) = \mathbf{P}_{ir}$. When $i \in (0, 1)$, it has [81]:

$$\Phi(i) = \mathbf{P}_{vis}\mathbf{U}_1\mathbf{\Gamma}_i - \mathbf{R}_{vis}\mathbf{U}_2\mathbf{\Sigma}_i \tag{4.10}$$

where $\mathbf{U}_1 \in \mathbb{R}^{d\times d}$ and $\mathbf{U}_2 \in \mathbb{R}^{(D-d)\times d}$ are orthogonal matrices. $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are $d \times d$ diagonal matrices, in which the diagonal elements are sine and cosine value of the principal angles [79] between $\mathbf{P}_{vis}$ and $\mathbf{P}_{ir}$. $\mathbf{U}_1$and $\mathbf{U}_2$ are computed by the following SVDs (singular value decompositions):

$$\mathbf{P}_{vis}^T\mathbf{P}_{ir} = \mathbf{U}_1\mathbf{\Gamma}\mathbf{V}^T,\ \mathbf{R}_{vis}^T\mathbf{P}_{ir} = -\mathbf{U}_2\mathbf{\Sigma}\mathbf{V}^T \tag{4.11}$$

More details on inferences on Grassmann manifold can be found in [82].

After the learning process, the constructed parameterized geodesic flow can characterize the smooth changes between two modalities of action patterns. The subspaces lying on the geodesic path are related to either the visible or the infrared actions depending on the parameters. In other words, the geodesic path can be regarded as the shortest to connect the visible and infrared action patterns on the Grassmann manifold. The mutual information between the two spectra is expected to be maximized by projecting the patterns using the subspaces on this geodesic path.

To represent the "transition" between the two modalities, i.e. visible and infrared in our HAR, a series of subspaces $S_n = [\Phi(i_1), \Phi(i_2) \ldots \Phi(i_n)], i \in [0, 1]$ is computed. Intuitively if i is close to 0, the subspace $\Phi(i)$ is more likely to one modality, while if i is close to 1, the subspace $\Phi(i)$ is more likely from the other modality. We can compute $\Phi(i)^T\mathbf{x}$ to project a feature vector $\mathbf{x}$ onto the subspace $\Phi(i)$. The action features from both the visible and infrared are projected onto the n subspaces, and a concatenation of all n projections is used to expand the original features. The new feature vector is of length $n \times d$. As a result, the action patterns from visible and infrared are changed to a new representation, which is suppose to be insensitive to modality changes.

Because the new feature vector consists of a series of subspaces on the Grassmann manifold, the dimensionality of the feature vector could be high. To deal with this, we apply the feature dimension reduction and discriminative learning methods, i.e., PLS_d, LDP, after the Grassmann learning procedure. The purpose is to improve the recognition accuracy for HAR.

4.6 Feature Representation for visible and infrared actions

Given the action data from visible and infrared modalities, we use spatiotemporal interest points (STIP) for feature extraction and representation. Specifically, the Cuboid detector with the Cuboid descriptor [83] is adopted to extract and represent the features for both visible and infrared action videos. Then vector quantization is conducted using the standard bag-of-words scheme by the K-means clustering method. In this way, each action video is quantized and represented by histogram bins, counting the occurrences of spatiotemporal features as different "key words". Empirically, we set the number of key words to 250. So, after the feature quantization, each action video is represented by an 250-dimension feature vector.

4.7 Experiments

4.7.1 Database

The experiments are conducted on a multi-spectral action database [1]. In this database, action videos are captured from two spectra by two synchronized cameras: a visible light camera and a thermal infrared camera. Each action video is about 21 seconds in average, with a 320×240 resolution and 30 frames per second (fps). In total there are 30 action categories performed by 30 different subjects in this database. The number of videos in this database is 1,800, all of which are used in our experiments. In specific, the 30 actions in the database include: fixinghair, handclapping, horizontalstretching, marching, squating, usingmicrowave, usingremotecontrol, wipingtable, boxing, kicking, movingbox, pickingup, running, openingdoor, organizingtable, walking, exercisejumping, wipingboard, knockingondoor, verticaljumping, drinking, handwaving, helpsignaling, reading, telephoning, typing, writing, sittingstanding, writingonboard, and checkingwatch. Fig. 4.1 shows some example images in both visible and infrared spectra in our database.

4.7.2 Experimental settings

First, the whole action database is divided into training and test sets. We use a cross-subject testing, where 10 subjects with all 30 actions in infrared (IR) are randomly selected from the database, which are used as the test set. The corresponding visible (VIS) videos of the same 10 subjects are discarded. Then the remaining 20 subjects with all 30 actions in both visible and infrared are used to construct the training set.

In our experiments, we explore the case where a fixed number of infrared samples with a different number of visible training samples for learning, to show the influence of the visible data size to the classification accuracy for actions in infrared.

Given the training and test sets, the next step of the experiments is feature extraction and representation. The spatiotemporal interest point based feature is used in our experiment. Specifically the Cuboid detector with the Cuboid descriptor [83] are adopted to extract and represent the features for both visible and infrared action videos. Then vector quantization is conducted using the standard bag-of-words scheme by the K-means clustering method. In this way, each action video is quantized and represented into a histogram of bins, counting the occurrences of spatiotemporal features lie into different "key words". Empirically, we set the number of "key words" to 250. Because of the randomness, K-means clustering is run 10 times to obtain the cluster centers. So, after the feature quantization, action videos are represented by a set of 250-dimension feature vectors.

For the first scheme based on correlations between visible and infrared, projection vectors are learned by PLS_c, using the features extracting from visible and infrared, respectively. The parameter (dimension) of PLS_c has been tried through 20 to 240 and finally set to 120, which has shown good recognition performances in our experiments. For the scheme that learns spectral transitions on Grassmann manifold, the PCA ratio is set to 0.98 to obtain the bases. Subspaces with parameters $[0.1 : 0.1 : 1]$ are used to obtain 10 subspaces with an interval of 0.1. For the discriminative learning method LDP , the dimensionality is set to 30. In action classification stage, the Supporting Vector Machine (SVM) with RBF kernel is used as the supervised learner. The RBF kernel is defined as:

$$K\left(x,\, x'\right) = exp\left(\gamma \|x - x'\|_2^2\right), \tag{4.12}$$

where x and x' are the training samples, γ is a free parameter. In our experiments, the fine-grid tuning is conducted to obtain the optimized parameters in SVM, γ and c. The experimental results in different groups are presented in the following.

4.7.3 Information Theoretic Measure of Modality Disparity

We conduct experiments to validate the proposed approaches from the information theoretic perspective. The proposed approaches learn the projections or the transitions between the visible light and infrared spectra. It is supposed that the modality gap between visible and infrared can be reduced, therefore the mutual information between visible and infrared features is increased. In the following, we conduct the experiments and analyze the results.

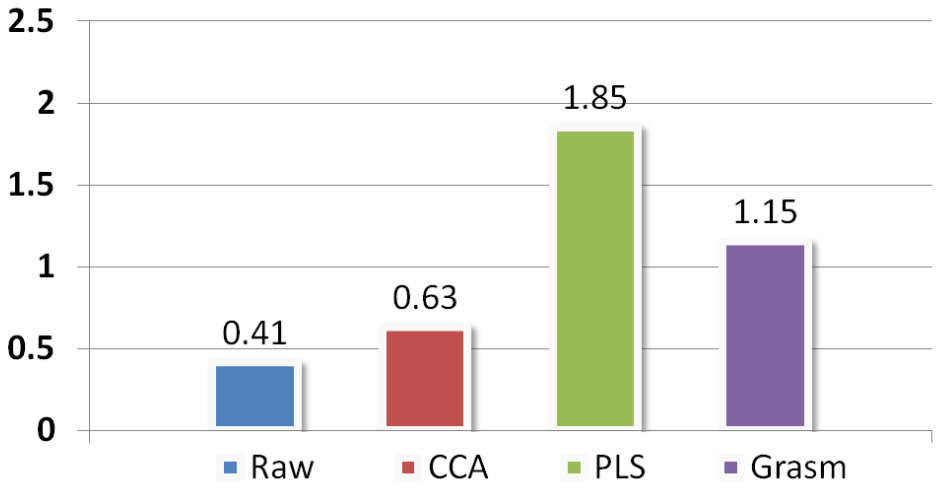

FIGURE 4.3: Mutual information computed between visible and infrared action pairs using different approaches. Totally 300 action pairs from visible and infrared are randomly selected from the dataset and average results are shown in the figure. "Raw" denotes the raw features extracted from visible and infrared, respectively. "Grasm" denotes the Grassmann manifold learning method.

From both visible and infrared data, 10 subjects with all the 30 actions are randomly selected in this experiment. Denote an "action pair" as one sample from visible and the corresponding one from infrared, these two action videos are of the same action category and performed by the same subject. Thus in the experiment there are totally 300 action pairs. Using the experimental settings mentioned above, we compute the Mutual Information(MI) between each action pairs and show the average results. The MI is calculated by the *log* function with base 2, so the unit of measurement is "bits". Figure 4.3 shows the MI that are computed between visible and infrared action pairs using different approaches. "Raw feature" means the MI is computed between the features extracted from visible and infrared without any learning, which shows a very low inherent dependence between the two spectra. On one hand, by applying the correlation learning approach PLS_c, the MI between the projected features are much higher than the baseline "Raw feature", resulting in the MI of 1.85 bits. We have also compare the MI between PLS_c and CCA, which shows that CCA is not good to increase the MI in our HAR problem. On the other hand, when the Grassmann approach is applied, the MI (1.15 bits) is also increased comparing to the baseline. It is higher than the CCA, but lower than the PLS_c. We observe that higher MI between the features show higher similarity between the two spectra, which also implies a potentially higher recognition accuracy, because the transformed features from the two spectra are getting "closer" to each other. In the following, we validate our approaches experimentally from action recognition aspect.

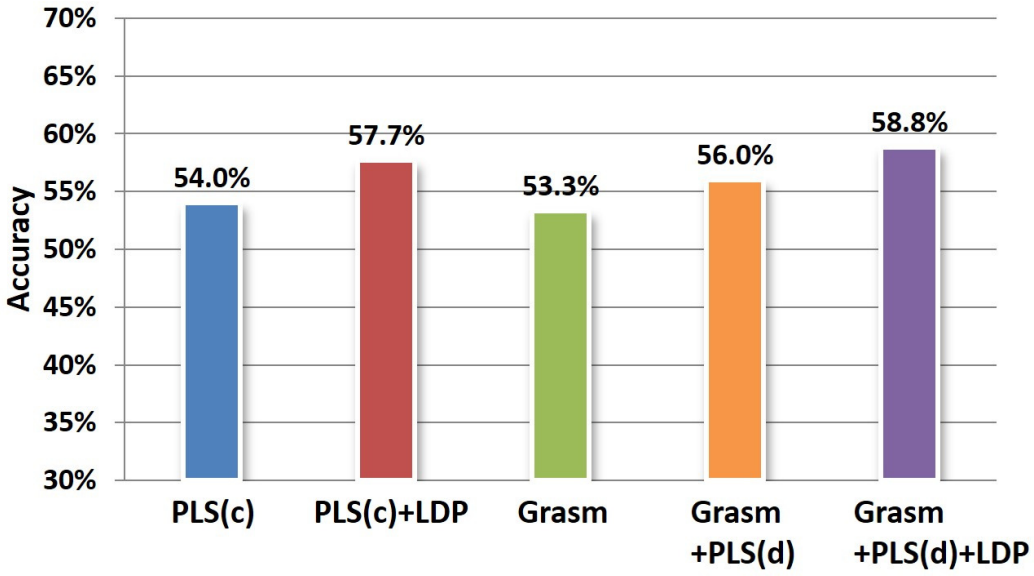

FIGURE 4.4: Recognition rates using different approaches when 5 subjects from visible data and 5 subjects from infrared data are used for training, 10 subjects from infrared data are used for testing. "PLS(c)" denotes that PLS is used for correlation learning, "PLS(d)" denotes PLS is used for dimension reduction. "Grasm" denotes the method that learns spectral transitions based on Grassmann manifold.

4.7.4 Heterogeneous Action Recognition Results

We present the experimental results for action classification. Experiments are conducted using different numbers of training samples, while the test set (10 subjects in IR) is kept the same. Only a small number of infrared samples are selected and used for training. In our experiments, the training set consists of two parts: (1) infrared data (one modality), where 5 subjects are randomly selected out of 20 on infrared data; (2) visible data (another modality), where the visible actions of 5, 10, 15, and 20 subjects, respectively, chosen for training.

First of all, the experiments are conducted to test the performance of using 5 subjects from infrared (no visible samples) for training, while the testing samples are fixed (10 subjects of infrared samples). This experiment on a single spectrum (infrared) can be regarded as the baseline of the infrared action recognition. Using the experimental settings presented above, the recognition rate achieves 51.3%. This demonstrates that using the spatiotemporal interest point features, action recognition in infrared spectrum can be performed, however the accuracy is not very high. In the next, experiments are conducted by utilizing the visible data, to help improve the infrared action recognition.

Fig. 4.4 shows the recognition accuracies using 5 subjects visible and 5 subjects infrared data for training, and tested on the fixed 10 subjects of infrared data. Totally 5 approaches are shown in the figure. The first 2 columns are correlation based approaches

(PLS_c, and PLS_c+LDP), the last 3 columns are manifold transition approaches (Grassmann, Grassmann+PLS_d, and Grassmann+PLS_d+LDP), respectively. From the results one can see that: (1) both correlation based and manifold learning approaches can improve the recognition accuracy, which is better than the baseline recognition rate (51.3%). The best result is 58.8%, obtained by Grassmann+PLS_d+LDP, which is higher than the best result based on correlation, PLS_c+LDP (the accuracy is 57.7%). These results show that the proposed approaches can learn from the visible spectrum, to help recognize the actions in infrared and the recognition accuracy is improved. (2) The recognition accuracy is 53.3% using Grassmann manifold method, which is higher than the baseline but lower than other methods shown in the figure. One possible reason is that the features after Grassmann manifold learning are not very discriminative. Therefore, we propose to further create discriminative features by utilizing PLS_d, or LDP. Experimental results show the recognition rates are improved to 56.0% by using PLS_d, and achieves the highest accuracy 58.8% using LDP. (3) The modality transition learned on Grassmann manifold performs slightly better than the correlation based approach. It can be seen from the figure that, although the Grassmann manifold only obtains a lower accuracy 53.3% than the PLS_c, a much higher accuracy (58.8%) is achieved by utilizing dimension reduction and discriminative learning (Grassmann+PLS_d+LDP). This demonstrates that the usefulness of the proposed approach for modality transition using Grassmann manifold for heterogeneous action recognition, while the discriminative mapping is needed. Similar observations can also be seen in the following experiments, where different numbers of visible samples are used.

4.7.5 HAR with Different Sizes of Training Data

To further investigate the performance of the proposed approaches, we conduct experiments using different numbers of training samples from the visible spectrum. Specifically, 5, 10, 15, and 20 subjects from visible data are used in the training set, respectively, along with 5 subjects from infrared data for training. The other parameter settings are kept the same. Recognition results are shown in Table 4.1. One can see that, in correlation based approaches, the PLS_c+LDP outperforms the PLS_c only method. The best accuracy of 62.7% is obtained by PLS_c+LDP when 20 subjects from visible samples are used, resulting in 4% higher than the PLS_c approach under the same settings. This demonstrates that by conducting the discriminative mapping on the correlated features, the recognition accuracy can be improved. On the other hand, in manifold learning approaches, experimental results (see last 3 rows in Table 4.1) also show that the discriminative mapping (e.g., LDP) applied with dimension reduction by PLS_d can further help improve the recognition accuracy. In particular, the accuracies increase 3% - 5%

TABLE 4.1: Accuracies w.r.t. different numbers of training samples. "*" denotes the schemes proposed in this paper. Note that in the training set, 5 subjects from infrared are kept the same. In testing, the 10 subjects from infrared are also kept the same.

	5 Subjects	10 Subjects	15 Subjects	20 Subjects
PLS	54.0%	55.0%	56.0%	58.7%
PLS+LDP *	57.7%	59.0%	60.7%	62.7%
Grassmann	53.3%	54.3%	55.7%	57.3%
Grassmann+PLS	56.0%	57.3%	60.0%	62.0%
Grassmann+PLS+LDP *	58.8%	60.5%	61.2%	63.7%

when PLS_d is applied for all the experimental groups (5th and 6th row in the table). The accuracies have a further 1% - 3% improvement, when LDP is applied after PLS_d. Finally, the highest accuracy is 63.7%, achieved by Grassmann+PLS_d+LDP, when 20 subjects from visible spectrum are used for training.

From the experimental results, we also analyze how the number of training samples from visible spectrum influences the infrared recognition results. Fig. 4.5 shows the accuracies by using different numbers of visible samples in training. The "IR Only" means that the training set contains only the 5 subjects' infrared action videos, which is considered as the baseline result (51.3%). This figure illustrates that the overall recognition accuracy increases when the number of visible training samples is increased, for all the proposed approaches. Note that the infrared training data are kept the same with more visible samples utilized. The highest accuracy of 63.7%, is achieved by Grassmann+PLS_d+LDP, when 20 subjects from visible samples are used, which is much higher than the baseline result 51.3%, where no visible data is used. This further validates the usefulness of heterogeneous action recognition, where the knowledge learned from visible action videos can be utilized to help recognize actions in infrared, and the recognition accuracy can be increased significantly.

Finally, we compare our approaches to others. Note that similar experimental settings are used in our previous work [1]. Particularly, 10 subjects from visible and 10 subjects infrared are used for training, 20 subjects from infrared are used for testing. The experiments run 10 times, the mean accuracies and standard deviations are computed. The experimental results are shown in Table 4.2. It can be seen that the recognition accuracies are significantly improved compared to the previous ones in [1]. For the discriminative learning, in comparison to the MFA (64.0%), our scheme that uses the LDP achieves a slightly higher accuracy (64.8%). The best result is obtained by Grassmann+PLS_d+LDP (67.3%($\pm$1.7%)), which is about 12% higher than the accuracy achieved by A-SVM (adaptive-SVM) in [1].

The confusion matrix of using the Grassmann+PLS_d+LDP approach is shown in Fig. 4.6. From the figure one can see that the actions like "writing on board" & "wiping

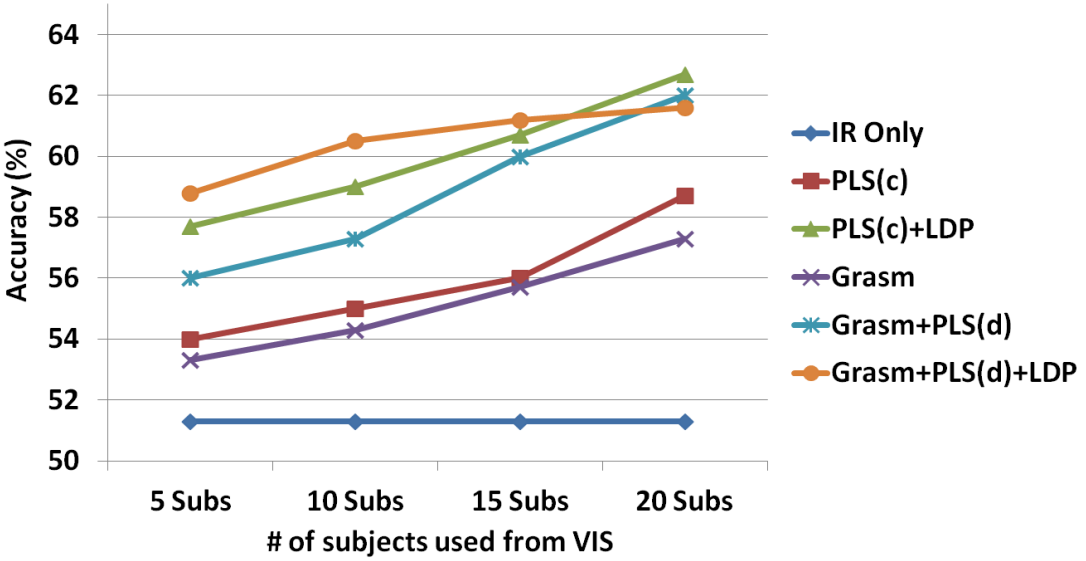

FIGURE 4.5: Line graph of different approaches. The recognition accuracy increases when the number of VIS training samples are increased. "PLS(c)" denotes that the PLS is used for correlation learning, "PLS(d)" denotes PLS is used for dimension reduction, "Grasm" denotes the manifold learning method. Note that in the training set, 5 subjects from infrared are kept the same. In the test set, the 10 subjects from infrared are also kept the same.

TABLE 4.2: Comparison with the approaches in [1], using the same experimental settings. "*" denotes the schemes proposed in this paper. Both the mean accuracy and the standard deviation are shown in the table.

Method	**Accuracy**
IR Only [1]	42.8%(±1.4%)
CCA [1]	31.2%(±3.1%)
A-SVM [1]	54.6%(±1.6%)
PLS_c	62.1%(±2.0%)
PLS_c+MFA	64.0%(±1.3%)
PLS_c+LDP *	64.8%(±1.4%)
Grassmann	61.7%(±1.3%)
Grassmann+PLS_d	66.4%(±1.0%)
Grassmann+PLS_d+LDP *	67.3%(±1.7%)

on board"; "writing" & "typing" & "reading"; and "knocking on door" & "opening door", are more likely misclassified to each other. That is because the appearance of such actions are similar when performed by the subjects, or the differences between these actions are mainly human-object interactions but not the motion, e.g., reading and writing, thus more difficult to characterize.

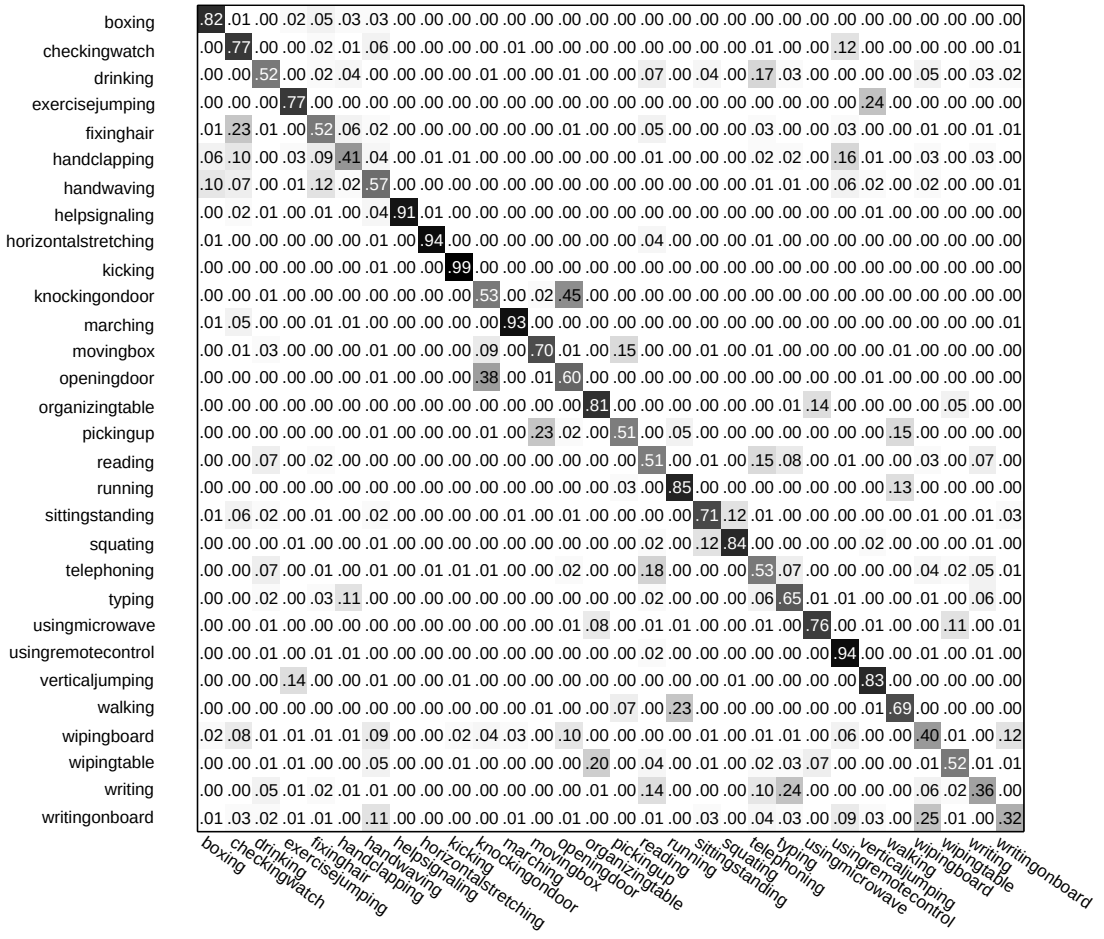

FIGURE 4.6: Confusion matrix of the approach, Grassmann + PLS_d + LDP, with the overall accuracy of 67.3%.

4.8 Conclusions

We have presented a relative new problem called heterogeneous action recognition (HAR). To address the problem, we have proposed approaches under the framework of maximizing the mutual information between actions in different modalities. Two schemes have been explored to address the HAR problem guided by the information theoretic measures. The first scheme is based on learning the correlations, to increase the mutual information between visible and infrared action patterns. The second scheme aims to learn the transitions between visible light and infrared actions on Grassmann manifold. The geodesic path on the manifold is considered as the "shortest" path connecting different modalities of actions. We have also shown that the discriminative mapping is needed for both schemes, in order to improve the action recognition accuracies. Experiments have been conducted on a relatively large database with 30 actions of different modalities to show the usefulness of our approaches for addressing the challenging HAR problem.

Chapter 5

Action Recognition in Thermal Infrared using Histogram of Spatiotemporal Sparse Codes

5.1 Abstract

Previous works on action recognition mainly focus on the use of visible light videos. Thermal infrared (IR) captures the temperature providing some advantageous over visible light for action recognition. Human actions in IR are insensitive to illumination changes, and can be captured at any time (day or night) and anywhere (indoor or outdoor). In this paper, we study recognizing human actions in IR. A new feature called the histogram of spatiotemporal sparse codes (HSSC) is proposed to characterize the IR action videos. The proposed method learns sparse representations directly from the IR data, taking into account both spatial and temporal information. Further, a saliency map is developed to incorporate spatial distribution of local features. From the experiments on a relatively large IR action database, a promising recognition accuracy is achieved. The proposed method is general and applicable to the traditional visible light actions. The recognition accuracy on the popular KTH dataset outperforms the state-of-the-art methods.

5.2 Introduction

Human action recognition as one of the important topics in computer vision, has gained tremendous research interests for decades. Automated recognition of ongoing human

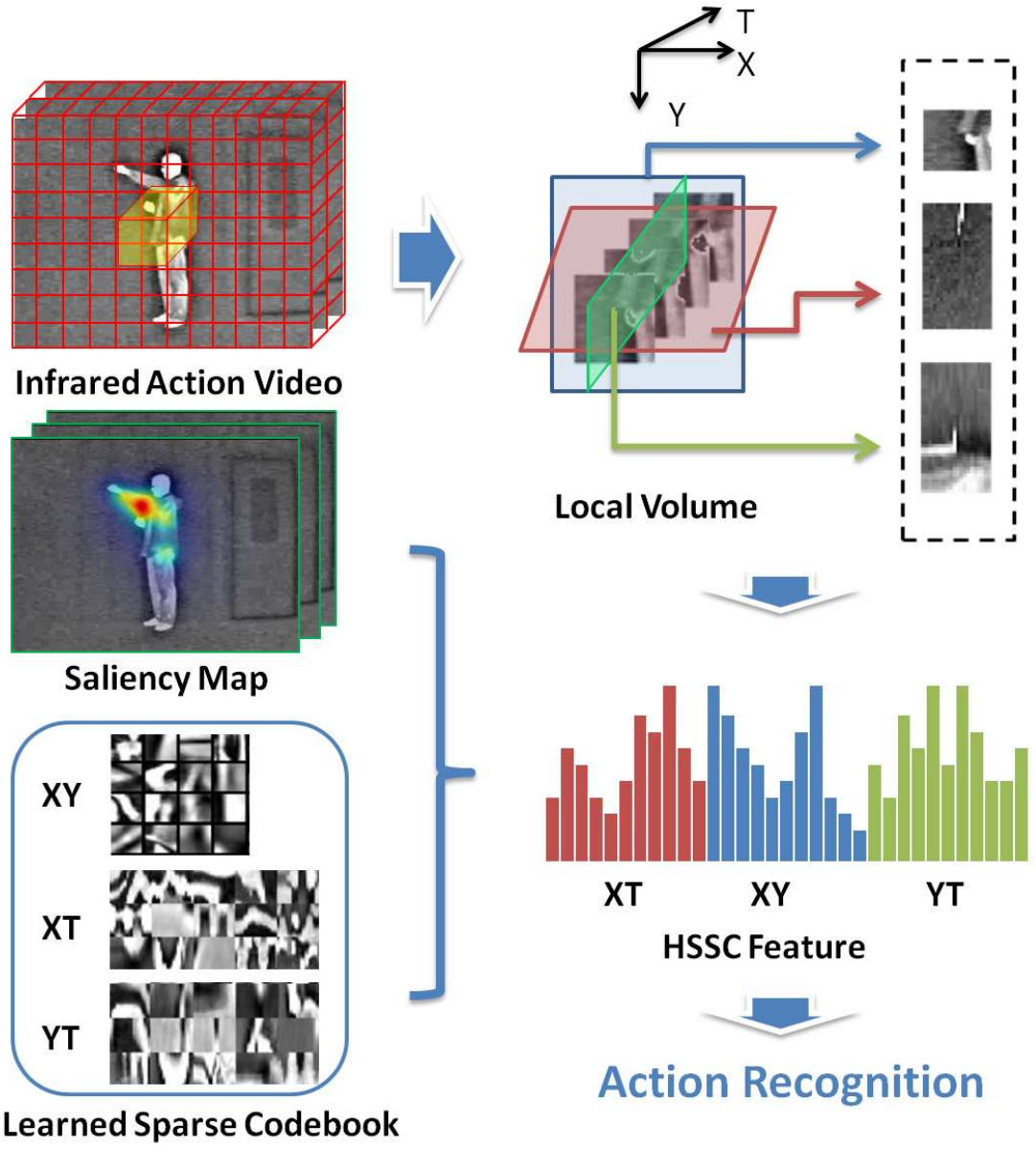

FIGURE 5.1: Schematic diagram of thermal infrared action recognition using histogram of spatiotemporal sparse codes.

actions has a wide range of applications, e.g., video analysis, intelligent surveillance, human-computer interaction and security [17]. A number of approaches have been proposed to extract representative features from the spatiotemporal action data. However, most of the studies are based on color or intensity imagery in the visible light spectrum. The action data are captured under good illumination conditions. Consequently, difficulties will be encountered when the illumination conditions are poor or at dark, where the visible light cameras cannot work well. Therefore the performance of action recognition would degrade dramatically. Recently action recognition on depth maps [23][84][2] emerges, which extends action recognition into a new domain beyond the visible light.

