8-2014

3D Robotic Sensing of People: Human Perception, Representation and Activity Recognition

Hao Zhang
University of Tennessee - Knoxville, hzhang29@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Hao Zhang entitled "3D Robotic Sensing of People: Human Perception, Representation and Activity Recognition." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Lynne E. Parker, Major Professor

We have read this dissertation and recommend its acceptance:

Michael W. Berry, Husheng Li, Wenjun Zhou

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

8-2014

3D Robotic Sensing of People: Human Perception, Representation and Activity Recognition

Hao Zhang
University of Tennessee - Knoxville, hzhang29@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Hao Zhang entitled "3D Robotic Sensing of People: Human Perception, Representation and Activity Recognition." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Science.

Lynne E. Parker, Major Professor

We have read this dissertation and recommend its acceptance:

Michael W. Berry, Husheng Li, Wenjun Zhou

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

3D Robotic Sensing of People: Human Perception, Representation and Activity Recognition

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Hao Zhang

August 2014

I would like to make a special dedication to my mother, Guicai Guo. She passed away February 27, 2012. My mother was the most inspiring and strong women I ever knew. She loved me unconditionally, raised me, took best care of me, and always encouraged me. Thanks mom – forever, you remain in my soul.

I also would like to dedicate this dissertation to other family members: my father, Zhichen Zhang, for his unconditional love, patience, and continued encouragement and support; my beautiful wife, Xiaolan Chen, for her unwavering love during our six years of marriage; my parents-in-law, Xinli Chen and Jinmei Gao, for their unconditional support, especially the help they provided during their visit; and my little boy, Samuel, who has brought endless happiness and joy to the entire family.

Acknowledgements

There are many people to whom I own a deep debt of gratitude for all the help I have received from them during this journey.

First of all, I wish to thank my advisor, Dr. Lynne E. Parker, who has inspired, challenged, advised, and supported me, and provided me a comprehensive training that has made me very well-prepared for an academic career.

I also thank Dr. Wenjun Zhou for her special insights that helped me throughout my Ph.D. research. In addition, I would like to thank my other committee members, Dr. Michael W. Berry and Dr. Husheng Li, for their time and expertise in serving on my committee and all their help during my job search process.

Last, but certainly not least, I would like to express my thanks to my fellow lab members in Distributed Intelligence Laboratory and all my friends in Knoxville.

Abstract

The robots are coming. Their presence will eventually bridge the digital-physical gap and dramatically impact human life by taking over tasks where our current society has shortcomings (e.g., search and rescue, elderly care, and child education). Human-centered robotics (HCR) is a vision to address how robots can coexist with humans and help people live safer, simpler and more independent lives.

As humans, we have a remarkable ability to perceive the world around us, perceive people, and interpret their behaviors. Endowing robots with these critical capabilities in highly dynamic human social environments is a significant but very challenging problem in practical human-centered robotics applications.

This research focuses on robotic sensing of people, that is, how robots can perceive and represent humans and understand their behaviors, primarily through 3D robotic vision. In this dissertation, I begin with a broad perspective on human-centered robotics by discussing its real-world applications and significant challenges. Then, I will introduce a real-time perception system, based on the novel concept of Depth of Interest, to detect and track multiple individuals using a color-depth camera that is installed on intelligent mobile robotic platforms. In addition, I will discuss human representation approaches, based on local spatio-temporal features, including new CoDe4D features that incorporate both color and depth information, a new SOD descriptor to efficiently quantize 3D visual features, and the novel AdHuC features, which are capable of representing the activities of multiple individuals. Several new algorithms to recognize human activities are also discussed, including the RG-PLSA model, which allows us to discover activity patterns without supervision, the MC-HCRF model, which can explicitly investigate certainty in latent temporal patterns, and the FuzzySR model, which is used to segment continuous data into events and probabilistically recognize human activities. Cognition models based on recognition results are also implemented for decision making that allow robotic systems to react to human activities. Finally, I will conclude with a discussion of future directions that will accelerate the upcoming technological revolution of human-centered robotics.

Table of Contents

List of Tables

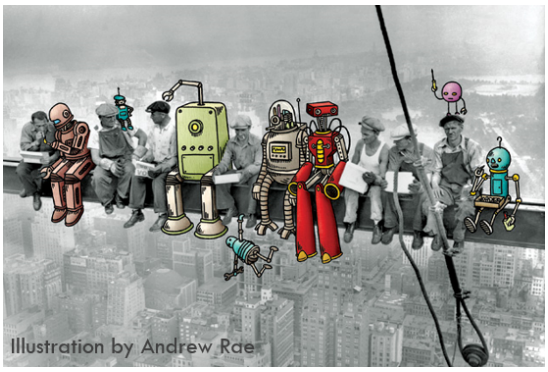
List of Figures

Chapter 1

Introduction

The central motivating theme of my Ph.D. research is human-centered robotics: *the use of robotic systems to help people live safer, simpler and more independent lives in human social environments.* Far beyond the boundary of traditional robotics research focusing on industrial applications that often manipulate objects in dirty, dull, or dangerous tasks, human-centered robotics is a vision to address how robots can live among us and take over tasks where our current society has shortcomings. Figure 1.1 provides an intuitive example of the difference between traditional industrial robotics and human-centered robotics. As illustrated in Figure 1.2, human-centered robotics has a wide variety of practical applications: elder and disabled care, child education, physical therapy, general assistance in daily life, entertainment, and search and rescue in disasters are some of the examples that will benefit from human-centered robotics in the near future.

(a) Traditional Industrial Robotics



(b) Human Centered Robotics

Figure 1.1: An illustration of the difference between industrial robotics and human-centered robotics.

At the core of human-centered robotics is perceiving people and understanding their behaviors to allow effective social interactions. Without the critical capability of perceiving and understanding people, no robotic systems can effectively interact

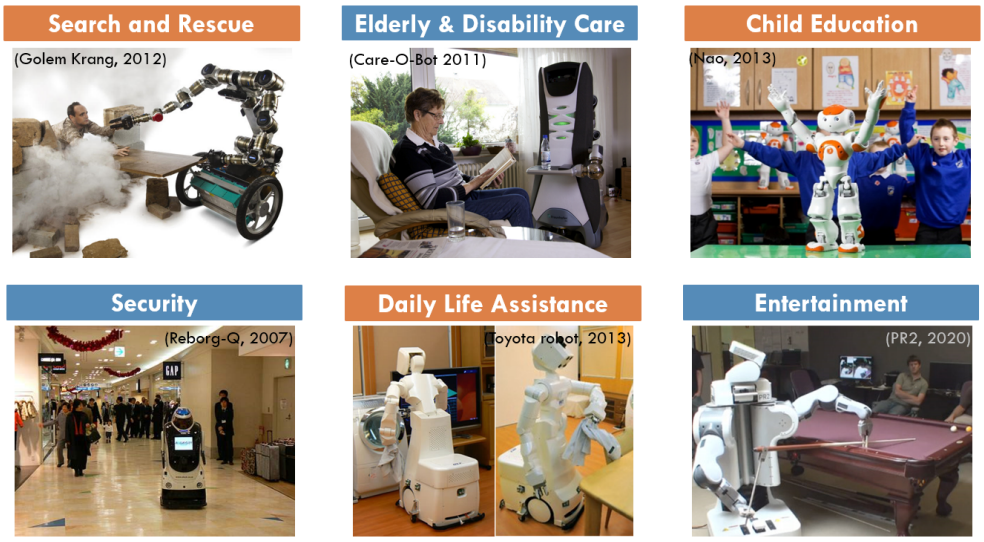


Figure 1.2: Real-world applications of human-centered robotics.

with humans. Although a large number of approaches have been proposed to perceive and represent humans and recognize their activities using color cameras, they do not make full use of one important piece of information that is now available—depth. Thanks to the emergence of affordable commercial color-depth cameras, such as the Microsoft Kinect and Asus Xtion Pro LIVE RGB-D cameras, it is now much faster, easier and cheaper to deploy a 3D vision system on a robot. Since humans act in 3D space, depth can be utilized along with color information to develop a more reliable and robust human perception, representation, and activity recognition system.

1.1 Problem Statement

This dissertation focuses on the research problem of *human perception, representation and activity recognition* in 3D space in complex real-world human social environments, mainly using color-depth cameras installed on intelligent mobile robots. An overview of the research topics discussed in this dissertation is graphically presented in Figure 1.3. The objective of human perception is to detect and track multiple humans who share the same workspace with the robotic system. Human representation addresses the problem of extracting low-level features from raw, noisy visual data to represent and encode humans and their activities in a compact fashion. The objective of human activity understanding is to recognize human activities*, especially from continuous visual data. My work generally focuses on using RGB-D data (e.g., from red, green, blue and depth channels) acquired from color-depth cameras, which are installed on intelligent robotic systems as shown in Figure 1.4. The ultimate objective of my Ph.D.

*In this dissertation, the terms *action* and *activity* are used interchangeably.

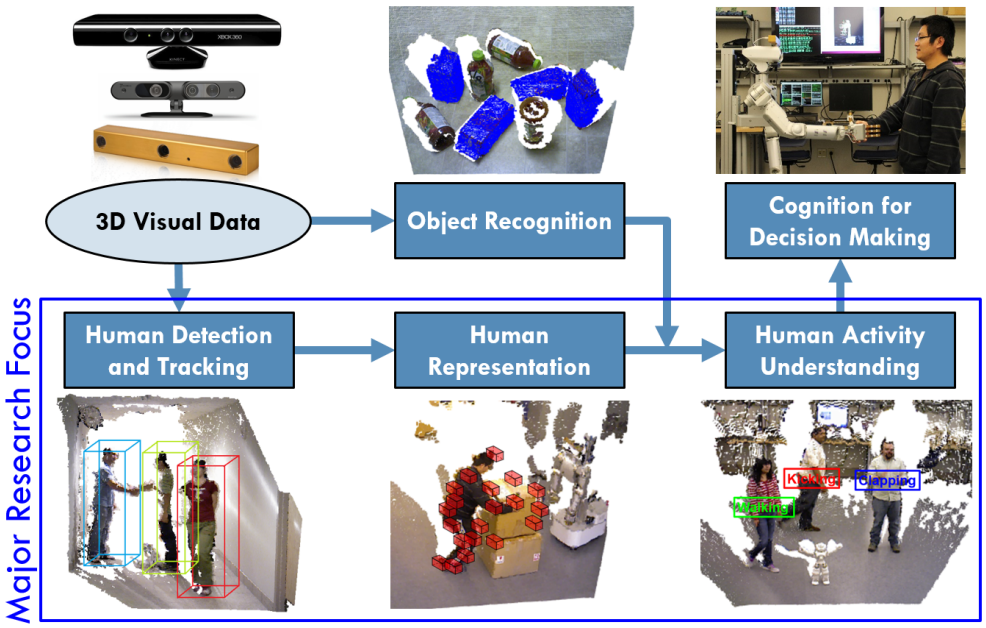


Figure 1.3: Overview of research topics.

work is to allow intelligent robotic systems to effectively and efficiently interact and cooperate with people in 3D space in practical human-centered robotic applications.

1.2 Challenges

As humans, we have a remarkable ability to perceive the world around us, localize people, and interpret their activities and even intentions. However, because human social environments are highly dynamic in nature, a variety of challenges to human-centered robotic systems can arise; as a result, it is a most difficult problem to endow intelligent robots with the critical capabilities of human perception, representation and activity recognition. Specifically, major challenges are discussed as follows:

- Complexity of human appearance, motion, and interaction: Human appearance can vary significantly, since humans can be a wide range of sizes, change poses, wear different clothes, and face arbitrary directions. Human can also move using different speeds and poses even when performing the same activity. In addition, humans often interact with objects and other people.

- Robot movement and dynamic background: 3D sensing of people with moving robotic systems introduces additional challenges. First, camera movement can result in a dynamic background, for which traditional motion or segmentation-based methods [Viola et al., 2005, Leibe et al., 2005] are no longer applicable. Second, a moving robot results in frequent changes in viewing angles of humans

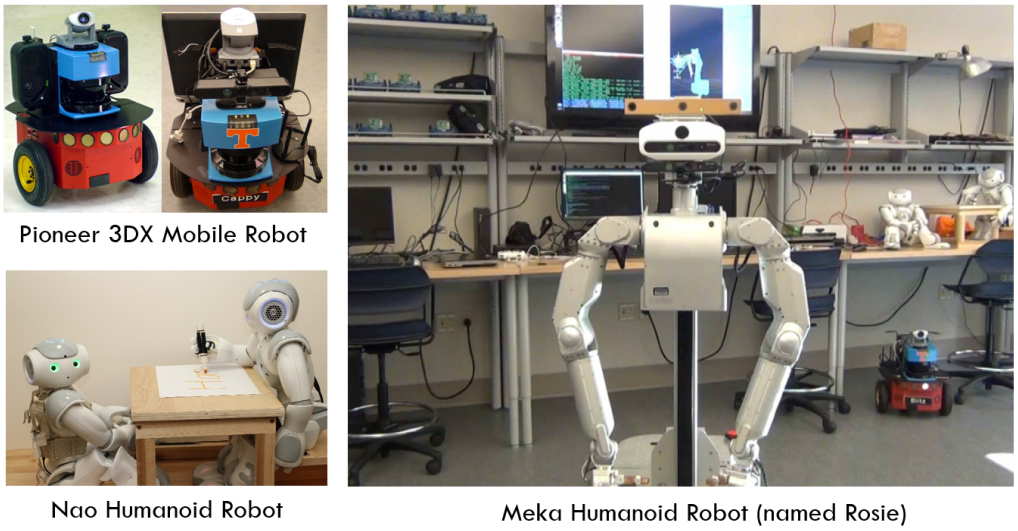


Figure 1.4: Intelligent robotic systems used in my human-centered robotics research.

(e.g., front, lateral and rear positions), and causes camera oscillations that can introduce additional noise.

- Time and safety constraints: Implementation of these critical sensing techniques on robotic systems adds additional spatial and temporal constraints. A moving robotic system needs to move safely and avoid collision with humans and other objects. A robot also needs to perceive humans, interpret their activities and react to human movements as quickly and safely as possible. Most importantly, all used sensing techniques must work live on real robotic systems with a limited computer with computational constraints.

- Traditional vision challenges: A human can be completely or partially occluded by objects, other humans, and even him or herself (i.e., self-occlusion). Linear perspective view changes (i.e., an object staying closer to a camera looks larger) always exist and need to be addressed. Finally, illumination variations usually occur in real-world applications.

1.3 Main Contributions

The contributions I have made to address the problems of human perception, human representation, and activity recognition are summarized as follows:

Human Perception

- I proposed the novel concept, called Depth of Interest, which is used to identify humans in 3D point cloud sequences and avoid the computationally expensive sliding window paradigm of previous approaches. Based on this concept, I also

made a practical contribution; that is the development of a real-time multiple human detection and tracking system, which is able to deal with the challenges of occlusion, robot movements, human-object interactions, etc. (Chapter 3)

Human Representation

- I introduced the first color-depth feature to represent humans in 4D space (3D spatial and 1D temporal dimensions), named the CoDe4D (i.e., 4-dimensional color-depth) feature, which combines both color and depth cues contained in a sequence of 3D point clouds that are acquired from a color-depth camera, such as Microsoft Kinect. (Chapter 4)

- I proposed the novel simplex-based orientation decomposition (SOD) algorithm to describe 3D local spatio-temporal features for human activity recognition. This descriptor avoids the singularity and limited discrimination power problems of traditional 3D descriptors, by quantizing and describing visual features in the simplex topological vector space. (Chapter 5)

- I implemented the new adaptive human-centered (AdHuC) feature to address the problem of multiple-individual activity recognition. This feature is able to identify feature affiliations, avoid extracting irrelevant features from dynamic backgrounds (e.g., caused by robot movement), and also compensate for linear perspective view changes. (Chapter 6)

Human Activity Recognition

- I implemented a new topic model, named RG-PLSA, which employs Gaussian Mixture Model (GMM) to regularize Probabilistic Latent Semantic Analysis (PLSA). In addition, an online expectation-maximization (EM) algorithm was developed to efficiently estimate model parameters in an incremental fashion. This model addresses the overfitting problem (as in original PLSA) and avoids the computationally expensive Bayesian learning (as used by Latent Dirichlet Allocation), which is essential for online robotic learning under computational constraints. (Chapter 7)

- I introduced the Maximum Certainty Hidden Conditional Random Field (MC-HCRF), a first discriminative probabilistic graphical model that explicitly models certainty in the latent temporal pattern of sequential data. I mathematically proved that inference of the model is tractable. This model achieves the state-of-the-art performance on recognition of sequential and reversal activities, such as sitting down and standing up. (Chapter 8)

- I proposed the idea of modeling human activity events in continuous visual data as fuzzy sets with fuzzy start and end time points. Then, I implemented this idea to construct a fuzzy temporal segmentation and probabilistic recognition

system (called FuzzySR), which is the first study to model gradual transitions between continuous human activities. (Chapter 9)

- I introduced two performance metrics to evaluate topic modeling, including the interpretability indicator, which is proposed to measure how topic modeling's results match human perspective, and the generalizability indicator, which is used to assess how topic models generalize over previously unseen observations. These indicators were applied to evaluate a most widely used topic model (i.e., Latent Dirichlet Allocation) and to construct reliable artificial cognitive system in human-centered robotics applications. (Chapter 10)

It is noteworthy that the approaches and contributions to address the problem of human perception are based on local spatio-temporal features, while the techniques proposed to recognize human activities are generally based on graphical models.