However, depth sensor still imposes some limitations which restrict its applications, e.g., the depth acquisition has a relatively small distance, with heavy noise, and is usually just applicable for indoor environment. Thermal infrared (IR) camera, which detects radiation in the range from 7-14μm, can capture human motion without the influence of illumination conditions. It can be used anytime (day or night) and anywhere (indoor or outdoor) for human motion acquisition. Thus IR has advantages over visible light or depth-based sensing for human action/activity acquisition and recognition.

Actually, thermal imaging has shown success in some other computer vision problems, e.g., face recognition [48][59][60], face expression analysis [61], human detection [62], and human gait analysis [49][63], however, it has not been truly applied to human action/activity analysis. Very recently, an approach was developed to utilize visible data to help action recognition in IR [1], but it does not focus on IR itself. In contrast, we study action recognition in pure IR without bothering the visible data.

To deal with infrared action recognition, we propose a novel feature called Histogram of Spatiotemporal Sparse Codes (HSSC). Firstly, the IR videos are densely sampled into local spatiotemporal volumes. Given the set of local volumes, sparse dictionaries are learned spatially and temporally representing action appearances and motions. Rather than using all pixels in each volume, three orthogonal planes are used to reduce the computational complexity. Then we construct the histogram feature for each IR video based on sparse codes similar to the bag-of-words scheme. In order to generate more discriminative histogram features, a saliency map of the IR video is computed to incorporate the spatial distribution of salient regions, which is a novel scheme to incorporate spatial information for histogram features, to the best of our knowledge. This step helps overcome the drawback of histogram features that usually ignore the spatial information.

Our action recognition experiments are conducted on a relative large thermal infrared action database with 30 actions [1]. Experimental results show significant improvement comparing to various methods. Besides, experiments are also conducted on the well-known visible light action dataset KTH [19]. Following the experimental protocols described in [19], our method outperforms the state-of-the-art methods. This further demonstrates that our method is general for human action recognition.

In the following, we introduce the related work in Section 5.3. The Histogram of Spatiotemporal Sparse Codes (HSSC) feature is presented in Section 5.4. Experiments are conducted and the experimental results are shown in Section 5.5. Finally, conclusions are drawn in Section 5.6.

5.3 Related Work

The space-time features which were originally developed for action recognition in visible light spectrum may not work well in videos of other domains. One category of the popular features is based on spatiotemporal interest points (STIP) [14]. Having detected a set of interest points, the bag-of-words strategy [85] [86] is often used with various local descriptors. One drawback of STIP features is that the detection is based on the local maximum of gradient on visible data, while IR videos are often with heavy noise which would have an impact on the interest point detection results. On the other hand, simply assigning the feature in the bag-of-words without considering the spatial distribution can lead to some loss of discriminative capability.

Sparse representation can capture the significant structure in the given signals. By learning a set of overcomplete basis vectors as the dictionary, the essential information of a signal can be efficiently represented using the linear combination of a small number of non-zero entries in the dictionary. Sparse representation provides a generalized way to learn the basis and potential patterns. In the literature, sparse coding has been shown to produce promising performance in various areas, e.g., image compression [87], image classification [88], object detection [89] and face recognition [90].

Recently, sparse coding has been explored for human action recognition in *visible* imagery. In [91], sparse representation is utilized on top of the spatiotemporal features. In comparison to our method, their learning procedure is more expensive because representing the local volumes using the HOG3D descriptor is required at the first step. As a consequence, their method brings not only high computational cost but also makes the sparse representation highly rely on the performance of HOG3D feature. Besides, comparing to the sparse coding, HOG3D feature is designed based on the gradient for the visible light data, which may not be appropriate on the thermal infrared data.

Zhang et al. [92] proposed to use sparse coding as a vector quantization step upon the feature in optical flow field. Their method may have the drift problem of optical flow when extracting trajectory features. More importantly, because of the heavy noise, methods based on optical flow may not work well in IR videos.

Our method is also different from the one proposed by Guha et al. [93]. In their work, two categories of features are extracted from the visible light action data: the spatiotemporal feature (Cuboids [5] with HOG descriptor) and local motion pattern (LMP) feature. Then a sparse dictionary learning is applied on the feature vectors. In contrast, our proposed method learns the sparse dictionary directly from the data in an unsupervised manner, thus our method is more general and suitable for IR data, and largely reduces the complexity in feature learning.

More recently, Ren et al. [89] proposed to use a histogram of sparse code feature for object detection in visible light images. They focus on comparing the sparse representation with the gradient and the designed feature similar to HOG. Experiments for object detection are conducted using the standard sliding window scheme. Different from their work, our method focus on the IR action recognition in videos. We design the spatiotemporal feature characterizing both action appearances and motions from the IR video data. Further, our method incorporates the spatial distribution of salient features based on a saliency map. Consequently, the developed histogram feature is more discriminative.

5.4 Histogram of Spatiotemporal Sparse Codes

In this section, we introduce the sparse coding learning and dictionary representation in Section 5.4.1. Then we propose the histogram of spatiotemporal sparse codes feature in Section 5.4.2. The method for constructing the histogram feature using the saliency map is presented in Section 5.4.3. Finally, the postprocessing for the histogram feature is presented in Section 5.4.4.

5.4.1 Sparse Code Learning and Representation

Given a set of densely sampled local spatiotemporal volumes, the first step of our method is to learn the sparse dictionaries and the corresponding sparse representation. K-SVD [87] is a standard dictionary learning algorithm in an unsupervised manner similar to the K-means [94]. In our method, the sparse dictionaries are directly learned from the IR volumes, which contain a set of overcomplete basis representing the initial pattern of the IR data.

Denote the learned overcomplete dictionary matrix as $\mathbf{D} \in \mathbb{R}^{n \times K}$, where the columns are the atom codewords. Given the local image patch or spatiotemporal volume $\underline{x} \in \mathbb{R}^n$, it can be represented as a linear combination of the codewords in the dictionary and the linear coefficients are sparse. Formally, given a set of spatiotemporal volumes $\mathbf{X} = [\underline{x}_1, \underline{x}_2 \cdots, \underline{x}_n]$, K-SVD learns a set of codewords (dictionary) $\mathbf{D} = [\underline{d}_1, \underline{d}_2 \cdots \underline{d}_m]$ and the sparse code matrix $\boldsymbol{\mu} = [\underline{\mu}_1, \underline{\mu}_2, \cdots, \underline{\mu}_m]$. Because the problem has infinite number of solutions if $n < K$ and $\mathbf{D}$ is a full-rank matrix, the constraint that each $\underline{x}$ contains K or fewer nonzero elements is used. The sparse approximation problem can be written as:

$$\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} ||\mathbf{X} - \mathbf{D}\boldsymbol{\mu}||_F^2, \ \textit{subject to} \ ||\underline{x}||_0 \leq K, \tag{5.1}$$

where $||\cdot||_F$ is Frobenius norm, and $||\cdot||_0$ is the ℓ_0 pseudo-norm that counts the number of nonzero entries in a vector, K is called the sparsity level. According to [87], the K-SVD algorithm has two major steps: sparse coding and dictionary update. One of the efficient solution to this NP-hard problem is using the greedy algorithm called Orthogonal Matching Pursuit (OMP). In [95], the dictionary update step of the K-SVD is modified, speeding up the sparse coding process.

Briefly, the OMP algorithm takes the leaned dictionary $\mathbf{D}$, the signal $\underline{x}$ and the sparsity K as the input, outputs the sparse representation $\underline{\mu}$ such that $\underline{x} \approx \mathbf{D}\underline{\mu}$. In every step, the algorithm greedily selects the atom which has the highest correlation: $\hat{a} = \arg\max_a |\underline{d}_a^{'}\underline{r}|$, where $\underline{r}$ is the current residual initialized to $\underline{x}$. Then the signal $\underline{x}$ is projected to the selected basis vectors orthogonally:

$$\underline{\mu}_k = \left(\mathbf{D}_k^{'}\mathbf{D}_k\right)^{-1}\mathbf{D}_k^{'}\underline{x}, \tag{5.2}$$

after the projection, the residual is updated by $\underline{r} = \underline{x} - \mathbf{D}_k\underline{\mu}_k$. These steps are executed iteratively until convergence.

After learning the dictionaries from IR data, the OMP [96] algorithm with its batch version [95] can be used to compute the pixel-level sparse code for each spatiotemporal volume. This algorithm can handle a large set of data and effectively speed up the sparse code learning process. Details of the OMP-Batch algorithm are referred to [95].

5.4.2 Histogram of Spatiotemporal Sparse Codes

A dense sampling scheme as described in [14] is used in our method for each video sequence $V(x, y, t)$, so that sufficient local information can be captured. A set of local blocks of size $n_x \times n_y \times n_t$ are densely sampled across the spatiotemporal domain throughout each video sequence. Each block centered at $p(x, y, t)$ can be viewed as a 3D spatiotemporal local volume. This spatial and temporal sampling can be done either with or without overlaps. In our experiment, both overlap and no overlap cases are evaluated with different sizes of volumes (see Section 5.5.3).

In each local volume centered at $p(x, y, t)$, three orthogonal planes that insect at the center pixel are used to incorporate the information in both spatial and temporal domain. Figure 5.1 shows an intuitive illustration of our scheme. Given a set of local volumes from the training data, we use K-SVD to find the sparse dictionary on X-Y, X-T, and Y-T planes. Three sets of dictionaries are obtained to characterize both the spatial and temporal local patterns. After the dictionaries are learned, corresponding sparse codes can be efficiently computed using the OMP algorithm.

In each local volume of an IR video, the sparse codes of three planes are computed to build the histogram. The size of the sparse codes for one plane is equal to the size of its corresponding dictionary. For each nonzero entry of the sparse codes, two ways can be used to fill the histogram bins: (1) use the absolute value [89]; (2) assign the occurrence of each nonzero entry. We found that the later scheme works better in our experiments. In this way three histograms are built for each video by counting the nonzero entries for all the local volumes. One histogram (X-Y) represents the spatial and the other two (X-T, Y-T) represent the temporal structures of data. L2 norm is then applied individually, and the three histograms are concatenated into one histogram of sparse code, which we call the Histogram of Spatiotemporal Sparse Codes (HSSC).

5.4.3 Histogram Binning using Saliency Map

Different from the standard techniques, e.g., spatial pyramid matching [97] and subsequences scheme [85], we propose to use the visual saliency map [98] [99] [100], taking into account the spatial information for the local features. In our method, we use the Graph-Based Visual Saliency (GBVS) algorithm proposed in [98]. GBVS is an effective algorithm based on a graph structure incorporating intensity, orientation and motion from the input data. The assumption of our method is that local volumes with higher "saliencies" have higher contributions to the action, thus higher weights can be assigned in building the histogram. Specifically, denote $vol\,(x, y, t)$ as the local volume centered at (x, y, t), where x, y are the spatial locations and t is the frame number. The saliency weight of this volume $w\,(x, y, t)$ is computed as:

$$w\,(x, y, t) = median\,(smap\,(x, y, t))\,, \tag{5.3}$$

where $smap\,(x, y, t)$ represents the saliency measures within the volume centered at (x, y, t) in the saliency map, with the same size as $vol\,(x, y, t)$. The saliency map was computed using the GBVS algorithm in [98]. Given the saliency weights, the histogram is computed as follows:

$$H_k = \Sigma_{x,y,t} g\,(\mu_k)\, w\,(x, y, t)\,,\ g\,(\mu_i) = \begin{cases} 0, & \mu_i = 0 \\ 1, & \mu_i \neq 0 \end{cases} \tag{5.4}$$

where μ_k is the kth entry of the sparse codes for $vol\,(x, y, t)$.

FIGURE 5.2: Example images in the thermal infrared human action database. The actions are: (top row) walking, help signaling, wiping table, hand waving and writing on board; (bottom row) picking up, typing, opening door, kicking and writing.

5.4.4 Histogram Postprocessing

To make the extracted feature more discriminative, the power normalization and dimension reduction techniques are applied on the feature vector before the classification stage. Power normalization (e.g., [101][102][89]) can improve the accuracy empirically. For each histogram feature $\underline{h}$, apply the power of each element as:

$$\underline{h} = \underline{h}^{\theta}, \ \theta \in (0, 1]. \tag{5.5}$$

Since the input is the video sequences, characterizing local patterns often results in a large dimensionality of the feature vector, e.g., in our experiments, when dictionary size is set to 2000, the HSSC feature is of 6000 dimensions. The standard Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are applied to reduce the dimensionality. After the above steps, each IR action video sequence is represented by a discriminative histogram of sparse codes, then can be fed into a multi-class classifier for action recognition.

5.5 Experiments

In this section, the infrared action database is introduced in Section 5.5.1. The experimental settings are presented in Section 5.5.2. In Section 5.5.3, the experimental results on the IR dataset are presented. Finally, experimental results on the KTH dataset are given to show the wide applicability of our method in Section 5.5.4.

5.5.1 Thermal Infrared Action Database

We conduct experiments for IR action recognition on the dataset proposed in [1]. Rather than cross-spectral action recognition [1], we focus on the thermal infrared spectrum only. The actions in infrared are captured by a Thermal-Eye 300D camera, with a 25mm lens and a pixel array of 320 by 240 for each image frame. The response wavelength of the IR camera is 7-14μm. The average video length is about 21 seconds, with 30 frames per second (fps). The total number of infrared videos is 900, including 30 action categories performed by 30 persons in each action. The 30 actions in the database are: fixinghair, handclapping, horizontalstretching, marching, squating, usingmicrowave, usingremote-control, wipingtable, boxing, kicking, movingbox, pickingup, running, openingdoor, organizingtable, walking, exercisejumping, wipingboard, knockingondoor, verticaljumping, drinking, handwaving, helpsignaling, reading, telephoning, typing, writing, sittingstanding, writingonboard, and checkingwatch. In our experiments, we subsampled each video into about 4 seconds and all the 900 infrared action videos are used with about the same length. Figure 5.2 shows some example image frames from the thermal infrared videos.

5.5.2 Experimental Settings

The sparse code dictionaries are leaned using K-SVD [87] algorithm on the corresponding three orthogonal planes (X-Y, X-T and Y-T), individually. We randomly select 200,000 local volumes sampled from the training data. Histograms of sparse codes from each plane are normalized and concatenated to represent each IR action. An exponent around 0.3 power transform [101] is applied to the histogram feature, which shows optimal performance in the experiments. Various dictionary sizes are tested in our experiments, ranging from 500 to 2000. Principal Component Analysis (PCA) [39] and Linear discriminant analysis (LDA) [103] are applied to reduce the feature dimensionality. In the classification stage, support vector machine (SVM) [41] with linear, RBF, Chi-square [85], and intersection [104]) kernels are tested. From the results, intersection kernel shows a higher accuracy comparing with others and all the results shown in the following are based on this kernel. A standard three-fold cross validation is performed where 20 subjects with all the 30 actions are used for training, the remaining 10 subjects for testing. This process is repeated three times and the overall accuracy is obtained by taking the average.

5.5.3 Experimental Results

Figure 5.3 shows the example dictionaries learned from the IR action data. The dictionaries for X-Y plane learned the rich patterns in the spatial domain, and X-T and Y-T plane capture temporal structures which can better represents the dynamics and motions of human action. This process of learning spatiotemporal dictionaries is simple and effective, and it can be implemented in parallel.

Sparsity level and the dictionary size. Figure 5.4 shows the recognition accuracies when the dictionary size and sparsity levels are changed. In this experiment, the local spatiotemporal volume is of size $18 \times 18 \times 20$, and sparsity level ranges from 1 to 4, dictionary sizes are empirically selected from 500 to 2000 (with an interval of 500). The saliency map is not used in this experiment. From the result we observe that, a small dictionary size, e.g., 500, may not catch enough information so that a spatiotemporal volume may not be represented well. A large dictionary, e.g., 2000, does not perform well either since a large histogram with high dimensionality might overfit the data. Another observation is that sparsity level 1 performs better when different dictionary sizes are used, which is consistent with [89] in still image analysis. The best result is achieved when dictionary size is 1500. In most of our following experiments, the sparsity level is set to 1 and the dictionary size is 1500.

Spatiotemporal volume size and dictionary size. We further investigate the selection of local volume size and the dictionary size. Table 5.1 shows the recognition accuracies as the volume size and dictionary size are changed. We first use the volume size $18 \times 18 \times 20$, which is a commonly used size for spatiotemporal features e.g., [14][91][34]. Smaller volume size might not catch rich enough local information while a larger volume size can learn richer patterns to help the feature representation. In the experiment, a larger volume size $36 \times 36 \times 20$ is also tested. Both non-overlapping and 50% overlapping volume sampling schemes are evaluated. From the result one can see that, the sampling with overlapping consistently performs better than non-overlapping, and too large volume size might not perform well. When the patch size is $18 \times 18 \times 20$ (with overlapping) and dictionary size is 1500, an optimal accuracy 81.3% is achieved. Under the same settings, the best result 86.7% is obtained, by utilizing the saliency map (described in Section 5.4), which suggests the advantage of our method incorporating the spatial structure into histogram features.

Recognition results in each plane. Figure 5.5 presents the recognition results using our HSSC representation. Three curves (X-Y, X-T and Y-T) give the performance using only one histogram from a single plane. The fourth curve shows the results obtained using the concatenated histograms. From the figure we can see that features

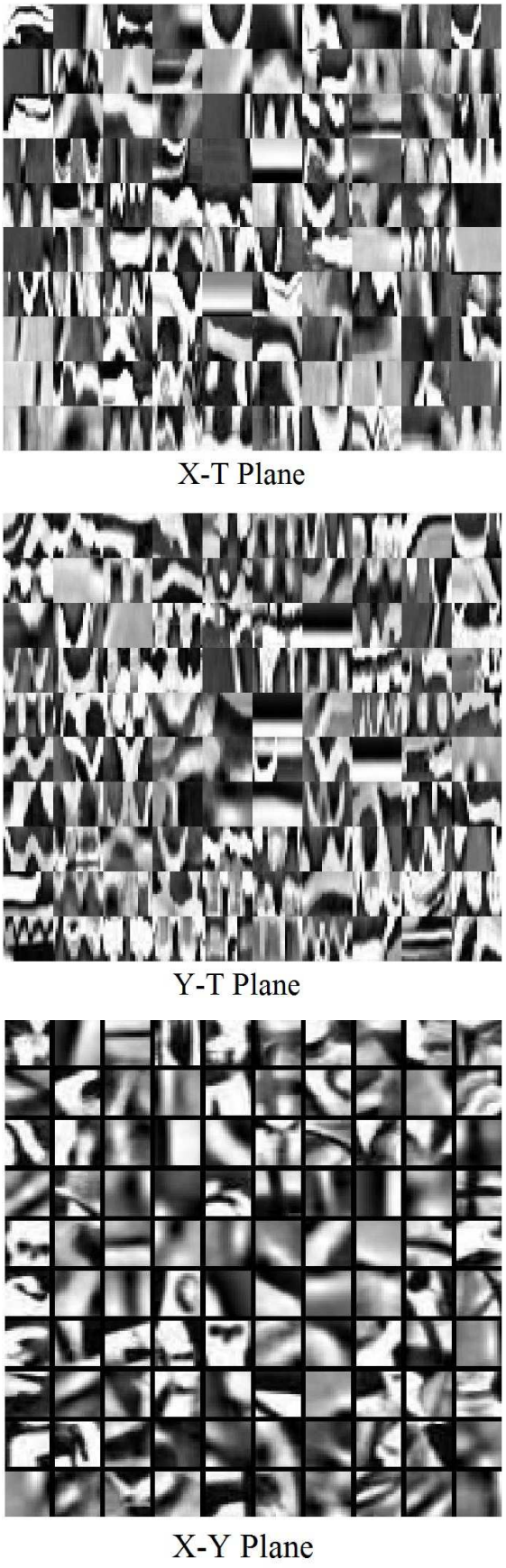

FIGURE 5.3: Spatiotemporal dictionaries learned through K-SVD for three orthogonal planes. Spatiotemporal volume size (x,y,t) is $18 \times 18 \times 20$. Complex patterns for both spatial and temporal are learned directly from the thermal infrared data and represented in the dictionary.

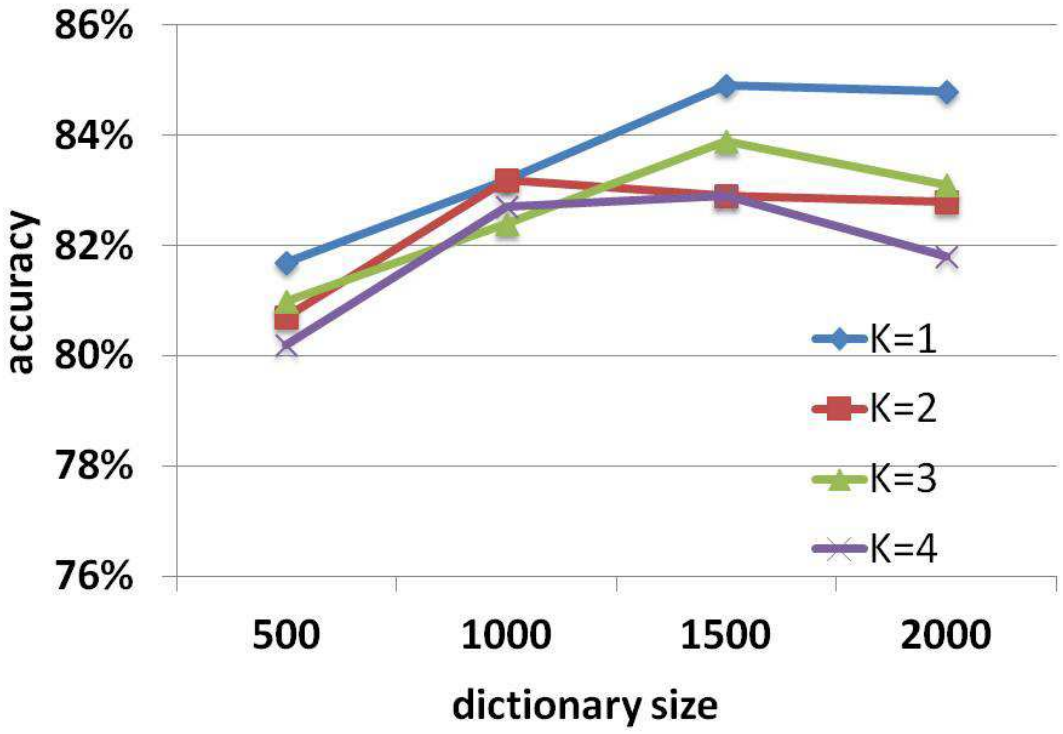

FIGURE 5.4: Recognition accuracies based on different sparsity levels (K) and dictionary sizes. When sparsity level is 1, the recognition works better, given different dictionary sizes.

TABLE 5.1: Recognition accuracies using different dictionary sizes and schemes. Dictionary size varies from 500 to 2000. In the first two rows, the local volumes are sampled without overlap. In the next two rows, there are 50% overlap between local volumes. In the last row, volumes are sampled with 50% overlapping, and the saliency map is used to construct the histogram feature.

Vol size	Dictionary Size			
	500	1000	1500	2000
18x18x20_no	67.9%	68.4%	65.3%	63.8%
36x36x20_no	56.6%	56.6%	57.2%	59.0%
18x18x20_50%	78.9%	80.9%	81.3%	80.2%
36x36x20_50%	75.1%	77.1%	77.0%	76.7%
18x18x20_50%_s	83.2%	83.8%	**86.7%**	83.3%

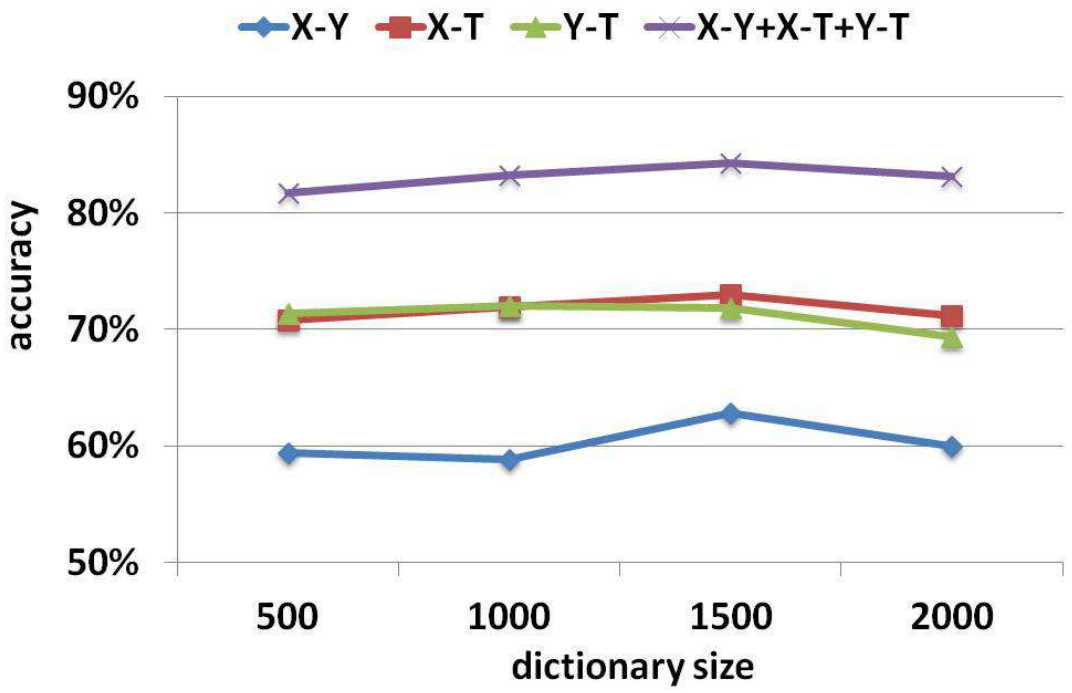

FIGURE 5.5: Recognition accuracies using single planes, and the integration of all three planes for our HSSC feature creation.

extracted from a single plane perform poorly, while combining the three planes brings great improvement for IR action recognition. In our experiments, the combination of three histograms is a direct concatenation, a weighted concatenation (e.g., [105]) could also be applied as an extension for assigning the features.

Comparison with other methods on IR data. We also compare the proposed HSSC representation with other features on our problem. In Table 5.2, four different methods are applied using the same experimental settings. Firstly we applied the Cuboids+Cuboids [5] feature. Then Harris3D+HOG3D [14] feature is used, to compare our spatiotemporal histogram of sparse code (HSSC) feature to the histogram of 3D gradient (HOG3D). Thirdly we compare our method to the LBP-TOP [105] feature, which also combines features from three planes. The experimental results show that the proposed HSSC feature is much better than the STIP or the LBP-TOP features, which indicates the superiority of our HSSC representation that learns sparse features directly from the IR data.

The confusion matrix in Figure 5.6 shows the best result (86.7%) using HSSC feature with saliency map on IR data. From the confusion matrix one can see the detailed recognition results, where most of the action categories are correctly classified and achieved 90% accuracy or above. A few actions are more difficult to recognize, e.g., 'wiping

TABLE 5.2: Comparison of thermal infrared action recognition accuracies between our method and other typical methods originally developed for visible light actions.

Method	**Accuracy**
Cuboid+Cuboid [5]	67.3%
Harris3D+HOG3D [14]	64.6%
LBP-TOP [105]	65.0%
Our Method (HSSC no saliency)	**81.3%**
Our Method (HSSC with saliency)	**86.7%**

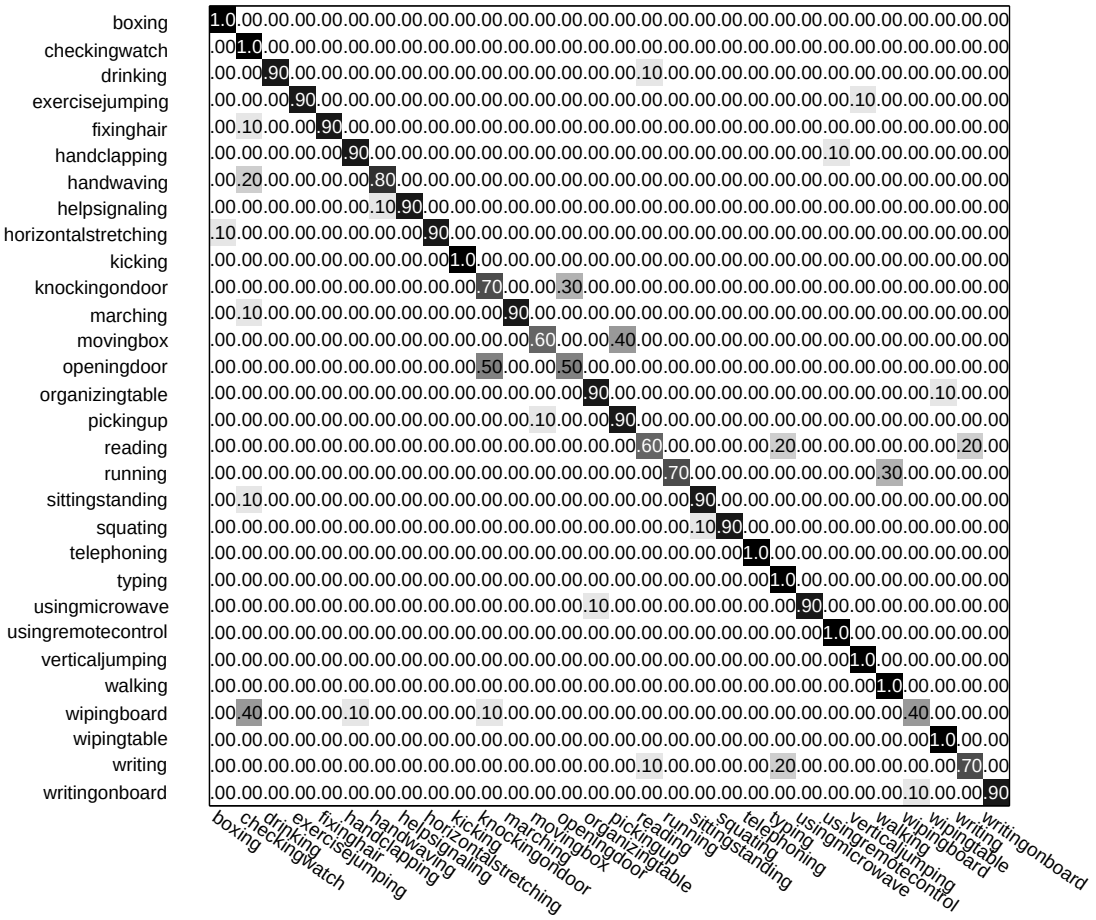

FIGURE 5.6: Confusion matrix of action recognition on a thermal infrared action database with 30 actions. Experimental settings: the spatiotemporal volume is $18 \times 18 \times 20$, dictionary size is 1500, and sparsity level is 1.

board', 'opening door', 'knocking on door', and 'reading'. Such difficulties might be caused by the similarities of motions and appearances for these actions.