1.4 Guide to the Dissertation

The remainder of this dissertation is structured as follows. Chapter 2 overviews the benchmark datasets used in the experiments to evaluate our algorithms' performance. Chapter 3 discusses the real-time multiple individual detection and tracking system from 3D visual data. Chapter 4–6 focus on the problem of human representation. Specifically, Chapter 4 describes the CoDe4D features; Chapter 5 discusses the novel SOD feature descriptor; and Chapter 6 presents the AdHuC features. After discussing the new algorithms to recognize human activities, including RG-PLSA in Chapter 7, MC-HCRF in Chapter 8 and FuzzySR in Chapter 9, Chapter 10 discusses how to use human perception techniques to construct an artificial cognitive system for decision making. Finally, I conclude this dissertation and point out potential future applications in Chapter 11. In order to provide specific comparisons of the introduced algorithms with existing techniques, previous works relating to different topics are reviewed in their respective chapters.

Chapter 2

Datasets

This chapter provides a comprehensive overview of the benchmark datasets used for evaluating human representation and activity recognition approaches. Two categories of datasets are employed in the experiments: datasets that contain traditional color videos and datasets that are collected using RGB-D cameras. The objective of using benchmark datasets is for fair comparisons with existing results reported in previous works. Because this dissertation focuses on human-centered robotics applications in human-social environments, the datasets used in the experiments generally consist of human daily activities. In the remainder of this chapter, the color and 3D (e.g., RGB-D) human activity datasets are reviewed in Section 2.1 and Section 2.2, respectively.

2.1 Color Human Activity Datasets

2.1.1 Weizmann Dataset

The Weizmann dataset [Blank et al., 2005] contains 93 segmented video clips with a resolution of 180×144 and is captured using a frame rate of 25 frames per second (FPS). This dataset is recorded using a static camera in an outdoor environment with a simple background. The dataset contains ten categories of human activities performed by nine individuals. The activities include: walking, running, jumping, siding, bending, one-hand waving, two-hands waving, jumping in place, jacking, and skipping. Representative frames showing these activities are depicted in Figure 2.1.

2.1.2 KTH Dataset

The KTH dataset [Schuldt et al., 2004] contains 600 color video sequences that were captured at 25 frames per second (FPS) with a resolution of 160×120. All videos are recorded using a static camera in a simple environment with homogeneous backgrounds. This dataset contains six human activities: walking, jogging, running, boxing, hand waving, and hand clapping. Each activity is performed by 25 human subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors

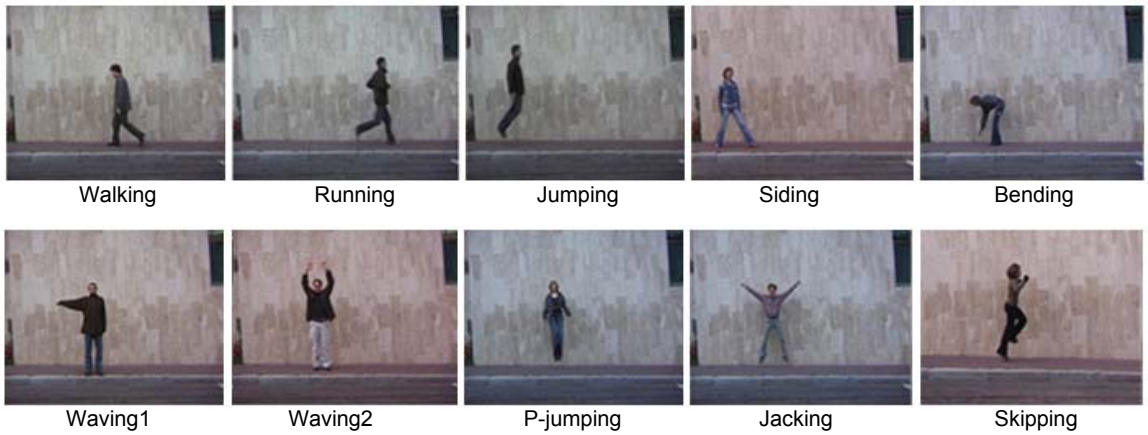


Figure 2.1: Exemplary frames of different human activities in the Weizmann dataset.

with different clothes, and indoors. Representative frames of each activity are depicted in Figure 2.2.

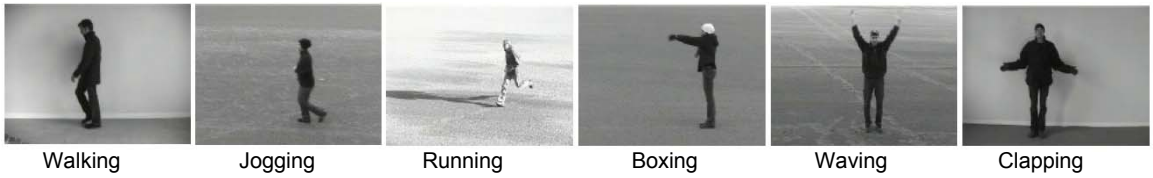


Figure 2.2: Representative frames of activities in the KTH dataset.

2.1.3 Hollywood-2 Dataset

The Hollywood-2 dataset [Marszalek et al., 2009] is collected from 69 different Hollywood movies, which contains twelve human daily activities, including answering phone, driving car, eating, fighting person, getting out of car, hand shaking, hugging person, kissing, running, sitting down, sitting up, and standing up. This dataset contains unconstrained activities from realistic scenes, which contains significant challenges including occlusion, camera movement, and lightening changes; different instances of each activity are often viewed from different camera angles. Exemplary frames of the human activity categories in the Hollywood-2 dataset are illustrated in Figure 2.3.

2.1.4 UCF Sports Dataset

The UCF Sports dataset [Rodriguez et al., 2008] contains ten different types of human activities: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf

Figure 2.3: Examples of human daily activities in the Hollywood-2 dataset, which contains significant challenges including view point changes, severe partial occlusions, etc.

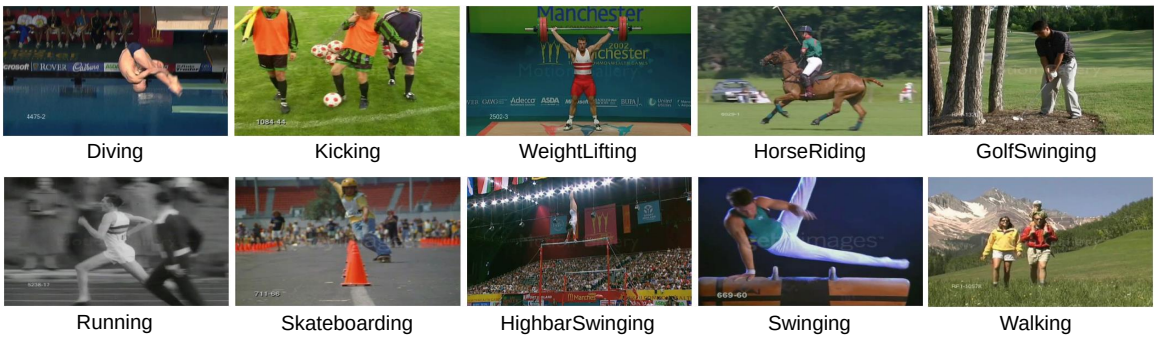


Figure 2.4: Examples of human activities in the UCF Sports dataset.

swinging and walking. The dataset consists of 150 video samples which show a large intra-class variability. Exemplary frames are illustrated in Figure 2.4.

2.2 3D Activity Datasets

2.2.1 Berkeley MHAD

The Berkeley MHAD dataset [Ofli et al., 2013] is a multi-modal human activity dataset that contains 11 activities performed by 7 male and 5 female subjects. Each activity was repeated 5 times, yielding around 550 activity video sequences. We employ the front-view Kinect data to evaluate our features in this work, which were captured with a resolution of 640×480 at a frame rate of 30 frames-per-second. Figure 2.5 shows snapshots of the activities in the Berkeley MHAD dataset.

2.2.2 ACT4^2

The ACT4^2 dataset [Cheng et al., 2012] is a large-scale multi-Kinect human activity dataset that contains 14 activities performed by 24 subjects in 6844 color-depth

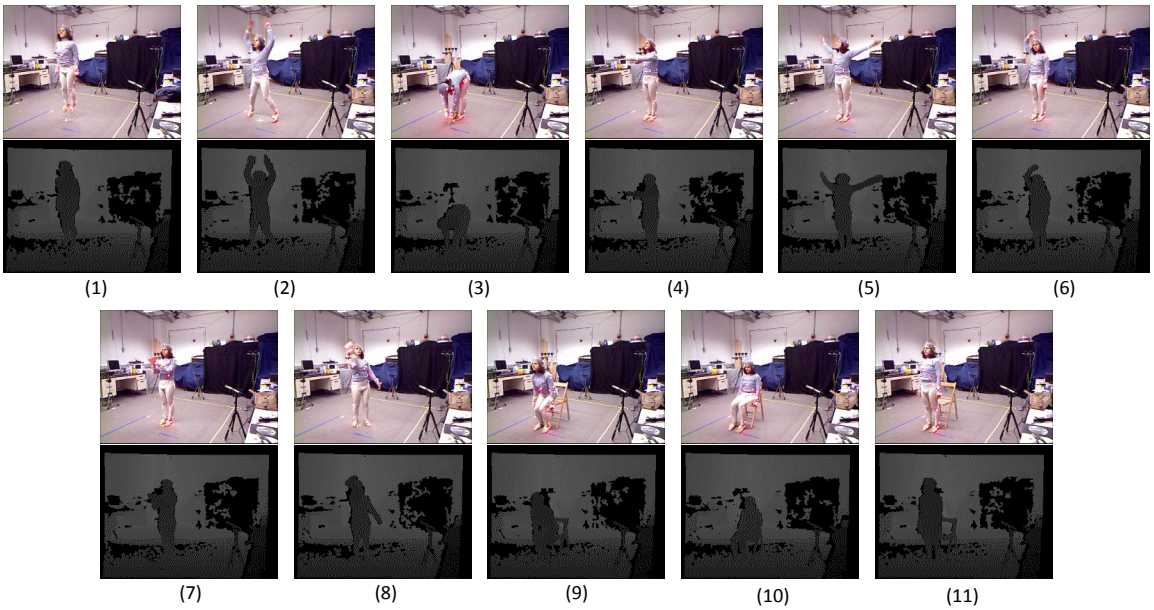


Figure 2.5: The Berkeley MHAD dataset contains eleven activities: (1) jumping in place, (2) jumping jacks, (3) bending, (4) punching, (5) two-hand waving, (6) one-hand waving, (7) clapping hands, (8) throwing a ball, (9) sitting down then standing up, (10) sitting down, and (11) standing up.

sequences. This RGB-D dataset was collected in a typical living room environment and has a focus on human daily activities. The color-depth dataset obtained from camera 4 is used, which shows side views of the human activities. The dataset was captured using a Kinect sensor with a resolution of 640×480 and a frame rate of 30 frames-per-second. Examples of each daily activity from the dataset are depicted in Figure 2.6.

2.2.3 MSR Daily Activity 3D

The MSR Daily Activity 3D dataset [Wang et al., 2012b] contains 16 human activities performed by 10 subjects in 320 color-depth sequences. Each subject performs each activity twice in a standing or sitting position in typical office environments. The color frames have a resolution of 640×320, while their respective depth frames have a 320×160 resolution. Exemplary color and depth frames of each daily activity from the dataset are depicted in Figure 2.7.

2.2.4 UTK Action3D

This dataset [Zhang and Parker, 2011] is an earliest RGB-D human activity dataset that is publicly available. This dataset contains six activities including sequential activities, repetitive activities and activities with small movements. Each activity

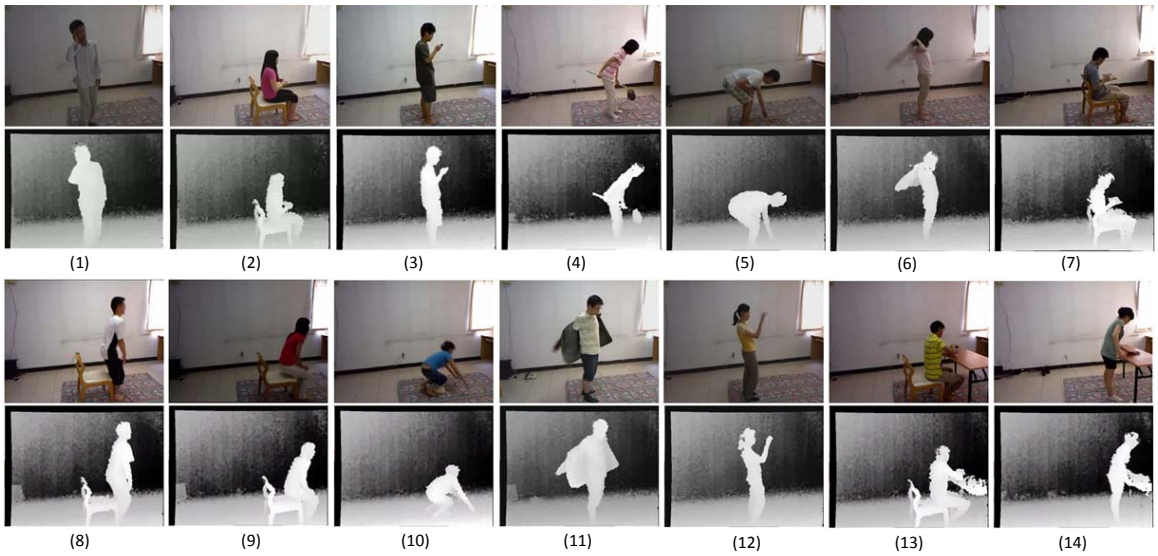


Figure 2.6: The ACT4^2 dataset contains fourteen human activities: (1) collapsing, (2) drinking, (3) making phone calls, (4) mopping floor, (5) picking up, (6) putting on, (7) reading book, (8) sitting down, (9) sitting up, (10) stumbling, (11) taking off, (12) throwing away, (13) twisting open, and (14) wiping clean.

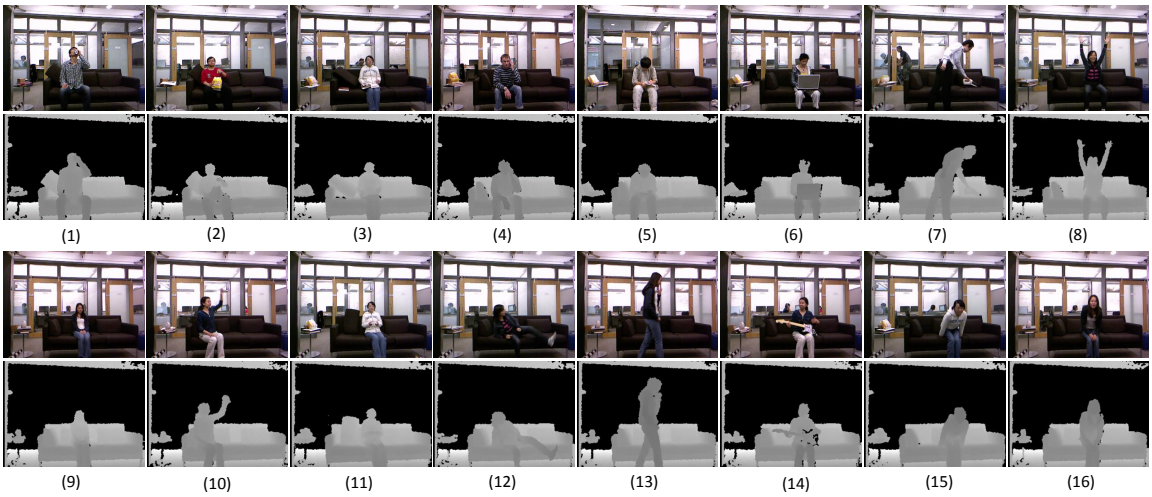


Figure 2.7: The MSR Daily Activity 3D dataset contains sixteen human activities: (1) drink, (2) eat, (3) read book, (4) call cellphone, (5) write on a paper, (6) use laptop, (7) use vacuum cleaner, (8) cheer up, (9) sit still, (10) toss paper, (11) play game, (12) lie down on sofa, (13) walk, (14) play guitar, (15) stand up, (16) sit down.

category contains 33 instances. Each instance consists of a color video and a calibrated depth video. This RGB-D activity dataset was collected using a Kinect sensor that is installed on a Pioneer 3DX mobile robot in human social environments such as office and home. The UTK Action3D dataset contains a wide variety of challenges including illumination changes, dynamic background and variations in human appearances and motions. Exemplary frames of the dataset are illustrated in Figure 2.8.

Figure 2.8: The UTK Action3D dataset contains six human activities: (1) lifting box, (2) removing box, (3) waving, (4) pushing box, (5) walking, and (6) signaling.

2.2.5 UTK-CAP Dataset

The UTK-CAP dataset [Zhang et al., 2014a] is collected by a Kinect color-depth camera that is installed on a Pioneer 3DX mobile robot. The dataset contains five color-depth videos. Each video has a duration of around 15 minutes and is recorded at a frame rate 15 Hz with a resolution of 640×480. Each video contains a sequence of continuous human activities that are performed in a natural way in 3D space. The dataset is collected in the scenario of a small gift store, in which a human actor plays a role of the store owner and performs a sequence of activities related to customer service. An autonomous robot is used to operate in the same environment to help the human improve productivity. The tasks that the store owner needs to accomplish include posting information and receiving messages on the internet, answering phone calls from customers and suppliers, writing inventory information on a white board, and preparing packages for customers. In this scenario, six activity categories are designed, as illustrated in Figure 2.9: (1) Grab box: grab an empty box from the storage area on the right side and bring it to the packing area; (2) Pack box: put required items into the box in the packing area in the center; (3) Push box: push the packed box from the packing area to the delivery area in the far left corner; (4) Use computer: operate a computer in the center area; (5) Write on board: write notes on a board on the right side; and (6) Answer phone: answer phone calls on the left side.

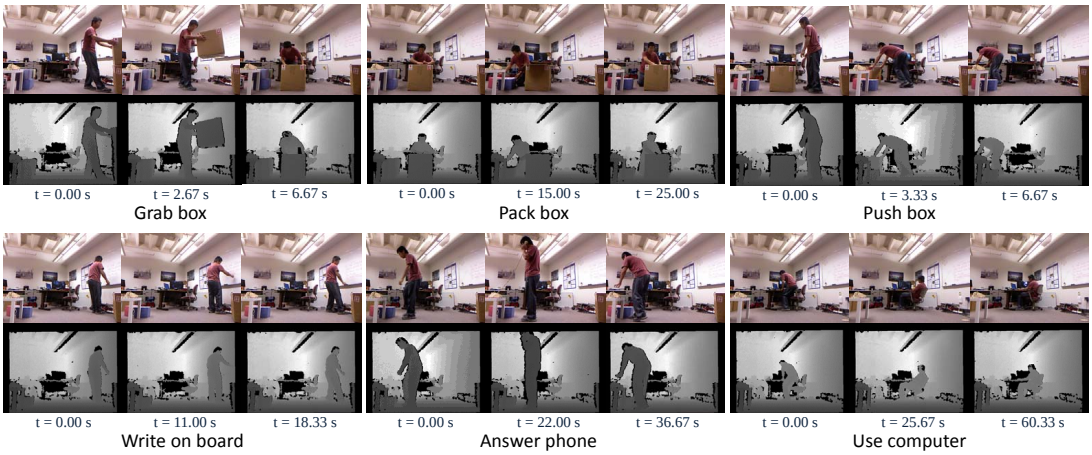


Figure 2.9: Typical sequences of the continuous human activities in the UTK-CAP dataset. Execution time is labeled under each frame to emphasize the difference in activity durations. In contrast to previous datasets, gradual transitions exist between temporally adjacent activities in our dataset.

2.2.6 UTK-ARMI Dataset

We collect a new 3D dataset for action recognition of multiple individuals (ARMI) in a group, with the objective of allowing researchers to benchmark new recognition algorithms that address the ARMI problem. To the best of our knowledge, this is the first color-depth dataset that contains different actions performed simultaneously by multiple individuals in the same scene.

Our ARMI dataset is captured using a PrimeSense Carmine 1.08 RGB-D camera that is installed on a Meka humanoid robot, as depicted in Figure 2.10a. The color-depth camera has a field of view $57.5 \times 45.0 \times 69.0$ degrees (horizontal, vertical and diagonal), a workable range of 0.5 m to 8.0 m, a xy spatial resolution of 3.4 mm at 2.0 m, and a depth resolution of 1.2 cm at 2.0 m. Each instance of our ARMI dataset is captured as a sequence of 3D point clouds that are saved as PCD [Rusu and Cousins, 2011] files with a frame rate of 30 frames per second (FPS). Each frame of an instance is a 3D point cloud that contains 307,200 points; each point has six values: its coordinates in 3D space and RGB values. In addition to the PCD format, we also provide separate color and depth videos of each instance saved as AVI files with a resolution of 640×480 at 30 Hz. The color and depth videos are converted from the instance's 3D point cloud, which allow for easy viewing and manipulation of the dataset. It is also noteworthy that compared with the color-depth videos, the point cloud format contains additional information, including the spatial location of each point in 3D space, viewpoint information, etc.

The ARMI dataset contains three subjects performing six actions: bend, clap, flap, kick, walk, and wave. The dataset contains 522 instances. Each data instance contains two or three subjects simultaneously performing different actions and has

(a) Meka robot **(b)** Flap and walk actions **(c)** Bend and wave actions **(d)** Kick and clap actions

Figure 2.10: Our ARMI dataset is acquired by a PrimeSense Carmine 1.08 color-depth camera that is installed in the sensor head of our Meka humanoid robot, as shown by the topmost camera on the Meka robot in Figure 2.10a. Our ARMI dataset contains six human actions that are performed simultaneously by two or three subjects in the same scene, as shown in Figure 2.10b, 2.10c and 2.10d.

a duration of around 2–3 seconds. The ground truth of each instance is manually labeled. Illustrative frames of each action are provided in Figures 2.10b, 2.10c and 2.10d, which depict both 3D point cloud and color-depth formats.

Chapter 3

Perception: Multi-Individual Detection and Tracking

3.1 Introduction

Efficient and robust detection and tracking of humans in complicated environments is an important challenge in human-centered robotics applications. For example, in human-robot teams [Loper et al., 2009], people can perform key functions; therefore, endowing robots with the ability to detect and track humans is critical to safe operation and efficient robot cooperation with humans. In this chapter, we address the task of human detection and tracking in complex, dynamic, indoor environments and in realistic and diverse settings, using a color-depth camera on a mobile robot.

Although a large number of sophisticated methods have been proposed to detect and track humans using color cameras [Dollár et al., 2012], they do not make use of one important piece of information that is now available—depth. Since humans act in the 3D space, depth can be utilized along with color information to develop a more reliable and robust human perception system.

In this chapter, we introduce a new, real-time human perception system to detect and track multiple humans in dynamic indoor environments, using a mobile robot that is equipped with a color-depth camera. Our system creates a new interleaved tracking-by-detection framework. To improve detection performance, the new concept of Depth of Interest (DOI) is introduced that enables us to efficiently obtain a set of possible human candidates in 3D point clouds. Then, a cascade of detectors is used to reduce the candidate set by rejecting non-human objects. The remaining candidates are handled by a decision process using a directed acyclic graph (DAG) to further distinguish between humans and objects and maintain object detection and human tracking information. Detection and tracking are interleaved in the sense that the tracking model utilizes a fine detector to classify new objects and humans, while the depths of tracked humans are fed back to the detection module to better allocate DOIs.

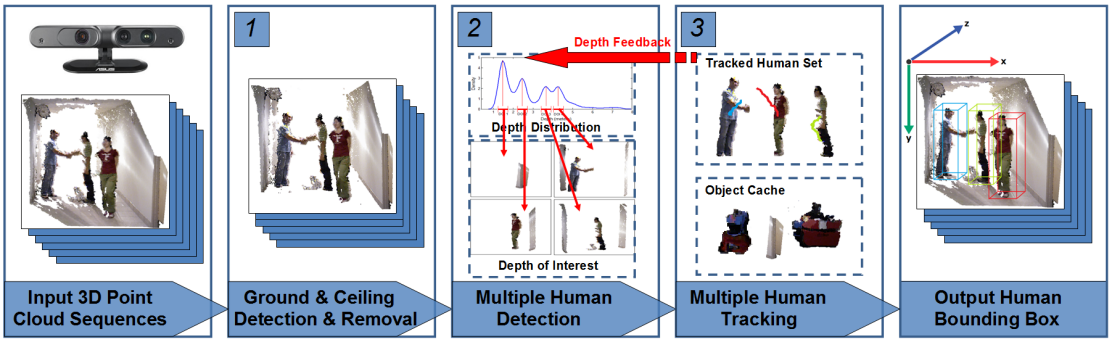


Figure 3.1: Description of the major procedures in our multiple human detection and tracking system. Starting with the input 3D point cloud sequences, the system: 1) identifies the ground and ceiling planes, and removes them from the point cloud, 2) employs DOIs along with a cascade of detectors to identify a set of candidates, 3) associates candidates with tracked individuals or detected objects, tracks humans, and feeds depth information back to guide candidate detection. Our system outputs tracking information for each human, such as a 3D bounding cube and the human's centroid.

System Overview

An algorithmic overview of our multiple human perception system is depicted in Figure 6.1, which clarifies our methodology by breaking it down into logical blocks. Our system takes 3D point clouds as input, which are acquired from a color-depth camera mounted on a robot, and outputs human tracking information. The major procedures for human perception are:

1. *Ground and ceiling plane detection and removal*: After the 3D cloud points are preprocessed, the ground and ceiling planes are detected based on a prior-knowledge guided plane fitting algorithm. Then, all points belonging to the planes are removed from the point cloud.

2. *Multiple human detection*: We first estimate the distribution of depth values in the point clouds and extract DOIs that are likely to contain humans but also may contain objects. Then, a set of candidates is identified by segmenting point clusters within each DOI. Finally, a cascade of detectors is applied to reject as many non-human candidates as possible.

3. *Multiple human tracking*: We use a decision DAG-based algorithm to efficiently handle the detected candidates. Candidate association with humans and non-human objects is achieved using a two-layer matching algorithm. Then, humans are tracked with extended Kalman filters. The depth values of tracked humans are also fed into the next detection step to guide candidate detection.

Contributions

Our system combines several novel and previously uncombined techniques to create a system that is capable of real-time tracking of multiple human targets and objects from a mobile robot. The contributions of this work include:

- The introduction of the new DOI concept for detecting humans in color-depth images, which allows us to avoid using the computationally expensive window scanning over the entire image and speeds up processing to help us achieve real-time performance.
- The new single-pass, decision DAG-based framework that incorporates human-object classification, data association, and tracking, which allows us to apply the most computationally expensive techniques only to the most difficult cases. This framework saves processing time and further makes our system perform in real time.
- The use of a detector cascade followed by the decision DAG over 3D point clouds provides an approach that explicitly addresses the previously unaddressed combination of human-human interaction, human-object interaction, humans assuming non-upright body configurations, and re-identification of tracked humans.

Together, our DOI concept, the use of a cascade of detectors, and our decision DAG-based framework allow us to construct a multiple human perception system that is robust to occlusion and illumination changes and operates in real time, on mobile platforms equipped with standard, consumer-grade computation capability and an RGB-D camera.

The remainder of this chapter is structured as follows. Section 10.2 overviews literature in the area of human detection and tracking. Section 3.3 introduces our approaches to ground/ceiling plane removal and detection and tracking of multiple humans in preprocessed 3D point clouds. Experimental results are presented in Section 7.5. Finally, Section 10.7 concludes the multi-human perception system.

3.2 Related Work

A large number of human detection and tracking methods have been proposed in the past few years. We begin with an overview of approaches using 2D cameras to detect humans in outdoor environments in Section 3.2.1. Then, Section 3.2.2 reviews previous work in human tracking. Finally, 3D-based human perception approaches are discussed in Section 3.2.3.

3.2.1 2D-Based Pedestrian Detection

Nearly all state-of-the-art human detectors are dependent on gradient-based features in some form. As a dense version of the SIFT [Lowe, 2004] features, Histogram of Oriented Gradients (HOG) was introduced by Dalal and Triggs [Dalal and Triggs, 2005] to perform whole body detection, which has been widely accepted as one of the most useful features to capture edge and local shape information [Dollár et al., 2012]. Other detectors to identify humans in 2D images include: 1) Shape-based detectors: Wu et al. [Wu and Nevatia, 2007] designed the edgelet features, which use a large set of short curve segments to represent local human shapes; 2) Part-based detectors: Bourdev et al. [Bourdev and Malik, 2009] developed poselet features, which employ a dictionary of local human parts with similar appearance and pose to represent pedestrians; 3) Motion-based detectors: Dalal et al. [Dalal et al., 2006] proposed the Histogram of Optical Flow (HOF) features that apply motions modeled by an optical flow field's internal differences to recognize moving pedestrians. Dollár et al. [Dollár et al., 2012] performed a thorough and detailed evaluation and comparison of these 2D-based detectors.

Pedestrian detectors generally assume that pedestrians are upright, which we do not require to be true in our application; we allow for humans to perform actions with a wide variety of body configurations. In addition, pedestrian detectors typically follow a sliding window paradigm, which applies dense multi-scale scanning over the entire image. This paradigm generally has a high computational complexity, and is therefore not suitable for the real-time requirement in our application. Our work addresses real-time human perception tasks using a color-depth camera on a moving platform in indoor environments with a complicated dynamic background.

3.2.2 Multiple Target Tracking

Many target tracking approaches [Lanz, 2006] from stationary cameras exist that are based on background subtraction [Stauffer and Grimson, 1999]. However, in applications with a moving camera, the tracking task becomes considerably harder, as it becomes extremely difficult to subtract the background reliably and efficiently. In these cases, tracking-by-detection appears to be a promising methodology to track multiple objects and is widely used by many state-of-the-art tracking systems [Breitenstein et al., 2011]. In the tracking-by-detection framework, objects are first detected independently in each frame. After per-frame detection is performed, data are associated across multiple temporal adjacent frames, and targets are typically tracked using classic tracking algorithms, including mean-shift tracking [Cheng, 1995] and dynamic Bayesian filters [Koller and Friedman, 2009], such as Kalman filters [Kalman, 1960] and particle filters [Arulampalam et al., 2002].

Several other approaches have also reported better tracking performance. Okuma et al. [Okuma et al., 2004] combined mixture particle filters with AdaBoost, and Cai et al. [Cai and Cai, 2006] further improved this method by applying independent particle sets to increase multiple tracking robustness. Zhang et al. [Zhang et al., 2008]

designed a graph-based formulation that allows an efficient global solution in complex situations. Ess et al. [Ess et al., 2009] developed a probabilistic graphical model to integrate different feature modules. To reduce drift, data association can be optimized by considering multiple possible associations over several time steps in multi-hypothesis tracking [Reid, 1979], or by finding best assignments in each time point to consider all possible associations in joint probabilistic data association filters [Fortmann et al., 1983]. Several recently proposed methods also explicitly deal with occlusions. Partial occlusion was addressed by a part-based model [Shu et al., 2012], and full occlusion was handled with approaches based on tracklet matching [Kaucic et al., 2005], visible and occluded part segmentation [Papadakis and Bugeau, 2011], or an explicit occlusion model [Zhang et al., 2008].

We introduce a new tracking-by-detection framework using a one-pass decision DAG, which is able to run in real time and address previously unaddressed issues, e.g., tracking occluded humans who are interacting with other humans or objects.