5.5.4 HSSC Feature on KTH Dataset

To measure how general the proposed HSSC feature could be, experiments are conducted on the KTH action dataset [19], too. KTH dataset is the most popular database

TABLE 5.3: Action recognition accuracies reported on KTH action dataset using various methods, comparing to our method.

Methods	**Accuracy**
Kellokumpu et al. [106]	93.80%
Mattivi et al. [107]	88.30%
Yeffet et al. [108]	90.17%
Harris3D + HOGHOF [34]	91.80%
Harrid3D + HOG3D [14]	89.00%
Cuboid + HOG3D [14]	90.00%
Niebles et al. [86]	81.50%
Jhuang et al. [109] split	91.70%
Fathi et al. [110] split	90.50%
Bregonzio et al. [111]	93.20%
Kovashka et al. [112]	94.50%
Ji et al. [113]	90.20%
Le et al. [114]	93.90%
Zhang et al. [92]	92.59%
Zhu et al. [91]	94.92%
Our Method	**95.00%**

for visible light action recognition. We provide an extensive comparison with various existing methods. KTH action dataset is captured using a visible camera with 6 actions performed by 25 subjects. Following the same settings as [19], all the 2391 video clips are used in our experiment, and 16 subjects are used for training and other 9 subjects for testing. HSSC feature (with saliency map) is applied to extract histograms representing the action sequences. Then SVM with RBF kernel [41] is used for action classification. Parameters for the HSSC feature: dictionary size is 1000, spatiotemporal volume size is $18 \times 18 \times 20$, and the sparsity level is 3.

Through the experiment, our HSSC feature achieves the accuracy 95.00% on KTH dataset. The reported recognition results of the state-of-the-art methods on this dataset are presented in Table 5.3, using the same settings. We compare our results to various category of methods, e.g., methods based on local binary patterns (LBP) (1-3 rows), methods based on spatiotemporal interest points (STIP) (4-6 rows), methods using deep learning (12-13 rows), methods using sparse coding (14-15 rows), and other state-of-the-art methods. Note that the recently reported results are all very close, e.g., 93.90% in [114], 94.50% in [112] and 94.92% in [91]. Using our HSSC feature, we still obtain the highest accuracy 95.00%, comparing to all the reported results on KTH dataset. This observation further demonstrates the generality and good performance of the proposed method, even on the traditional visible light action database.

5.6 Conclusion

We have investigated human action recognition in thermal infrared spectrum. A new feature based on sparse coding is proposed. Our histogram of spatiotemporal sparse codes (HSSC) feature learns the sparse dictionary from the thermal infrared data directly, using histogram representations. It has advantages to use HSSC feature for action recognition in IR, comparing to some representative features originally developed for visible light actions. We proposed to construct the spatiotemporal histogram effectively through three orthogonal planes, integrating both spatial and temporal structures of the action videos. A saliency map is also computed to incorporate the spatial distribution of local features. Our studies have shown that richer sparse representations can be learned to improve the performance using proper dictionary size and local spatiotemporal volume. Experiments demonstrate the proposed HSSC feature can perform well on IR action videos. Besides, on the popular visible light action dataset KTH, our HSSC feature has also achieved the highest accuracy, compared to the state-of-the-art methods. Our work also suggests that learning features directly from data is very promising, which can simplify the computation especially on a large video database.

Chapter 6

Evaluating Spatiotemporal Interest Point Features for Depth-based Action Recognition

6.1 Abstract

Human action recognition has lots of real-world applications, such as natural user interface, virtual reality, intelligent surveillance, and gaming. However, it is still a very challenging problem. In action recognition using the visible light videos, the spatiotemporal interest point (STIP) based features are widely used with good performance. Recently, with the advance of depth imaging technology, a new modality has appeared for human action recognition. It is important to assess the performance and usefulness of the STIP features for action analysis on the new modality of 3D depth map. In this paper, we evaluate the spatiotemporal interest point (STIP) based features for depth-based action recognition. Different interest point detectors and descriptors are combined to form various STIP features. The bag-of-words representation and the SVM classifiers are used for action learning. Our comprehensive evaluation is conducted on four challenging 3D depth databases. Further, we use two schemes to refine the STIP features, one is to detect the interest points in RGB videos and apply to the aligned depth sequences, and the other is to use the human skeleton to remove irrelevant interest points. These refinements can help us have a deeper understanding of the STIP features on 3D depth data. Finally, we investigate a fusion of the best STIP features with the prevalent skeleton features, to present a complementary use of the STIP features for action recognition on 3D data. The fusion approach gives significantly higher accuracies than many state-of-the-art results.

6.2 Introduction

Human actions convey a significant amount of information for human interaction with the environment, human-to-human communication and human-to-machine interaction. Human action recognition is a very active research topic in computer vision, aiming to automatically recognize and interpret ongoing human actions. The ability to recognize complex human actions from videos enables the construction of several important applications such as natural user interfaces, virtual reality, intelligent surveillance and gaming [7, 8].

Although human action recognition is very important for many real-world applications, it is still a challenging problem. A number of methods have been proposed to solve the action recognition problem [7]. Among various methods, the spatiotemporal interest point (STIP) based features have shown good performance for action recognition in RGB videos [14].

Very recently, depth imaging technology has made a significant progress, which brings a broader scope for human action recognition. Using a consumer depth sensor, e.g., the Kinect [13], depth information can be captured simultaneously with the RGB videos. Moreover, from the depth maps the geometric positions of skeleton points can also be detected effectively [13]. As a result, the depth data provides a promising modality for action recognition.

In traditional RGB video-based action recognition, several spatiotemporal features have been proposed to characterize human actions using local motions in a space-time volume. Local features possess many advantages, e.g., it can avoid possible problems caused by inaccurate segmentation or partial occlusions. In the literature, many spatiotemporal feature detectors [5, 34, 109, 115] and descriptors [35, 116–118] have been proposed and shown promising performance for action recognition in RGB videos. However, it has not been well studied yet on whether these spatiotemporal interest point (STIP) features can be useful or not for depth-based action recognition.

In this paper, we perform a comprehensive evaluation of different spatiotemporal interest point features for depth-based human action recognition. In particular, three interest point detectors and six local descriptors are adopted, in total there are 14 different detector/descriptor combinations adopted for the evaluation. Experiments are conducted on four challenging depth action databases with the same experimental setup for each feature. Besides, we also extend the capability of using spatiotemporal features by utilizing the corresponding RGB videos, and the skeleton joints positions, in order to have a deep understanding of the STIP features on depth data. Two different interest points

refinement approaches are examined. Moreover, a feature-level fusion method is presented to combine the best spatiotemporal features on each database with the skeleton joints features. From the experimental results and comparisons with the state-of-the-art approaches for depth-based action recognition, we show the usefulness of spatiotemporal features for action recognition in depth videos.

The rest of the paper is organized as follows: the related work on depth-based action recognition is reviewed in Section 6.3. Different spatiotemporal interest point features are introduced in Section 6.4. Four different depth action/activity databases are presented in Section 6.5. Experiments are conducted and presented in Section 6.6. Two STIP refinement approaches are introduced and evaluated experimentally. A fusion of the best STIP features with skeleton features is shown in Section 6.7. Finally, we draw conclusions.

6.3 Related work on Depth-based Action Recognition

The depth sensors offer several advantages over traditional video cameras, e.g., working in low light conditions, giving a real 3D measure invariant to surface color and texture, resolving silhouette ambiguities in pose [13], etc. Depth sensors can significantly simplify the task of background subtraction and human detection. Because of the advantages, the depth sensors, e.g., the Kinect, have attracted researchers' attentions from many areas including 3D modeling, object recognition, gesture analysis, etc. Recently, action analysis and recognition in depth videos have become a very active topic. In this quite novel area, different approaches have been proposed. Here we give a brief overview of the methods for depth-based action recognition.

Li *et al.* [23] proposed a sampling of 80 representative 3D points to describe a salient posture. In order to select the representative points, each depth map was projected onto three orthogonal Cartesian planes: xy, xz and zy, and then a specified number of 2D points were sampled at equal distance along the contours of the projected depth data. An action graph was used to model the dynamics of actions. Their method has smaller error rates than using 2D silhouettes.

Xia *et al.* [119] proposed to use histograms of 3D joint locations (HOJ3D) for action recognition. In order to be view invariant, they aligned the spherical coordinates with the person's specific direction. The hip center joint served as the center of the coordinate system. By projecting the vector from left-hip center to the right-hip center to the horizontal plane, the horizontal reference vector was obtained. The zenith reference vector passes through the coordinate center and is perpendicular to the ground plane.

According to different joint's contribution to the body motion, they chose 9 joints to compute the 3D spatial histogram by partitioning the 3D space into 84 bins. After that, the LDA was performed to extract the dominant features, so that each frame will have a $n-1$ dimensional feature vector, where n is the number of classes. The K-means clustering was performed to represent each posture as a visual word. A discrete HMM was trained for action recognition.

Vieira *et al.* [120] proposed the Space-Time Occupancy Patterns (STOP) to represent sequences of depth maps. In their representation, the space and time axes were divided into multiple segments so that each depth map sequence was embedded in multiple 4D grids. They computed occupancy feature in each cell. After that, they employed a Nearest Neighbor classifier based on the cosine distance for action recognition.

Yang and Tian [121] combined static posture, motion property, and overall dynamics to form an action feature descriptor called EigenJoints. In order to remove noisy frames and reduce computational cost, they performed informative frame selection based on Accumulated Motion Energy (AME). A non-parametric Naive-Bayes-Nearest-Neighbor (NBNN) classifier was used for action classification.

In order to make skeleton representation invariant to sensor orientation and global translation of the body, Miranda *et al.* [122] proposed a pose descriptor vector in a torso-based coordinate system. A predefined key pose set was used to build SVM classifiers. Because each gesture can be viewed as a sequence of key poses, a decision forest was used to search for key pose sequences. In recognition stage, the key pose classifiers can recognize key poses performed by the user and then determine the corresponding gesture class.

Yang *et al.* [123] proposed to generate three 2D Depth Motion Maps (DMM) from each 3D depth frame according to front, side, and top views. The HOG feature is computed from DMM to represent an action video. They used a linear SVM classifier to recognize actions.

In [84], Wang *et al.* extracted two features, pairwise relative positions and Local Occupancy Patterns at each joint. Each skeleton joint i has 3 coordinates $F_i(t) = (x_i(t), y_i(t), z_i(t))$ at frame t, the pairwise relative position features are extracted for joint i as: $p_i = \{p_{ij} | i \neq j\} = \{p_i - p_j | i \neq j\}$. In order to model the interaction between human subject and objects, they computed the LOP feature based on the 3D point cloud around a particular joint. After that, Fourier temporal pyramid was used to represent the temporal dynamics of the frame-level features. In order to deal with the errors of the skeleton tracking and better characterize the intra-class variations, they defined an actionlet as a conjunction structure on base features. One base feature is the Fourier

pyramid feature of each joint. A data mining algorithm was used to find discriminative actionlets for action recognition.

In [124], Sung *et al.* used all three channels, i.e., RGB, depth and skeleton positions, for human activity recognition. They extracted hand position information, body pose features and motion from skeleton joints. For both RGB and depth images, they used the Histogram of Oriented Gradients (HOG) feature in two settings. One is to compute HOG in both the RGB and depth within the bounding box of the person. The other is to get the bounding boxes for the head, torso, left arm, and right arm, based on the skeleton locations, and compute the HOG in RGB and depth with each of the four bounding boxes. A two-layered maximum-entropy Markov model was trained to capture the hierarchies of human activities and transitions between sub-activities over time.

Wang *et al.* [125] proposed a semi-local feature called Random Occupancy Patterns (ROP). A depth sequence is treated as a 4D volume. Given a subvolume, the ROP feature was computed as: $o_{xyz} = \delta(\Sigma_{q \in bin_{xyzt}} I_q)$, where $I_q = 1$ if the point cloud has a point in the location q and $I_q = 0$ otherwise. $\delta(\cdot)$ is a sigmoid normalization function: $\delta(x) = \frac{1}{1+e^{-\beta x}}$. Because the sizes of the 4D subvolume are extremely large and the features are highly redundant, a weighted sampling method was applied to reduce the complexity and obtain the discriminative features. They also utilized a sparse coding method to robustly encode those features. The SVM classifier was used for classification.

More recently, Oreifej *et al.* [126] represented the depth sequence using a histogram capturing the distribution of the surface normal orientation in 4D space of time, depth, and spatial coordinates (HON4D feature). A 600-cell polychoron with 120 vertices was used to quantize the 4D space and represent possible directions of the 4D normals. The SVM classifier was used for action classification.

Koppula *et al.* [127] proposed to jointly model the human activities and object affordances as a Markov Random Field where the nodes represent objects and sub-activities, and the edges represent the relationships between object affordances, their relations with sub-activities, and their evolutions over time. In order to find atomic movements in an activity, they also performed temporal segmentation of the frames. They used a multi-class SVM classifier for action recognition.

Ni *et al.* [128] proposed the Depth-Layered Multi-Channel STIP (DLMC-STIP) and Three-Dimensional Motion History Images (3D-MHIs). For DLMC-STIP, after getting local feature descriptors in a video, they introduced a set of (M) depth layers $L_1^z = [z_1^l, z_1^u]$, $L_2^z = [z_2^l, z_2^u]$, ..., $L_M^z = [z_M^l, z_M^u]$, with lower and upper boundaries denoted as z_M^l and z_M^u for the $m-th$ depth layer, so a detected spatio-temporal interest point by Harris3D detector would be located in one specific layer. In this way they formed

multi-channel histograms for feature description using the HOGHOF descriptor. The 3D-MHIs are motion history images (MHIs), including both forward-DMHIs (fDMHIs) and backward-DMHIs (bDMHIs). The SVM classifiers were used for action recognition.

Zhao *et al.* [129] explored the combination of RGB channel and depth for action recognition. They extracted interest points from RGB videos. They proposed local depth pattern (LDP) to represent each local video volume at each interest point position extracted from visible light videos, and adopted to the corresponding depth videos. Given an interest point p, its local region is partitioned into $N_x \times N_y$ spatial cells. Each cell is of size (S_x, S_y) pixels. For each cell, they computed an average depth value and then the difference of average depth values between every cell pair to form the LDP feature. For each interest point p, which can be detected by the Harris3D detector on either RGB video or depth sequence, the output feature vector can be denoted as $S_p = (x, y, t, F)$, where (x, y, t) denote the coordinates and time of interest point, and action feature F could be obtained either by HOGHOF descriptor or LDP. They explored different combinations of RGB and depth map features and used the SVM classifiers.

Inspired by Dollar's work on local features [5], Zhang *et al.* [130] developed a 4D local spatio-temporal feature which combines both intensity and depth information. They first applied separate filters along the 3D spatial dimensions and the temporal dimension to detect interest point. Then they computed and concatenated the intensity and depth gradients with a 4D hyper cuboid to obtain features for an action sequence. The Latent Dirichlet Allocation with Gibbs sampling was used as the classifier.

Also inspired by Dollar's local interest point detector [5], Xia and Aggarwal [2] proposed a spatiotemporal interest point detector on depth map, which effectively eliminate the noise ('value jumps' and 'holes') appear on depth maps. They extended the Cuboid detector [5] to the fourth dimension. A depth cuboid similarity descriptor is proposed to describe the local feature, based on the similarity between all pair of blocks in the 3D cuboid. Finally a feature selection process based on F-score is applied to generate the feature vector, and then used for action classification.

From the overview of related works on depth-based action recognition, we show that many of the approaches were motivated by the methods originally developed for RGB action recognition, e.g., motion history and the spatiotemporal interest point features. Although the STIP features prevail for analyzing color/intensity actions with good performance, only very limited types of STIP features were applied to depth-based action recognition. It has not been well studied yet on the performance of the typical STIP features on 3D depth actions. Thus it is important to evaluate the representative STIP features, so that a better understanding of the STIP features can be obtained for 3D

depth-based action analysis. Our goal is to measure the usefulness of the STIP features for 3D action recognition, and build benchmark results of these features on several depth-based action databases.

In the following, we briefly introduce the STIP features that we used for the evaluation, and then present the databases and the evaluation results.

6.4 Spatiotemporal Interest Point Features

Different Spatiotemporal Interest Point (STIP) features have been proposed for action characterization in RGB videos with good performance [14]. For example, Laptev and Lindeberg [37] used some effective methods to make STIP velocity-adaptive as well as spatially and temporally invariant. Willems *et al.* [35] presented a method to detect features under scale changes, in-plane rotations, video compression and camera motion, the extended SURF descriptor was also proposed in this work. Dollar *et al.* [5] proposed the cuboids detectors and descriptors for action analysis. Jhuang *et al.* [109] used local descriptors with space-time gradients as well as optical flow. Klaser *et al.* [36] compared space-time HOG3D descriptor with HOG and HOF descriptors [85]. Recently, Wang *et al.* [14] conducted an evaluation of different detectors and descriptors on four RGB/intensity action databases. Shabani *et al.* [131] evaluated the motion-based and structured-based detectors for action recognition in color/intensity videos. However, there is no evaluation of the STIP features on 3D depth videos.

In Wang *et al.*'s work [14], it was observed that although the spatiotemporal interest point features perform differently on different databases, their performances are quite similar on the same database. Our evaluation will show that the STIP features perform quite differently on the same depth database (See Section 6.6). In the following, we introduce the specific STIP features that are used in our evaluation.

6.4.1 Interest points detectors

The Harris3D detector was proposed in [34]. It locates the spatiotemporal volumes with large variations along space and temporal directions in a video sequence. A spatiotemporal second-moment matrix is used to model a video sequence f,

$\mu = g(\cdot) \times \begin{pmatrix} L_x^2 & L_xL_y & L_xL_t \\ L_xL_y & L_y^2 & L_yL_t \\ L_xL_t & L_yL_t & L_t^2 \end{pmatrix}$, where $g(\cdot)$ is a Gaussian function for weighting and L is the convolution of f with a spatiotemporal Gaussian derivative kernel. The

interest point locations are determined by computing the local maxima of the response function $H = det(\mu) - k \cdot trace^3(\mu)$.

The Cuboids [5] detector computes the interest point location by the local maxima of the response function R, which is defined as: $R = (I * g * h_{ev})^2 + (I * g * h_{od})^2$, where g is the 2D Gaussian smoothing kernel, h_{ev} and h_{od} are a quadrature pair of 1D Gabor filter, which are computed by $h_{ev} = -cos(2\pi t\omega)e^{-t^2/\tau^2}$ and $h_{ev} = -sin(2\pi t\omega)e^{-t^2/\tau^2}$.

Willems *et al.* [35] proposed the Hessian detector, which measures the strength of each interest point using the Hessian matrix. The response function is defined as $S = |det(H)|$, where H is the Hessian matrix.

6.4.2 Local feature descriptors

Given a set of interest point locations, various feature descriptors can be applied to characterize the local space-time content. Given the spatial scale σ and temporal scale τ at each interest point location, a local volume is used to extract features.

Kläser *et al.* extended the histograms of oriented gradient (HOG) to HOG3D, which is the histogram of 3D gradient orientations. Integral videos are computed for efficiency.

HOG/HOF descriptor was proposed by Laptev *et al.* [85], using the combination of histogram of gradient (HoG) and histogram of optical flow (HoF) accumulated from the local volume.

The Cuboids descriptor was proposed along with the Cuboids detector in [5]. For each detected point (x, y, t, σ, τ), a feature descriptor is computed in a 3D patch centered at (x, y, t). The gradient at each spatiotemporal location is computed within the cuboid and the histogram is computed as the feature vector. The PCA can be applied to reduce the dimensionality.

The extended SURF (ESURF) descriptor [35] was proposed with the Hessian detector, which is an extension of the SURF [38]. For each local volume, the feature vector is computed using the sum of uniformly sampled responses of Haar-waveletes along three directions.

We will evaluate the above three interest point detectors and six local descriptors for 3D action recognition. Although there exist some works using the STIP features for depth-based action recognition [2, 129, 130], only very limited types of STIP features were investigated. Through the evaluation of several representative STIP features on multiple depth databases, we will not only provide the benchmark results of STIP features on

TABLE 6.1: Depth-based action/activities databases. In the 4th column, RGB denotes color images, DEP denotes depth maps, and SK denotes skeleton joints positions. The 5th column shows the average length of each video in the dataset.

Database	**# of act.**	**# of subj.**	**# of vid.**	**# of channels**	**Vid Len**
MSR-Action3D	20	10	557	DEP, SK	˜1s
MSRDailyActivity3D	16	10	320	RGB, DEP,SK	˜6s
UTKinect-Action	10	10	200	RGB, DEP,SK	˜3s
CAD-60	12	4	60	RGB, DEP,SK	˜45s

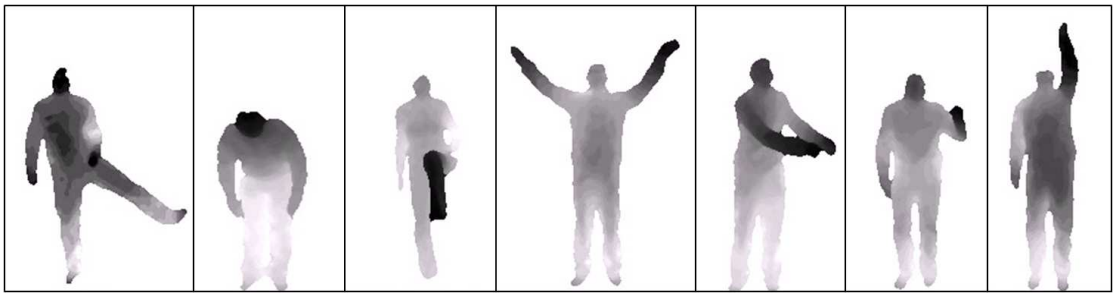

FIGURE 6.1: Some samples from MSRAction3D Dataset. 7 depth images are showed. The actions shown are (from left to right): side kick, bend, jog, high arm wave, golf swing, pickup&throw and high throw.

depth data, but also find the best, appropriate STIP features that may help to improve the accuracies significantly [132] for depth-based action recognition.

6.5 Databases

In order to perform a comprehensive evaluation, we conduct experiments on four different depth databases, which were captured under different scenarios and/or environments. The evaluation on these databases can provide a thorough test of various STIP features on depth data. Table 6.1 shows a brief description of the four depth-based action/activity databases. More details of these databases are given as follows.

6.5.1 MSR-Action3D Dataset

MSR-Action3D Dataset [23] was captured by a depth camera similar to the Kinect sensor. This dataset contains 20 actions, and each action was performed by 10 subjects three times. Two channels of data are provided: depth sequences at 15 frames per second (fps) with resolution of 640×480, and skeleton joint positions in each frame. The 20 actions are: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend,*

TABLE 6.2: Three subsets of actions used for the experiments on MSRAction3D dataset.

AS1	**AS2**	**AS3**
Horizontal arm wave	High arm wave	Hight throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

FIGURE 6.2: Sample depth images from MSRDailyActivity3D Dataset. Actions in the top row (left to right): use laptop, use vacuum cleaner, cheer up, and lay down on sofa. Action classes in the bottom row: toss paper, stand up, walk, and play guitar.

forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, and *pick up & throw* (see Figure 6.1 for some example images).

6.5.2 MSRDailyActivity3D Dataset

This dataset was collected for human daily activities by a Kinect device [84]. In total there are 16 activities in this dataset: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up,* and *sit down.* Each subject performed an activity twice, one "sitting on sofa" and the other "standing". The total number of videos is 320. Three channels of data, i.e., RGB, depth and skeleton joint positions are provided in this dataset. See Figure 6.2 for some examples of depth images in this dataset.

FIGURE 6.3: Sample images from UTKinect-Action Dataset. Action classes in the top row: walk, wave hands, sit down, and throw. Action classes in the bottom row: pick up, clap hands, carry and push.

6.5.3 UTKinect-Action Dataset

The action videos of the UTKinect-Action Dataset [119], were collected by a single stationary Kinect with the distance ranges from 4 to 11 feet. There are totally 10 action classes performed by 10 subjects. Each subject performed each action twice. The RGB, depth and skeleton joint locations are synchronized and all three channels are provided. Some examples of depth images are shown in Figure 6.3. The resolution of RGB images is 640×480, the depth image resolution is 320×240. The 10 action classes are: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands,* and *clap hands.*

6.5.4 CAD-60 Dataset

Cornell Activity Dataset-60 (CAD-60) [124], contains 60 RGB-D videos collected by a Kinect sensor with the distance ranges from 1.2m to 3.5m, the resolution of the depth sequences is 640×480, and captured at 15 fps. There are 4 different subjects and 12 different actions. The action videos were captured in five different locations, with 3 to 4 common activities performed at each location. The five locations are: office, kitchen, bedroom, bathroom and living room. Figure 6.4 shows some example depth images from this dataset. All the RGB, depth and skeleton data are provided in this dataset.

6.6 Evaluations

We present the experimental settings in Section 6.6.1, the evaluation results for various combinations of detectors and descriptors in Section 6.6.2, and two STIP refinement

FIGURE 6.4: Examples depth images from the CAD-60 Dataset to illustrate the actions.

approaches along with the corresponding results in Section 6.6.3.

6.6.1 Experimental settings

The bag-of-words representation is used for the spatiotemporal interest points. First, different STIP detectors are applied to the depth sequences. Given the detected locations, different local descriptors are used to characterize the space-time volume around each interest point. These local features are then quantized into visual words, so that a depth action sequence can be represented as a histogram of the visual words. In our evaluation, vocabularies are constructed using the K-means clustering technique. We empirically set the vocabulary size to be 200, 300, 850 and 1550, respectively, for the MSRDailyActivity3D, MSRAction3D, CAD-60 dataset and UTKinect-Action datasets, depending on the database size and empirical performance. After quantization, the histograms of visual words are used as the features for action classification. The multi-class support vector machines (SVMs) are used for action learning, with a linear kernel for the CAD-60 dataset and χ^2-kernel for the other three datasets, based on our empirical comparisons between different kernels. The χ^2-kernel is defined by:

$K(H_i, H_j) = exp\left(-\frac{1}{2A}\Sigma_{n=1}^{V}\frac{(h_{in}-h_{jn})^2}{h_{in}+h_{jn}}\right)$, where $H_i = \{h_{in}\}$ and $H_j = \{h_{jn}\}$ are the frequency histograms of the visual word occurrences, and V is the vocabulary size. A is the mean value of distances between all training samples.

For different feature representations, we utilize the implementations or source code provided by the authors, mostly with the default parameter settings, since some executable code cannot be modified. All the experiments were conducted on a 64-bit operating system DELL Optiplex 790 PC, with i7 3.4GHz CPU and 12G RAM.

Specifically, for the Harris3D detector, we used the original implementation with the default parameter settings: $k = 0.0005, \sigma^2 = [4, 8, 16, 32, 64, 128]$ and $\tau^2 = [2, 4]$. For the Cuboids detector [5], we ran the authors' implementation and the default scale values $\sigma = 2, \tau = 4$ were used in our evaluation. The UTKinect-Action dataset has typically shorter video clips, we used $\sigma = 2, \tau = 2$ for the Cuboids detector. For the Hessian detector [35], the executable code was used with the default parameter setting.

For the HOG/HOF descriptor, we followed [85] and adopted the grid parameters $n_x = n_y = 3, n_t = 2, \sigma^2 = 4$ and $\tau^2 = 2$. For the HOG3D descriptor [36], we used the parameters $n_x = n_y = 5, n_t = 4, \sigma = 2$ and $\tau = 2$ for the UTKinect-Action dataset and $n_x = n_y = 2, n_t = 5, \sigma = 2$ and $\tau = 4$ for the other three datasets in our evaluation. For the Cuboid descriptor [5], we applied the descriptor size $\Delta_x(\sigma) = \Delta_y(\sigma) = 2\sigma + 1, \Delta_t(\tau) = 2\tau + 1$, where $\sigma = 2$, $\tau = 4$. The PCA was applied to reduce the feature dimensions to 100. For the ESURF descriptor, we used the executable code with default parameter settings [35]: $\Delta_x(\sigma) = \Delta_y(\sigma) = 3\sigma, \Delta_t(\tau) = 3\tau$.

For all depth databases, the depth sequences are firstly transformed and stored into gray level videos (depth videos). The skeleton joint positions are also stored for each frame. Then the spatiotemporal features are extracted from the depth videos for each database.

6.6.2 Evaluation Results

The evaluation results are presented in the following, using all four datasets.

6.6.2.1 On MSRAction3D Dataset

MSRAction3D is a commonly used dataset for 3D action recognition. We followed the same settings as [23], where the dataset is divided into 3 subsets, each consisting of 8 actions (see Table6.2). Then a cross-subject scheme is used in our evaluation, with half of the subjects for training and the remaining half for testing. The overall accuracy is obtained by taking the average of three subsets. The results of different

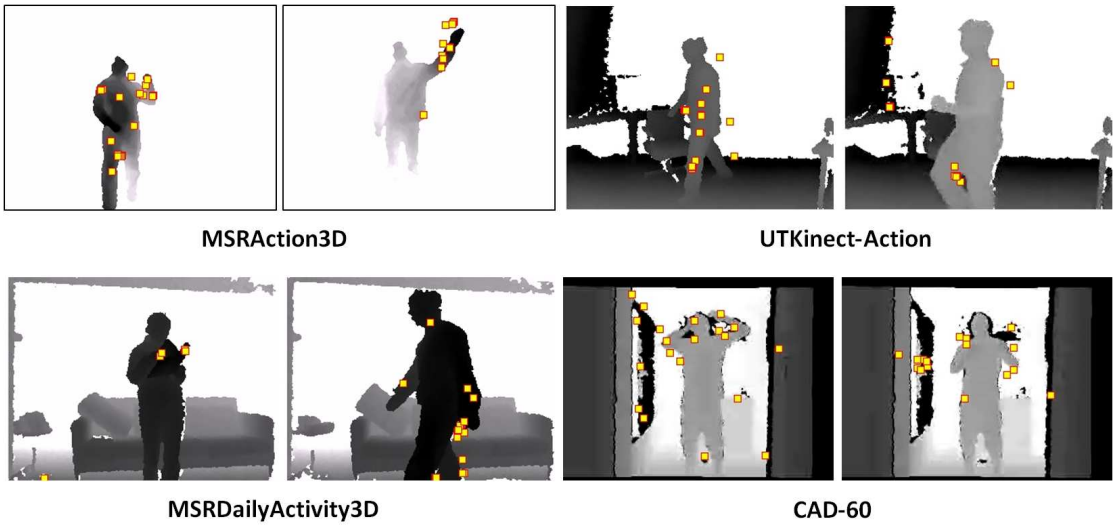

FIGURE 6.5: Illustration of the spatiotemporal interest points detected on depth sequences from four datasets.