3.2.3 3D-Based Human Detection and Tracking

Several human detection and tracking approaches based on 3D sensing systems have also been discussed, which can be categorized in terms of depth sensing technologies: 1) 3D lasers: Spinello et al. [Spinello et al., 2010] suggested a pedestrian detection system using 3D laser range data that involves dividing a human into parts with different height levels and learning a classifier for each part; 2) Stereo cameras: A dense stereo vision system [Keller et al., 2011] was designed to detect pedestrians using HOG features and Support Vector Machine (SVM) classifiers, and a different system was suggested in [Muñoz Salinas et al., 2007] to use Kalman filters with color features to track moving humans; 3) Time-of-flight cameras: A method using relational depth similarity features [Ikemura and Fujiyoshi, 2011] was proposed to detect humans by comparing the degree of similarity of depth histograms in local regions, and Xu et al. [Xu and Fujimura, 2003] developed a method based on a depth split and merge strategy to detect humans; 4) RGB-D cameras: Salas [Salas and Tomasi, 2011] designed a method that combines appearance-based detection and blob tracking to detect upright pedestrians in an indoor environment with a static background, Xia et al. [Xia et al., 2011] created another human detector by identifying human heads from depth images acquired by a static camera, and Luber et al. [Luber et al., 2011] detected pedestrians indoors using an off-line a priori detector with on-line boosting and tracked humans with a multi-hypothesis Kalman filter.

The work most closely related to ours was conducted by [Choi et al., 2011a], which proposed a particle filter-based method to fuse observations from multiple independent detectors, and track humans with a Kinect camera on a mobile robot. However, the detection in this research was based on a sliding window technique over 2D images, which is highly computationally expensive. In addition, in human-robot teaming, the ability to re-identify humans and discriminate in human-human and

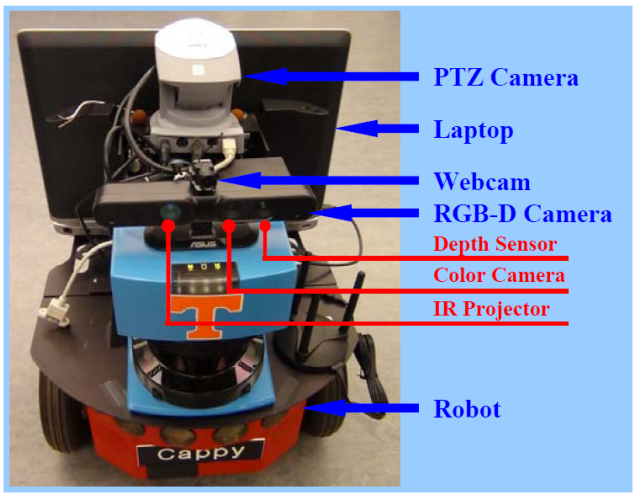


Figure 3.2: Installation of Asus Xtion Pro LIVE RGB-D camera on a Pioneer 3DX robot.

human-object interaction scenarios is of significant importance, as robots often must work with a group of co-workers who can repeatedly leave and enter the robot's view and interact with each other and objects. We address all of these issues which were not incorporated in the previous work.

3.3 Multiple Human Detection

As discussed above, our objective in this work is to develop a robust human perception system, with an ultimate goal of allowing a mobile robot to efficiently interact and cooperate with humans in human-robot teaming. Our human perception system is based on the methodology of tracking-by-detection. We begin the discussion by describing 3D point cloud preprocessing procedures, and a guided sample consensus approach to identify and remove the ground and ceiling planes. Then, we discuss our interleaved tracking-by-detection approach to efficiently track humans in real time. Finally, we describe our system's implementation.

3.3.1 Camera Calibration and Pixel-Level Preprocessing

We use an Asus Xtion Pro LIVE color-depth camera to acquire 3D point clouds. The color-depth camera is installed on top of a Pioneer 3DX mobile robot, as depicted in Figure 3.2. Before acquiring 3D point cloud data, the color-depth camera must be calibrated to obtain its intrinsic parameters, such as focal distances, distortion coefficients and image centers. Because RGB-D cameras acquire color and depth information separately, the camera must be calibrated to accurately map between depth and color pixels. Then, a 3D point cloud is formed using the color and depth

(a) 3D point cloud **(b)** Color & depth images

Figure 3.3: An example 3D point cloud with corresponding color and depth images, which are obtained from the RGB-D camera on a mobile robot moving in a hallway.

information. Figure 6.2 depicts a 3D point cloud along with its color and depth images.

The raw color and depth images acquired by the Xtion camera have a resolution of 640×480. To reduce computation costs, each 3D point cloud is first downsampled to a smaller size by resizing color and depth images to 320×240. The Xtion camera captures depth by projecting infra-red (IR) patterns on the scene and measuring their displacement. Due to the limitations of this depth sensing technology, the depth data is very noisy, and contains a significant amount of null or missing values, which can result from the occlusion of the IR camera's point of view or the absorption of the IR light by objects. The points without depth information and the noisy points, i.e. those with few neighbors, are removed from the 3D point cloud. Then, histogram equalization is applied to the color pixels to remove the effect of sudden intensity changes resulting from the auto white balancing technology.

3.3.2 Ground and Ceiling Plane Removal

We assume humans and robots exist and operate on the same ground plane, and that a ceiling plane is viewable above them. Since our color-depth camera is installed on a mobile robot at a small tilt angle, these planes generally consist of a significant amount of points that gradually change depth. The points on the ground usually connect objects that are located on the floor. In order to eliminate this connection, ground plane detection and removal is an important operation to separate candidate objects with similar depth values. Using the same technique, the ceiling plane is likewise detected and removed to increase processing speed.

To perform this task, we use a random sample consensus (RANSAC) approach [Fischler and Bolles, 1981], which is an iterative method to estimate the parameters of a mathematical model from a set of observations that contains outliers. We also combine the RANSAC algorithm with our prior knowledge: 1) the ground and ceiling planes should be at the bottom and top of the 3D point cloud, and 2) each plane's surface norm is a vertical vector. Because the physical oscillations of the moving robot cause slight changes in each plane's location in the 3D point clouds, the plane's parameters should be re-estimated for each point cloud. We also observe that, because there is only a slight change between temporally adjacent point clouds, previous parameters can be used to guide parameter estimation in the current point cloud. Considering this knowledge, we introduce a new extension of the standard RANSAC algorithm, shown in Algorithm 1, that is very robust and efficient.

Given distance tolerance ϵ, maximum tolerance ϵ_{max}, maximum iterations I_{max}, and the plane's previous parameters $\boldsymbol{A}^{t-1}$ that are estimated from previous 3D point cloud at $t-1$, Algorithm 1 estimates the parameters of the current plane, i.e., $\boldsymbol{A}^t = [a^t, b^t, c^t, d^t]$, from prior knowledge and current observations $\boldsymbol{X}$. The parameter ϵ_{max} is a predefined maximum distance tolerance used to select search regions of the plane in order to compensate for robot oscillations. Then, all points satisfying $\text{dis}(\boldsymbol{x}, \boldsymbol{A}^t) \leq \epsilon$ are defined to belong to the plane, where the distance between a point to the plane in the 3D space is computed by:

$$\text{dis}(\boldsymbol{x}, \boldsymbol{A}) = \frac{|ax + by + cz + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{3.1}$$

The initial parameters $\boldsymbol{A}^0$ of the ground and ceiling planes are computed using the robot's geometric information. Then, all points in these planes are removed from the current observation for further processing. As an example, given the input point cloud as shown in Figure 6.2, Algorithm 1 is applied to detect the ground and ceiling planes, and the resulting point cloud, with these planes removed, is illustrated in Figure 3.4.

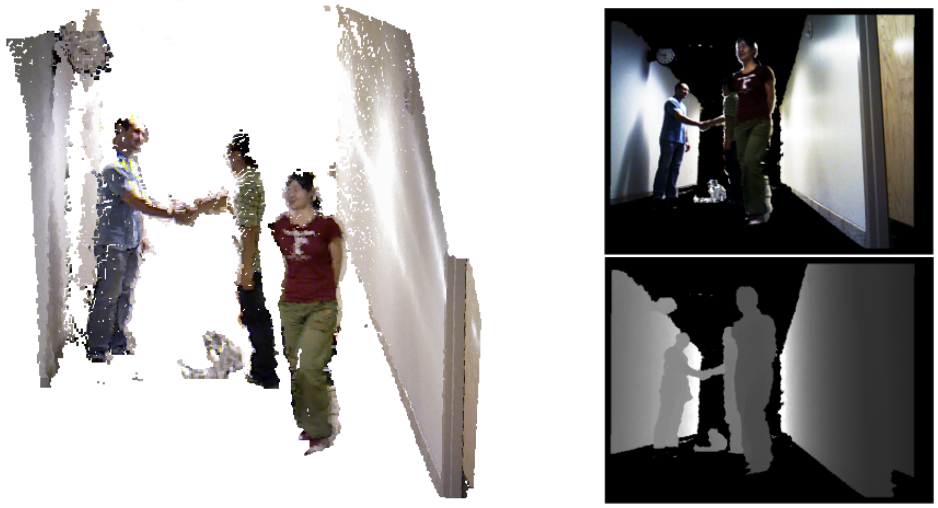
3.3.3 Candidate Detection

Our human detection approach is based on a new concept: *Depth of Interest* (DOI). Analogous to the concept of region of interest (ROI), which is defined as a highly probable rectangular region of object instances [Kim and Torralba, 2009], a DOI is defined as a highly probable interval of human or object instances in the 3D point cloud depth distribution. A DOI is identified by finding a local maximum in the depth distribution and selecting a depth interval centered at that maximum. The correctness of DOI is supported by the observation that any object in a point cloud includes a set of points with similar depth, or several spatially adjacent sets. Each DOI has a high probability to contain objects that we are interested in, which can correspond to humans or non-human objects. Since 3D point clouds captured by

Algorithm 1: Prior-knowledge guided RANSAC

Input : I_{max}, ϵ, ϵ_{max}, $\boldsymbol{X}^t$, and $\boldsymbol{A}^{t-1}$
Output: $\boldsymbol{A}^t = [a^t, b^t, c^t, d^t]$

Extract a set of 3D points belonging to the initial plane:
$\boldsymbol{C}_0 = \{\boldsymbol{x} \in \boldsymbol{X}^t : \mathrm{dis}(\boldsymbol{x}, \boldsymbol{A}^{t-1}) \leq \epsilon_{max}\}$;
for $i \leftarrow 1$ **to** I_{max} **do**
  Randomly select three points that are not on a line: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} \in \boldsymbol{C}_{i-1}$;
  Estimate the parameters $\boldsymbol{A}_i^t$ with $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}$;
  Extract a set of points belonging to the plane:
  $\boldsymbol{C}_i = \{\boldsymbol{x} \in \boldsymbol{X}^t : \mathrm{dis}(\boldsymbol{x}, \boldsymbol{A}_i^t) \leq \epsilon\}$;
  if $|\boldsymbol{C}_i| < |\boldsymbol{C}_{i-1}|$ **then** Set $\boldsymbol{C}_i = \boldsymbol{C}_{i-1}$; ;
end
Estimate $\boldsymbol{A}^t$ that best fits all points in $\boldsymbol{C}_{I_{max}}$;
return $\boldsymbol{A}^t$



(a) 3D point cloud **(b)** Color & depth images

Figure 3.4: Resulting 3D point cloud with corresponding color and depth images after removing ground and ceiling planes.

color-depth sensors can contain multiple objects located at various depth ranges, the depth distributions of different clouds generally have different shapes with a different number of local maximums. Because the underlying density form is therefore unknown, a non-parametric method is required. To estimate the depth distribution, a 3D point cloud is first downsampled to a small size (e.g., 500 points), then the non-parametric Parzen window algorithm [Parzen, 1962] is applied on the downsampled cloud. Our estimate is based on a Gaussian kernel function [Parzen, 1962] with a bandwidth of 0.15 meters, which we tuned through empirical testing. As an example, the estimated depth distribution of the 3D point cloud in Figure 3.4 is depicted in Figure 6.3.

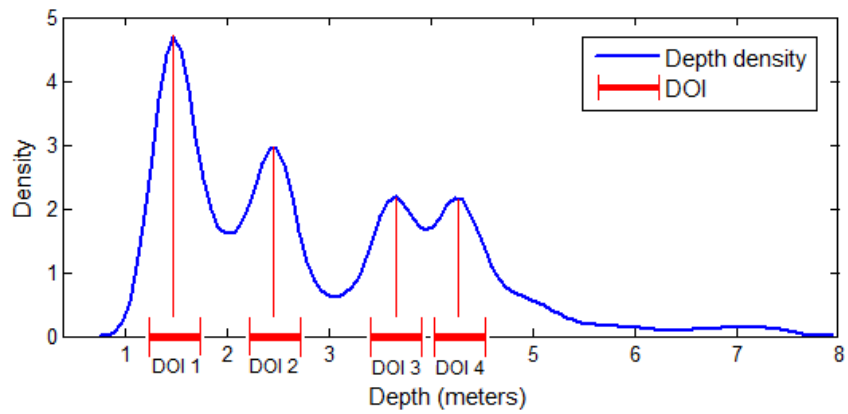
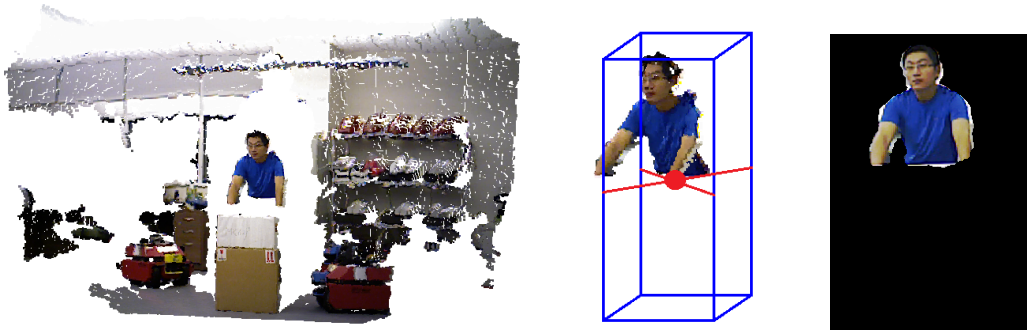


Figure 3.5: Depth distribution of the point cloud in Figure 3.4a with four extracted DOIs. The density is estimated using the Parzen window method with Gaussian kernels.

(a) DOI 1 **(b)** DOI 2 **(c)** DOI 3 **(d)** DOI 4

Figure 3.6: Candidates detected from the 3D point cloud in Figure 3.4a, using the DOIs shown in Figure 6.3.

To efficiently generate candidates from a 3D point cloud, the following procedures are conducted in parallel at each DOI:



(a) (b) (c)

Figure 3.7: Computation of the height and centroid of an occluded object. Figure 6.4a shows a raw 3D point cloud. The height of an occluded object is defined as the distance between the highest point to the ground, as shown by the height of the bounding cube in Figure 6.4b. The object centroid is drawn with a red dot in the center of the bounding cube in Figure 6.4b. When the object's point cluster is projected to a 2D color image of size 96×64, the object is placed in the center of the image according to its real size, instead of the blob size, as shown in Figure 6.4c.

1. Depth filtering: The 3D point cloud is filtered along the depth dimension by selecting all points within each DOI, and a depth image is computed from the filtered cloud.

2. Connected component detection: A binary mask is computed from the depth image to indicate whether a depth pixel is within each DOI. Then, connected components are detected using a connectivity of eight. Each connected component is then given a unique index.

3. Candidate generation: Each cluster of 3D points, whose depth pixels belong to the same connected component, is extracted to form a candidate.

To reduce false negatives, depth values of all currently tracked humans are fed back from the tracking to the detection module. If a depth value being examined does not exist in the current DOIs, a new DOI is created, centered on the depth value, and the candidate generation process above is applied to the new DOI to generate additional candidates. Using this DOI-based candidate generation process drastically reduces the number of candidates and avoids the need to scan the entire cloud, greatly saving processing time in our real-time system.

To preserve the 3D point clusters that contain only human candidates, a cascade of detectors is used to reject candidates that contain only non-human objects. In the detector cascade framework [Viola et al., 2005], simple detectors are first applied to reject the majority of candidates before more complex detection is performed. A positive result from the first detector triggers the evaluation with a second detector. Cascades of detectors have been shown to greatly increase detection performance

by lowering the false positive ratio, while radically reducing computation time [Viola et al., 2005]. Moreover, the detector cascade can be applied in parallel on each candidate to further reduce computation time. Thus, using a cascade of detectors not only improves the accuracy of our system, but also makes it more able to function in real time. In our system, we use a sequence of heuristic detectors and a HOG-based detector to form a detector cascade in order to reject most of the non-human candidates. Our detector cascade includes the following:

1. Height-based detector: The height of a candidate point cluster is defined as the distance between the point with the largest height value and the ground plane, which can be computed using Eq. (3.1). Figure 6.4b illustrates the definition of the height feature. A candidate is rejected if its height is smaller than a min-height threshold, or larger than a max-height threshold.

2. Size-based detector: The size of a candidate point cluster can be estimated with: $s(\boldsymbol{d}) = n(\boldsymbol{d})/k(z_{DOI})$, where $n(\boldsymbol{d})$ is the number of points in candidate $\boldsymbol{d}$, z_{DOI} is the average depth value of the DOI that contains $\boldsymbol{d}$, and $k(\cdot)$ is the conversion factor in units of points/m^2, which is a function of depth and is used to take into account visual linear perspective, i.e., an object contains more points when it gets closer to the camera. A candidate is rejected if its size is greater than a max-size threshold. However, it should be noted that in order to allow for occlusion, our system does not reject small-sized candidates.

3. Surface-normal-based detector: This detector is used to reject planes, such as walls and desk surfaces. Given three randomly selected points in a candidate point cluster: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\} \in \boldsymbol{d}$, a 3D surface normal $\boldsymbol{v} = [x, y, z]$ of the candidate can be computed by:

$$\boldsymbol{v}(\boldsymbol{d}) = (\boldsymbol{x}_2 - \boldsymbol{x}_1) \times (\boldsymbol{x}_3 - \boldsymbol{x}_1) \tag{3.2}$$

   If $\boldsymbol{v}(\boldsymbol{d})$ is in the x-z plane, i.e., $y \approx 0$, then the candidate is detected as a vertical plane, e.g., a wall. If $\boldsymbol{v}(\boldsymbol{d})$ is along the y-coordinate, i.e., $x \approx 0$ and $y \approx 0$, then it is detected as a supporting plane, e.g., a table or desk top. The surface normal of a candidate is computed multiple times with different points, and majority voting is used for a robust decision.

4. HOG-based detector: The detector applies a linear SVM and the HOG features, as proposed by Dalal and Triggs [Dalal and Triggs, 2005]. Their recommended settings are also used for all parameters except that our detection window has a size of 96×64. The candidate point cluster is projected onto a color image of size 96×64 to enable single-scale scanning to save computation. It is desirable that the color image contains the whole candidate, including the parts that are occluded. When a candidate is partially occluded, we set the distance between the candidate's highest pixel and the bottom of the projected color image to be proportional to the candidate's height, as illustrated in Figure 6.4. By using a

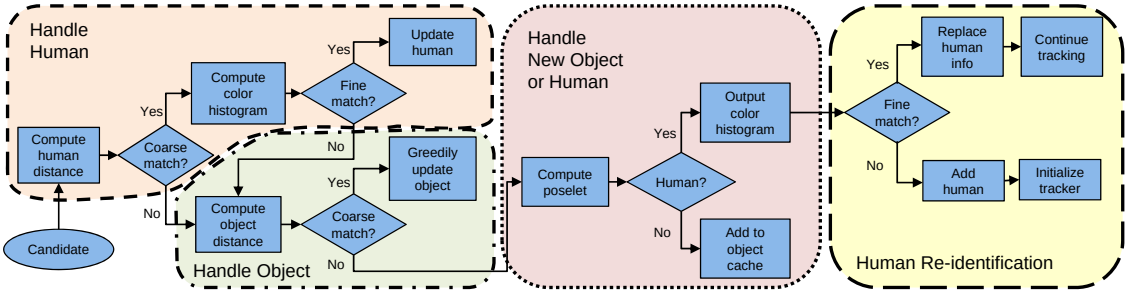


Figure 3.8: Illustration of our candidate handling decision DAG that efficiently integrates human-object classification, data association, and multiple target tracking. This framework also simultaneously handles tracked humans, detected objects, and new humans and non-human objects, and performs human re-identification.

height closer to the actual height rather than the blob height we obtain a more reliable detection result.

The parameters of the heuristic detectors are manually tuned according to empirical observation and prior knowledge, while the HOG-based detector requires a training process to learn model parameters. The candidates that survive the detector cascade are passed to the tracking module. The HOG features of each candidate are also passed to the tracking module, which are used to further distinguish humans and non-human objects to allow for more robust and efficient tracking.

3.3.4 Multiple Human Tracking

In human-robot teaming, most tasks, such as human action recognition and navigation among humans, require a robot to perceive human trajectories. In these scenarios, single-frame detection is insufficient and human tracking across multiple consecutive frames becomes essential. In this work, we implement a decision DAG-based candidate handling algorithm to simultaneously handle tracked humans, detected non-human objects, and new humans and objects, and re-identify humans who enter the camera's view after leaving the camera's view for a period of time, as illustrated in Figure 3.8. One key advantage of our decision DAG-based algorithm is that it allows us to divide the types of candidates into separate cases and only apply the most computationally expensive techniques where necessary, thus increasing the speed of our overall system to achieve real-time performance.

Human-Object Classification

In order to further separate humans and non-human objects and explicitly address partial occlusion, the poselet-based human detector [Bourdev and Malik, 2009], a state-of-the-art body part-based detector, is applied in our human-object classification module. Poselets are defined as human parts that are highly clustered in both

appearance and configuration space. This detector separately classifies human parts using trained linear SVMs with poselet features, and combines their outputs in a max-margin framework. Although this detector can alleviate the occlusion problem by relying on the unoccluded parts to recognize a human, it is very time consuming to compute poselet features. Because of the time constraints of our real-time system and our desire to use consumer-grade computation hardware, this key disadvantage prevents us from simply applying this technique to every candidate in all point cloud frames. As a result, a poselet-based detector cannot be used as a part of the detector cascade in our candidate detection module, and instead must be used only where most necessary.

In order to use the poselet-based detector most effectively for our real-time system, our decision DAG-based candidate handling algorithm applies this technique only on a subset of candidates. To achieve this goal, we first introduce the *object cache*, which is defined as the set of non-human candidates. Given the object cache, the poselet technique is used only on the candidates that do not match with any tracked humans or non-human objects in the cache. Thus, for each object, including both humans and non-human objects, application of the poselet-based classification is a one-time procedure, even if the object stays in the robot's view over a long time period, across multiple frames.

The object cache is maintained in the following way. A new candidate in the robot's view, classified by the poselet-based detector as a non-human object, is added to the object cache. Alternatively, if a candidate is not new, i.e., it matches coarsely with an object in the cache, that object is replaced by the candidate. If an object in the cache does not match any candidate for a period of time, it is removed. It should be noted that no tracking is performed over the non-human objects in the object cache, and the object cache evolves by replacing old objects with new ones. The object cache plays an important role in improving the efficiency and accuracy of our tracking module. It not only reduces the number of poselet-based detection procedures to significantly reduce computation time, but also provides negative instances to discriminatively update human models during run-time for robust fine matching of candidates with tracked humans, as discussed next.

Data Association

This module is applied to match candidates with tracked humans or detected non-human objects in the object cache, based on the assumption that at most one candidate is matched with at most one human or detected non-human object. Our data association process is divided into coarse and fine matching phases.

Coarse Matching: Coarse matching between detected candidates and tracked humans is based on position and velocity information. Formally, the Euclidian distance in the 3D space between a candidate $\boldsymbol{d}$ and a human $\boldsymbol{t}$ is first computed

by:

$$dis(\boldsymbol{d}, \boldsymbol{t}) = \|(\boldsymbol{c_t} + \dot{\boldsymbol{c}}_{\boldsymbol{t}} \Delta t) - \boldsymbol{c_d}\| \tag{3.3}$$

where $\boldsymbol{c_d}$ and $\boldsymbol{c_t}$ are the positions of $\boldsymbol{d}$ and $\boldsymbol{t}$, respectively, $\dot{\boldsymbol{c}}_{\boldsymbol{t}}$ is the velocity of the human, and Δt is the time interval between frames. If this distance is smaller than a predefined threshold ϵ_{ct}, then there is a coarse match between the candidate and human. Because objects in our system are detected but not tracked, their velocity information is not available, so only position information is used to coarsely match a candidate and object. Similarly, if the Euclidian distance $\|c_{\boldsymbol{d}} - c_{\boldsymbol{o}}\| < \epsilon_{co}$, the candidate $\boldsymbol{d}$ and the non-human object $\boldsymbol{o}$ are coarsely matched, where $c_{\boldsymbol{o}}$ is the position of the non-human object $\boldsymbol{o}$, and ϵ_{co} is a predetermined distance threshold.

Fine Matching: Fine matching is applied to further match candidates with tracked humans, and also to re-identify humans when they re-enter the camera's view.

We use color information to create an appearance model for each tracked human, which is learned and updated in an online fashion using an online AdaBoost algorithm for feature selection, as proposed by Grabner et al. [Grabner and Bischof, 2006]. We train a strong classifier for each human $\boldsymbol{t}$ to determine whether a candidate $\boldsymbol{d}$ matches a human, which is a linear combination of selectors:

$$h_{\boldsymbol{t}}^{strong}(\boldsymbol{d}) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i h_i^{sel}(\boldsymbol{d})\right) \tag{3.4}$$

where sgn is the signum function, N is the number of selectors to form a strong classifier, and α is the weight for the selector h^{sel}, which chooses the weak classifier with the lowest error from a pool of M weak learners. A weak learner h^{weak} represents a feature $f(\boldsymbol{d})$ that is computed on the candidate $\boldsymbol{d}$. A color histogram in the RGB color space is used for our features, and is computed from the candidate's color image that is projected from its 3D point cluster, as shown in Figure 6.4c. We use nearest neighbor classifiers, with a distance function D, as our weak learners:

$$h^{weak}(\boldsymbol{d}) = \text{sgn}(D(f(\boldsymbol{d}), \boldsymbol{p}) - D(f(\boldsymbol{d}), \boldsymbol{n})) \tag{3.5}$$

where $\boldsymbol{p}$ and $\boldsymbol{n}$ are cluster centers for positive and negative instances. The weak learner is updated from a positive and a negative instance in each learning process. Each positive instance to the human-specified classifier $h_{\boldsymbol{t}}^{strong}$ is provided by the tracked human $\boldsymbol{t}$. Each negative instance is randomly sampled from other tracked humans or non-human objects in the object cache.

Our fine matching approach has several advantages. First, it creates an adaptive human appearance model that provides a natural way to adapt to human appearance changes caused by occlusions and different body configurations. Moreover, our matching approach is based on a discriminative classification framework, which selects

the most discriminative features to distinguish a specific human from other tracked humans and non-human objects in a more reliable and robust way. Finally, our color histogram features are an accurate representation of a candidate, since the background is masked out in our color images by applying DOIs, as shown in Figure 6.4c. Together, these advantages improve our system's performance through the reduction of errors.

Extended Kalman Filtering

Humans in the robot's field of view are tracked locally in our human tracking module, i.e., human positions and velocities are tracked relative to the robot. Based on the assumption that humans and robots move smoothly in the global coordinates, humans also move smoothly in the local coordinates. The centroid of each human is tracked in the 3D space, using the extended Kalman filter (EKF) [Einicke and White, 1999]. EKF is able to track non-linear movements with a low computational complexity, making it suitable to address non-linear tracking tasks in real-time applications.

The following procedures for initialization, update and deletion for the EKF process were integrated into our candidate handling framework (Figure 3.8):

Initialization: A new human tracker is created if a candidate is detected as a human that is not currently tracked. However, to address human re-identification tasks, when a non-tracked human is detected, instead of immediately initializing a new tracker, the deactivated trackers are first checked to detect whether the human has already been observed. If the human matches a previously tracked subject, the deactivated tracker is reactivated instead, re-identifying the human.

Update: At each frame, EKF predicts each tracked human's current state, and corrects this estimated state using the observation that is provided by the data association module. Then, the updated estimate is used to predict the state for the next frame. If a tracked human is not associated with any candidate, the tracker is updated with the previous observation.

Termination: A human tracker instance only persists for a predefined period of time without being associated by any candidates. After this threshold is passed, it is automatically terminated. However, in order to allow for recovering the identity of a human who re-enters the camera's field of view after leaving for a short period of time, the trackers are terminated by deactivation instead of deletion.

3.3.5 Implementation

In the candidate detection module, the parameters of the height-based detector are manually set; the min-height threshold is set to 0.4 meters, and the max-height threshold is set to 2.3 meters. The max-size threshold in the size-based detector is set to 3 meters^2. Our HOG-based detector is modified from the HOG implementation in [Dalal and Triggs, 2005]. Our detector is trained using bootstrapping. We first train an initial detector with the H3D dataset [Bourdev and Malik, 2009], using all

of the positive and a subset of the negative samples. Then, we apply the initially trained detector on samples of our newly created datasets, as described in Section 3.4.2, and collect samples leading to false positives and false negatives. Finally, we do a second round of training by including these samples in the training set. In the multiple human tracking module, we use the pre-trained poselet-based classifier, which is implemented as described by Bourdev et al. [Bourdev and Malik, 2009]. The coarse matching threshold is set to be 1 meter for humans and candidate pairs, and 0.5 meters for object and candidate pairs. When performing fine matching with online AdaBoost, we use $N = 30$ selectors that select color histogram features from a feature pool of size $M = 250$. The EKF termination threshold for human trackers is set to 5 minutes.

3.4 Experimental Results

We performed experiments using our human perception system that is implemented with a mixture of MATLAB and C++ with the PCL library [Rusu and Cousins, 2011], without taking advantage of GPU processing, on a laptop with an Intel i7 2.0GHz CPU (quad core) and 4GB of memory (DDR3). We created a new dataset suitable for the task of multiple human detection and tracking, consisting of 3D point clouds obtained using an RGB-D camera. Half of the samples in our dataset were used to train the HOG-based detector in a bootstrapping fashion, and half were used to evaluate our system's performance.

3.4.1 Datasets

At the time of this work, there is no publicly available 3D human detection and tracking dataset that is collected with an RGB-D camera. Thus, we collected a large-scale dataset to evaluate the performance of our human perception system. Our dataset was recorded with an Asus Xtion Pro LIVE RGB-D camera in an indoor laboratory environment. The camera was installed on a Pioneer 3DX mobile robot, as illustrated in Figure 3.2, and a laptop was mounted on the robot to record 3D point cloud data. Because the problem of following a target human at an appropriate and safe distance is outside the scope of this work, the robot was remotely teleoperated by a human, who could only observe the robot's surrounding environment through the robot sensors, i.e., the operator could only perceive what the robot perceives. The webcam on top of the RGB-D camera has a similar field of view as the RGB-D camera, which allows the operator to identify and track human subjects without interfering with data recording. The PTZ camera was used to observe behind the robot for safety purposes. The robot's on-board PC was used to control the robot and handle the webcam and PTZ cameras. Although they were needed for conducting experiments, it is noteworthy that the webcam and PTZ cameras do not provide any information

Table 3.1: Characteristics of our datasets with varying difficulties. Check marks indicate the challenge exists.

	Dataset 1	Dataset 2	Dataset 3
Number of samples	8	8	4
Frames per sample	300	540	1800
Has occlusion	✓	✓	✓
Has robot motion			✓
With non-upright human		✓	✓
With human re-entrance	✓		✓
With human-object interaction		✓	✓
With human-human interaction			✓

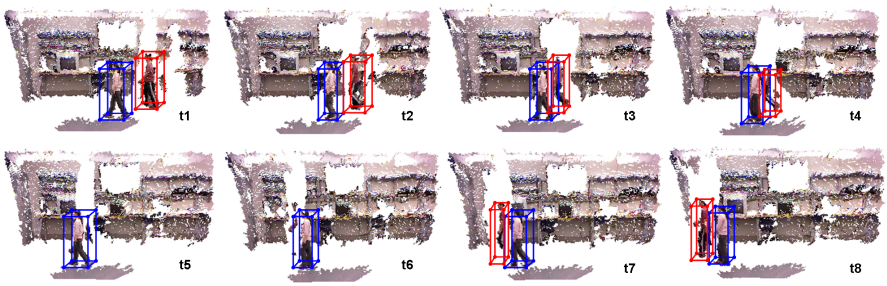
to our human perception system, and thus are not pertinent to the essence of this work.

Our dataset considers three scenarios with increasing difficulties. In Dataset 1, humans act like pedestrians with simple (linear) trajectories. In Dataset 2, humans conduct the task of lifting several humanoid robots and putting them away. In Dataset 3, humans pick up an object, exchange it, and one delivers the object from a laboratory down a hallway to an office room, passing and interacting with other humans on the way. The robot follows the human delivering the object during the entire task. The statistics of our datasets are summarized in Table 3.1, with a breakdown of the increasing difficulty aspects. Each sample in our dataset is a sequence of 3D point clouds that are saved as PCD [Rusu and Cousins, 2011] files with a frame rate of 30 FPS. Each 3D point cloud contains $307,200$ points, corresponding to 640×480 color and depth images, and each point has six values: its coordinates in the 3D space and RGB values.