TABLE 6.3: Accuracies of different STIP features on MSRAction3D dataset. Different detectors and descriptors are combined. Some combinations cannot be realized because of the non-separable executable code.

	HOG3D	**HOG/HOF**	**HOG**	**HOF**	**Cuboids**	**ESURF**
Harris3D	76.1%	**80.8%**	72.3%	77.3%	-	-
Cuboids	77.3%	78.7%	68.5%	71.0%	70.0%	-
Hessian	60.3%	55.9%	47.3%	44.9%	-	47.1%

detectors/descriptors on this dataset are showed in Table 6.3. One can see that the STIP features have very different accuracies on the same database, ranging from 47.1% to 80.8%, when different detectors and descriptors are used. This observation is very different from the results on color/gray level action videos [14], where the different STIP features have similar accuracies on the same database. This evaluation indicates the significant difference between 3D depth and color/gray level videos in action recognition.

The highest accuracy is achieved by Harris3D+HOG/HOF feature with a recognition accuracy of 80.8%. This accuracy is comparable to some state-of-the-art approaches, but lower than the highest in the literature by more than 10% (see Table 6.10 for the state-of-the-art results on MSRAction3D). Note that in [84] the skeleton joints information was used while in our evaluation of STIP features, only the depth videos are used. One reason that might impact the accuracy is that the interest points cannot be detected for several depth sequences where the lengths of the sequences are quite short.

TABLE 6.4: Subsets of actions used for the experiments on MSRDailyActivity3D dataset.

AS1	**AS2**
Read book	Drink
Write on a paper	Eat
Use laptop	Call cellphone
Use vacuum cleaner	Cheer up
Sit still	Lay down on sofa
Toss paper	Walk
Play game	Stand up
Play guitar	Sit down

TABLE 6.5: Accuracies of various STIP features on MSRDailyActivity3D dataset.

	HOG3D	**HOG/HOF**	**HOG**	**HOF**	**Cuboids**	**ESURF**
Harris3D	60.6%	67.5%	63.8%	59.4%	-	-
Cuboids	68.8%	**70.6%**	68.1%	58.1%	64.4%	-
Hessian	**70.6%**	63.8%	61.9%	63.1%	-	65.6%

6.6.2.2 On MSRDailyActivity3D Dataset

The MSRDailyActivity3D dataset contains 16 activities performed by 10 subjects in two scenarios: sitting and standing. Similar to the partition in [23], we divided this dataset into 2 subsets, and evaluate the performance considering two different scenarios, sitting and standing, respectively. We consider the activities in each subset according to the motions: subset 1 (AS1) contains activities without much motion and subset 2 (AS2) with obvious motion. Table 6.4 shows how we divide the subsets. In our evaluation, we adopt the cross-subject test scheme, using half of the subjects for training and the remaining half for testing. The final results are obtained by averaging accuracies over the subsets.

The evaluation results on MSRDailyActivity3D dataset using different combinations of detectors and descriptors are presented in Table 6.5. Again, the STIP features achieved very different accuracies. The highest accuracy is obtained by Cuboids+HOG/HOF and Hessian+HOG3D, with an accuracy of 70.6%. The result is lower than the reported results, e.g., Oreifej *et al.* got 80% accuracy with HON4D feature in [126]. The highest accuracy from previous approaches is 85.8% obtained in [84]. In our evaluation, all the combinations of detector/descriptors are above 58%. In the subset with more motion, the performance of STIP is much better ($\sim$ 80%) than the subset with less motion ($\sim$ 50%). This demonstrates that the STIP features can characterize actions with significant motions, but not static actions like sitting. Further, the STIP features cannot represent the human-object interaction. There are several activities in this dataset with similar motion but different objects, e.g., reading and writing, eating and drinking, etc. We also

FIGURE 6.6: Examples of interest points that are detected from the background (MSRActivity3D dataset).

observe that many of the interest points are detected on depth sequences irrelevant to the actions (see Figure 6.6). This inspires us to evaluate some refinement schemes for the STIP features (to be shown later).

6.6.2.3 On UTKinect-Action Dataset

The evaluation results on the UTKinect-Action dataset are showed in Table 6.6. Note that because many depth sequences in this dataset are of length about 10 frames, which is too short for space-time interest point detection. Thus a preprocessing is conducted for the depth videos where 10 frames are copied to expand the length of video from both the starting and ending frames.

From the results, the best accuracy is 81%, obtained by Harris3D+HOG3D. This result is lower than the result 90.9% in [119], and the highest accuracy 91.5% in [133]. Note that in [119] and [133] the *leave-one-out cross-validation* scheme was applied but we use half of the subjects for training and the other half for testing. Figure 6.7 shows the confusion matrix of the best STIP feature. Most of the actions are correctly recognized, while the

TABLE 6.6: Accuracies of various STIP features on UTKinect-Action dataset. Note that we use half subjects for training and the remaining half for testing. There are 100 samples in total in the test set.

	HOG3D	**HOG/HOF**	**HOG**	**HOF**	**Cuboids**	**ESURF**
Harris3D	**81.0%**	80.0%	66.0%	69.0%	-	-
Cuboids	65.0%	65.0%	56.0%	57.0%	67.0%	-
Hessian	69.0%	56.0%	57.0%	53.0%	-	65.0%

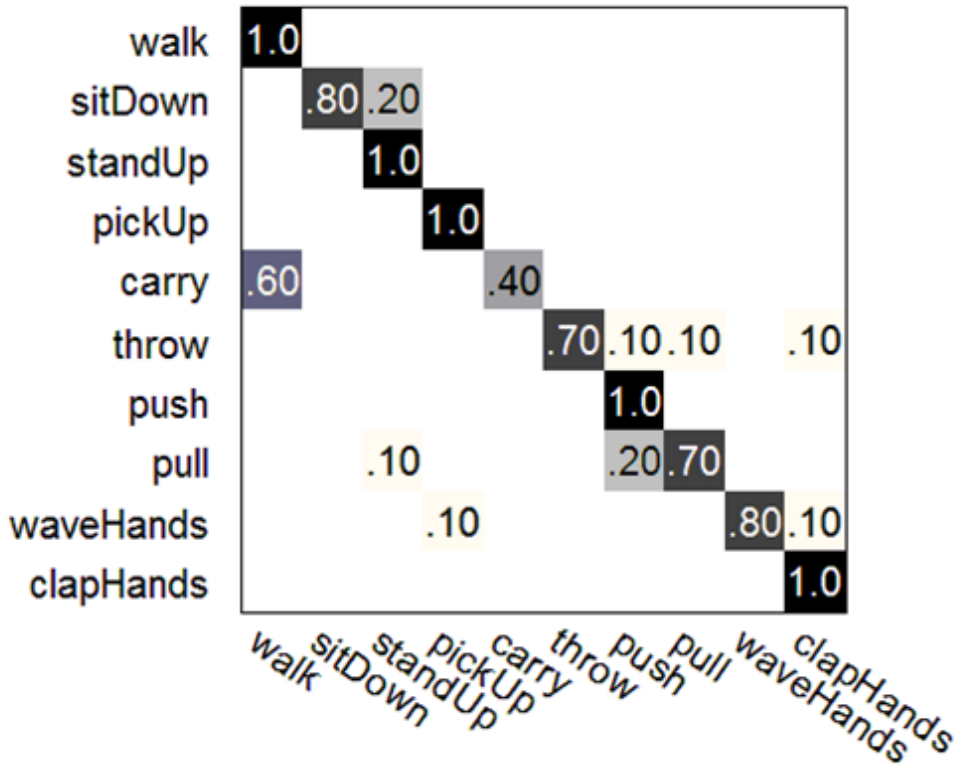

FIGURE 6.7: Confusion matrix for the feature Harris3D+HOG3D on UTKinect-Action dataset.

action "carry" has a much lower recognition rate, i.e., 60% of the testing samples are incorrectly classified as "walk". These two actions are quite similar in the dataset, since "carrying" is performed by a "walking" subject who is holding an object, while the STIP features might not correctly detect the corresponding object regions thus the object feature might not be well-represented, which might cause the misclassifications for these two actions.

TABLE 6.7: Accuracies of various STIP features on CAD-60 dataset.

	HOG3D	**HOG/HOF**	**HOG**	**HOF**	**Cuboids**	**ESURF**
Harris3D	43.8%	50.0%	43.8%	37.5%	-	-
Cuboids	50.0%	31.3%	37.5%	37.5%	43.8%	-
Hessian	43.8%	50.0%	56.3%	43.8%	-	**62.5%**

6.6.2.4 On Cornell Activity Dataset (CAD-60)

For the CAD-60 dataset, all the depth videos are sampled to 500 frames in our evaluation. All the activity categories (12 desired activities and a random activity) in this dataset are used in our evaluation as in [124]. The same experimental settings are adopted, i.e., three subjects for training, while the remaining for testing.

The evaluation results are shown in Table 6.7. Among the various features, the Hessian+ESURF gives the highest accuracy 62.5%. From the confusion matrix (Figure 6.8), one can see that some of the similar activities on depth sequence are incorrectly recognized, e.g., talkOnCouch and relaxOnCouch, and the random activity in this dataset also influences the recognition rate, where the talkOnPhone activity is recognized incorrectly as the random activity.

In [124], the precision/recall is reported as the performance measurement (67.9%/55.5%). Yang *et al.* [121] reported 71.9%/66.6% on this dataset. Koppula *et al.* [127] reported the 80.8%/71.4%. We also compute precision/recall for the feature Hessian+ESURF. The result achieves 66.7%/59.0%. Note that in our experiment we do not divide the different environment into different subset as [127]. The noisy background in depth sequences (see Figure 6.5) impact the detection of interest points with many interest points detected from the background. This drawback can be overcome when human segmentation is applied. We will investigate some refinement to reduce the effect of background noise on depth-based action recognition.

6.6.3 Refinements of the STIP features

In the above experiments, various STIP features are evaluated on depth videos with recognition accuracies reported. The best accuracies on each database are comparable to, but lower than some state-of-the-art methods that are developed especially for 3D action analysis. Note that the synchronized RGB videos and the human skeleton joints positions [13] are usually provided with the depth sequences. Intuitively these different sources of data can be used as the complementary information for human action recognition. Thus in our evaluation, we attempt to further utilize the RGB videos and the skeleton joints positions, to enhance the performance for action recognition on depth

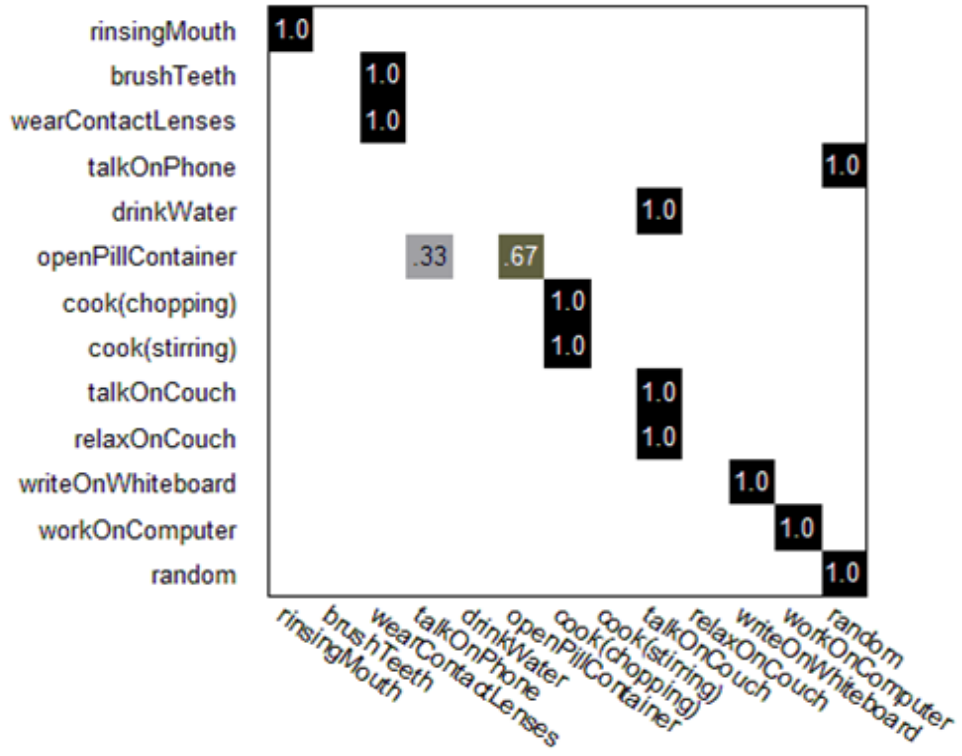

FIGURE 6.8: Confusion matrix for the feature Hessian+ESURF on CAD-60 dataset.

data. In this way, we can understand the STIP features deeper in depth videos. Two approaches are investigated in the following.

6.6.3.1 STIP feature refinement using Skeleton Joints

Shotton *et al.* [13] developed an efficient technique for human skeleton detection with 20 joint positions. Since the STIP features have a drawback, i.e., the spatial relations or distributions of the interest points cannot be utilized. From the above experiments, we observe that the detected interest points on depth images can be in the background or not accurate because of the noise in depth data. Therefore, we demonstrate that on depth images, the refinement of interest point detection could be done by using the skeleton. It is based on constraining the locations of STIP according to the skeleton joints. The idea is different from the work [2], but aims at the same goal—interest points refinement. Specifically, we define a bounding box around the subject at each frame t. The bounding box at frame t is obtained by the temporal images from time $t-5$ to $t+5$, and the maximum boundaries are selected and shifted by 30 pixels to each side to construct the new bounding box. Then the STIP which are detected on the whole depth sequences are constrained within the new box. STIP detections which lie outside

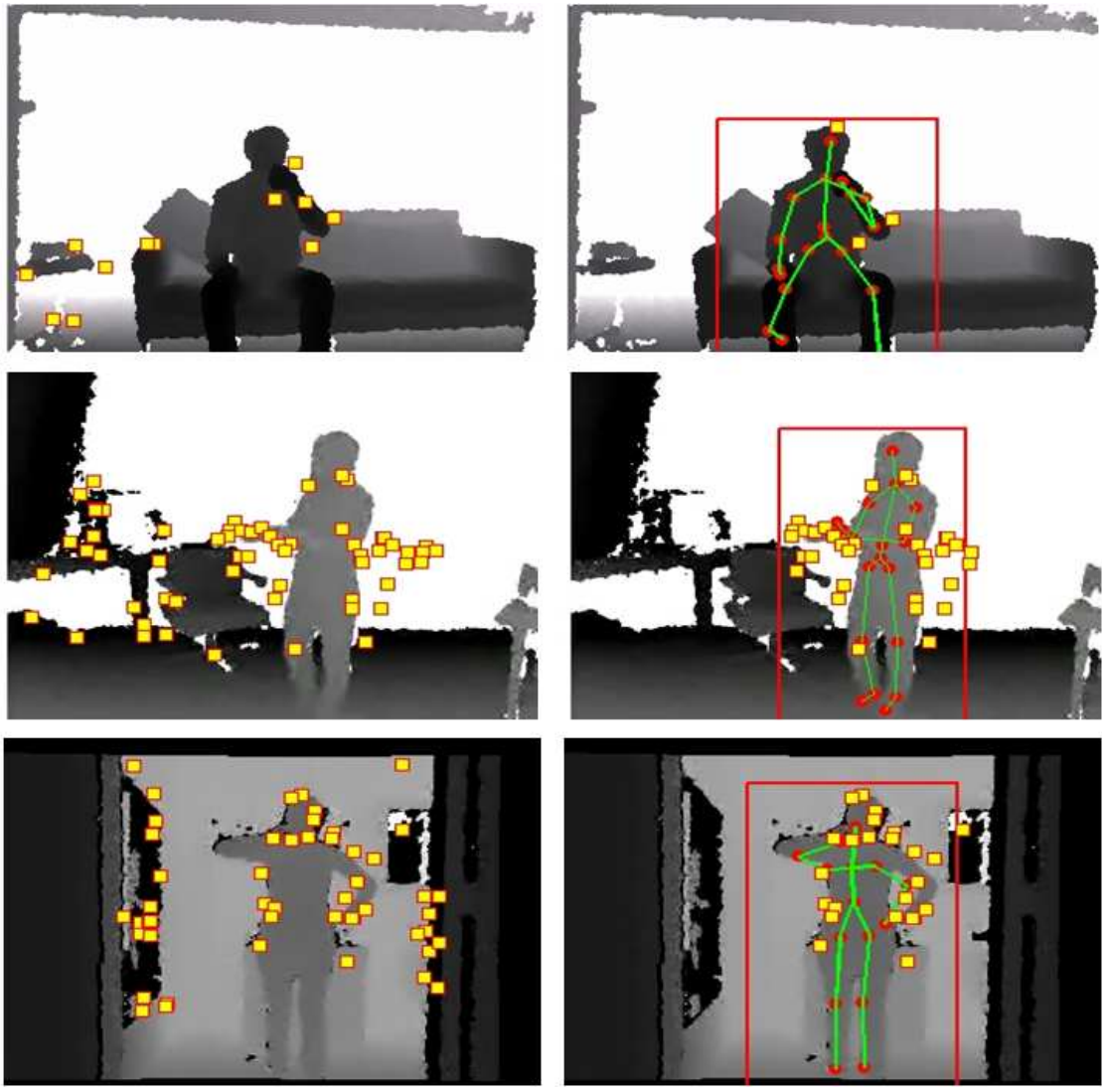

FIGURE 6.9: Examples of STIP refinement on different datasets. Left column shows the original interest points detected, right column shows the interest points after refinement by the human bounding box derived from the skeleton joints.

the bounding box are considered as from the background, and thus are eliminated (see Figure 6.9). Finally, we do the evaluation again using the same experimental settings as previous, only a smaller K in K-means clustering because of the reduced number of interest points.

The evaluation results using this STIP refinement scheme on four datasets are shown in Figure 6.10. From the results we observe that (1) most of the features can get better results when applied the STIP refinement, e.g., on MSRAction3D dataset, the accuracy of Cuboids + Cuboids feature increases by 4.2% after the refinement; on MSRDailyActivity3D dataset, an 11.9% increase is achieved for the Hessian + ESURF feature; and on UTKinect-Action dataset, the accuracy is increased by 13% for Cuboids + HOG/HOF feature. We also notice that on the CAD-60 dataset, the STIP refinement method does

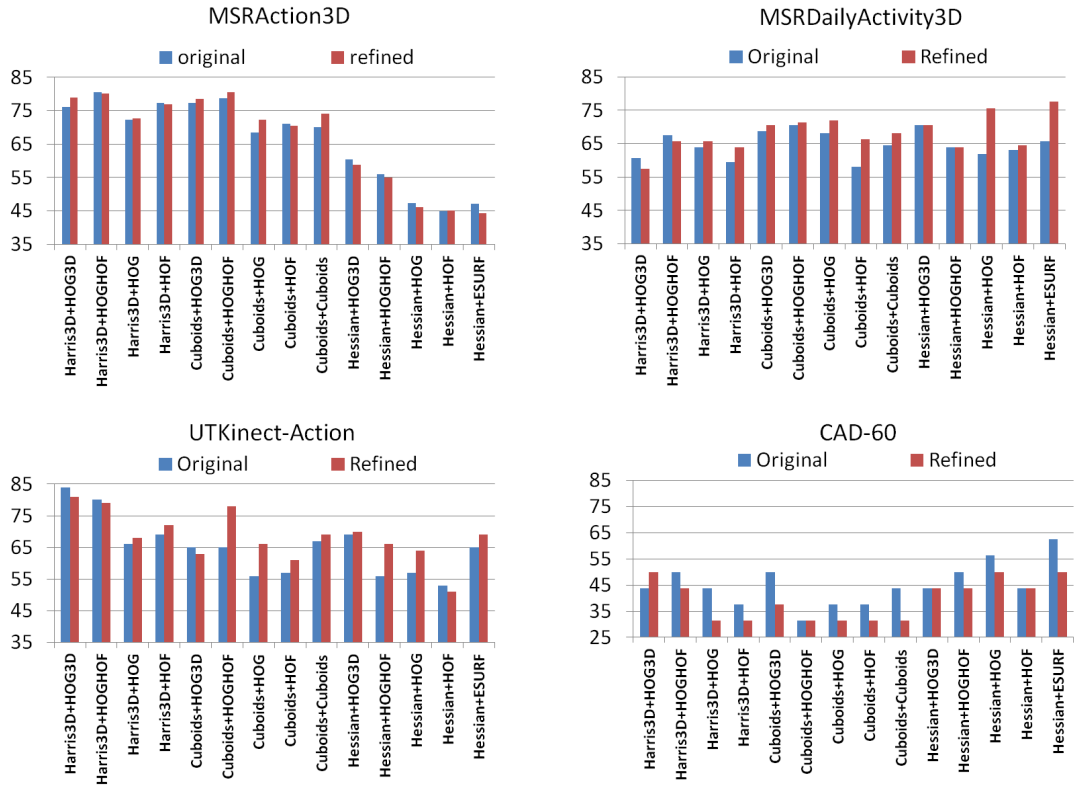

FIGURE 6.10: Bar graph of the recognition accuracies before and after the refinements on different datasets. The vertical axis denotes the recognition accuracy (%).

TABLE 6.8: Accuracies using skeleton and RGB refinement approaches. Two cells have no results since the MSRAction 3D dataset does not contain RGB data.

	MSRDailyActivity3D	UTKinect-Action	CAD-60	MSRAction3D
Original	70.6%	81.0%	56.3%	80.8%
RGB Refined	75.6%	85.0%	68.8%	–
Skeleton Refined	72.5%	84.0%	50.0%	81.7%
Sk+RGB Refined	77.5%	85.0%	62.5%	–

not improve the accuracies. One reason might be that the dataset was collected in five different locations and certain actions are "correlated" to some specific scene/location, e.g., the action 'cooking' is performed in kitchen, while the action 'brushing teeth' is performed in bathroom, etc. The eliminated STIPs, which are mainly from the background, could contain some helpful information for action encoding. Eliminating the interest points from background will "lose" the scene or context information, thus the refinement may have some negative impact on action analysis; (2) The overall accuracies on MSRAction3D and MSRActivity3D datasets increase after applying the STIP refinement. On MSRAction3D dataset, the refined accuracy is 80.5%, comparing to the original accuracy 78.7%, on MSRActivity3D dataset, the best accuracy is 77.5%, which is higher than the original 70.6%, after the refinement.

6.6.3.2 STIP feature refinement using RGB images and Skeleton Joints

We have shown above that in most cases the STIP refinement with the 20 skeleton joint positions can increase the action recognition rates. However, the performance is still highly relied on the interest point detection accuracy. When the interest point detection performs poorly on the depth maps because of the noisy depth data, the skeleton constraints may not help too much. Based on this consideration, we pursue another refinement scheme. The idea is to adopt the interest point detection on RGB videos, i.e., using the STIP locations detected in RGB videos for depth sequences. In other words, the interest point detection is conducted on RGB sequences, and just duplicated to the depth maps. The feature descriptors are still executed on the depth videos.

Experiments are conducted on three datasets except the MSRAction3D because it does not have the RGB data. We use the same settings as previous. The evaluation results are shown in Table 6.8. The best STIP feature on each dataset are selected (because separate implementation of ESURF descriptor is not available, we chose the 2nd best STIP feature instead). From the results, one can see that the accuracies are improved significantly after using RGB refinement approach, either the skeleton refinement is applied or not. On MSRDailyActivity3D dataset, the accuracy is increased from 70.6% to 75.6%, on CAD-60 dataset, the accuracy is improved from 56.3% to 68.8%, and on the UTKinect-Action dataset, the accuracy is improved from 81.0% to 85.0%, when using the RGB refinement approach.

For the refinement with skeleton joints, the accuracies can be improved or keep the same on the MSRDailyActivity3D and UTKinect-Action datasets, but reduced on the CAD-60 dataset. The reason could be that the interest points located in the background or scene may help to improve the action recognition accuracies (the CAD-60 dataset contains different actions in different scenes), while the removal of those interest points (constrained by the skeleton joints) can reduce the recognition performance.

The refinement results show that it may not be accurate enough to use the detected locations of interest points on depth sequences directly, because of the noisy depth values.

6.7 Fusing spatiotemporal features and skeleton joints for action recognition

In the above, two approaches have been presented to refine the STIP features. These approaches can be viewed as posing constraints to the interest point locations on depth videos, by using either RGB videos or the skeleton joints. On the other hand, the skeleton joints positions extracted from the depth videos can be used as another feature, representing human posture information. In this section we want to evaluate the performance of combining the STIP features with the skeleton joints feature. This evaluation can tell if the STIP features can complement the skeleton joints features, and if the combination can improve the accuracies significantly. If the accuracies can be improved greatly, it can indicate the usefulness of the STIP features from another aspect.

Specifically, the combination approach has four major steps, which has been presented in a workshop [132]. Firstly, the STIP features are extracted on depth sequences. Then skeleton joints features are computed from the skeleton joint positions. A quantization is performed for the two features respectively to encode the action sequences with histograms. Finally, a feature-level fusion is executed for action recognition using the random forests method [134]. We chose the detector/descriptor combinations which performs the best based on our evaluation presented above. The evaluation of the STIP features in Section 6.6 is the basis for our fusion approach [132].

We use the histogram of the skeleton joints features proposed in [121] to combine with the best STIP features on each database. Different from [121] where the Naive Bayes classifier was used, we compute the histogram of the joints to combine with the STIP features by the random forests method.

The features from joint locations consist of three parts: (1) current posture: pair-wise joint distances in current posture; (2) motion: joints difference between current posture and the original (in the first frame); and (3) offset: joints differences between current posture and the previous one. A concatenation of the three feature vectors is taken to represent the feature. The PCA method is applied for dimensionality reduction.

To represent each action sequence, we quantize the STIP features and the skeleton joints features, respectively, based on the K-means clustering. The cluster centers are used as the keywords to construct the histogram bins. These features are used in the next step for feature-level fusion and action classification.

In order to perform the fusion and feature selection of spatiotemporal features and the skeleton joints features, the random forests (RFs) method [134] is used. RFs are usually considered as a classifier using tree predictors in which each tree splits the data

TABLE 6.9: Accuracies of the fusion method compared to each single feature on four datasets. RFs denotes the random forests method.

MSRAction3D	**Acc.**
STIP (Harris3D+HOG/HOF)	77.5%
Skeleton Joint Features	90.9%
Combined features with RFs	**94.3%**
UTKinect-Action	**Acc.**
STIP (Harris3D+HOG3D)	80.8%
Skeleton Joint Features	87.9%
Combined features with RFs	**91.9%**
CAD-60	**Acc.**
STIP (Hessian+ESURF)	75.0%
Skeleton Joint Features	81.3%
STIP + Skeleton	**87.5%**
MSRDailyActivity3D	**Acc.**
STIP (Hessian+HOGHOF)	70.6%
Skeleton Joint Features	73.8%
Combined features with RFs	**80.0%**

depending on the randomly selected features. And there are many nice properties to use the random forests: (1) robustness to noise, (2) efficiency for classification, and (3) the improvement of accuracy by growing multiple trees and vote for the most popular class. Here we use the RFs for fusion of distinct features and action classification together.

The experiments are conducted on the four datasets (MSRAction3D, UTKinect-Action, CAD-60, and MSRDailyActivity3D) while three of them were used in our study in [132]. Our fusion approach can improve the recognition rates to 94.3%, 91.9%, 87.5%, and 80.0%, respectively, on the four databases, which are significantly higher than the STIP feature or skeleton. This result shows that the STIP features can be useful to complement the often-used skeleton features for action recognition.

We also compare the fusion results to other results reported in the literature on the four datasets. Table 6.10 shows all reported results that we can find on the MSRAction3D dataset. Under the same experimental settings, it can be seen that the fusion result of 94.3% accuracy is the second best result among all of the previous methods. Our result is only 0.5% lower than the best result in [135]. On the UTKinect-Action dataset from Table 6.11, the fusion approach has an accuracy of 91.9% which is higher than the DSTIP+DCSF feature [2], and slight higher than the HOJ3D feature in [119] (90.9%) and the space-time pose representation in [133] (91.5%). Note that we used the same settings as [2], which is more challenge than the settings in [119] and [133]. On the CAD-60 dataset, same experimental settings are kept the same as [124] and the presicion/recall of our fusion method is computed for a direct comparison with other

TABLE 6.10: Comparisons of different methods on MSRAction3D dataset.

Method	**Accuracy**
High Dimensional Convolutional Network [125]	72.5%
Action Graph on Bag of 3D Points [23]	74.7%
HOJ3D feature [119]	79.0%
Key Pose Learning [122]	80.3%
Eigenjoints [121]	82.3%
STOP feature [120]	84.8%
Random Occupancy Patterns [125]	86.2%
Actionlet [84]	88.2%
HON4D [126]	88.9%
DSTIP+DCSF [2]	89.3%
Depth Motion Maps [123]	91.6%
Space-time Pose Representation [133]	92.8%
JAS (Cosine)+MaxMin+HOG2 [135]	94.8%
STIP + Skeleton	94.3%

TABLE 6.11: Comparisons of different methods on UTKinect-Action dataset.

Method	**Accuracy**
DSTIP+DCSF [2]	85.8%
HOJ3D [119]	90.9%
space-time pose representation [133]	91.5%
STIP+Skeleton	**91.9%**

TABLE 6.12: Comparisons of different methods on CAD-60 dataset.

Method	**Precision/Recall**
J. Sung *et al.* [124]	67.9%/55.5%
X. Yang *et al.* [121]	71.9%/66.6%
Koppula *et al.* [127]	80.8%/71.4%
STIP + Skeleton	**93.2%/84.6%**

methods, shown in Table 6.12. Our fusion approach obtained a much higher accuracy than the state-of-the-art results on this dataset. Finally, Table 6.13 shows the results on the MSRDailyActivity3D dataset, an accuracy 80.0% is obtained using our fusion approach. Slight different settings are used in our experiment, the result is comparable but about 8% lower than the highest accuracy. Note that all the 16 activities are used in our experiment, while in [2], four activities (with less motion) were removed in their experiment.

From the comparison with various methods in the literature, we demonstrate the usefulness of the STIP features for depth-based action recognition, when combined with the skeleton feature.

TABLE 6.13: Comparisons of different methods on MSRDailyActivity3D dataset.

Method	**Accuracy**
NBNN + parts + time [136]	70.0%
Local HON4D [126]	80.0%
DCSF [2]	83.6%
RGGP + Fusion [137]	85.6%
Actionlet [84]	85.8%
DCSF+Joint [2]	88.2%
STIP+Skeleton	80.0%

6.8 Conclusions

We have presented a comprehensive evaluation of the spatiotemporal interest point features for action recognition in 3D. The evaluated STIP features include three spatiotemporal interest point detectors and six descriptors. The combinations of these detectors and descriptors form 14 different features. These STIP features have been evaluated on four different depth action/activity databases. The comparisons to the state-of-the-art methods have shown that the STIP features are still useful for depth-based action recognition.

From the evaluation, we have shown that most of the results are comparable to the current state-of-the-art approaches. However, under the bag-of-words framework, the extracted features do not contain the spatial distribution of the interest points in depth maps, this is one reason that limits the performance. We have also shown that the noisy depth data and background have a great impact on interest point detection. Moreover, the interest point detection may not perform well on actions without much motion, resulting in lower accuracies.

The evaluation has shown that different STIP features perform quite differently on depth actions. It discovers that the feature with Harris3D and HOG/HOF performs the best on the MSRAction3D dataset, the Cuboids detector with HOG/HOF descriptor performs the best on the MSRDailyActivity3D dataset, while the Harris3D detector combined with HOG3D descriptor is the best on UTKinect-Action dataset. On the CAD-60 dataset, the Hessian detector with ESURF descriptor gives the highest accuracy.

Two interest points refinement schemes have been presented for the STIP features, based on constraining the STIP features using skeleton joint positions and/or the detection in RGB videos. We have shown that the STIP features can be refined to achieve better performance in most cases. We have also proposed a fusion scheme to combine the best

STIP features with the skeleton joint features in each database. Significant improvements of the recognition accuracies have been achieved on all four databases. Overall, we have explored the STIP features for 3D action recognition from different aspects.

Chapter 7

Fusing Multiple Features for Depth-based Action Recognition

7.1 Abstract

Human action recognition is a very active research area in computer vision and pattern recognition with important applications. Recently, human action recognition using the depth data captured by the emerging RGB-D sensors has shown a great potential in action analysis, compared to the traditional color video-based approaches. Several features and/or algorithms have been proposed for depth-based action recognition. Some questions are raised: Can we find some complementary features for depth-based action analysis? Can we fuse these complementary features to improve the recognition accuracy significantly? To address these questions and have a better understanding of depth videos for action recognition, we advocate the study of fusing different features for depth-based action analysis. Although data fusion has shown great success in many areas, such as multimedia analysis and biometrics, it has not been well studied yet on whether the fusion is helpful or not for depth-based action recognition, or how to do the fusion properly. In this paper, we study different fusion schemes comprehensively, using diverse features for action characterization in depth videos. Two different levels of fusion schemes are investigated, i.e., feature-level and decision-level. Various methods are explored at each fusion level. Four different features are considered to characterize the depth action patterns from different aspects. The experiments are conducted on four challenging depth action databases, in order to evaluate and find the best fusion methods generally. Our experimental results show that the four different features investigated in the paper can complement each other, and appropriate fusion methods can improve the recognition accuracies significantly over each individual feature. More importantly,

our fusion-based action recognition outperforms the state-of-the-art approaches on these challenging databases.

7.2 Introduction

Human action recognition has been an active research topic for more than two decades. It has a wide range of applications in the real world, such as Human Computer Interaction (HCI), video surveillance, video retrieval and security [7]. Most of the work has focused on action recognition using the videos captured in the visible spectrum [9, 10, 46]. Very recently, with the emerging, low-cost RGB-D sensors, e.g., the Kinect, human action recognition in 3D data has gained great attentions in computer vision. Depth maps provide many advantages over traditional color images/videos. For example, firstly the depth maps provide the 3D structure and shape information which makes several problems easier to deal with, such as segmentation, detection, etc. Second, depth images/videos are insensitive to illumination changes. Third, a quite accurate estimation of 3D human skeleton joint positions can be obtained from the depth data [13]. Therefore, using the Kinect sensor, three channels (RGB, depth and skeleton joint positions) of data are provided, which not only bring great benefits for robotics and human centered computing, but also give a broader scope for action recognition as well [18].

7.2.1 Related Work on Depth-based Action Recognition

Depth-based action recognition has been actively studied since 2010 [23]. Several algorithms and/or features have been proposed in the literature.

There are several representative works for action recognition with 3D depth data. Li *et al.* [23] proposed an action graph for depth action recognition. A bag of 3D points sampled on depth data is used to encode the action posture, and the action graph is used to model the dynamics of actions. Wang *et al.* [84] proposed to combine the skeleton feature and local occupation feature, then learned an actionlets ensemble model to represent actions. A multiple kernel learning method is used to combine the actionlets. Wang *et al.* also proposed a semi-local feature called Random Occupancy Patterns (ROP), which is extracted from 4D volumes [125]. Sparse coding is utilized to encode the features and the SVM is used for classification. Vieira *et al.* [120] proposed the space-time occupancy patterns to represent depth sequences. Both space and time axes are divided into multiple segments. Occupancy feature is computed in each cell, and a nearest neighbor classifier is applied for recognition. Different approaches based on Motion History Images (MHI) are proposed by Yang *et al.* [121] [138] [139]. The main

idea is to use accumulated depth maps and compute histogram of gradients (HOG) features to represent human actions. More recently, Oreifej and Liu [126] proposed a method called HON4D to describe the depth sequence as a histogram captured in the 4D space of time, depth and spatial coordinates. A 600-cell polychoron is used to quantize and represent the features. They used the SVM classifier and showed a good performance for action recognition. In [2], a modified spatiotemporal feature based on Cuboids is proposed to capture the action motion and eliminate the flip noise on the depth video. A feature selection scheme is applied to the proposed features and then the selected features are fed into the SVMs for classification.

On the other hand, by modeling the skeleton joints, human actions in a depth video can be represented by the sequence of human postures and can be fed into the learning based algorithms, such as Hidden Markov Models (HMMs) and Dynamic Time Warping (DTW). In [140] and [141], similar ideas were proposed where feature vectors defined by skeleton joints are used to modal the human action, and DTW is applied to the resulted feature vector for action recognition. Yang et al. [138] proposed another skeleton feature to modal the human action posture frame by frame, based on computing posture feature, motion feature, and the offset feature, according to the relative frames in the action video. In [84], pairwise skeleton joint positions were computed in each frame to shape the motion of the human body. Xia *et al.* [119] proposed an alternative feature called HOJ3D based on the skeleton joints. A coordinate based on skeleton joints is constructed, and multiple 3D bins are used to extract histogram features, by counting the number of joints in each bin. A Hidden Markov Model is used for action classification. Similarly, Miranda *et al.* [122] used the pose descriptor in a torso-based coordinate system and the SVM classifier to learn key poses. A decision forest is then used to recognize the action classes.

There are also works using both the RGB videos and depth maps for action recognition [124], [128] and [129]. The histogram of oriented gradient (HOG) feature was used as the descriptor for both RGB and depth images in [124]. The hand positions, body pose and motion features were also extracted from skeleton joints. A two-layer maximum-entropy Markov model is trained for classification.

7.2.2 Related Work on Data Fusion

Data fusion has been studied extensively, and shown great performance in many areas, including multi-sensor system [142], multimedia analysis [143], human identification [144], face recognition [145], handwriting recognition [146], biometrics [147], etc. Various methods have been proposed and investigated for data fusion.

In [143], a survey of multimodal fusion for multimedia analysis was conducted. A categorization of different fusion methods with a thorough review of literature was given. They argued that the linear weighted fusion and SVM fusion methods are more often used because of the efficiency of these methods. Kittler [148] proposed the theoretical framework for combining different classifiers, and [149] investigated various rule-based fusion methods and experimentally validated the performance. Later on, a more extensive study was conducted in [150] on classifier fusion strategies. In [151], a decision template is proposed to represent different fusion methods. In [147], different levels of fusion methods were presented for biometrics applications.

The usefulness of data fusion in other areas motivated us to explore fusion-based approaches for depth-based action recognition, which has not been well studied yet, to the best of our knowledge. In this paper, fusion methods are explored on two levels: Feature-level and Decision-level. For the Feature-level, Random Forests, Joint Mutual Information and Conditional Mutual Info Maximization approaches are studied. For the Decision-level fusion, the Majority Voting, Naive-Bayes Combination, Rule-based fusion, SVM-Based fusion and Multi-Agent System approaches are studied. These approaches are further described in Section 7.4.

7.2.3 Our Approach

We study whether and how fusion-based approaches can help to improve the action recognition accuracies in depth videos. The underlying assumption is that there are complementary features that can be extracted in depth videos. For the purpose of fusion, the complementary features should be extracted and combined together appropriately, otherwise the overall accuracy might not be improved even with multiple features. In our preliminary work [132], the spatiotemporal features and skeleton features were combined using the random forest method [134]. This fusion approach improves the accuracies of depth-based action recognition significantly. In this paper, we will further explore our fusion-based idea, by combining more features with diversity, investigating and evaluating a variety of fusion methods comprehensively, and using more databases to validate the fusion methods for generality.

Several representative data fusion methods are explored for our problem. We compare different fusion methods and find the best ones to solve the specific problem of depth-based action recognition.

The major contributions of our work include: (1) Evaluation of different features on depth-based action recognition, using the same experimental setting. There are four different features chosen for the evaluation. Two of them capture local motions and

were originally proposed for action recognition in RGB videos, the third one extracts features according to the skeleton joints positions, and the last one extracts features from 3D surface normal distributions; (2) Exploration of two levels of fusion schemes, i.e., the feature-level and decision-level fusions. Several methods are explored at each fusion level; (3) Validation of the capability of different fusion methods and finding the appropriate for depth-based action recognition on four challenging databases.

The remaining of the paper is organized as follows: In Section 7.3, we introduce four different features for depth-based action characterization. Conceptually these features represent the action patterns from different aspects. In Section 7.4, we describe different fusion methods belonging to two different fusion levels. The experiments are conducted in Section 7.5 with comparisons to the state-of-the-art methods. Finally, we draw conclusions.

7.3 Feature Extraction and Description on Depth Data

Feature extraction and representation is an important step for action recognition. To develop our fusion-based approach to depth-based action recognition, we use multiple, diverse representations to characterize the action patterns. In our preliminary work [132], the spatiotemporal interest point features (STIP) and skeleton features are extracted and fused for action recognition in 3D. Here we expand the preliminary work by integrating more features, and executing a systematic study of various fusion methods. The 4D descriptor (HON4D) can characterize the normal distributions of the 3D surfaces in action performing [126], and the space-time auto correlation (STACOG) feature can represent more details of human actions [152]. Totally there are four different features to investigate for our developing of fusion-based recognition framework, characterizing the depth action patterns from different aspects. We introduce these features in the following.

7.3.1 Spatiotemporal Interest Point Features

Spatiotemporal interest point (STIP) features capture the complex motion of human actions. These features are quite popular for action representation in color videos [131], but not often in depth data. Here we adapt some STIP features to depth sequences. We attempted several combinations of the detectors and descriptors and find the best ones for depth action characterization. Because of the space limits, we briefly describe the STIP features that we used and only the best one will be reported in each dataset experimentally (see Section 7.5).

The Harris3D detector [37] computes the locations of the interest points based on a second-moment matrix of gradients with the convolution of spatiotemporal gaussian kernel in the video sequence. It locates the spatiotemporal volumes where large variations of gradient exist along space and temporal directions. Specifically, a spatiotemporal second-moment matrix is computed from a video sequence f,

$$\mu = g(\cdot) \times \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}, \tag{7.1}$$

where $g(\cdot)$ is a Gaussian weight function and L is the convolution of f with the spatiotemporal gradient. The interest point locations are determined by computing the local maxima of the response function $R = det(\mu) - k \cdot trace^3(\mu)$.

Another detector applied in this paper is the Hessian detector [35]. The Hessian detector uses the response function $S = |det(H)|$ to measure the strength of each interest point, where H is the Hessian matrix.

Given the detected locations, various descriptors can be used to characterize the local motion patterns. The *HOG/HOF* descriptor was proposed in [85] to describe local human motion in RGB videos. It computes the histogram of gradient (HOG) and histogram of optical flow (HOF) in each local volume. Klaser *et al.* [36] extends the HOG to HOG3D descriptor, which computes the 3D gradient and constructs a histogram as the feature vector. It computes the histogram of 3D gradient orientations. The integral videos can be pre-computed to efficiently compute the gradients and combine both shape and motion information at the same time. The *extended SURF (ESURF)* descriptor [35] is an extension of the SURF [38] for action representation.

7.3.2 Space-Time Auto-Correlation of Gradients (STACOG)

A method called the Space-Time Auto-Correlation of Gradients (STACOG) was proposed with the bag-of-frame-features computation in [152] to extract motion features from RGB action videos. It is computed with the frame-based STACOG features sampled densely along the time axis. In order to extract the feature, space-time gradient vector is calculated by taking derivatives (I_x, I_y, I_t) at each local space-time volume, around each space-time point. The gradients can be represented by the angles $\theta = \arctan(I_x, I_y)$ and $\phi = \arcsin(I_t/m)$, where $m = \sqrt{I_x^2 + I_y^2 + I_t^2}$ is the magnitude. Then a histogram is constructed by binning the gradients in a unit sphere. The histogram is defined as space-time orientation coding (STOC) vector. The auto-correlation functions can be

computed for the space-time gradients:

$$F_0 = \Sigma_r m(r)h(r), \tag{7.2}$$

$$F_1(a_1) = \Sigma_r min[m(r), m(r+a_1)]h(r)h(r+a_1)^T, \tag{7.3}$$

where r is the reference point (x, y, t), h is the STOC vector, and a_1 is the displacement vector from the reference point, and F_0 and F_1 are the zero and first order auto-correlations. We adapt the STACOG features from RGB to depth data.

7.3.3 EigenJoints Feature

Human skeleton joints can be detected fast on depth data [13]. The skeleton joint positions can be viewed as an alternative modality for action characterization. Features can be computed from skeleton joint positions to represent the action patterns, which are usually not available in color videos. Several features extracted from skeleton joints are proposed for depth-based action recognition such as [84], [119], [138]. We implemented these features and found that the method in [121] gives a better representation. Thus we chose the histogram of the skeleton joints features to represent human actions.

Specifically, the features consist of three parts: (1) current posture: pair-wise joint distances in current posture compared in the current frame; (2) motion: joints differences between current posture and the previous one; and (3) offset: joint differences between current posture and the original (in the first frame). Denote each 3D skeleton joint by $p_i = (x_i(t),\ y_i(t),\ z_i(t))$ at frame t. The number of skeleton joints in each frame is denoted as N. The feature vector can be computed by:

$$f = [f_{current}\ f_{motion}\ f_{offset}], \tag{7.4}$$

$$f_{current} = \{p_i - p_j \mid i \neq j;\ i, j = 1..N\}, \tag{7.5}$$

$$f_{motion} = \{p_i(t) - p_i(t-1) \mid i = 1..N\}, \tag{7.6}$$

$$f_{offset} = \{p_i(t) - p_i(0) \mid i = 1..N\}, \tag{7.7}$$

where $p(0)$ denotes the original posture in each action sequence.

7.3.4 Histogram of Oriented 4D Normals (HON4D)

The depth data can be represented as a surface in 4D space with a set of points (x, y, t, z), where z is the depth value of the point. Then the normal to the surface can be computed

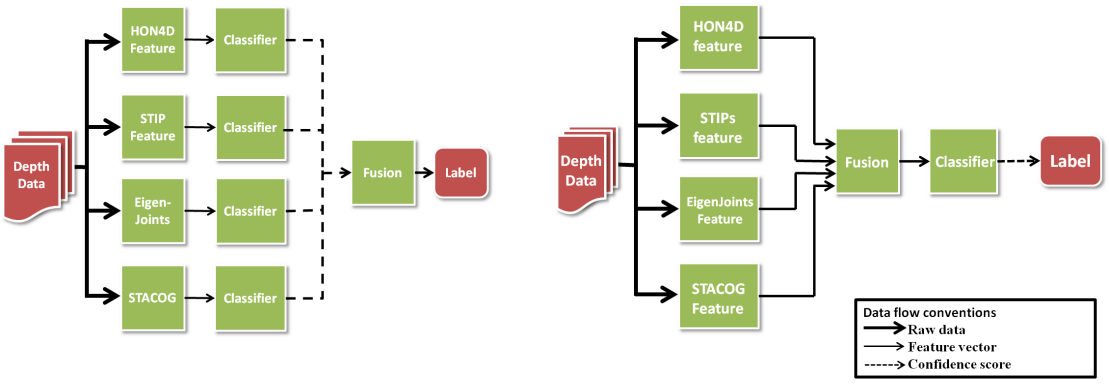

FIGURE 7.1: Illustrate the schemes of the decision-level/late fusion (left) and feature-level/early fusion (right) in combining different features for 3D action recognition.

by:

$$n = \nabla S = (\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t}, -1)^T. \tag{7.8}$$

The surface normals over all voxels in the depth sequence can be used for action representation [126]. A 600-cell polychoron in 4D space is used to quantize the 4D normals to derive the feature. The HON4D will be combined with other features together to develop our fusion-based approach.

7.4 Fusion Methods

Data fusion has gained much attention in recent years. It can be accomplished at different levels [153], e.g., early fusion (sensor, feature levels), where fusion is conducted before matching, and late fusion (rank, score, decision levels), where fusion is executed after matching. We present several fusion methods at both the decision and feature levels to solve our problem of depth-based action recognition (see Fig. 7.1 for an illustration).

7.4.1 Feature-Level Fusion

According to [147], feature level fusion is usually conducted through feature normalization and feature selection or transformation, because of the relationship between different feature sets and the curse of dimensionality [154]. The objective of feature-level fusion is to combine different feature sets to generate a new feature vector. For feature selection, we adopt two representative approaches from [155]. Totally we explore three methods for feature-level fusion to deal with the problem of depth-based action recognition.

7.4.1.1 Random Forests (RFs)

Random Forests [134] are usually considered as a classifier (or regressor) using tree predictors in which each tree splits the data depends on the randomly selected features. Random Forests can be used as a fusion method which is based on randomness of the split in each node and the forest structure. There are many nice properties of the random forests method: (1) robustness to noise, (2) efficiency for classification, and (3) the improvement of accuracy by growing multiple trees and voting for the most possible class. Here we use the RFs for fusion of distinct features and action classification jointly.

Let the feature vector be $v \in \mathbb{R}^N$, where the number of the features for each sample is N. A number $n < N$ is specified at each node of the tree, where n features are randomly selected to determine the split of that node. The randomly selected n features are used in the tree node.

The best split is determined by the information gain using these features,Several decision trees are growing to generate a forest, and each tree grows until it reaches the maximum tree depth max_{dep}, or the tree node receives the given number of minimum samples min_{node}. In the leaf nodes, the probabilistic distribution for each class is computed. In this way, the feature fusion is executed randomly and naturally in the tree building process.

In recognition phase, each new observation x goes down to one of the leaf nodes in each tree, denoted as $l(t, x)$, which contains the distribution P_n of all classes. Random forests classifier chooses the class label which gets the most votes over all the trees:

$$\hat{c} = \arg\max_j \frac{1}{T} \sum_{t=1}^{T} p^j_{l(t,x)}, \tag{7.9}$$

where $\hat{c}$ is the predicted class label, T is the total number of trees, $l(t, x)$ is the leaf node of tree t where the test sample x falling into. $p^j_{l(t,x)}$ is the posterior probabilities for class j at leaf node $l(t, x)$, $p^j_n = \frac{|S_j|}{|S|}$, where $|S|$ is the total number of samples in this leaf node and $|S_j|$ is the number of samples of class j in S.

7.4.1.2 Joint Mutual Information (JMI)

JMI was proposed to select a discriminative feature subset from the feature pool [156]. In our case, different features are normalized and concatenated to construct the feature pool. We investigate if the feature selection by JMI can fuse different features for our action recognition in 3D.

The mutual information between X and Y can be defined [156] by,

$$I(X;Y) = H(X) - H(X|Y) = \Sigma_{x\in X}\Sigma_{y\in Y} p(xy) \log \frac{p(xy)}{p(x)p(y)}, \tag{7.10}$$

where $H(X)$ denotes the entropy of the random variable:

$$H(X) = -\Sigma_{x\in X} p(x) \log p(x), \tag{7.11}$$

and the conditioned form of entropy $H(X|Y)$ can be written as:

$$H(X|Y) = -\Sigma_{y\in Y} p(y) \Sigma_{x\in X} p(x \mid y) \log p(x \mid y). \tag{7.12}$$

As a feature selection method, the JMI can be viewed as using a criterion J to measure how useful a feature or feature subset is when used by a classifier. This criterion is defined as:

$$J_{jmi}(\mathbf{v}_k) = \Sigma_{\mathbf{v}_k\in S} I(\mathbf{v}_k \mathbf{v}_j; Y), \tag{7.13}$$

where S is the previously selected feature set, Y is class label, and $\mathbf{v}_k$ is the kth feature in the feature vector $\mathbf{v}$.

The JMI pairs the candidate features in $\mathbf{v}_k$ with each newly selected feature to increase the complementary information between features [155].

7.4.1.3 Conditional Mutual Info Maximization (CMIM)

CMIM is an alternative feature selection method [157]. Different from JMI, the CMIM method adds a new feature only if the optimal value based on a criterion is larger than using the features already selected, such that the information has not been brought by any already selected features. This criterion is given by

$$J_{cmim}(\mathbf{v}_k) = min_{\mathbf{v}_k\in S} \left[I(\mathbf{v}_k; Y \mid \mathbf{v}_j) \right], \tag{7.14}$$

which can be equally written as:

$$J_{cmim}(\mathbf{v}_k) = I(\mathbf{v}_k; Y) - max_{\mathbf{v}_k\in S} \left[I(\mathbf{v}_k; \mathbf{v}_j) - I(\mathbf{v}_k; \mathbf{v}_j \mid Y) \right], \tag{7.15}$$

where S is the previously selected feature set, Y is class label, and $\mathbf{v}_k$ is the k-th feature in the feature vector $\mathbf{v}$.

Using the feature selection procedures, different feature sets can be fused together to feed into a classifier for action recognition.

7.4.2 Decision-Level fusion

Different from feature-level fusion, the decision-level fusion or late fusion deals with the fusion process on the decision level, where classifier outputs are combined to make the final decision.

Let $\mathbf{v} \in \Re^n$ be a feature vector extracted from an input pattern, and let $\{\omega_1,\ \omega_2,\ ...,\ \omega_c\}$ be the class labels of c classes. For a *classifier* D, the output of D given the input pattern $\mathbf{v}$ can have two representations: $D(\mathbf{v}) = [d_1(\mathbf{v}),\ d_2(\mathbf{v}),\ ...d_c(\mathbf{v})]$, where $d_i(\mathbf{v}) \in [0, 1],\ i = 1..c$, is an estimate of the posterior probability $P(\omega_i|\mathbf{v})$ offered by classifier D. The other is $D(\mathbf{v}) = \omega_i, i \in \{1..c\}$, where ω_i is the class label given by classifier D.

When there exist totally L classifiers, denoted by $\{D_1, \cdots, D_L\}$, the representations for multiclassifiers can be given [151] by

(1) Decision Profile:

$$DP(\mathbf{v}) = \begin{bmatrix} d_{1,1}(\mathbf{v}) & \cdots & d_{1,j}(\mathbf{v}) & \cdots & d_{1,c}(\mathbf{v}) \\ \cdots & & \cdots & & \cdots \\ d_{i,1}(\mathbf{v}) & \cdots & d_{i,j}(\mathbf{v}) & \cdots & d_{i,c}(\mathbf{v}) \\ \cdots & & \cdots & & \cdots \\ d_{L,1}(\mathbf{v}) & \cdots & d_{L,j}(\mathbf{v}) & \cdots & d_{L,c}(\mathbf{v}) \end{bmatrix}, \tag{7.16}$$

where $d_{i,j}(\mathbf{v})$ denotes the estimate of posterior probability of class j made by classifier D_i, $i \in [1,\ L]$, $j \in [1,\ c]$.

(2) Decision Vector:

$$DV(\mathbf{v}) = \begin{bmatrix} \omega_1^c & \cdots & \omega_i^c & \cdots & \omega_L^c \end{bmatrix}, \tag{7.17}$$

where ω_i^c is the class label given by classifier D_i. Given above representations, we will present specific fusion methods for decision-level fusion as follows.

Popular methods for decision-level fusion include (weighted) majority voting, Naive-Bayes Combination, weighted Sum, Minimum, Maximum, Median, Product, SVM-based fusion and Multi-agent system [151] [147] [148] [143] [143] [158].

7.4.2.1 Majority Voting

Majority voting is one of the most common approaches for decision-level fusion [148]. The idea is to assign the final class label by "voting" over the different classifiers, and select the one that the majority classifiers agree on. For each *classifier* D_i *in* L *classifiers,* the output of D_i given an input pattern $\mathbf{v}$ is a predicted class label ω_i^c, and the final

class label is assigned according to which class label is the majority in the decision vector $DV(\mathbf{v}) = \begin{bmatrix} \omega_1^c & \cdots & \omega_i^c & \cdots & \omega_L^c \end{bmatrix}$. If more than one label occurs, the class label will be randomly selected from those labels.

It is reasonable to assign different weights to the decisions made by different classifiers, when the performance of these classifiers is quite different. Larger weights can be assigned to the decisions made by more accurate classifiers. So the discriminant function for class ω_k can be rewritten as:

$$g_k = \Sigma_{i=1}^{L} w_i s_i^k, \tag{7.18}$$

where $w_i$is the weight of classifier D_i, and $s_i^k$is an indicator function defined as:

$$s_i^k = \begin{cases} 1, & if\ the\ classifier\ D_i\ outputs\ class\ label\ \omega_k \\ 0, & otherwise \end{cases} \tag{7.19}$$

The weight of each classifier can be determined by a training process in a supervised manner.

7.4.2.2 Naive-Bayes Combination

The Naive-Bayes fusion method relies on transforming decision labels into probabilities, under the assumption that different classifiers are mutually independent in the multi-classifier system [146]. The first step is to construct confusion matrix CM_i for each classifier D_i. Each element on the k-th row and s-th column denotes the number of patterns of the training data set of which the true label is ω_k but is assigned to class ω_s by D_i. The next step is to construct the Label Matrix LM_j for each classifier, where each element is defined by:

$$lm_{k,s}^i = \hat{P}(\omega_k \mid D_i(\mathbf{v}) = \omega_s) = \frac{cm_{k,s}^i}{cm_{\bullet,s}^i}, \tag{7.20}$$

where $cm_{k,s}^i$ denotes the element on the k-th row and s-th column of CM_i, $cm_{\bullet,s}^i$ denotes the sum of the s-th column of CM_i.

For each pattern $\mathbf{v}$, classifier D_j outputs a class label, the estimated probability of the class label ω_i is computed by:

$$\theta_i(\mathbf{v}) = \Pi_{j=1}^{L} P(i \mid D_j(\mathbf{v}) = s_j) = \Pi_{j=1}^{L} lm_{i,s_j}^j. \tag{7.21}$$

We found that the above multiplication in [146] cannot work well for our problem. Replacing it with summation can result in much better results:

$$\theta_i(\mathbf{v}) = \Sigma_{j=1}^{L} lm_{i,s_j}^{j}. \tag{7.22}$$

7.4.2.3 Sum, Minimum, Maximum, Median and Product Rules

These fusion methods can be categorized as rule-based methods [148] [150]. These basic rules are defined to combine multiple classifiers and can generally perform well if the quality of temporal alignment between different modalities is good [143].

Denote θ the predicted class label, $P(\omega_j \mid \mathbf{d}_i)$ the *posteriori* probability of θ assigned as class ω_j by the measurement vector $\mathbf{d}_i$ from the i-th classifier. We have

(i) Sum rule: Assign $\theta \rightarrow \omega_j$ if

$$(1-L)P(\omega_j) + \Sigma_{i=1}^{L} P(\omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \left[(1-L)P(\omega_j) + \Sigma_{i=1}^{L} P(\omega_j \mid \mathbf{d}_i)\right] \tag{7.23}$$

(ii) Maximum Rule: Assign $\theta \rightarrow \omega_j$ if

$$\max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) \tag{7.24}$$

(iii) Minimum Rule: Assign $\theta \rightarrow \omega_j$ if

$$\min_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} \min_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) \tag{7.25}$$

(iv) Product Rule: Assign $\theta \rightarrow \omega_j$ if

$$P^{-(L-1)}(\omega_j)\Pi_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} P^{-(L-1)}(\omega_j)\Pi_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) \tag{7.26}$$

(v) Median Rule: Assign $\theta \rightarrow \omega_j$ if

$$\max_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) = \max_{j=1}^{c} median_{i=1}^{L} P(\theta = \omega_j \mid \mathbf{d}_i) \tag{7.27}$$

7.4.2.4 SVM-Based Fusion

SVM-Based fusion is a decision-level fusion method which combines multiple individual SVM classifiers by a new SVM classifier using their confidence scores. Basically, given

the training samples $\mathbf{x}_i \in \Re^n$ and the class labels $y_i \in \{-1, 1\}$, the Support Vector Machine (SVM) [47] optimize the following problem:

$$\min_{W,b,} \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\Sigma_{i=1}^{l}\xi_i, \ s.t. \ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, ..., l. \tag{7.28}$$

The SVM-based fusion approach is based on a two-layer structure, where the input to the higher layer SVM is the confidence scores given by individual lower-layer SVM classifiers [143]. Each lower-layer SVM classifiers train one feature and output the confidence scores instead of the class labels. The confidence scores of all individual classifiers are then combined into a new feature vector, and then fed into the second-layer SVM for final classification.