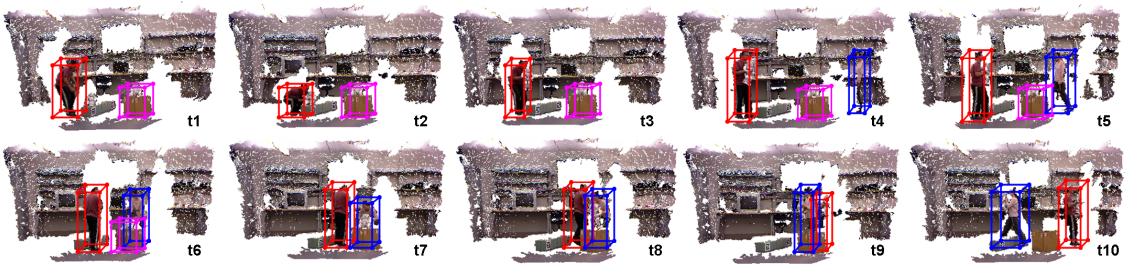
To establish ground truth, our dataset is manually annotated using 2D depth images as follows: First, a representative pixel on a human in a depth image is manually selected to determine the DOI that applies to the human. Using the proper DOI, we mask out the background, leaving the pixels belonging to the same human clustered together as a blob. Then, a bounding box is manually added around each human blob to indicate its x and y coordinates in the depth image. Finally, the bounding box and the DOI are converted to a bounding cube in the 3D space, which is used as ground truth, and the center of a bounding cube is considered the centroid of a human.

3.4.2 Qualitative Analysis

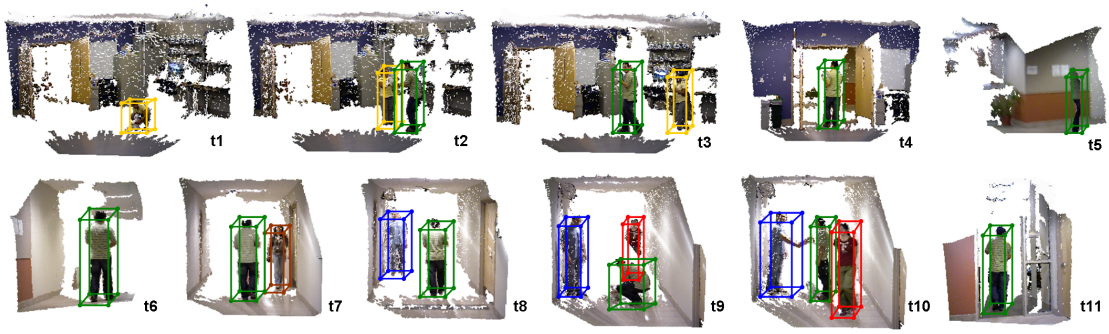
We first analyze the tracking results from our human perception system to demonstrate its effectiveness and robustness in handling different challenges in human



(a) Dataset 1: Humans move like pedestrians with linear trajectory.



(b) Dataset 2: Humans act with complicated body-configurations.



(c) Dataset 3: A human performs a delivery task followed and observed by a moving robot.

Figure 3.9: Experimental results of the proposed human perception system over our datasets.

detection and tracking tasks. For each tracked human, a bounding cube with a consistent shape is manually drawn in the 3D point cloud, according to the cube's vertices that are output by our system. Human identities are represented with different colors, i.e., the same human is represented with the same color in a dataset. The tracking results are illustrated in Figure 7.2.

Dataset 1: Humans act like pedestrians in Dataset 1; they always have an upright pose and generally move with a linear trajectory. It can be observed from Figure 3.9a that non-occluded humans in Dataset 1 are detected and tracked perfectly by our system. When a slight partial occlusion occurs, e.g., Figure 3.9a (t4) and Figure 3.9a (t7), humans are still detected, but the accuracy of the bounding cube might decrease. However, when severe or full occlusion occurs, e.g., in Figure 3.9a (t5), the

Table 3.2: Evaluation results of our 3D-based human perception system using the CLEAR MOT metrics.

	MOTP	MOTA	FN	FP	ID-SW
Dataset 1	56 mm	95.39%	2.77%	1.84%	0
Dataset 2	122 mm	85.48%	4.27%	10.25%	0
Dataset 3	83 mm	94.26%	3.45%	1.19%	0

occluded human cannot be detected, which results in a false negative. Despite the fact that the mostly or fully occluded human cannot be identified, the location of the occluded human's centroid is still updated by the EKF algorithm (for a predefined period of time), using the observation from the previous time point. The advantage of this is that after a human re-appears in the camera's field of view, our system is able to coarsely match the human and continue to use the same tracker to track the human, as shown in Figure 3.9a (t7), which both saves processing time and improves accuracy.

Dataset 2: In this dataset, humans move with a complicated but approximately linear trajectory, in which they switch positions as shown in Figure 3.9b (t7–t9). Our EKF-based tracking algorithm performs well in this situation. Humans also exhibit a variety of body configurations in this dataset, e.g., crouching as shown in Figure 3.9b (t2), and interacting with objects, as illustrated in Figure 3.9b (t7). In these situations, humans can be detected using our detector cascade along with the poselet-based detector, even with partial occlusions as shown in Figure3.9b (t9). In some cases from Dataset 2, a false positive is detected and incorrectly tracked, as indicated by the magenta-colored bounding cube in Figure 3.9b (t1–t6), which is induced by the human-shape robot sitting on a big box in the center. The other humanoid robot sitting on a small box is not detected, as it is rejected by our height-based detector.

Dataset 3: Dataset 3 involves a variety of challenges, as listed in Table 3.1. First, because the robot is moving and humans are tracked in the robot's local coordinate system, human trajectories are no longer linear. We observe that the EKF algorithm still tracks humans with high accuracy in this case, as shown in Figure 3.9c. Second, humans can leave the robot's field of view for a certain period of time. For example, the robot loses the target when the tracked human goes through the door and turns right, as shown in Figure3.9c (t5). Our system addresses this problem; when the human re-enters the robot's field of view, the human re-identification module, using online human specific appearance models, is activated and continues to track the human with the correct index, as shown in Figure 3.9c (t6). Third, humans perform very complicated actions, including human-object and human-human interactions. For instance, a person is passing a humanoid robot to another person in Figure 3.9c (t2), and two persons are shaking hands in Figure 3.9c (t10). In most cases, the interacting humans are separated into different candidates, as illustrated in Figure

3.6c and Figure 3.6d for the hand-shaking interaction. However, when interacting humans have very similar depth values (e.g., less than 0.1 meters), they can be incorrectly extracted as a single candidate, which can then be rejected by the size-based detector. This incorrect rejection would result in a false negative.

3.4.3 Quantitative Evaluation

We follow the CLEAR MOT metrics [Bernardin and Stiefelhagen, 2008] to quantitatively evaluate the performance of our multiple human perception system, which consists of two scores: multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA). The MOTP distance indicates the error between the tracking results and the actual target, and thus reflects the ability of the tracking module to estimate target positions and keep consistent trajectories. The MOTA score combines the errors that are made by the perception system, in terms of false negatives (FN), false positives (FP), and the number of identity switches (ID-SW), into a single accuracy metric. A false negative occurs when a human is annotated in ground truth, but not detected by the perception system. This usually happens for persons that are severely occluded, or on the boundary of the camera's field of view. A false positive occurs when the candidate that is detected as a human does not have a match with any annotated humans in ground truth. In our system, this happens with the non-human objects that have a similar height, size, shape and surface property to a human. An identity switch occurs when a tracked human changes its identity. This can happen when a new human enters the scene who is similar in appearance to a human who has just left the robot's field of view, or when two humans with similar appearances switch positions. Our human perception system is evaluated using the metric threshold of 50 cm, as suggested in [Bernardin and Stiefelhagen, 2008]. The evaluation results are listed in Table 3.2.

Examining our test results, several important observations should be highlighted. First, our human perception system has a very low (perfect) number of ID switches, which is one of the most important properties of our tracking system, since differentiating humans in human-robot teaming applications is essential, especially when, e.g., different human coworkers can have distinct preferences and habits. Minimizing ID switch ratio is achieved by combining the following concepts: 1) The background is masked out by the DOI information, which results in a highly accurate human appearance model; 2) An online algorithm is used to continuously update appearance models in real time; 3) Human appearance models are trained discriminatively, which helps maximize the difference between positive and negative instances; 4) The 'difficult' objects that survive the detector cascade are saved in our object cache. As negative examples to update human appearance models, 'difficult' objects are more representative than other 'easy' objects that are rejected by the cascade of detectors. Second, our system also performs fairly well when localizing targets, in terms of the MOTP scores. We discovered that occlusion usually decreases the object localization ability of our system, and we have greatly relieved this problem

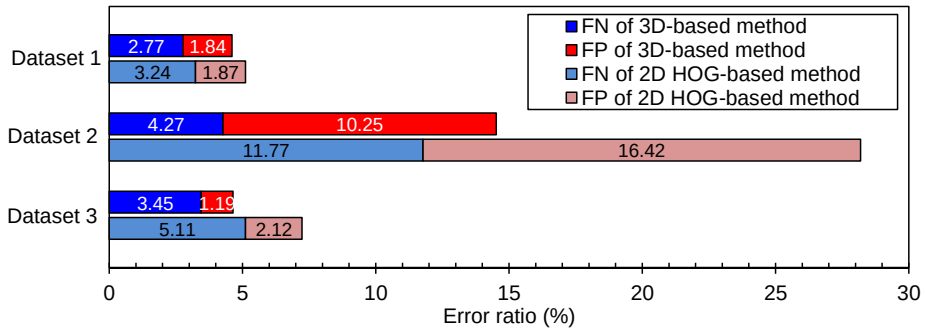


Figure 3.10: Comparison of error ratios (i.e., FN and FP) between our 3D-based approach and 2D baseline method of [Dalal and Triggs, 2005].

by centering a human in a projected 2D color image according to its real height. Third, we achieve very good results with our most complex and difficult dataset, Dataset 3. One reason for this is that in a large number of frames, there is only one human in the scene without any occlusions. In these cases, the human is detected and tracked perfectly. Finally, our human perception system does not perform as well with Dataset 2 as the other sets, because the humanoid robot on the big box causes a large number of false positives. Moreover, our system has the highest false negative ratio on Dataset 2, due to the fact that humans have the longest occlusion duration.

3.4.4 Comparison to 2D Baseline

To provide a baseline comparison for the detection aspect of our approach, the most widely used 2D HOG-based detector [Dalal and Triggs, 2005] was implemented on the same hardware. For this baseline, the detector uses a sliding window paradigm and a sparse scan with 800 windows [Dalal and Triggs, 2005]. For input to the baseline detector, color images were converted from the point cloud data in Datasets 1–3. The baseline detection results are compared in Fig 6.6.

Comparison of the 2D baseline detector with our detection results shows that the addition of depth information provides a clear increase in accuracy. As discussed in Section 3.3.3, this is because depth information allows for accurate estimation of a candidate's height, size, and surface norm; this heuristic information can be used by our cascade of detectors to greatly reduce false positives by rejecting non-human candidates. Depth information also greatly helps to reduce false negatives by feeding DOI information from the tracking module to the detection module to provide assistance locating humans in a new observation. In addition, using the candidate's height helps to detect partially occluded humans, as shown in Figure 6.4.

In obtaining the results for accuracy shown above, the 2D baseline detector yields a frame rate of 0.893 FPS. However, detection is only a part of the entire perception system. The additional tracking step would add additional non-trivial time and further decrease the frame rate of any system into which the baseline detector was

incorporated. Because of this, using a 2D detector such as this baseline in a real-time perception system would be impractical. Because the baseline detector was less accurate and performed so slowly, we did not undertake a comparison between our system and a full system using a 2D detection component.

In comparison, our complete system, including detection and tracking, achieves a processing rate of 7–15 FPS, which is suitable for real-time applications. Our processing rate is improved using the following techniques: 1) Prior knowledge is used to guide the RANSAC algorithm to efficiently detect ground and ceiling planes; 2) The detector cascade efficiently rejects the majority of the candidates, which can be applied on multiple objects in parallel to further save computation time; 3) Window scanning over entire images is avoided by applying DOIs; 4) HOG features are computed with a single-scale scanning over the projected 2D color image that contains a candidate blob; 5) Computed features in previous steps are reused in the current step (e.g. the process to compute HOG features reuses a candidate's height and size features, and the process to compute poselet features reuses HOG features); 6) A decision DAG-based candidate handling framework provides a one-pass process that efficiently combines object-human classification, data association, and multiple human tracking. We observe that a larger number of clusters generally results in more DOIs with more candidates, which typically need more time to process. Therefore, while our experiments were conducted in an academic building and the environments were not manipulated in any way to improve our system's performance, it is certainly possible to conceive of an extremely cluttered environment that would negatively impact computation time.

3.5 Summary

In this chapter, we presented a system for perceiving multiple individuals in three dimensions in real time using a color-depth camera on a mobile robot. Our system consists of multiple, integrated modules, where each module is designed to best reduce computation requirements in order to achieve real-time performance. We remove the ground and ceiling planes from the 3D point cloud input to disconnect object clusters and reduce the data size. In our approach, we introduce the novel concept of Depth of Interest and use it to identify candidates for detection thereby avoiding the computationally expensive sliding window paradigm of other approaches. To separate humans from objects, we utilize a cascade of detectors in which we intelligently reuse intermediary features in successive detectors to reduce computation costs. We represent our candidate tracking algorithm with a decision DAG, which allows us to apply the most computationally expensive techniques only where necessary to achieve best computational performance. Our novel approach was demonstrated in three scenarios of increasing complexity, with challenges including occlusion, robot motion, non-upright humans, humans leaving and re-entering the field of view (i.e., the re-identification challenge), human-object and human-human interaction. Evaluation of

the system's performance using CLEAR MOT metrics showed both high accuracy and precision. The implementation achieved a processing rate of 7–15 FPS, which is viable for real-time applications. Our results showed that through use of depth information and modern techniques in some new ways, it is possible to use a color-depth camera to create an accurate, robust system of real-time, 3D perception of multiple humans by a mobile robot.

Chapter 4

Representation: CoDe4D LST Features

4.1 Introduction

Since visual data directly acquired from cameras usually contain a significant amount of noise, the problem of human representation is to extract features from raw data in order to provide a clean, compact representation of the humans in the data.

Among different approaches for representing human activities [Choi et al., 2011b, Lan et al., 2012, Khamis et al., 2012, Xia and Aggarwal, 2013], local spatio-temporal (LST) features have recently become a most popular representation, which are inspired by a human visual attention mechanism that allows a human to use salient body appearances and movements to rapidly recognize activities in complex, cluttered scenes [Treisman and Gelade, 1980]. LST features are designed to capture variations of characteristic textures, shapes and poses in visual data and thereby to provide a descriptive representation of human activities in a video. These features are typically defined as spatio-temporal pixels, referred to as *interest points*, which maximize a user-defined saliency function. LST features are often described using local appearance and motion information in the neighbourhood of each selected interest point. Since LST features are relatively invariant to image rotation, scaling, and translation, partially invariant to illumination changes, and robust to partial occlusion [Lowe, 2004, Wang et al., 2009], they are widely used to encode human activities in color videos [Zhang and Parker, 2011, Dollár et al., 2005, Laptev, 2005, Kläser et al., 2008, Chakraborty et al., 2012, Wang et al., 2013a]. In addition, because LST features are directly extracted from raw visual data, they avoid potential failures of preprocessing steps such as human detection and tracking.