7.4.2.5 Multi-Agent System

A multi-agent system (MAS) was proposed to solve the multiclassifier classification problem [158]. An auction-based negotiation is used for classification. The idea is that all agents/classifiers are considered to be the buyers who are trying to reach an agreement in relation to an input pattern. Specifically, the confidence scores of each agent/classifier D_i given an input pattern $\mathbf{v}$ is computed in the first step, denoted as: $D_i(\mathbf{v}) = [d_{i,1}(\mathbf{v}), \ d_{i,2}(\mathbf{v}), \ ...d_{i,c}(\mathbf{v})]$, and the class with maximum confidence of each agents is selected as the chosen class for that agent, the maximum confidence value of each agent is denoted as: $[d_{1,m_1}, d_{2,m_2} \cdots d_{L,m_L}]$. Given L agents, the cost for the agents are defined as a vector $\mathbf{c^j} = \{c_1^j, c_2^j, \cdots, c_L^j\}$. The cost for the i-th agent is computed by

$$c_i^j = \begin{cases} d_{j,m_j} - d_{j,i} & i \neq j \\ d_{j,i} - \Sigma_{k \neq i}^{c} d_{i,k} & i = j \end{cases}. \tag{7.29}$$

The agent with the highest cost $\arg_j \max_{j=1}^{L} c_j^j$ is considered the loser. Then the confidence of the chosen class of all agents are changed according to the difference between the current confidence and their responding cost: $d_{i,j} = d_{i,j} - c_j^j$. After the confidence values of each agent have been updated, the agent can decide whether or not to keep the chosen class, according to the current confidence values. When an agent loses twice in succession, it is then discarded from the negotiation. The remaining agents continue this process, until there is only one agent remains in the auction.

7.5 Experiments

In this section, we conduct experiments on four challenging depth-based action databases, using four different features and various fusion methods. First, we transformed the depth data into gray level depth videos and projected all the skeleton joint positions into image coordinates. After this preprocessing, feature extraction is conducted on each database. Every action sequence is represented by four different feature vectors, i.e., the STIP, STACOG, EigenJoints, and HON4D, respectively. For feature quantization, the K-means clustering method is used to derive histograms for each feature in each action video. Before investigating the comprehensive fusion-based framework, we analyze the performance of the individual feature for action recognition. Note that the same training and test sets are used for both individual features and various fusion methods. The SVM is used as the supervised classifier for each individual feature vector. After evaluating the individual features, various fusion methods are investigated. On feature-level fusion, different feature vectors are normalized first before fusion. Random forests can both select features and execute the classification task. For other feature-level fusion methods, the SVM is used as the classifier. On decision-level fusion, the SVM classifiers are trained for each feature independently, from which multiple decisions are made for each test pattern. The confidence scores or the intermediate decision class labels are transmitted into the fusion engine for fusion with different methods.

In the following, we introduce the four databases first, followed by some experimental settings, and then the experimental results. We also provide some analysis and discussions about the experimental results.

7.5.1 Databases

Four challenging 3D action databases are used in our experiments to evaluate the performance of different fusion approaches. In brief, these four databases capture various human actions/activities under different circumstances (viewpoints, locations, and backgrounds, etc.) with different considerations (# of actions, # of human subjects, or different scenarios, etc.) and complexity. And also, the performed actions are quite different in these databases. More details are given below.

MSRAction3D dataset [23] captures 20 human actions using a depth camera similar to the Kinect sensor. In total 10 subjects were asked to perform 20 action classes 3 times each. Each video clip is of resolution 640×480 at 15fps. We used all of the 557 video clips, along with the skeleton joint locations provided by [23]. In our experiment, we follow the same settings of "cross-subjects" as in [23]. The whole dataset was divided

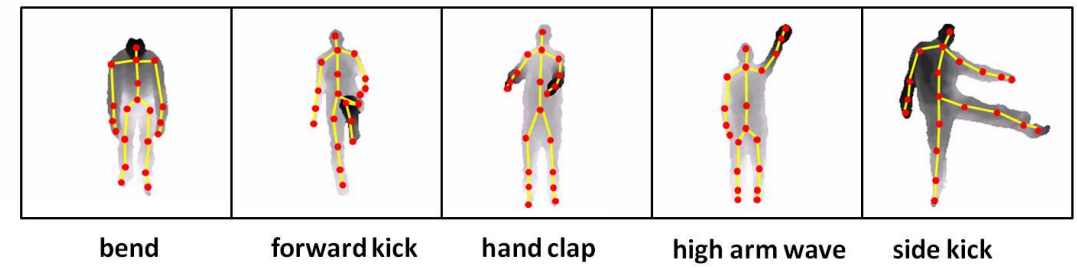

FIGURE 7.2: Some examples (with skeleton joints shown) in the MSRAction-3D dataset.

FIGURE 7.3: Some examples (with skeleton joints shown) from the MSRActivity3D dataset. The actions (from left to right) are: cheer up, drink, stand up, play guitar, and walk.

into 3 subsets, half of the subjects are used for training while the other half of subjects are used for testing. The final accuracy on this dataset is the average of the accuracies over the three subsets. See Fig. 7.2 for some example images in this dataset.

MSRDailyActivity3D dataset [84] was collected with human daily activities by the Kinect. In total there are 16 activities in this dataset: *drink, eat, read book, call cellphone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, sit down.* Each subject performed an activity in two scenarios, one "sitting on sofa" and the other "standing". The number of activity videos is 320. Three types of data, i.e., the RGB, depth and skeleton joint positions are provided in this dataset. The specific subject IDs which are used in training and testing are listed in Table 7.1. See Fig. 7.3 for examples of depth images in this dataset.

UTKinect-Action dataset [119] contains 10 different action classes performed by 10 subjects, collected by a stationary Kinect sensor. The 10 action classes are: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands.* Depth sequences are provided with resolution 320×240, and skeleton joint locations are also provided in this dataset. In our experiments, we used the cross-subjects scheme with half of the subjects for training while the remaining for testing (See Table 7.1), which is different from the leave-one-out scheme used in [119] where more subjects were used for training. Some example images of this dataset are shown in Fig. 7.4.

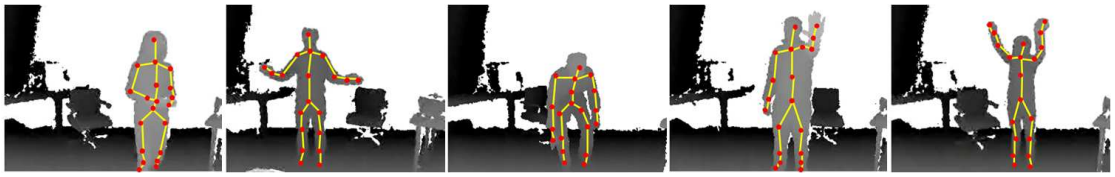

FIGURE 7.4: Some example images (with skeleton joints shown) in the UTKinect-Action dataset. The actions (from left to right) are: carry, clap hands, pickup, push, and wave.

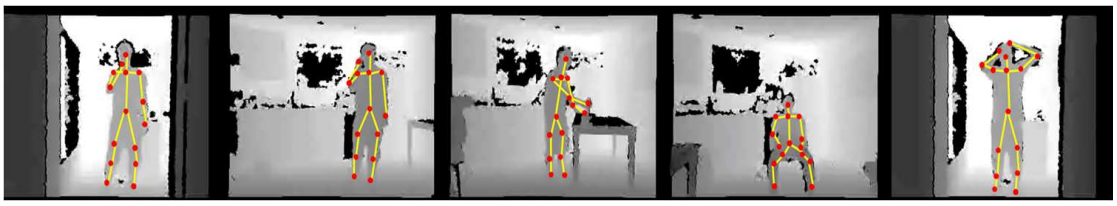

FIGURE 7.5: Some example images (with skeleton joints shown) from the CAD-60 dataset. The actions (from left to right) are: brush teeth, talk on phone, cook, relax on couch, and wear contact lens.

TABLE 7.1: Subject IDs which are used for training and testing in each database.

Dataset	**Training Subject IDs**	**Testing Subject IDs**
MSRAction3D	2, 3, 5, 7, 9	1, 4, 6, 8, 10
MSRDailyActivity3D	1, 6, 8, 9, 10	2, 3, 4, 5, 7
UTKinect-Action3D	1, 2, 3, 4, 8	5, 6, 7, 9, 10
CAD-60	1, 3, 4	2

Cornell Activity Dataset-60 (CAD-60) [124] has 60 RGB-D sequences collected by the Kinect , each video is of length about 45s. In this dataset, four different subjects performed 12 different activities in five locations. The five locations are: *office, kitchen, bedroom, bathroom and living room.* To reduce the computational complexity, we first sub-sample each video to the length about 500 frames. Then we follow the same procedure of "new person" as in [124] for training and testing (See Table 7.1). See Fig. 7.5 for some example images of this dataset.

7.5.2 Experimental Settings

We follow the same experimental settings as our conference paper [132]. Specifically, for the STIP feature extraction, Harris3D detector with HOG/HOF descriptor are used for MSRAction3D dataset. Harris3D detector and HOG3D descriptor are used for UTKinect-Action dataset. On CAD-60 dataset, Hessian detector and ESURF descriptor are adopted. On MSRDailyActivity3D dataset, Hessian detector and HOG3D descriptor are used to extract local features. These are the best STIP features in each dataset,

based on a systematic evaluation. Because of the space limit, we do not present the detailed evaluations here. The K-means clustering method is applied to quantize the STIP features into histograms. Empirically we set K = 100 to get the clusters or keywords. For skeleton joints feature, the bag-of-words scheme is used for quantization. In order to get the STACOG feature, we follow the settings in [152], adopting a hemisphere for coding the gradients. Four orientation bins along the longitude are arranged on each of five layers along the latitude, and one bin is located at pole, totally there are $B = 21$ bins. We restrict $N \in \{0, 1\}$, where $0th$ order F_0 and the $1st$ order feature F_1 are considered. The dimensionality of STACOG features is $d = B + 13B^2 = 5754$. The Linear discriminant analysis (LDA) is performed for dimension reduction. For the HON4D feature, each video sequence is divided into $5 \times 4 \times 3$ spatiotemporal cells ($5 \times 4 \times 2$ cells for UTKinect-Action dataset because this dataset has typically shorter video clips) and separate HON4D feature is obtained for each cell. The final descriptor is a concatenation of the HON4Ds obtained from all the cells. Note that since the subjects' locations and the temporal motions are changing significantly in MSRDailyActivity3D dataset, local HON4D descriptor is adopted to represent shape and motion information. Because the dimensionality of HON4D feature is much higher than the other three features, we use PCA to reduce the HON4D feature dimension to 100 in our experiments. For feature normalization, we employ the Gaussian normalization scheme. For the classifiers used in the experiment, the SVMs with χ^2 kernel are used. For the random forests the number of trees can be selected from $[1, 500]$, and the number of features used in each split can be selected from $[3, 60]$. The related parameters were adjusted in a tuning set, which is about 20% of the training examples in each training dataset.

7.5.3 Gaussian Normalization

After the feature representation, the data range of different features might be very different, a direct fusion of such features might not perform well. The Gaussian normalization is used to map different features into a comparable range. Suppose there are M video sequences in the database, the four types of features can form an $M \times N$ feature matrix $F = f_{ij}$, where f_{ij} is the jth feature component in feature vector $f_{i,\cdot}$, each feature vector is of N dimensions. Our goal is to normalize the entries in each column $f_{\cdot,j}$ to the same range so as to ensure that each individual feature component receives equal weight in determining the similarity between two vectors. We compute the mean μ_j and standard deviation σ_j of the sequence and then normalize the original sequence into a normal distribution $N \sim (0, 1)$ as follows:

$$f'_{ij} = \frac{f_{ij} - \mu_j}{\sigma_j} \tag{7.30}$$

then, the probability of a feature component value in the range of [-1, 1] is approximately 99%. An additional shift will guarantee that 99 percent of feature values are within [0,1]:

$$\tilde{f_{ij}} = \frac{f_{ij}^{'} + 1}{2} \tag{7.31}$$

After this shift, we can consider that all of the feature component values are within the range of [0,1]. Therefore, this normalization process ensures the same range of the feature components when different types of feature are used.

7.5.4 Experimental Results

We present the experimental results of individual features first, and then the fusion results based on different fusion methods.

7.5.4.1 Results of Individual Features

We first investigate the individual features on the depth action datasets. The bag-of-words approach is used for histogram construction and the SVM is used as the classifier. In order to explore the capability of different features, we use the bag-of-feature approach for the STACOG feature, other than the bag-of-frame as in [152]; for HON4D feature, we adopted the uniform settings, without using the skeleton information for local non-uniform quantization as in [126]; Different from [138] where the Naive Bayes nearest neighbor classifier was used, we extract the skeletons and then construct the histogram features for the SVM classifier.

The experimental results on four databases using four different features are shown in Fig. 7.6. The HON4D feature performs the best on the MSRAction3D (Accuracy: 92.0%) and MSRDailyActivity3D database (Accuracy: 75.6%), while on the other two databases, its accuracies are lower than some other features. On the other hand, the EigenJoints feature achieves the best results on UTKinect-Action (Accuracy: 87.9%) and CAD-60 (Accuracy: 81.3%) databases. This feature performs the second best in other two databases. It can also be observed that the STIP feature and the STACOG feature exhibit comparable performance although they are not the best on these four databases. This fair comparison of different features has not been carried out in previous research. Our evaluation tells that no single feature can perform the best in all databases. This is also one of the reasons why we are interested in studying the fusion-based approach for depth-based action recognition.

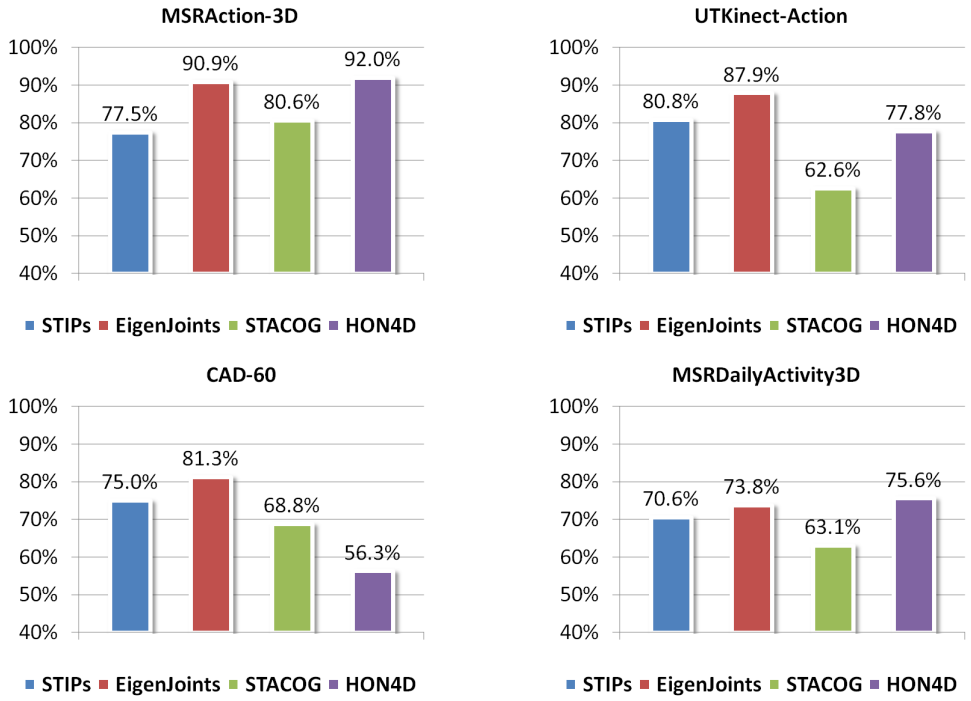

FIGURE 7.6: An evaluation of the individual features on four databases: MSRAction3D, MSRDailyAcitivity3D, Kinect-Action and CAD-60. The same training and test data are used for each feature to have a fair comparison.

After evaluating the individual features on the depth action databases, we conduct experiments that applying various fusion methods, and show the performance using the same data (training and test sets) on the four databases.

7.5.4.2 Fusion Results on MSRAction-3D Dataset

The experimental results on MSRAction-3D dataset using various fusion methods are shown in Table 7.2. The highest accuracy of 98.2% is achieved by the sum rule based fusion method, which is significantly better than any single features. For example, the accuracy of the best single feature HON4D is 92.0%. We can also observe that in the feature-level fusion scheme, the random forests fusion approach achieve the accuracy of 97.3%, close to the best result obtained by the sum rule based decision-level fusion method. The confusion matrix is shown in Fig. 7.7, where most of the actions can be separated well.

7.5.4.3 Fusion Results on UTKinect-Action Dataset

The results of different fusion methods with the four individual features on the UTKinect-Action dataset are shown in the second column of Table 7.2. From the results we can see

TABLE 7.2: The recognition accuracies of individual features and various fusion methods on four datasets. The decision-level fusion methods include the MAS, MAJ, SVM, SUM, MIN, MAX, MED, and PRODUCT, and the feature-level fusion methods include the RFs, JMI and CMIM. (See text for the meaning of each fusion method.)

Method		**Accuracy**			
		MSRAction3D	**UTKinect**	**CAD-60**	**MSRActivity**
Single Feature	STIPs	77.5%	80.8%	75.0%	70.6%
	EigenJoints	90.9%	87.9%	81.3%	73.8%
	STACOG	80.6%	62.6%	68.8%	63.1%
	HON4D	92.0%	77.8%	56.3%	75.6%
Decision-Level Fusion	MAS	93.3%	83.8%	68.8%	59.4%
	MAJ	96.3%	**92.9%**	**87.5%**	**88.1%**
	SVM	97.3%	86.9%	81.3%	79.4%
	SUM	**98.2%**	91.9%	68.8%	85.6%
	MIN	90.6%	61.6%	50.0%	58.8%
	MAX	95.2%	88.9%	68.8%	72.5%
	MED	96.3%	90.9%	62.5%	70.0%
	PRODUCT	96.4%	86.9%	68.8%	72.5%
Feature-Level Fusion	RFs	**97.3%**	**92.9%**	**87.5%**	**88.8%**
	JMI	94.2%	85.9%	81.3%	69.4%
	CMIM	94.6%	85.9%	81.3%	70.0%

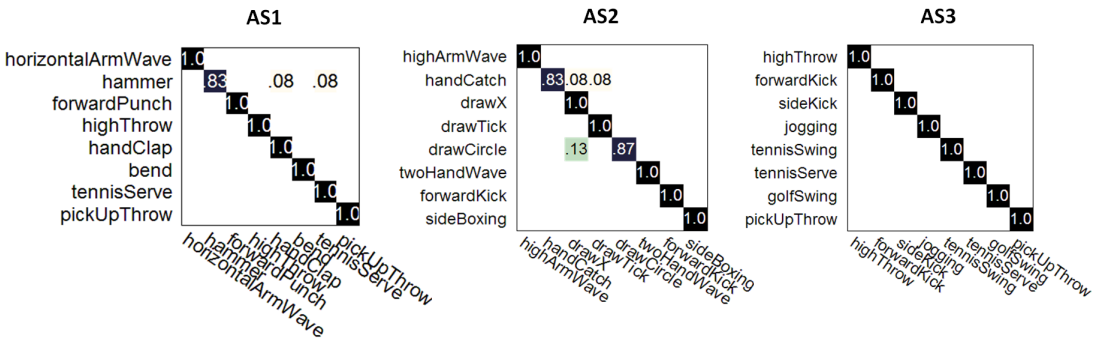

FIGURE 7.7: The confusion matrix of the SUM rule based fusion on MSRAction3D dataset.

that the decision-level fusion gets an accuracy 92.9% by applying the majority voting method. Comparable accuracies are achieved by the sum rule and median rule based fusion methods, which are 91.9% and 90.9%, respectively. On the other hand, random forests exhibits a better accuracy (92.9%) than the other feature-level fusion methods. The confusion matrix of the majority voting method is shown in Fig. 7.8, which shows that the actions "carry" and "throw" impact the overall accuracy because several samples are incorrectly classified as other (similar) actions, e.g., the action "carry" is actually a walking subject carrying an object, which confuses the system to classify it as walking. The ambiguity may also happen between actions "throw" and "push" in this dataset, thus classifying such actions is still challenging.

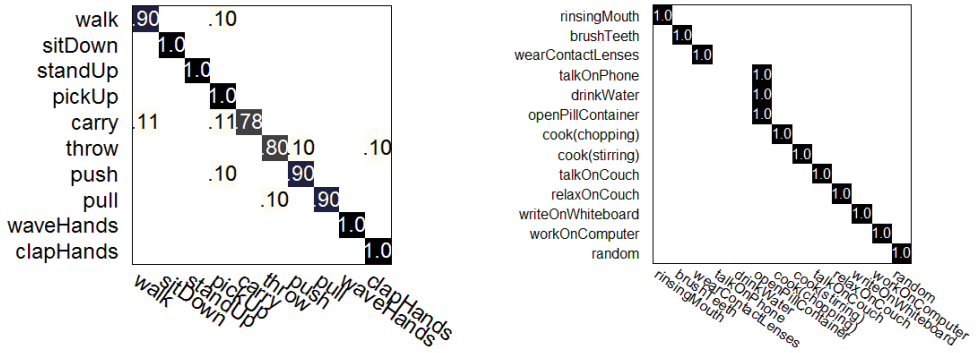

FIGURE 7.8: The confusion matrix of the majority voting fusion method on UTKinect-Action (left) and CAD-60 dataset (right).

7.5.4.4 Fusion Results on CAD-60 Dataset

On the CAD-60 dataset, the decision-level and feature-level fusions achieve the same best accuracy of 87.5% based on the MAJ rule and RFs, respectively. This accuracy is much higher than each of the individual features. From Table 7.2, one can observe that most of the decision-level fusion methods perform poorly, some even lower than the individual features. The feature-level fusion methods (except the RFs) have the same accuracy of 81.3% as the best individual feature, i.e., the EigenJoints feature. These results show clearly that some fusion methods cannot work well, depending on the input data and the specific fusion methods. That is why we need to investigate the different fusion methods carefully, in order to find the workable methods for the specific problem and special data.

7.5.4.5 Fusion Results on MSRDailyActivity3D Dataset

Results of different fusion methods on MSRDailyActivity3D dataset are shown in the last column of Table 7.2. The random forest method as a feature-level fusion scheme achieves the highest accuracy of 88.8%, higher than any other fusion methods on this dataset. Among the decision-level fusion methods, the majority voting has an accuracy of 88.1%, very close to the random forest method. The recognition accuracies of these two fusion methods are higher than each of the individual features, as shown in the top rows in Table 7.2.

7.5.5 Comparison with the State-of-the-art Methods

We further compare our fusion-based approach with the state-of-the-art methods for depth-based action recognition on the four challenging datasets. In all our experiments,

TABLE 7.3: Comparison of the recognition accuracies between our fusion-based approaches and all state-of-the-art methods on MSRAction3D dataset.

Method	Accuracy
High Dimensional Convolutional Network[125]	72.5%
Action Graph [23]	74.7%
HOJ3D [119]	79.0%
Key Pose Learning[122]	80.3%
EigenJoints[121]	82.3%
STOP [120]	84.8%
ROP [125]	86.2%
Actionlet [84]	88.2%
HON4D [126]	88.9%
DSTIP+DCSF [2]	89.3%
Part-set [159]	90.2%
Depth Motion Maps [139]	91.6%
DS-SRC [160]	93.6%
JAS (Cosine)+MaxMin+HOG^2 [135]	94.8%
STIP+Joint+RFs [132] (Our Preliminary)	94.3%
Decision Level Fusion (SUM Rule)	**98.2%**
Feature Level Fusion (Random Forests)	**97.3%**

TABLE 7.4: Comparison of the recognition accuracies between our fusion-based approaches and all state-of-the-art methods on the UTKinect-Action dataset. Note that, we used a less number of training examples, while the leave-one-out setting was used in [2].

Method	**Accuracy**
Posture Word [2]	79.57%
DSTIP+DCSF [2]	85.8%
HOJ3D [119]	90.9%
DS-SRC [160]	91.0%
STIP+Joint+RFs [132] (Our Preliminary)	91.9%
Decision Level Fusion (Majority Voting)	**92.9%**
Feature Level Fusion (Random Forests)	**92.9%**

TABLE 7.5: Performance comparison of our fusion-based approaches with the state-of-the-art methods on the CAD-60 dataset.

Method	**Precision/Recall**
[124]	67.9%/55.5%
[138]	71.9%/66.6%
[127]	80.8%/71.4%
[132] (Our Preliminary)	93.2%/84.6%
Decision Level Fusion (Majority Voting)	**96.4%/84.6%**
Feature Level Fusion (Random Forests)	**90.9%/84.6%**

TABLE 7.6: Performance comparison between our fusion-based approaches and the state-of-the-art methods on MSRDailyActivity3D dataset. Note that, all the actions are used in our experiment, while in [2] four actions (with less motion) were removed from the dataset in their experiment.

Method	**Accuracy**
NBNN+parts+time [136]	70.0%
Local HON4D [126]	80.0%
DCSF [2]	83.6%
RGGP+Fusion [137]	85.6%
Actionlet [84]	85.8%
DCSF+Joint [2]	88.2%
Decision Level Fusion (Majority Voting)	**88.1%**
Feature Level Fusion (Random Forests)	**88.8%**

the cross-subjects action recognition is conducted, since it is more appropriate in practical applications. We list all the published results on the four databases, to the best of our knowledge. In specific, on the MSRAction3D dataset, half of the subjects are used for training and the remaining half for testing. Table 7.3 shows the reported results in the literature on the MSRAction3D dataset. We can see that the sum rule based fusion can achieve an accuracy of 98.2% and the random forests feature-level fusion can get an accuracy of 97.3%, both are much higher than all of the previous reported results. On the UTKinect-Action dataset, the results are shown in Table 7.4, where the majority voting and Random Forests methods have the same accuracy of 92.9%, which is also higher than all the state-of-the-art methods on this dataset. For the CAD-60 dataset, the same "new person" setting is used as previous approaches in our experiment. The precision/recall were computed as the performance measure to have a direct comparison with the previous methods. The results are shown in Table 7.5. One can see that both the majority voting and random forests methods can get the same recall value, however, the majority voting has a higher precision value than the random forests. Finally, we compare our results with the state-of-the-art on the MSRDailyActivity3D dataset. From Table 7.6 one can see that, the random forests has the highest accuracy of 88.8%, which outperforms the DCSF+Joint approach in [2]. The majority voting can get an accuracy of 88.1%, which is lower than the 88.2% reported in [2], however, four actions were eliminated in their experiments, while we used all 16 actions.

Through the comparisons with the state-of-the-art methods, our fusion-based approaches perform the best on all four challenging datasets. The appropriate fusion methods have been found based on our exploration, at both the decision and feature levels, while some other fusion methods cannot work well for our problem. The comprehensive results demonstrate that *proper fusions* of different features are important that can significantly improve the action recognition performance in depth videos.

7.6 Conclusions

We have presented a comprehensive study of fusing diverse features for depth-based action recognition. Both the decision-level and feature-level fusion schemes have been explored with different methods at each fusion level. A number of experiments have been conduced on four depth databases. Experimentally we have shown that the four different features that we investigated can be complementary to each other, characterizing the depth actions from different aspects. Given the diverse features, different fusion methods perform quite differently in action recognition. Based on a systematic evaluation, the appropriate fusion methods have been found to significantly improve the recognition accuracies over each individual feature. We have also shown that our fusion-based action recognition in depth videos can outperform the state-of-the-art methods on all four challenging databases.

Chapter 8

Computational Depression Diagnosis Analysis using Deep Learning Approach

8.1 Abstract

As a severe psychiatric disorder disease, depression is a state of low mood and aversion to activity, which prevents a person from functioning normally in both work and daily lives. Recently, one of the approaches to track the patients with depression is monitoring through human-computer interaction. In this paper, we study the problem of automatically analyzing the depression diagnosis. A new approach to predict the Beck Depression Inventory II (BDI-II) values from video data is proposed based on the deep learning networks. The proposed framework is designed in a two stream manner, aiming at capturing both the facial appearance and dynamics. Besides, we employ the joint tuning layers that can implicitly integrate the appearance network and dynamics network. Experiments are carried out on two databases, AVEC2013 and AVEC2014 depression databases, and the experimental results (mean absolute error and root mean square error) show that our proposed approach significantly improve the depression value prediction, compared to the other visual-based approaches.

8.2 Introduction

Major depression disorder (MDD) is one of the prevalent causes of disability which heavily threatens the mental health of human among all age groups [161]. Depression

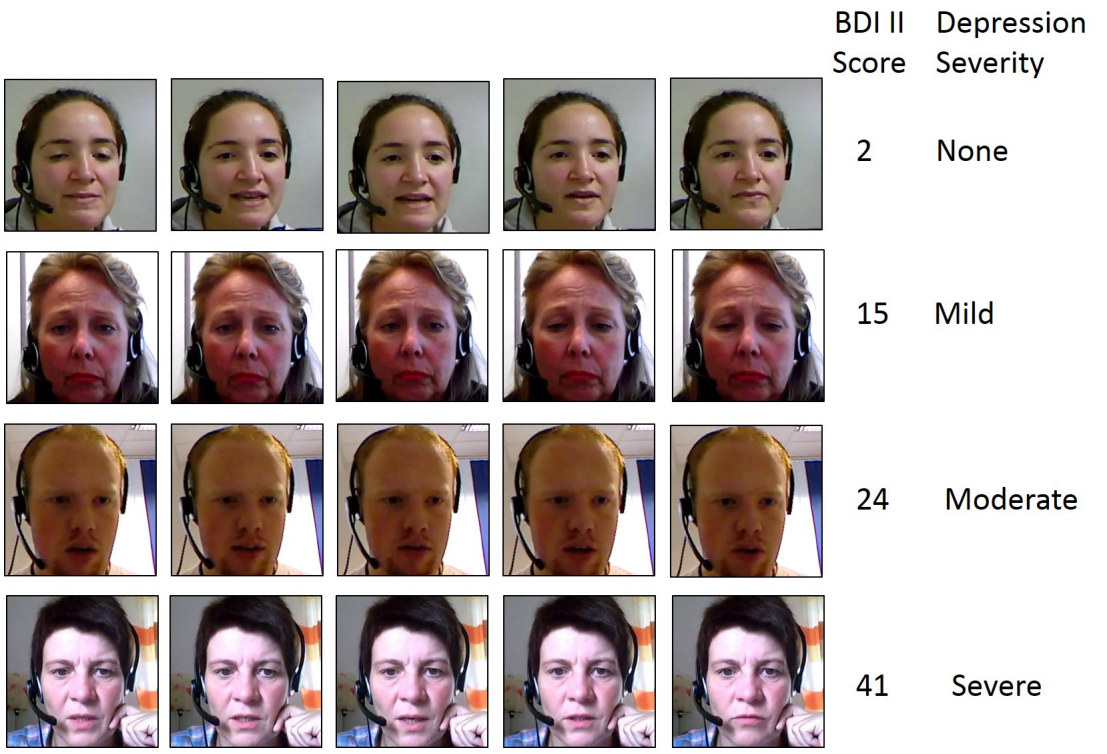

FIGURE 8.1: Example image frames with depression value score (BSDII score) and depression severity categories from AVEC2014 database.

disorder, with a 10-20% for women and 5-12% for men lifetime risk, can severely affect person's thoughts, behavior, feelings, and ability to work. Depressed people may feel sad, helpless, anxious, hopeless, worried, irritable, or restless, even in the worst scenario, severe depression could even lead to suicide[162] [163]. Fortunately, through proper medication, psychological counseling and other clinical methods, MDD is treatable despite of its severity. Currently, the diagnosis of MDD mostly requires comprehensive assessment by a experienced professional. It is largely constrained by individual subjective observation and lack of real-time measurements. As the increasing number of people suffering from MDD, it also brings the burden to the accurate diagnosis. Therefore, machine learning based methods is expected to provide a subjective assessment and a fast diagnosis, which can aid the MDD therapy.

The study on automatic mental health assessment have been given increasing attention in recent years. One way of keep track of patients with depression is online monitoring through human computer interaction and affective computing technologies. Particularly, machine learning methods of automatically analyzing affect and expressive behavour, are directly related to depression diagnosis. Evidence has shown that speech production differs in people with depression [164][165], thus many methods have been proposed utilizing audio cues for depression diagnosis [166–170]. It is also suggested that nonverbal cues is indicative of depression severity, such as gestures and expressions [171, 172].

TABLE 8.1: Beck Depression Inventory-II (BDI-II) score and depression severity.

BDI-II Score	Depression Severity
0 - 13	None
14 - 19	Mild
20 - 28	Moderate
29 - 63	Severe

Studies have showed that more than a half visual-based nonverbal behavior is around facial region in human communication activities [173–176]. Accordingly, in this work we focus on the visual-based nonverbal behavior for the depression diagnosis.

From the machine learning perspective, the depression diagnosis can be modeled as a regression problem, e.g., in AVEC2013 and AVEC2014 depression recognition challenge, the goal is to predict the depression value called Beck Depression Inventory-II (BDI-II score [177], see table 8.1) for the subject in each video. To deal with this problem, the facial appearance and dynamics in the video clips are often considered very useful for an depression diagnosis system. In this work, we study the depression recognition and propose a new approach to model the facial appearance and dynamics, based on deep convolutional neutral network (DCNN). Our approach is designed in a two stream manner, combined with joint-tuning layers for depression prediction. Specifically, facial appearance representation is modeled through a very deep neural network, with face frames as the input. Facial dynamics are modeled by another deep neural network, with face "flow images" as the input. Face "flow image" are generated by computing within the video sub-volumes using the optical flow, to capture the facial motions. The two deep networks are then integrated by joint-tuning layers into one deep network, which can further improve the overall performance. To the best of our knowledge, our proposed approach is the first time employing deep learning technology for the problem of depression diagnosis. Extensive experiments conducted on two depression databases AVEC2013 [178]and AVEC2014 [179] show that, our approach achieved better results than the other state-of-the-art visual-based methods for depression recognition.

The rest of our paper is organized as following: Firstly, in Section 8.3 previous work on depression diagnosis is described. Then in Section 8.4, our proposed method and network architecture are presented in details. Next, experiments are conducted on two databases and the results are shown in Section 8.5. Finally, some discussions and future work are given in Section 8.6.

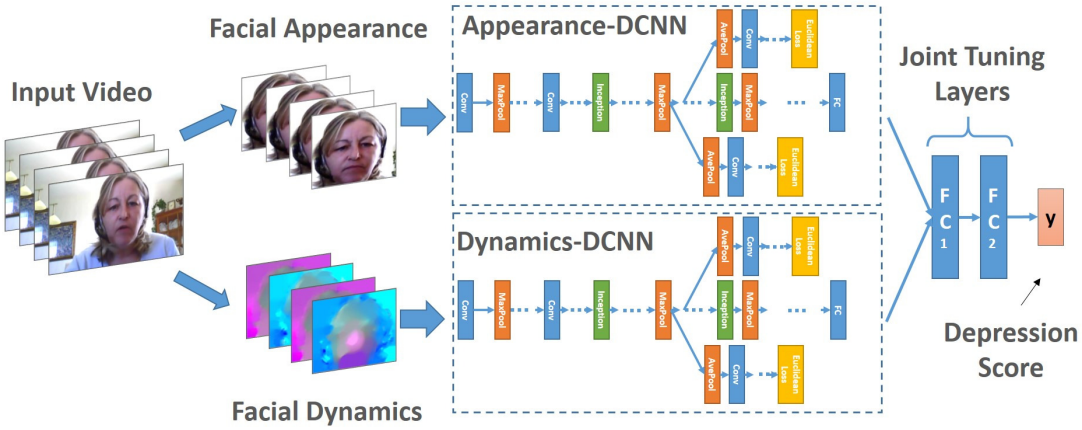

FIGURE 8.2: Schematic illustration of proposed method for depression recognition using deep learning approach.

8.3 Previous work

The Audio-Visual Emotion Challenge and Workshop 2013 and 2014 (AVEC2013 and AVEC2014) held the competition event for depression recognition as one of its sub challenges. Formulated as a regression problem, the diagnosis depression values are tested on the collected audio-video database (see Section 8.5.1 and 8.5.2 for more details about the database). Since our focus is on video-based learning approach, where audio clue are not utilized, in the following, we briefly describe the competing visual based methods in the AVEC2013 and AVEC2014 competitions.

Baseline features for AVEC2013 was using the Local Phase Quantization (LPQ)[180], which has shown good performance in facial expression recognition. Specifically for the AVEC2013 depression recognition, the face detection, fitting and alignment were firstly performed for each video frame. Then the dense LPQ features were extracted from those facial regions. Facial feature for each frame is represented by concatenating histograms of different blocks within the face region. Finally, the Support Vector Regressor (SVR) is applied for the prediction.

In Cummins et al. 's work [168], two different features are compared: the Space-Time Interest Points (STIPs) [85], and Pyramid of Histogram of Gradients (PHOG) [181]. In their method, face tracking is firstly applied for each video frames to obtain the face region. Then, both STIPs and PHOG features are extracted from the aligned face images. Those features are further generated as histograms by using the bag-of-word scheme, respectively. Finally, SVR with histogram intersection kernel was used for the training and testing. In their experiments, PHOG has shown better results than STIPs.

Meng et al. in their work [167] utilized Motion History Histogram (MHH) [182] to characterize motion information of each pixel in the video. Totally there were 5 MHH

based images were generated from each video frame. Then Edge Orientation Histogram (EOH) and Local Binary Patterns (LBP) [183] features were extracted from each MHH based image. Finally, a Partial Least Squares (PLS) [184] regressor was used for the regression. In their method, the MHH based descriptor to some extend can reflect all behaviors but temporal information is still not well-encoded.

In our previous work [185], the temporal dynamic is captured by the LPQ-TOP features from facial region sub-volumes. Then a behavior pattern dictionary is learned through sparse coding schemes. The sparse codes are calculated for each LPQ-TOP feature separately. Finally, a discriminate mapping method and decision level fusion were applied to further improve the accuracy for depression diagnosis.

In the AVEC2014, local dynamic appearance descriptor LGBP-TOP [186] has been adopted as the baseline video features. LGBP-TOP utilize a number of Gabor filters on a block of consecutive frames as input, then apply LBP feature extraction from three different orthogonal slice of the block: XY, XT and YT. The resulting patterns are further histogrammed and concatenated into the final feature representation. Support Vector Regressor (SVR) is used for the prediction, which is the same as AVEC2013.

In the video based approach from [187], the authors detected the face within each video frame firstly, then utilized three motion related features: motion history image, motion static image and motion average image from the detection face region. The feature were also combined with the relative differences of the face and eye coordinates. Finally, the extracted features were fed into a SVR for the prediction.

In [188], the authors firstly detected and cropped the faces within each video frame, then three features were extracted: Local Binary Patterns (LBP), Edge Orientation Histogram (EOH) and Local Phase Quantization (LPQ). Instead of generating from a image sequence, they proposed an 1-D Motion History Image (MHH) that extracts the changes on each component in a feature vector sequence. Then histogram features are used to represent all the components of the feature vector in one video. Partial Least Squares (PLS) regression is applied for the final prediction.

In [189], the authors proposed to utilize both LGBP-TOP and LPQ features for the video representation. They focus on the inner facial regions that correspond to eyes and mouth for the feature extraction. Then the Canonical Correlation Analysis (CCA) is applied on the feature vectors, and the two features are combined to generate the final regression results.

Most of the above mentioned methods were based on hand-crafted features which were proposed for facial analysis or expression recognition. These features may not be appropriate for for the task of depression analysis. Therefore it is necessary to explore a

more robust representation for the depression data, which can better capture the appearance and dynamics cues. In this work, we propose a new approach that based on deep learning networks, for the depression diagnosis prediction.

8.4 Network architectures for depression recognition

Video data can be naturally viewed into two components, i.e., spatial and temporal components. For the problem of depression recognition, on one hand, the spatial part carries the appearance information about the face and static expressions for the subject in the video. On the other hand, the temporal part, captures the motion across the frames, contains the facial dynamics such as the expression and micro expression changes of the subject. Therefore, we explore the architecture from video data accordingly, by treating it into appearance and dynamics models. As shown in Figure 8.2, each part is implemented using a DCNN. Moreover, joint layers are proposed to combine the two streams for the final depression recognition.

8.4.1 Appearance-DCNN

Deep convolutional neutral networks (DCNN) are known to be very effective to learn face representations given large number of face samples. However, for the specific task of depression recognition, usually the size of data available is very limited. To handle this issue, we utilize a cascaded way to train the facial appearance deep model by two steps: a pre-training step and a fine-tuning step.

In the pre-training step, a deep network (e.g., GoogLeNet [190]) is trained from scratch by utilizing large number of face samples with identity labels. After this step, the deep network is expected to effectively capture rich facial structures, which can be considered as a base deep model for facial representations. Since this pre-trained network aims at minimizing the identification error, it is still necessary to fine-tune the network for depression recognition, i.e., a regression task. Figure 8.3 shows the detailed deep network architecture that is applied in our framework.

Next, in the fine-tuning step, the aim is to adapt the pre-trained network with depression data so that the network is capable of predicting the depression values given the input of image frames. Because depression prediction serves as a regression problem, the network loss in this step is changed to Euclidean loss for regression, other than the softmax loss used in pre-training step. Mathematically, the Euclidean loss function E computes the

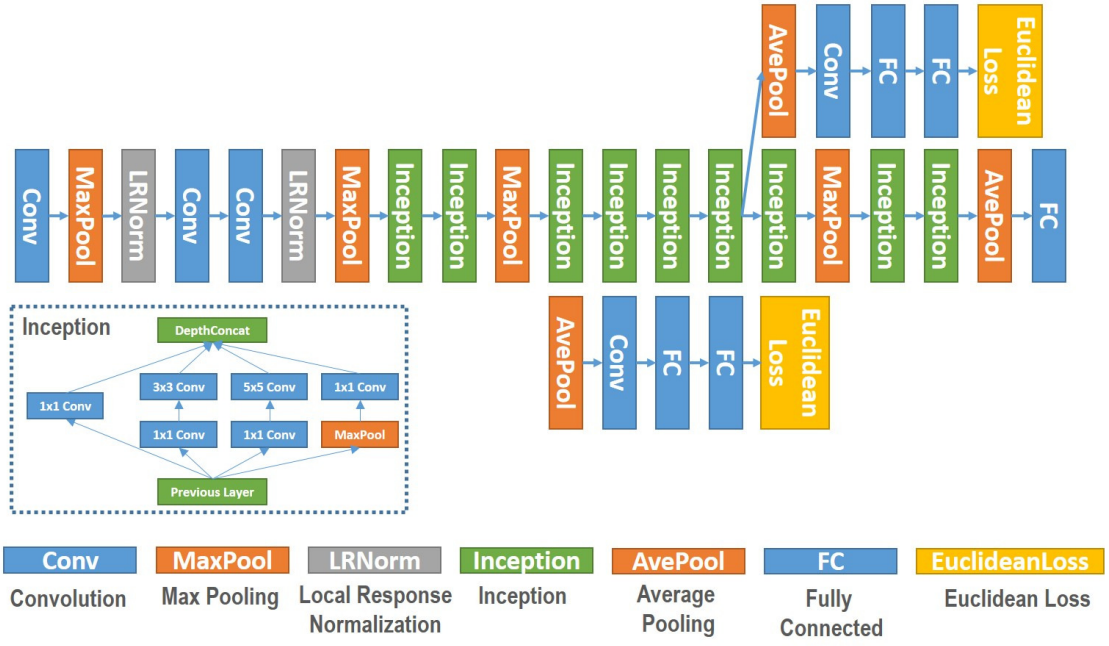

FIGURE 8.3: Network architecture (GoogLeNet) of proposed method for depression recognition.

the sum of squares of differences of its two inputs, which can be written as:

$$E = \frac{1}{2N}\Sigma_{i=1}^{N}||\hat{y}_i - y_i||^2, \tag{8.1}$$

where N is the number of samples, $\hat{y}_i$ is the output from the network and y_i is the ground truth.

Then the training images frames with their depression values are fed into modified pre-trained network for the fine-tuning. After this step, the network is capable of learning the depression representations given the input of image frames.

8.4.2 Dynamics-DCNN

In this section, we describe the architecture to model facial dynamics, by utilizing the optical flow between video frames, which is named dynamics-DCNN. Unlike the appearance model described above, the input of this model is formed by optical flows between several consecutive video frames. In this way, the input itself captures the motion between frames caused by the movement of the subject, so that the network does not to estimate the motion implicitly.

Specifically, for each frame in the video, we compute the optical flow displacements between several consecutive frames. Since in the depression recognition we focus on the face region in each video frame, however the changes of face region between two frames

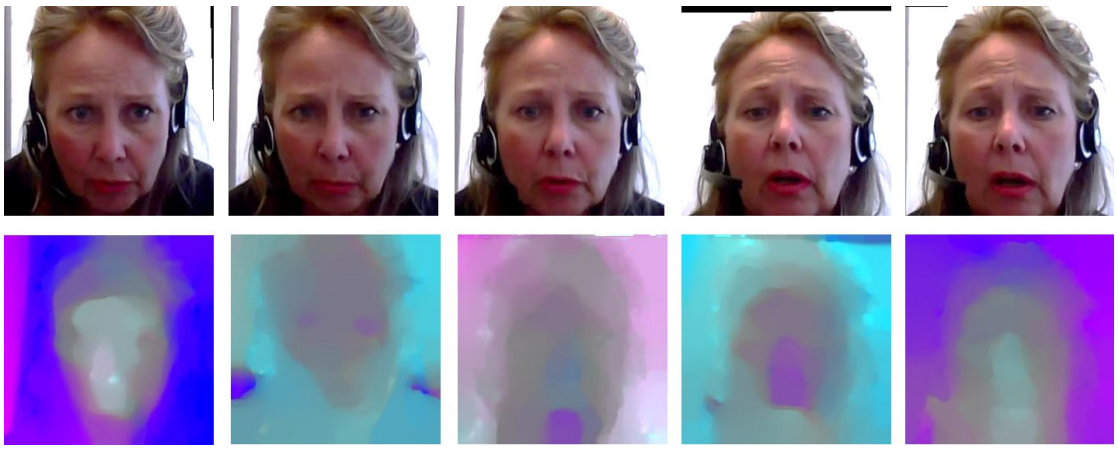

FIGURE 8.4: Example image frames (top row) and generated flow images (bottom row) from AVEC2014 dataset.

are usually too subtle. Therefore, we compute optical flow between several consecutive frames (e.g., every 10 frames), so that the motion of the face can be well captured at the same time the video redundancy can be reduced.

Then the optical flow computed from each image is transformed into a "flow image" [191]. The three channels of the "flow image" are constructed by the horizontal and vertical components: x flow values, y flow values, as well as the flow magnitude. The values of each channel are then centered and normalized between 0 and 255, respectively. Figure 8.4 shows some examples of RGB image frames and generated flow images.

Given the "flow images" computed from video frames, a DCNN is trained for modeling the facial dynamics. The architecture and configurations remain largely the same as that used in the appearance DCNN, without the pre-training step. Figure 8.3 shown the illustration of the networks.

8.4.3 Joint tuning layers

In our approach, the appearance DCNN and the dynamics DCNN are capable of predicting the depression values separately. In order to further improve the performance and integrated the two individual deep networks, we propose to construct joint tuning layers with a fine-tuning step, aiming at combining the appearance and dynamic models. Specifically, two fully connected layers are constructed with different number of hidden units (e.g., 512 and 256, respectively), connecting the concatenated feature layers of both appearance and dynamics networks. The final loss function is still kept the same Euclidean loss for regression task. The gradually decreasing number of hidden units in joint tuning layers, is designed to better convergence for the single value regression. During the training, the two DCNNs are trained respectively, which can be viewed as

a pre-training step. Then the final fine-tuning is conducted using the architecture with joint tuning layers as shown in Figure 8.2, where the input are the RGB video frames as well as its computed "flow image".

8.5 Experimental Results

The experiments are conducted on two databases: the Audio/Visual Emotion Challenge (AVEC) 2013 [178] and 2014 [179] depression sub-challenge databases. In this section, firstly we briefly describe the two databases that are used in our work. Then we show and analyze the experimental results. Finally the comparison with other state-of-the-art methods is presented.

8.5.1 AVEC2013 Depression Database

AVEC2013 depression database [178] is collected in the wild which contains 340 video clips from 292 subjects. A subset of the audio-visual depressive language corpus (AViD-Corpus) from AVEC2013 database is used for the depression sub-challenge. This subset contains video clips of subjects performing a human-computer interaction task, which is collected by a webcam as well as a microphone. There is only one subject in each video clips with no constrains when being recorded. The average length of each video clip is about 25 minutes. The age range of the subjects is from 18 to 63 years old with a mean age 31.5 years. Some example images of this database are shown in Figure 8.1. Specifically for the depression sub-challenge, totally there are 150 videos from 82 subjects are used and split into three partitions: training set, development set, and test set. Each of the three set contains 50 video clips. For each video clip, a depression severity is assigned as the label, which was accessed using a standardized depression questionnaire, the Beck Depression Inventory-II (BDI-II) [178]. BDI-II scores ranges from 0 to 63, where 0-3 indicates minimal depression, 14-19 indicates mild depression, 20-28 indicates moderate depression, and 29-63 indicates sever depression. In our experiments, all the data from training set and development set are used for training our proposed deep model, while the test set is used to evaluate the overall performances for depression recognition.

8.5.2 AVEC2014 Depression Database

AVEC2014 depression database is proposed for the Audio/Visual Emotion Challenge 2014 [179], where a subset of the audio-visual depressive language corpus (AViD-Corpus)

is used for the depression sub-challenge. For the AVEC2014 challenge, two of the 12 tasks from AViD-Corpus are used, which are referred as Freeform and Northwind tasks. For both tasks, the recorded videos are split into three partitions: training, development, and tests of 50 videos, respectively. In our experiments, we merge the training and development set from both Freeform and Northwind data as one training set. The overall performances are reported by testing video clips from the test set. Some example images of this database are shown in Figure 8.4 (top row).

8.5.3 Experimental Settings

8.5.3.1 Face region detection and alignment

In order to extract facial representations from the videos, the first step is to apply face detection and facial landmark localization for each video frames. In our experiments, we use dlib library [192] in this step. Then within each video frames, the facial region are cropped and aligned by the eye locations with an image size of 256×256. This setting is kept for all face images for both training set and test set.

8.5.3.2 Facial dynamics computation

After the above step, for each video clip in the dataset, a sequence of facial regions are extracted where faces are also aligned according to the eye locations. To compute the facial dynamics, we applied optical flow computation between two frames, with an interval of 10 frames, which is empirically selected and shows good performances in our experiments. A "flow image" is generated for each frame by taking the x and y flow values as the first and second channel. The third channel is created by calculating the magnitude of optical flow. Those values are also centered around 128 and normalized between 0 to 255.

8.5.3.3 Subsampling

In order to reduce the large number of frames in each video clip, we applied a subsampling that take video frames with an interval of 100 frames and frames for AVEC2013 and 10 frames for AVEC2014 databases, respectively. Totally there are about 380,000 video frames extracted from the AVEC2013 database, while on AVEC2014 the number of extracted video frames is about 50,000.

8.5.3.4 Deep convolutional neural network

The layer configuration of our appearance and dynamics deep network is schematically shown in Figure 8.2, and the details are described in Section 8.4. For appearance and dynamic DCNN, the model architecture are similar to the GoogLeNet model [190]. In our experiments, the joint tuning layers are designed as two fully connected layers with 512 and 256 hidden units, respectively. The networks are trained with stochastic gradient using caffe deep learning toolbox [190] with batch size 32. In appearance DCNN, the model is fine-tuned from our pre-trained deep face model. This pre-trained model is trained on CASIA WebFace Database [193] with 494414 images of 10575 subjects. While in the dynamics DCNN, the training starts from scratch. The loss function is set to Euclidean loss for regression. The number of iterations for appearance model and dynamics model are set to 400,000 and 600,000, respectively. The base learning rate are set to 0.001 and reduced by polynomial with gamma equals to 0.5. The momentum is set to 0.9 with weight decay equals to 0.0002. For joint tuning, the number of iterations is set to 200,000, with base learning rate 0.0001. All experiments are conducted using Titan-X GPU with 12GB memory.

8.5.3.5 Performance Measurement

For each video, the predicted depression value is computed by averaging the predicted values from both appearance model and dynamics model for each frame (subsampled) in the video. The overall performance is measured using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

The MAE is computed by:

$$MAE = \frac{1}{N}\Sigma_{i=1}^{N}|y_i - \hat{y}_i|. \tag{8.2}$$

And the RMSE is computed by:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{8.3}$$

where N is the number of data samples, y_i denotes the ground truth of $i-th$ sample and $\hat{y}_i$ is the predicted value of $i-th$ samples.

TABLE 8.2: Depression recognition results of the proposed methods on AVEC2013 (Test set). Ave. means score level fusion by taking average.

Our Methods	**RMSE**	**MAE**
Facial Appearance Model	10.19	7.88
Facial Dynamics Model	10.02	7.87
Appearance & Dynamics (Ave.)	9.91	7.74
Appearance & Dynamics (Joint Tuning)	**9.82**	**7.58**

8.5.4 Performances of individual models for depression recognition

The results of depression recognition on AVEC2013 and AVEC2014 databases are shown in table 8.2 and 8.3, respectively. Firstly, we explored the performance using individual deep model (appearance and dynamics models) without any joint tuning procedure. From table 8.2 one can see that, on AVEC2013 database, when only the appearances model is used, the MAE and RMSE achieved 7.88 and 10.19, respectively. While the MAE and RMSE are 7.87 and 10.02 when using dynamics model, which is comparable to the appearance model. From table 8.3 (AVEC 2014), when using appearance model the MAE and RMSE are obtained 7.82 and 10.36, respectively. Comparable MAE and RMSE are also obtained (7.52 and 9.80 respectively) when using dynamics model. These results show the effectiveness of appearance model as well as dynamics models, both of which are capable of learning the facial representations for depression recognition from the video frames.

8.5.5 Overall Performance by fusing the individual models

We also compute the performance by fusing the appearance and dynamics models. This fusing is conducted on score level and the results are computed by averaging the results for both appearance and dynamics models. The experimental results are shown in table 8.2 and 8.3 for AVEC2013 and AVEC2014, respectively. From table 8.2, one can see that, the results after fusing the models obtained the MAE 7.74 and RMSE 9.91 on AVEC2013 database. On AVEC2014 database (see table 8.3), the fusion results achieved the MAE 7.53 and RMSE 9.73, respectively. Both results performs better than those using the individual model. This observation shows that by fusing the appearance and dynamics models, the overall performance can be improved than using individual model, which further implies the necessity of utilizing both facial appearances and dynamics for the depression recognition.

TABLE 8.3: Depression recognition results of the proposed methods on AVEC2014 (Test set). Ave. means score level fusion by taking average.

Our Methods	**RMSE**	**MAE**
Facial Appearance Model	10.36	7.82
Facial Dynamics Model	9.80	7.52
Appearance & Dynamics (Ave.)	9.73	7.53
Appearance & Dynamics (Joint Tuning)	**9.55**	**7.47**

TABLE 8.4: Depression recognition result comparison to other methods on AVEC2013 (Test set). Note that the listed results are using video data only.

Methods	**RMSE**	**MAE**
Baseline [178]	13.61	10.88
team-australia [168]	10.45	N/A
Uni-Ulm [167]	11.19	9.14
Wen [185]	10.27	8.22
Our Method	**9.82**	**7.58**

8.5.6 Overall Performance by Joint Tuning

Next, we conduct experiments using the proposed joint tuning approach. The results are shown in forth columns in table 8.2 and 8.3, for AVEC2013 and AVEC2014, respectively. It can be seen from the table that, when joint tuning is applied, the MAE and RMSE obtained are 7.58 and 9.82 respectively on AVEC2013 database. These results are significantly better than that obtained by individual models. Besides, the joint tuning results are also better than that obtained by score-level fusion (MAE 7.74 and RMSE 9.91) of the two models. Similar observations can also be found on AVEC2014 database (see table 8.3), the best result is achieved by using joint tuning, where the MAE is 7.47, and RMSE is 9.55. These results illustrate that, the proposed joint tuning approach can better utilizing both the appearance and dynamics models, and the performance is significantly improved. Moreover, the comparison with score-level fusion also shows the effectiveness of the proposed joint tuning approach.

8.5.7 Comparison with pervious methods

Finally, we compare our approach to other methods on both AVEC2013 and AVEC2014 database. For a fair comparison, we show the results that are only using video data for depression recognition in table 8.4 and 8.5. From the table one can see that, our approach achieves better performance than the other listed methods on both AVEC2013 and AVEC2014 database. This further shows the effectiveness of our proposed approach for depression recognition.

TABLE 8.5: Depression recognition result comparison to other methods on AVEC2014 (Test set). Note that the listed results are using video data only.

Methods	RMSE	MAE
Baseline [179]	10.86	8.86
UUIMSidorov [194]	13.87	11.20
InaoeBuap [187]	11.91	9.35
Brunel [188]	10.50	8.44
BU-CMPE [189]	9.97	7.96
Our Method	**9.82**	**7.58**

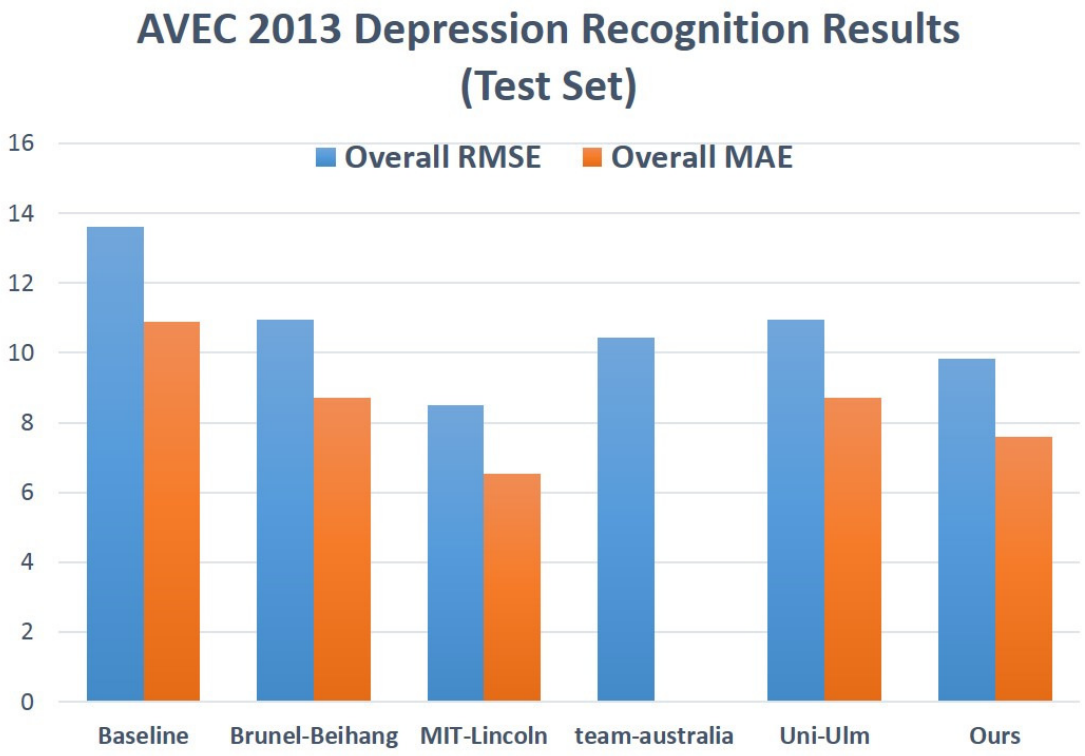

FIGURE 8.5: Comparison of depression recognition results on AVEC2014 competition. Note that several of the listed methods are utilizing audio data while our method only use visual data.

8.6 Discussion and Conclusions

Since the AVEC2013 and AVEC2014 databases are also used for the depression competition, in this section, we show our results with comparison to the competition results. Note that, our approach only utilized the video data without using audio clues, however in the listed competition results, audio based approaches are also utilized in many of those methods. We believe that by combining audio based approach, our results could be further improved. In this work, our focus is exploring visual-based approaches for depression analysis.