Although LST features have shown promising performance on human activity recognition from color videos, due to the limitation of the sensing device (e.g., color cameras), most of previous LST features do not make use of one important piece of information that is now available—depth. Because humans act in 3D space, depth can be utilized along with color cues to implement more distinctive, robust salient

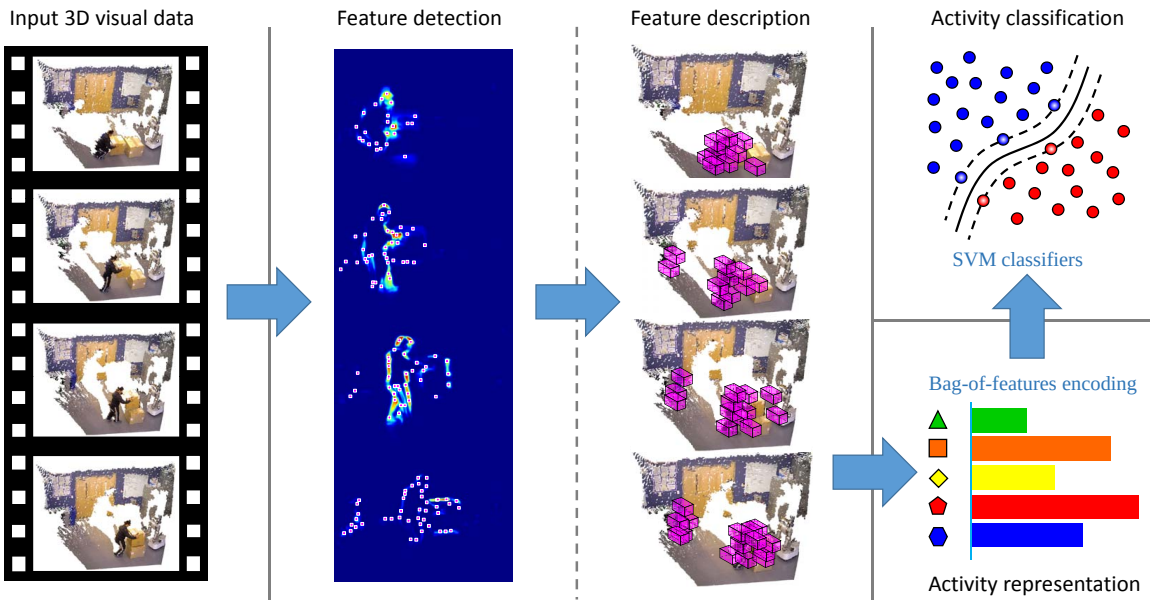


Figure 4.1: Illustration of major components to recognize human activities using our 4D color-depth local spatio-temporal features. Given a sequence of 3D (xyz) frames (e.g., 3D point clouds or color-depth images), our feature detection algorithm constructs a sequence of saliency maps, which characterize texture, shape and pose variations using both color and depth information, and extracts spatio-temporal interest points from the saliency map sequence. Then, our multi-channel feature description algorithm centers an adaptive support region at each interest point to incorporate information in its neighborhood and encodes intensity and depth image gradients within the support region to form a feature vector. To recognize human activities using the proposed CoDe4D LST features, we construct a complete system using the bag-of-features representation and the SVM classifier.

features. The depth sensor has several advantages over color cameras. First, depth sensors provide 3D structure information of the scene, which significantly alleviates the limitation of traditional vision systems that only acquire information in 2D space. Second, depth sensors are generally not sensitive to illumination changes and can work in darkness, which allows for obtaining observations at night.

In this chapter, we introduce the *4-d*imensional *Co*lor-*De*pth (CoDe4D) LST features that are extracted in $xyzt$ space (i.e., 3D spatial and 1D temporal) and incorporate both color and depth information contained in a sequence of RGB-D frames or 3D point clouds. The objective of developing such CoDe4D features is to provide a compact, discriminative representation of human activities when depth information is available along with color information, in order to improve activity recognition performance in realistic, complex scenes.

An overview of our CoDe4D LST feature extraction method along with how the features are employed to construct a human activity recognition system is graphically

summarized in Figure 6.1. Given a sequence of 3D point clouds or color-depth frames, we construct a spatio-temporal saliency map that incorporates both color and depth information, and define interest points as the local maxima of the saliency map. Then, we place a 4D hyper-cuboid as the feature's support region at each detected interest point in 4D dimension (i.e., 3D space and 1D time), which adapts its size to the interest point's depth value; then we use the Multiple-Channel Orientation Histogram (MCOH) descriptor to incorporate color and depth cues to form a final feature vector. To perform human activity recognition using our CoDe4D features, we apply the standard Bag-of-Features (BoF) model, which quantizes our CoDe4D LST features into discrete visual words and represents each input sequence as a frequency histogram of the words. Then, a non-linear Support Vector Machine (SVM) with a χ^2-kernel is applied to perform multi-class activity classification.

The contributions of this work are summarized as follows:

1. We propose a new CoDe4D multi-channel feature detector based on a salience map, which considers both color and depth cues to extract local spatial-temporal interest points in $xyzt$ space.

2. We implement a new feature descriptor, called *Multiple-Channel Orientation Histogram* (MCOH), which is able to encode both color and depth cues and adapt the support region size to visual linear perspective variations.

3. We empirically validate that our CoDe4D LST features extracted using the multi-channel detector and MCOH descriptor are highly discriminative to represent human activities, which results in state-of-the-art activity recognition performance.

The remainder of this chapter is organized as follows. In Section 4.2, we provide a comprehensive overview of previous features to represent human activities in 3D space. Our multi-channel feature detector to detect the CoDe4D LST features is proposed in Section 4.3. Then, the MCOH descriptor applied to quantize our CoDe4D LST features is introduced in Section 4.4. Section 4.5 briefly describes our activity recognition approach based on BoF models and SVMs. Evaluation results of our CoDe4D LST features for activity recognition tasks are presented in Section 7.5. Finally, 10.7 summarizes this work.

4.2 Related Work

A large number of features have been proposed to represent and recognize activities from visual data [Turaga et al., 2008, Aggarwal and Ryoo, 2011, Borges et al., 2013]. We review different categories of feature extraction methods with a focus on approaches working with 3D visual data and LST features.

4.2.1 Activity Representation in 3D Space

Although most previous features for human activity representation are based on 2D videos [Yu and Aggarwal, 2009, Chakraborty et al., 2012, Wang et al., 2013a], several methods using 3D visual data were proposed in the past few years, which can be generally classified into four groups. A naive human activity representation is based on the 3D centroid trajectory, in which a human subject is represented as a point that indicates the 3D location of the human subject in the visual data [Chowdhury and Chellappa, 2003a, Brdiczka et al., 2009]. In general, features based on the centroid trajectory are only suitable for representing a human that occupies a small region in an image. Another representation of human activities using 3D visual data is based on human shape information, including a history of 3D human silhouette [Veeraraghavan et al., 2005, Yan et al., 2008, Singh et al., 2008]. A third category of representation to recognize human activities is based on 3D human models, such as a 3D human skeleton model [Sung et al., 2011] and a 3D articulated body-part model [Ben-Arie et al., 2002, Knoop et al., 2006, Schwarz et al., 2010]. The robustness of the features based on 3D human shapes and body models relies heavily on the performance of foreground human segmentation and body part tracking, which are hard-to-solve problems due to camera motions, dynamic background and occlusions [Zhang et al., 2013]. Different from the discussed three categories of features that are extracted globally from 3D visual data, the last category of features do not require global information (e.g., human locations) to compute.

4.2.2 LST Feature Detection

LST features, which represent global human activities with local texture, shape and pose changes, have recently become a most popular activity representation due to their promising performance on human activity classification. The first LST feature detector, referred to as Spatio-Temporal Interest Point (STIP) detector, was introduced by Laptev et al. [Laptev, 2005], which is based on generalized Harris corner detectors with a set of multi-scale spatio-temporal Gaussian derivative filters. Dollar et al. [Dollár et al., 2005] detected LST features, often referred to as Cuboid features, from color videos through applying separable filters in spatial and temporal dimensions (i.e., Gaussian filters along spatial dimension and Gabor filters along temporal dimension) and selecting interest points with maximum responses in the motion saliency map. Other LST features were also developed based on an extended Hessian saliency measure [Willems et al., 2008], a salient region detector [Oikonomopoulos et al., 2005], or global information [Wong and Cipolla, 2007].

These approaches extract LST features only based on color information and ignore the important depth information that is available in color-depth videos. Recently, several features were introduced to extract LST features from depth images. Cheng et al. [Cheng et al., 2012] applied the STIP detector directly on depth images obtained from color-depth sensors. Ni et al. [Ni et al., 2011] introduced

the Depth-Layered Multi-Channel STIPs (DLMC-STIPs) by applying the standard STIP detector on multiple depth layers. Xia et al. [Xia and Aggarwal, 2013] proposed the Depth-STIPs (DSTIPs) through applying the Cuboid detector on depth images. Although these feature detection methods can extract visual cues from depth images, they do not make use of color or intensity information and therefore ignore important texture information. Different from previous feature detectors that are based on either color [Dollár et al., 2005, Laptev, 2005] or depth [Cheng et al., 2012, Xia and Aggarwal, 2013] cues, we introduce a multi-channel LST feature detector that is capable of incorporating both color and depth information during the detection process and extract LST features from color-depth visual data.

4.2.3 LST Feature Description

After an interest point is detected, a descriptor is required to encode the information in the neighbourhood of the interest point to construct a final feature vector. Nearly all LST feature descriptors used to represent human activities in color videos are based on image gradients. Dollar et al. [Dollár et al., 2005] concatenated gradients of intensity images into a feature vector. Scovanner et al. [Scovanner et al., 2007] implemented the SIFT3D descriptor, an extension of the well-known Scale-Invariant Feature Transform (SIFT) [Lowe, 2004] descriptor to color videos, to describe gradients in space-time dimensions using spherical coordinate based quantization methods. Klaser et al. [Kläser et al., 2008] implemented the HOG3D descriptor, an extension of the well-known Histogram of Oriented Gradients (HOG) descriptor [Dalal and Triggs, 2005], to describe spatio-temporal gradients computed from color image sequences using regular polyhedron based quantization approaches. Laptev et al. [Laptev et al., 2008] implemented the HOG/HOF (i.e., Histogram of Optical Flow) descriptor to characterize local shapes and motions for activity recognition from color videos. Another popular type of feature descriptors investigate trajectories of interest points based on optical flow [Wang et al., 2013a, Wang and Schmid, 2013]. The low-level local features were also aggregated to construct more complicated middle-level human activity representations, such as the motionlet [Wang et al., 2013b].

With the emergence of depth sensors (e.g., Kinect), several LST descriptors were developed to quantize local information contained in depth image sequences. A most commonly used methodology to describe visual features from depth frames is to extend the existing color/intensity descriptors. For example, Ni et al. [Ni et al., 2011] directly applied the HOG/HOF descriptor on their DLMC-STIPs; Ofli et al. [Ofli et al., 2013] also employed HOG/HOF descriptors to quantize features detected from depth frames for activity recognition. To represent depth information for human action analysis, Cheng et al. [Cheng et al., 2012] introduced the Comparative Coding Descriptor (CCD), which describes the structure of the depth cuboid using sequential codes. Xia et al. [Xia and Aggarwal, 2013] implemented the Depth Cuboid Similarity Feature (DCSF) descriptor that uses self-similarity to encode the spatio-temporal shape of the cuboids extracted from depth image sequences.

Previous feature description algorithms are based on either color or depth cues. Different from these descriptors, we aim at developing a multi-channel descriptor that can simultaneously encode both color and depth cues and adapt to linear perspective view variations. To this end, we significantly improve our previous descriptor introduced in [Zhang and Parker, 2011] by adapting the support region size and designing spherical coordinate based methods to quantize visual cues that are extracted from both color and depth channels.

4.3 CoDe4D Feature Detection

Color-depth data obtained from RGB-D cameras generally contains massive amounts of information in the form of spatio-temporal color and depth variations. Most of the information, such as pixels representing floors and background clusters, is not directly relevant to human subjects or informative enough to represent human activities. Accordingly, it is highly desirable to extract compact, discriminative features from color-depth visual data to effectively encode human activities. Our 4-dimensional color-depth LST feature is introduced to address this important problem. The feature is defined in 4D space in the sense that it characterizes local pose, shape, and texture variations in 3D spatial dimension (i.e., xyz) and 1D temporal dimension (i.e., t).

4.3.1 Noise Reduction

Color-depth visual data obtained from the RGB-D camera usually contains a considerable amount of noise. Accordingly, noise reduction is an important process before extracting LST features. We identify three major noise sources as follows:

- Color-depth misalignment: RGB-D cameras acquire color and depth information independently; as a consequence, the obtained color and depth images can be missaligned. To reduce this misalignment noise, a color-depth camera should be calibrated by adjusting its intrinsic parameters, such as focal distances, distortion coefficients and image centers, in order to accurately map between depth pixels and color pixels. For example, as depicted in Figure 4.2, the depth pixels are mapped to their respective color pixels.

- Improper auto white balance: the color sensor of RGB-D cameras uses an auto white balance mechanism, which usually causes a significant fluctuation of the RGB value of a pixel under minor variations in the light. To handle this noise source, histogram equalization is applied over RGB images to reduce the white balance fluctuation.

- Depth sensing defect: The depth sensor of RGB-D cameras captures depth by projecting discrete infrared (IR) patterns on the scene and measuring their displacement. Due to the limitation of this depth sensing technology, acquired

depth data often contains a large number of pixels with missing values, which can result from occlusions of the depth camera's point of view or the absorption of the IR light by objects. To handle this type of noise, erosion and dilation [Gonzalez and Woods, 2007] are performed to remove noisy pixels and small structures in depth images; then, hole filling using morphological reconstruction [Gonzalez and Woods, 2007] is applied on black regions to estimate depth for pixels with missing depth values.

The resulting color-depth visual data serves as the input to our multi-channel LST feature extraction algorithm to compute the CoDe4D LST features in $xyzt$ space.

4.3.2 Spatio-Temporal Filtering

We denote the color-depth visual data (e.g., 3D point cloud sequences or color-depth videos) as a sequence of 3D frames $\{I_1, \cdots, I_T\}$. The 3D frame at time point t is denoted by $I_t = (x, y, z, i, t)$, $\forall t \in [1, T]$, where x and y represent pixel locations in the image; z is the pixel's depth value in the range of 0 to 255 that is typically mapped from physical range of 0 to 8 meters obtained by depth cameras; and i is the intensity value computed from its respective RGB values. It is noteworthy that depth values can be considered as a function $\mathbb{R}^3 \rightarrow \mathbb{R} : z = z(x, y, t)$, which constitutes a hyper-surface in 4D space represented as $S(x, y, t, z(x, y, t)) = 0$.

The first step to detect LST interest points is to incorporate space-time information. To achieve this objective, we propose a separable filtering algorithm that employs independent 3D spatial and 1D temporal filters on each intensity-depth pixel to consider spatio-temporal variations in $xyzt$ space. To incorporate spatial variations, a pass-through filter and a Gaussian filter are applied to spatially smooth intensity and depth values of each 3D frame along xyz dimensions:

$$i_s(x, y, t) = \big(i(x, y, t) \circ f(z|\delta)\big) * p(x, y|\sigma) \tag{4.1}$$

$$z_s(x, y, t) = \big(z(x, y, t) \circ f(z|\delta)\big) * p(x, y|\sigma), \tag{4.2}$$

where '$*$' denotes convolution, and '$\circ$' represents Hadamard product (entry-wise matrix multiplication). $f(z|\delta)$ is the pass-through filter parameterized by δ, which controls the spatial scale along the depth dimension and is applied to prune pixels falling outside of the depth range:

$$f(z|\delta) = \mathbb{1}(\big|z(x, y, t) - z)\big| \leqslant \delta). \tag{4.3}$$

The function $p(x, y|\sigma)$ is a 2D Gaussian filter applied along x and y spatial dimensions. The parameter σ of the Gaussian filter controls its spatial scale:

$$p(x, y|\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|x^2+y^2\|}{2\sigma^2}}. \tag{4.4}$$

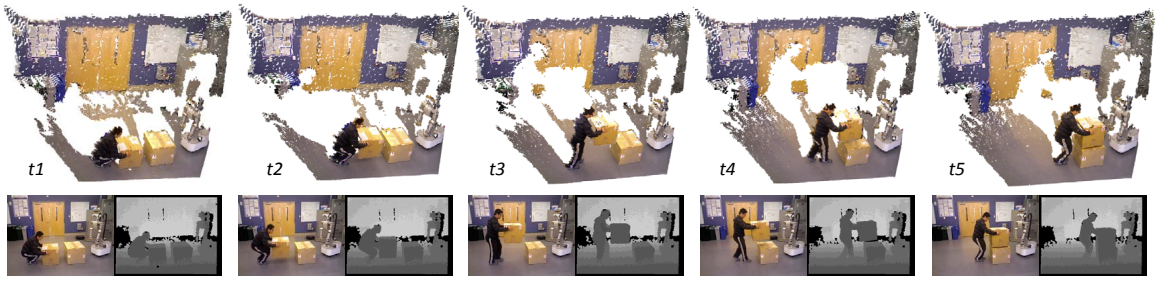


Figure 4.2: An exemplary input sequence of 3D frames (i.e., point clouds and their respective color-depth frames). Noise reduction is applied on the visual data, which aligns color and depth pixels, reduces auto-balancing fluctuation and removes depth noise. In this example, a human subject is performing a box-lifting activity in a human-robot collaboration application.