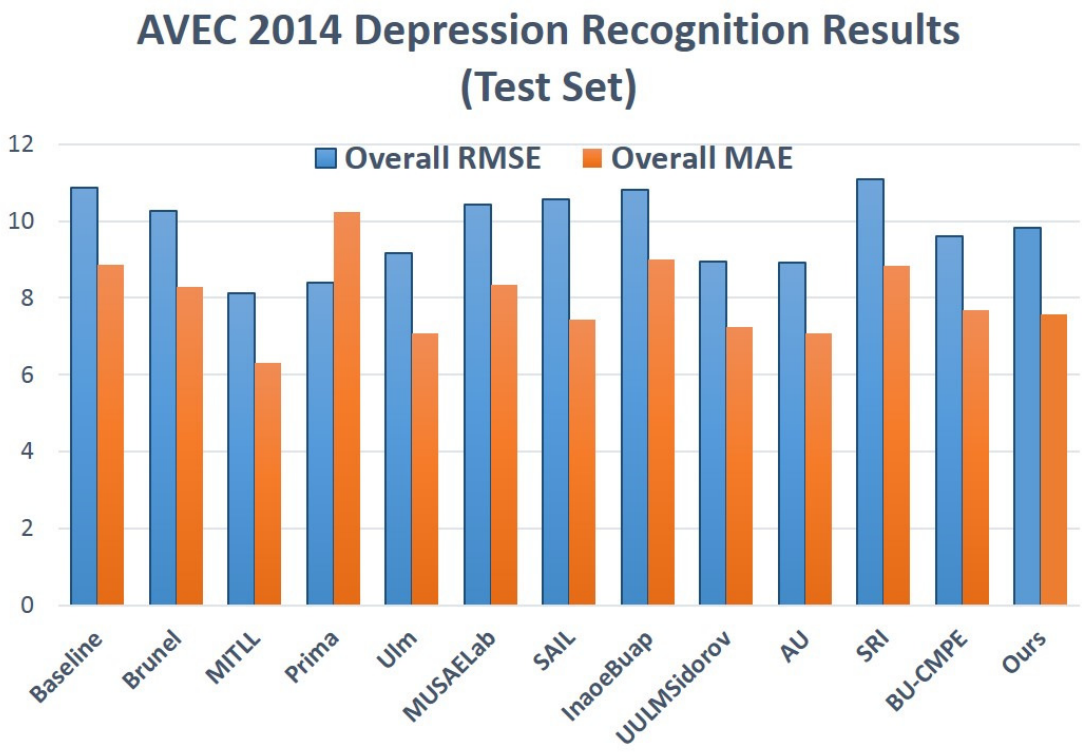

FIGURE 8.6: Comparison of depression recognition results on AVEC2014 competition. Note that several of the listed methods are utilizing audio data while our method only use visual data.

The results of the AVEC2013 and AVEC2014 challenges are shown in figure 8.5 and 8.6. Note that, in these tables, most of the methods are utilizing both video and audio data for the depression recognition, while in our approach, only video data are used. From figures 8.5, one can see that, our approach performs better than four methods on the AVEC2013 database, and comparable to the best results from [166], where both audio and video data are used. On the AVEC2014 database (see figure 8.6), our approach also achieved promising results, which is comparable to the top methods where both audio and video data are utilized.

In summary, we investigated the problem of depression value prediction from video data. In order to model both facial appearance and dynamics for depression recognition, we proposed a new approach based on deep learning, which is the first time employing deep representations for depression analysis, to the best of our knowledge. In our proposed deep network, a two steam manner is designed to take facial images and facial flows as input to model the depression information, which we called appearance and dynamics DCNN, respectively. Then, we proposed to construct joint tuning layers, to combine the appearance and dynamics DCNN, and further improve the performance. Experimental results on two depression databases, AVEC2013 and AVEC2014 shown that, our

approach achieved better results compared to other visual based approaches for depression prediction. Moreover, our result obtained by video data only obtained comparable performance to the state-of-the-art approaches in the AVEC competition, where most of the methods were utilizing both video and audio data.

Chapter 9

Summary

In this dissertation, the multi-modality human action recognition is studied. Two main aspects are investigated: multi-spectral action recognition, depth-based action recognition. Besides, the special category of action recognition: facial action analysis for depression recognition is also studied. The objective of this dissertation is to investigate these relatively new topics, to extend the research on action recognition, and propose new approaches handling the challenges and improve the overall performance. In this chapter, the summary of the contributions of this dissertation are presented. Then some future extensions of the current work are described.

9.1 Summary

This dissertation has presented and studied several topics for multi-modality action recognition. Specifically the multi-spectrum action recognition, RGB-D action recognition, and facial action analysis for depression recognition are investigated and new approaches are proposed to handle this problems. We summarize our work and discuss conclusions as following.

9.1.1 Multi-Spectral Action Database

We proposed a new database for the study of multi-modality action recognition. The collection was using three cameras that capturing the data in visible, near infrared, and infrared spectrum, respectively. This database contains a large number of samples which was collected with 30 action classes from 30 subjects. This is the first human action database that have three different spectrum: visible, infrared, and near infrared, to the best of our knowledge. Also, this database can serve as an useful benchmarking

for the study of action recognition in many aspects, e.g., action recognition in different individual modality, action recognition cross different modalities, and action recognition combining the different modalities, etc.

9.1.2 Visible to Infrared Action Recognition

We explored the new problem of visible to infrared action recognition, and introduced an approach for such problem. The idea of visible to infrared action recognition is to better utilize the action data in visible, to help recognize the actions in thermal infrared. We adopted the adaptive SVM in our approach to train the model using both visible and infrared data and test on the infrared data. Experimental results compared to the correlation based approaches shown that, our approach significantly improved the performance for the visible to infrared action recognition.

9.1.3 Heterogeneous Action Recognition for Infrared Action Recognition

We investigated the heterogeneous approaches for action recognition from visible to infrared. Two new approaches were proposed under the framework of maximizing the mutual information between actions in different modalities, i.e., visible and thermal infrared. To achieve this goal, one approach we proposed is based on correlation mapping, while the second approach is based on manifold learning and discriminative mapping for heterogeneous action recognition. Experiments have been conducted on a relatively large database with 30 actions of different modalities shown the usefulness and effectiveness of our approaches for addressing the challenging problem.

9.1.4 Infrared Action Recognition using Sparse Coding Approach

We investigated the problem of action recognition in thermal infrared spectrum. We propose a new method based on spatial temporal sparse coding. The proposed approach is based on learning the sparse dictionary from the thermal infrared data directly, using histogram representations. In order to integrating both spatial and temporal structures of the action videos, a spatiotemporal histogram is extracted from three orthogonal planes. Besides, a saliency map is computed to incorporate the spatial distribution of local features. Experimental results demonstrate the proposed feature achieved promising performance on IR action recognition.

9.1.5 Evaluation of Spatial-Temporal Interest Points Features for RGB-D Action Recognition

We conducted an evaluation of local spatial-temporal features on RGB-D action data. We have evaluating and compared several existing approaches which was proposed for visible action data, on a relatively new topic: RGB-D action recognition. Firstly, we conducted evaluations using different spatial temporal features for action recognition in depth data. Totally there are 14 features that were used in our evaluation. Further, two schemes are proposed to refine the features. One scheme is to utilize the skeleton joints modality to constrain the STIP locations, the other scheme is to utilize the RGB modality for the STIP detection. Experiments wrere carried out on 4 RGB-D databases. Our evaluations concludes that on different depth database, spatial temporal features perform quite differently. By combining the features using our proposed schemes, the overall performance can be improved.

9.1.6 Fusion Approaches for Depth based Action Recognition

We explored and studied the fusion methods on spatial-temporal features for RGB-D action recognition. Fusion approaches for the RGB-D action recognition has not been well studied yet, therefore we were aiming at combining different features on different levels for the RGB-D action recognition, in order to improve the performance compare to individual approaches. To accomplish this goal, we introduced 11 different fusion methods based on both feature level and decision level fusion scheme. For the RGB-D features, totally there are four different type of features that have been applied, aiming at representing actions from different aspects. Experimental has been done in 4 different databases, and the results shown that for the feature level fusion, random forests shows good results, and for the decision-level fusion, majority voting approach performs better than the other methods. These results suggests that different features can be complementary to each other for the depth based action representation, by combining the features using appropriate fusion method the performance can be significantly improved over each individual feature.

9.1.7 Facial Action Analysis for Depression Recognition

We explored deep learning approach on facial action analysis for the problem of depression recognition. To the best of our knowledge, we have proposed a new approach based on deep convolutional neural network, which is the first time introducing the deep approach for depression recognition based on facial video data. A two stream architecture

of deep network is proposed dealing with both facial appearance and facial dynamics from the video data. Experimental results on two databases shown that our proposed approach achieved significantly better results. These results suggest the feasibility of automatic prediction of depression based on facial action analysis using the deep learning approach.

9.2 Future Work

In this section, we propose several future research topics based on our study of multi-modality human action. These future work are summarized as following.

For the action recognition problem, it is more still very challenging to recognize the real world actions or activities. More robust and power techniques towards to the real applications has not been well studied yet. One of the major challenges lies in the fundamental theory of the action recognition, such as the mathematical definition of vision based human actions, etc. It is also necessary to collect more practical and general action databases for the future action recognition research.

For the action recognition beyond visible, we have studied the action patterns that are performed by single subject. In the future, it is necessary to study more complex actions/activities, for the practical considerations. For examples, action detection in the dark using IR camera, for the unusual actions, action recognition for night vision video surveillance system and action recognition for group of people are all interesting but challenging problems that can be issued in the future.

For the multi-modality action recognition problem, in this work, we have proposed approaches for the heterogeneous action recognition problem in different spectra. In our study, two categories of methods, i.e., correlation mapping mapping and manifold learning, are utilized in recognizing actions from different modalities. In addition to these methods, metric learning is also another direction, in which a metric between different modalities is learned so that the samples can be matched effectively.

Toward realistic RGB-D action recognition. In this dissertation, we have discussed the classification performance of depth based action recognition, specific by evaluating the spatial-temporal features, and also combining different local and global features. The fusion methods shows benefits when multiple action representations are extracted from video clips. Consequently, it seems necessary to study more effective action features/-models, from different aspects. One possible exploration, is based on deep convolutional neural network, which has show promising performance in many computer vision topics. However, in action recognition, the superiority of deep features is still not significant

comparing to the traditional features, also how to modeling the temporal information in action clips is still under-studied. Specially in RGB-D data, how to design effective and robust deep networks is also an important task to develop a deep RGB-D action recognition system.

Modeling temporal information in action recognition. In Chapter VIII, we shown that temporal information is modeled through optical flow computed in video clips, to help the depression representation. Since the motion/temporal information is very important in action analysis, it is still necessary to study how to more efficiently and effectively model the temporal information. One interesting direction is based on the deep learning approach. Specifically, recurrent neural network (RNN) is proposed to handle the sequence data, and its further extension LSTM has shown promising results in action recognition. However, how to integrate the LSTM with DCNN, and improve the performance is still not well-studied for action recognition, this should be investigated in the future.

Bibliography

[1] Yu Zhu and Guodong Guo. A study on visible to infrared action recognition. *Signal Processing Letters, IEEE*, 20(9):897–900, 2013.

[2] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

[3] Ying Wu and Thomas S Huang. Vision-based gesture recognition: A review. *Urbana*, 51:61801, 1999.

[4] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1):259–275, 2003.

[5] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

[6] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2):90–126, 2006.

[7] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[8] Dariu M Gavrila. The visual analysis of human movement: A survey. *Computer vision and image understanding*, 73(1):82–98, 1999.

[9] Pavan Turaga, Rama Chellappa, Venkatramana S Subrahmanian, and Octavian Udrea. Machine recognition of human activities: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1473–1488, 2008.

[10] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[11] Liang Wang, Weiming Hu, and Tieniu Tan. Recent developments in human motion analysis. *Pattern recognition*, 36(3):585–601, 2003.

[12] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[13] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. IEEE, 2011.

[14] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, 2009.

[15] Aaron F Bobick. Movement, activity and action: the role of knowledge in the perception of motion. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 352(1358):1257–1265, 1997.

[16] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90–102. IEEE, 1997.

[17] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[18] Lulu Chen, Hong Wei, and James M Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 2013.

[19] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[20] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE, 2005.

[21] M.D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[22] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2929–2936. IEEE, 2009.

[23] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–14. IEEE, 2010.

[24] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013.

[25] HA Jhuang, HA Garrote, EA Poggio, TA Serre, and T Hmdb. A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2011.

[26] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[27] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(3):257–267, 2001.

[28] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, 1962.

[29] Eli Shechtman and Michal Irani. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 405–412. IEEE, 2005.

[30] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal shape and flow correlation for action recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[31] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176. IEEE, 2011.

[32] Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 144–149. IEEE, 2005.

[33] A Yilma and Mubarak Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 150–157. IEEE, 2005.

[34] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[35] Geert Willems, Tinne Tuytelaars, and Luc Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. *Computer Vision–ECCV 2008*, pages 650–663, 2008.

[36] Alexander Klaser, Marcin Marszałek, Cordelia Schmid, et al. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference, 2008*, 2008.

[37] Ivan Laptev and Tony Lindeberg. Velocity adaptation of space-time interest points. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 1, pages 52–56. IEEE, 2004.

[38] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[39] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[40] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[41] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3): 27, 2011.

[42] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

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

[44] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[45] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[46] JK Aggarwal and Michael S Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[47] V. N. Vapnik. Statistical learning theory. 1998.

[48] Seong G Kong, Jingu Heo, Besma R Abidi, Joonki Paik, and Mongi A Abidi. Recent advances in visual and infrared face recognition—a review. *Computer Vision and Image Understanding*, 97(1):103–135, 2005.

[49] Ju Han and Bir Bhanu. Human activity recognition in thermal infrared imagery. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 17–17. IEEE, 2005.

[50] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.

[51] Jun Yang, Rong Yan, and Alexander G Hauptmann. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.

[52] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4): 321–377, 1936.

[53] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[54] Tae-Kyun Kim, Shu-Fai Wong, and Roberto Cipolla. Tensor canonical correlation analysis for action classification. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[55] Annan Li, Shiguang Shan, Xilin Chen, and Wen Gao. Maximizing intra-individual correlations for face recognition across pose differences. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 605–611. IEEE, 2009.

[56] Abhishek Sharma and David W Jacobs. Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 593–600. IEEE, 2011.

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

[58] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC, British Machine Vision Conference*, 2009.

[59] George Bebis, Aglika Gyaourova, Saurabh Singh, and Ioannis Pavlidis. Face recognition by fusing thermal infrared and visible imagery. *Image and Vision Computing*, 24(7):727–742, 2006.

[60] Seong G Kong, Jingu Heo, Faysal Boughorbel, Yue Zheng, Besma R Abidi, Andreas Koschan, Mingzhong Yi, and Mongi A Abidi. Multiscale fusion of visible and thermal ir images for illumination-invariant face recognition. *International Journal of Computer Vision*, 71(2):215–233, 2007.

[61] Leedham Wang, G Leedham, and S-Y Cho. Infrared imaging of hand vein patterns for biometric purposes. *IET computer vision*, 1(3):113–122, 2007.

[62] Li Zhang, Bo Wu, and Ramakant Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[63] Zhaojun Xue, Dong Ming, Wei Song, Baikun Wan, and Shijiu Jin. Infrared gait recognition based on wavelet transform and support vector machine. *Pattern recognition*, 43(8):2904–2910, 2010.

[64] Brendan F Klare and Anil K Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1410–1422, 2013.

[65] Zhifeng Li, Dihong Gong, Yu Qiao, and Dacheng Tao. Common feature discriminant analysis for matching infrared face images to optical face images. *Image Processing, IEEE Transactions on*, 23(6):2436–2445, 2014.

[66] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 513–520. IEEE, 2011.

[67] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[68] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, pages 34–51. Springer, 2006.

[69] William Robson Schwartz, Huimin Guo, and Larry S Davis. A robust and scalable approach to face identification. In *Computer Vision–ECCV 2010*, pages 476–489. Springer, 2010.

[70] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S Davis. Human detection using partial least squares analysis. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 24–31. IEEE, 2009.

[71] Guodong Guo and Guowang Mu. Simultaneous dimensionality reduction and human age estimation via kernel partial least squares regression. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 657–664. IEEE, 2011.

[72] H. Wold. Path models with latent variables: The nipals approach. In H. M. Blalock and et al, editors, *Quantitative Sociology: Internnational perspectives on mathematical and statistical model building*, pages 307–357. Academic Press, 1975.

[73] Risheng Liu, Zhouchen Lin, Zhixun Su, and Kewei Tang. Feature extraction by learning lorentzian metric tensor and its extensions. *Pattern Recognition*, 43(10): 3298–3306, 2010.

[74] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Access Online via Elsevier, 1990.

[75] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. Graph embedding and extensions: a general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, 2007.

[76] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. *Neural Networks, IEEE Transactions on*, 17(1): 157–165, 2006.

[77] X Niyogi. Locality preserving projections. In *Neural information processing systems*, volume 16, page 153, 2004.

[78] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(10):1713–1727, 2008.

[79] Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[80] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Kernel analysis on grassmann manifolds for action recognition. *Pattern Recognition Letters*, 2013.

[81] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[82] Kyle A Gallivan, Anuj Srivastava, Xiuwen Liu, and Paul Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *Statistical Signal Processing, 2003 IEEE Workshop on*, pages 315–318. IEEE, 2003.

[83] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE Int'l Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[84] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012.

[85] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[86] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[87] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[88] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[89] Xiaofeng Ren and Deva Ramanan. Histograms of sparse codes for object detection. *CVPR 2013*, 2013.

[90] John Wright, Allen Y Yang, Arvind Ganesh, Shankar S Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[91] Yan Zhu, Xu Zhao, Yun Fu, and Yuncai Liu. Sparse coding on local spatial-temporal volumes for human action recognition. In *Computer Vision–ACCV 2010*, pages 660–671. Springer, 2011.

[92] Xiaojing Zhang, Hua Zhang, and Xiaochun Cao. Action recognition based on spatial-temporal pyramid sparse coding. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1455–1458, 2012.

[93] Tanaya Guha and Rabab K Ward. Learning sparse representations for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1576–1588, 2012.

[94] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[95] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008.

[96] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[97] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[98] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2006.

[99] Chenlei Guo and Liming Zhang. A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression. *Image Processing, IEEE Transactions on*, 19(1):185–198, 2010.

[100] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1): 171–177, 2010.

[101] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[102] Ren Xiaofeng and Liefeng Bo. Discriminatively trained sparse code gradients for contour detection. In *Advances in Neural Information Processing Systems*, pages 593–601, 2012.

[103] William R Dillon and Matthew Goldstein. *Multivariate analysis*. Wiley New York, 1984.

[104] Subhransu Maji, Alexander C Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[105] Guoying Zhao and Matti Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, 2007.

[106] Vili Kellokumpu, Guoying Zhao, and Matti Pietikäinen. Human activity recognition using a dynamic texture based method. In *BMVC*, pages 1–10, 2008.

[107] Riccardo Mattivi and Ling Shao. Human action recognition using lbp-top as sparse spatio-temporal feature descriptor. In *Computer Analysis of Images and Patterns*, pages 740–747. Springer, 2009.

[108] Lahav Yeffet and Lior Wolf. Local trinary patterns for human action recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 492–497. IEEE, 2009.

[109] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio. A biologically inspired system for action recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[110] Alireza Fathi and Greg Mori. Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[111] Matteo Bregonzio, Shaogang Gong, and Tao Xiang. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1948–1955. IEEE, 2009.

[112] Adriana Kovashka and Kristen Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2046–2053. IEEE, 2010.

[113] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1): 221–231, January 2013. ISSN 0162-8828.

[114] Quoc V Le, Will Y Zou, Serena Y Yeung, and Andrew Y Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pages 3361–3368. IEEE, 2011.

[115] Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic. Spatiotemporal salient points for visual recognition of human actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(3):710–719, 2005.

[116] Shu-Fai Wong and Roberto Cipolla. Extracting spatiotemporal interest points using global information. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[117] Ivan Laptev and Tony Lindeberg. Local descriptors for spatio-temporal recognition. *Spatial Coherence for Visual Motion Analysis*, pages 91–103, 2006.

[118] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, pages 357–360, 2007.

[119] L. Xia, C.C. Chen, and JK Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012.

[120] Antonio Vieira, Erickson Nascimento, Gabriel Oliveira, Zicheng Liu, and Mario Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 252–259, 2012.

[121] Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, page In Press, 2013.

[122] Leandro Miranda, Thales Vieira, Dimas Martinez, Thomas Lewiner, Antonio W. Vieira, and Mario F. M. Campos. Real-time gesture recognition from depth data through key poses learning and decision forests. *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, 0:268–275, 2012. ISSN 1530-1834. doi: http://doi.ieeecomputersociety.org/10.1109/SIBGRAPI.2012.44.

[123] Xiaodong Yang, Chenyang Zhang, and YingLi Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1057–1060, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1089-5. doi: 10.1145/2393347.2396382. URL http://doi.acm.org/10.1145/2393347.2396382.

[124] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012.

[125] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3d action recognition with random occupancy patterns. In *Computer Vision–ECCV 2012*, pages 872–885. Springer, 2012.

[126] Omar Oreifej, Zicheng Liu, and WA Redmond. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

[127] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *International Journal of Robotics Research (IJRR)*, page In Press, 2013.

[128] Bingbing Ni, Gang Wang, and Pierre Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1147–1153. IEEE, 2011.

[129] Yang Zhao, Zicheng Liu, Lu Yang, and Hong Cheng. Combing rgb and depth map features for human activity recognition. In *Signal Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, pages 1–4, Dec.

[130] Hao Zhang and Lynne E Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2044–2049. IEEE, 2011.

[131] Amir H Shabani, David A Clausi, and John S Zelek. Evaluation of local spatio-temporal salient feature detectors for human action recognition. In *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pages 468–475. IEEE, 2012.

[132] Y. Zhu, W.B. Chen, and G.D. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013.

[133] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, Alberto Del Bimbo, et al. Space-time pose representation for 3d human action recognition. In *ICIAP Workshop on Social Behaviour Analysis*, 2013.

[134] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[135] Eshed Ohn-Bar and Mohan M Trivedi. Joint angles similiarities and hog2 for action recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Computer Society Conference on*. IEEE, 2013.

[136] Lorenzo Seidenari, Vincenzo Varano, Stefano Berretti, Pietro Pala, and Alberto Del Bimbo. Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In *Proc. of CVPR Int. Workshop on Human Activity Understanding from 3D Data (HAU3D),2013*, 2013.

[137] Li Liu and Ling Shao. Learning discriminative representations from rgb-d video data. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI), 2013*, 2013.

[138] X. D. Yang and YL. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 14–19, 2012.

[139] X. D. Yang, CY. Zhang, and YL. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM Int'l Conf. on Multimedia*, pages 1057–1060. ACM, 2012.

[140] S. Sempena, N. U. Maulidevi, and PR Aryan. Human action recognition using dynamic time warping. In *Int'l Conf. on Electrical Engineering and Informatics (ICEEI)*, pages 1–5. IEEE, 2011.

[141] M. Reyes, G. Domínguez, and S. Escalera. Featureweighting in dynamic time-warping for gesture recognition in depth data. In *Computer Vision Workshops (ICCV Workshops)*, pages 1182–1188. IEEE, 2011.

[142] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[143] P. K. Atrey, M. A. Hossain, A. El S., and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379, 2010.

[144] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Fusion of face and speech data for person identity verification. *IEEE Trans. on Neural Networks*, 10(5):1065–1074, 1999.

[145] K. Chang, K. Bowyer, and P. Flynn. Face recognition using 2d and 3d facial data. In *ACM Workshop on Multimodal User Authentication*, pages 25–32. Citeseer, 2003.

[146] L. Xu, A. Krzyzak, and C. Y. Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Trans. on Systems, Man and Cybernetics*, 22(3):418–435, 1992.

[147] A. Ross and A.K. Jain. Information fusion in biometrics. *Pattern recognition letters*, 24(13):2115–2125, 2003.

[148] J. Kittler. Combining classifiers: A theoretical framework. *Pattern analysis and Applications*, 1(1):18–27, 1998.

[149] F. M. Alkoot and J Kittler. Experimental evaluation of expert fusion strategies. *Pattern Recognition Letters*, 20(11):1361–1369, 1999.

[150] L. I. Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(2):281–286, 2002.

[151] L. I Kuncheva, J. C Bezdek, and R. PW Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, 2001.

[152] T. Kobayashi and N. Otsu. Motion recognition using local auto-correlation of space–time gradients. *Pattern Recognition Letters*, 33(9):1188–1195, 2012.

[153] A. A Ross and R. Govindarajan. Feature level fusion of hand and face biometrics. In *Defense and Security*, pages 196–204. International Society for Optics and Photonics, 2005.

[154] D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

[155] G. Brown, A. Pocock, M.J. Zhao, and M. Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13:27–66, 2012.

[156] H Yang and J. Moody. Feature selection based on joint mutual information. In *Proc. of Int'l ICSC symposium on advances in intelligent data analysis*, pages 22–25. Citeseer, 1999.

[157] F. Fleuret. Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research*, 5:1531–1555, 2004.

[158] MC Da C. A. and M. Fairhurst. Analyzing the benefits of a novel multiagent approach in a multimodal biometrics identification task. *Systems Journal, IEEE*, 3(4):410–417, 2009.

[159] C. Y. Wang, Y. Z. Wang, and A.L. Yuille. An approach to pose-based action recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 915–922, 2013.

[160] I. Theodorakopoulos, D. Kastaniotis, G. Economou, and S. Fotopoulos. Pose-based human action recognition via sparse representation in dissimilarity space. *Journal of Visual Communication and Image Representation*, 2013.

[161] RH Belmaker and Galila Agam. Major depressive disorder. *New England Journal of Medicine*, 358(1):55–68, 2008.

[162] Ronald C Kessler, Patricia Berglund, Olga Demler, Robert Jin, Doreen Koretz, Kathleen R Merikangas, A John Rush, Ellen E Walters, and Philip S Wang. The epidemiology of major depressive disorder: results from the national comorbidity survey replication (ncs-r). *Jama*, 289(23):3095–3105, 2003.

[163] Sandra Salmans. *Depression: questions you have-answers you need*. Peoples Medical Society, 1995.

[164] Daniel J France, Richard G Shiavi, Stephen Silverman, Marilyn Silverman, and D Mitchell Wilkes. Acoustical properties of speech as indicators of depression and suicidal risk. *Biomedical Engineering, IEEE Transactions on*, 47(7):829–837, 2000.

[165] James C Mundt, Adam P Vogel, Douglas E Feltner, and William R Lenderking. Vocal acoustic biomarkers of depression severity and treatment response. *Biological psychiatry*, 72(7):580–587, 2012.

[166] James R Williamson, Thomas F Quatieri, Brian S Helfer, Rachelle Horwitz, Bea Yu, and Daryush D Mehta. Vocal biomarkers of depression based on motor incoordination. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 41–48. ACM, 2013.

[167] Hongying Meng, Di Huang, Heng Wang, Hongyu Yang, Mohammed AI-Shuraifi, and Yunhong Wang. Depression recognition based on dynamic facial and vocal expression features using partial least square regression. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 21–30, 2013.

[168] Nicholas Cummins, Jyoti Joshi, Abhinav Dhall, Vidhyasaharan Sethu, Roland Goecke, and Julien Epps. Diagnosis of depression by behavioural signals: a multimodal approach. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 11–20, 2013.

[169] Ying Yang, Catherine Fairbairn, and Jeffrey F Cohn. Detecting depression severity from vocal prosody. *Affective Computing, IEEE Transactions on*, 4(2):142–150, 2013.

[170] Jeffrey F Cohn, Tomas Simon Kruez, Iain Matthews, Ying Yang, Minh Hoai Nguyen, Margara Tejera Padilla, Feng Zhou, and Fernando De La Torre. Detecting depression from facial actions and vocal prosody. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–7. IEEE, 2009.

[171] IVOR H JONES and MONIKA PANSA. Some nonverbal aspects of depression and schizophrenia occurring during the interview. *The Journal of nervous and mental disease*, 167(7):402–409, 1979.

[172] Heiner Ellgring. *Non-verbal communication in depression*. Cambridge University Press, 2007.

[173] Ray L Birdwhistell. Toward analyzing american movement. *Nonverbal communication*, pages 134–143, 1974.

[174] Jeffrey M Girard, Jeffrey F Cohn, Mohammad H Mahoor, S Mohammad Mavadati, Zakia Hammal, and Dean P Rosenwald. Nonverbal social withdrawal in depression: Evidence from manual and automatic analyses. *Image and vision computing*, 32 (10):641–647, 2014.

[175] Niall Firth. Computers diagnose depression from our body language. *New Scientist*, 217(2910):18–19, 2013.

[176] Sharifa Alghowinem, Roland Goecke, Michael Wagner, Gordon Parker, and Michael Breakspear. Eye movement analysis for depression detection. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 4220–4224. IEEE, 2013.

[177] Aaron T Beck, Robert A Steer, Roberta Ball, and William F Ranieri. Comparison of beck depression inventories-ia and-ii in psychiatric outpatients. *Journal of personality assessment*, 67(3):588–597, 1996.

[178] Michel Valstar, Björn Schuller, Kirsty Smith, Florian Eyben, Bihan Jiang, Sanjay Bilakhia, Sebastian Schnieder, Roddy Cowie, and Maja Pantic. Avec 2013: the continuous audio/visual emotion and depression recognition challenge. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 3–10. ACM, 2013.

[179] Michel Valstar, Björn Schuller, Kirsty Smith, Timur Almaev, Florian Eyben, Jarek Krajewski, Roddy Cowie, and Maja Pantic. Avec 2014: 3d dimensional affect and depression recognition challenge. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM, 2014.

[180] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[181] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408, 2007.

[182] Hongying Meng and Nick Pears. Descriptive temporal template features for visual motion recognition. *Pattern Recognition Letters*, 30(12):1049–1058, 2009.

[183] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[184] Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.

[185] Lingyun Wen, Xin Li, Guodong Guo, and Yu Zhu. Automated depression diagnosis based on facial dynamic analysis and sparse coding. *Information Forensics and Security, IEEE Transactions on*, 10(7):1432–1441, 2015.

[186] Timur R Almaev and Michel F Valstar. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 356–361, 2013.

[187] Humberto Pérez Espinosa, Hugo Jair Escalante, Luis Villaseñor-Pineda, Manuel Montes-y Gómez, David Pinto-Avedaño, and Veronica Reyez-Meza. Fusing affective dimensions and audio-visual features from segmented video for depression recognition. In *Proceedings of the ACM 4th International Workshop on Audio/Visual Emotion Challenge*, pages 49–55, 2014.

[188] Asim Jan, Hongying Meng, Yona Falinie A Gaus, Fan Zhang, and Saeed Turabzadeh. Automatic depression scale prediction using facial expression dynamics and regression. In *Proceedings of the ACM 4th International Workshop on Audio/Visual Emotion Challenge*, pages 73–80, 2014.

[189] Heysem Kaya, Fazilet Çilli, and Albert Ali Salah. Ensemble cca for continuous emotion prediction. In *Proceedings of the ACM 4th International Workshop on Audio/Visual Emotion Challenge*, pages 19–26, 2014.

[190] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going

deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[191] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

[192] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[193] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint: 1411.7923*, 2014.

[194] Maxim Sidorov and Wolfgang Minker. Emotion recognition and depression diagnosis by acoustic and visual features: A multimodal approach. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 81–86. ACM, 2014.