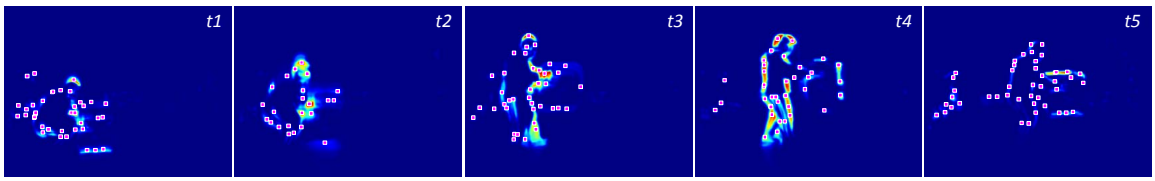


Figure 4.3: The spatio-temporal saliency map that combines both color and depth information to characterize space-time variations of poses, textures and shapes. Warmer colors in the saliency map represent stronger variations. Our spatial-temporal interest points are defined as the local maxima, which have the strongest local variations, of the saliency map in xyt space, as depicted by the magenta boxes with white edges.

To combine variations of intensity-depth pixel values across frames, a Gabor filter is applied along the time dimension over the spatially filtered 3D frames:

$$i_{st}(x, y, t) = i_s(x, y, t) * g(t|\tau, \omega) \tag{4.5}$$

$$d_{st}(x, y, t) = d_s(x, y, t) * g(t|\tau, \omega), \tag{4.6}$$

where $g(t|\tau, \omega)$ is a complex-valued Gabor filter given by:

$$g(t|\tau, \omega) = \frac{1}{\sqrt{2\pi}\tau} \cdot e^{-\frac{t^2}{2\tau^2}} \cdot e^{i(2\pi\omega t)}, \tag{4.7}$$

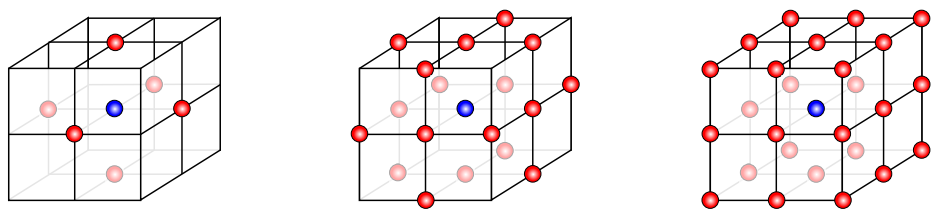
where τ controls the temporal scale of our feature detector. Throughout this work, we assign $\omega = 0.6/\tau$, which empirically shows good human activity representation and classification performance.

4.3.3 Interest Point Detection

In order to identify space-time interest points, we construct a spatio-temporal saliency map from the responses of intensity and depth filters, as follows:

$$r(x, y, t) = (1 - \alpha) \cdot \|i_{st}(x, y, t)\|^2 + \alpha \cdot \|d_{st}(x, y, t)\|^2 \tag{4.8}$$

where α is a mixture weight to balance between intensity and depth information. The spatio-temporal saliency map generally represents variations of textures, shapes and poses, because any region undergoing such variations induces responses. For example, the saliency map of the box-lifting activity in Figure 4.2 is illustrated in Figure 4.3, where warmer colors denote stronger responses, indicating that the frame has larger texture, shape and pose variations. It is noteworthy that our saliency map is defined in 3D space, which encodes variations of pixel values in xyt space.



(a) 6 connected **(b)** 18 connected **(c)** 26 connected

Figure 4.4: Pixel connectivity defined to compute the local maxima of the 3D saliency map. The blue dot denotes the query pixel; the red dots represent its connected neighbors.

Given the saliency map, our spatio-temporal interest points are defined as the local maxima of the map; that is, the pixels having the most significant variations. We employ an approach based on connected neighbors to compute local maxima of the 3D saliency map, using 6 neighbors (pixels that touch one of the faces of the query pixel, as shown in Figure 4.4a), 18 neighbors (pixels that touch one of the faces or edges, as shown in Figure 4.4b), or 26 neighbors (pixels touching one of the faces, edges, or corners, as shown in Figure 4.4c). As an example, the spatio-temporal interest points detected from the saliency map of the box-lifting activity (as shown in Figure 4.2) are illustrated in Figure 4.3, which are computed based on 18-connected neighbor pixels. Because the introduced feature detection algorithm is able to incorporate both color and depth information to select LST interest points in $xyzt$ space, our detector is referred to as the 4-dimensional color-depth (CoDe4D) LST feature detector.

4.4 CoDe4D Feature Description

After interest points are detected in space-time dimensions, which represent locations of our CoDe4D features, a feature descriptor is required to incorporate the information

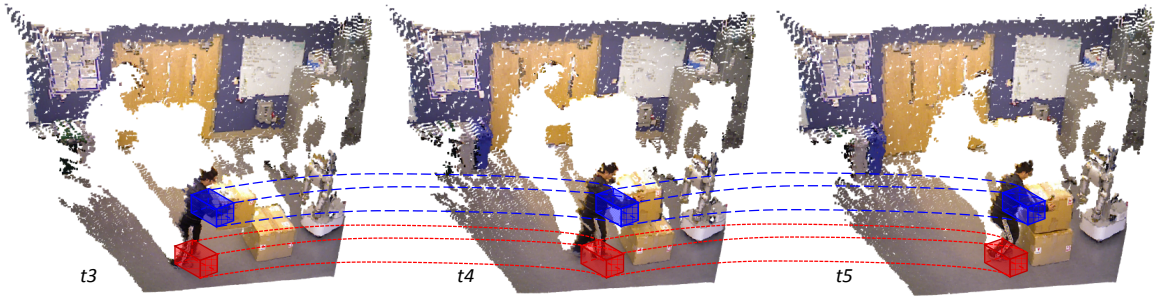


Figure 4.5: Illustration of spatio-temporal support regions in 4D ($xyzt$) space, which are used in our feature description algorithm to incorporate color-depth information in the neighborhood of detected interest points. Given a spatio-temporal interest point, which is detected within a specific time span, its 4D support region is constructed by centering a 3D (xyz) cuboid at the point of each frame within the time span (e.g., the blue and red cubes in each 3D frame) and connecting these 3D cuboids across multiple frames along 1D time dimension (as illustrated by the dashed lines across multiple 3D frames).

contained in the neighborhood of each detected interest point in order to form a final feature vector. The neighborhood of an interest point is typically encoded by a support region that is centered at the point. In this work, we define our support region $\boldsymbol{S}$ as a hyper-cuboid in $xyzt$ dimensions, which is parameterized by an octuple, i.e., $\boldsymbol{S} = (x, y, z, t, s_x, s_y, s_z, s_t)$, where (x, y, t) is the location of the spatio-temporal interest point extracted from the saliency map in xyt space; z is the depth value of the interest point, i.e., $z = z(x, y, t)$; and (s_x, s_y, s_z, s_t) represents the support region's size along 3D spatial and 1D temporal dimensions. Examples of the 4D support regions in $xyzt$ space are illustrated in Figure 4.5.

4.4.1 Adaptive Support Region

Based only on color cues, adapting the support region's size is generally a hard-to-solve problem due to the difficulties of estimating depth values from color cues. Taking advantage of the color-depth sensing technology, we can use the available depth information provided by the depth sensor to estimate 3D geometry structures of a scene, and thereby adapt the size of a support region to linear perspective view variations, i.e., an object closer to the camera seems to have a larger size. This phenomenon is illustrated in Figure 4.6. When the human is walking toward the color-depth camera, the size of the support regions should remain the same in 3D (xyz) physical space. This is because these support regions are used to incorporate information contained in local regions, such as left shoulder and right foot of the human in Figure 4.6, whose size is generally not changed. However, when the support regions are mapped onto 2D images, their size is changed due to linear perspective

view changes. In order to address this important but not well studied issue, our adaptive support region is introduced.

Since LST interest points are usually detected on boundaries (e.g., corners and edges), a number of detected points can fall out of a human blob (i.e., a region of the 3D frame that only contains pixels from a human subject), even though they are generated by humans and represent human pose, texture and shape variations. To handle this issue, we introduce a method to estimate the more accurate depth of an interest point, with the objective to adapt support region sizes to linear perspective view changes. Given scales of the space-time filters $(\sigma, \sigma, \delta, \tau)$ that are applied to detect interest points, for each interest point located at (x, y, z, t) (where z can be inaccurate if the interest point falls out of human blobs), we estimate a more accurate depth for the point using the following two steps:

1. Construct the spatial-temporal detection cuboid $\boldsymbol{C} = (x, y, z_t, t, 2\sigma, 2\sigma, 2\delta, 2\tau)$ in $xyzt$ space, which is centered at (x, y, z_t, t), where $z_t = z(x, y, t)$ is the depth value of the pixel (x, y) at time t;

2. Estimate a new depth value for the point (x, y, z_t, t) by calculating the minimum depth of the points within the spatio-temporal detection cuboid $\boldsymbol{C}$, which is mathematically defined as:

$$z(\boldsymbol{C}) = \min_{\substack{z \in [z_t - \delta, z_t + \delta] \\ \forall i \in [x-\sigma, x+\sigma] \\ \forall j \in [y-\sigma, y+\sigma] \\ \forall k \in [t-\tau, t+\tau]}} z(i, j, k). \tag{4.9}$$

Then, the estimated depth value $z(\boldsymbol{C})$ is used as the depth of the interest point. Our depth estimation approach is based on the plausible assumption of foreground humans, which is a typical situation for most color-depth camera applications in indoor environments, such as gaming [Bloom et al., 2012] and human-robot social interaction [Fanello et al., 2013]. The plausibility of the assumption can also be observed from benchmark color-depth human activity datasets, including UTK Action3D [Zhang and Parker, 2011], Berkeley MHAD [Ofli et al., 2013], ACT4^2 [Cheng et al., 2012], and MSR Daily Activity 3D [Wang et al., 2012b] datasets, in which human subjects always stay in the foreground.

After estimating $z(\boldsymbol{C})$, the support region is placed at $z(\boldsymbol{C})$ in the depth dimension, i.e., $\boldsymbol{S} = (x, y, z(\boldsymbol{C}), t, s_x, s_y, s_z, s_t)$. Then, we can adapt the size of the support region to compensate for linear perspective view changes along xy dimensions, as follows:

$$s_x = s_y = \frac{\sigma_0 \sigma}{z(\boldsymbol{C})} \tag{4.10}$$

where σ_0 characterizes the support region's relative spatial size along xy dimensions. Since the depth dimension is not affected by the linear perspective view variation, we

define $s_z = \delta_0 \delta$, where δ_0 encodes the relative spatial size in the z dimension. Similarly, the support region's temporal size is not affected by spatial linear perspective view variations, and thus we define $\tau_s = \tau_0 \tau$, where τ_0 characterizes the relative temporal size.

4.4.2 Multi-Channel Orientation Histogram

We introduce a multi-channel (color and depth) descriptor, based on image gradient orientations, to quantize visual cues within a support region in $xyzt$ space. Because a visual cue's orientation is independent of its magnitude that is affected by image noise and illumination changes, orientation quantization has proved to be a robust methodology for feature description [Dalal and Triggs, 2005, Lowe, 2004, Scovanner et al., 2007, Wang et al., 2009].

Given the support region $\boldsymbol{S}$ in $xyzt$ space that contains a set of pixels with intensity-depth values, we first decompose $\boldsymbol{S}$ into sequences of intensity and depth image patches in xyt space, i.e., $i_p(x, y, t)$ and $z_p(x, y, t)$. Then, we compute spatio-temporal gradients of the intensity patch sequence along x, y and t dimensions as follows:

$$\nabla i_p = (\frac{\partial i_p}{\partial x}, \frac{\partial i_p}{\partial y}, \frac{\partial i_p}{\partial t}), \tag{4.11}$$

where the gradient along each dimension is computed using the finite difference approximation:

$$\begin{aligned} \frac{\partial i_p(x,y,t)}{\partial x} &= i_p(x+1,y,t) - i_p(x-1,y,t) \\ \frac{\partial i_p(x,y,t)}{\partial y} &= i_p(x,y+1,t) - i_p(x,y-1,t) \\ \frac{\partial i_p(x,y,t)}{\partial t} &= i_p(x,y,t+1) - i_p(x,y,t-1). \end{aligned} \tag{4.12}$$

Spatio-temporal gradients of the depth image patch sequence, i.e., ∇z_p, can be computed in the same way.

We quantize the gradients of image patch sequences in the support region using a spherical coordinate based approach. For each spatio-temporal intensity image gradient vector, we compute its azimuth $\theta(\nabla i_p)$ and elevation $\varphi(\nabla i_p)$ angles to characterize its 3D orientations in xyt space, as follows:

$$\theta(\nabla i_p) = \arctan \frac{\partial i_p}{\partial y} \Big/ \frac{\partial i_p}{\partial x} \tag{4.13}$$

$$\varphi(\nabla i_p) = \arctan \frac{\partial i_p}{\partial t} \Big/ \sqrt{\frac{\partial^2 i_p}{\partial y} + \frac{\partial^2 i_p}{\partial x}}\,. \tag{4.14}$$

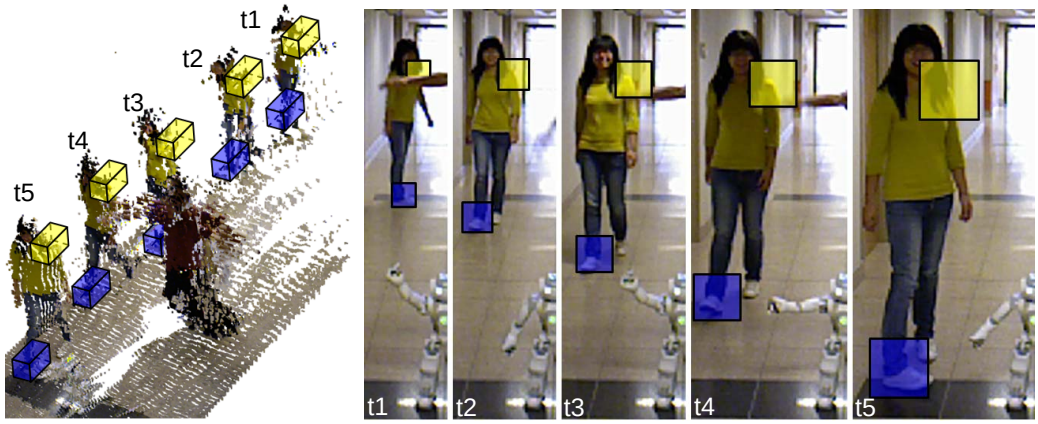


Figure 4.6: Illustration of linear perspective view changes. Support regions that have the same size in 3D (xyz) physical space have different projected sizes when they are mapped onto 2D (xy) images, due to linear perspective view changes, as shown by the smaller support regions on 2D images when the human subject performs activities further away from the camera.

An intuitive explanation of azimuth and elevation computation is illustrated in Figure 4.7a. Then, the azimuth θ and elevation φ angles of each image gradient are discretized using 2D bins, as graphically explained in Figure 4.7b. Finally, a 1D histogram is formed through concatenating all entries of the 2D bins. A histogram of 3D gradient orientations of depth patch sequences can be computed using the same procedure.

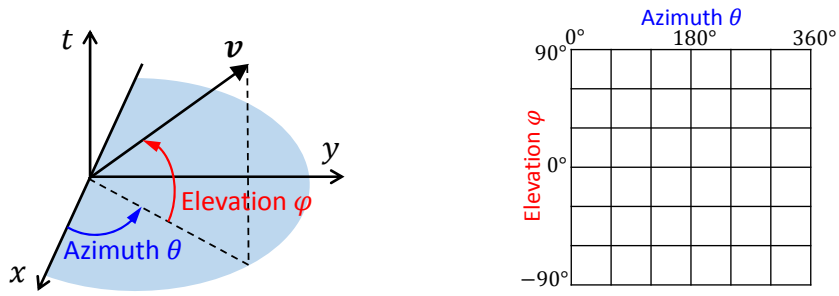
In order to construct a final feature vector that contains both intensity and depth information, we implement the *Multiple-Channel Orientation Histogram* (MCOH) descriptor, based on the histograms of the intensity and depth image patch gradient orientations. To deal with adaptive support region size, which can lead to a different number of elements in the histograms, we apply normalization on the histograms. Specifically, given the histograms of intensity and depth gradient orientations, $\boldsymbol{h}_i$ and $\boldsymbol{h}_z$, the final feature vector $\boldsymbol{h}$ is constructed by:

$$\boldsymbol{h} = \left(\frac{\boldsymbol{h}_i}{2N_i}, \frac{\boldsymbol{h}_z}{2N_z} \right), \tag{4.15}$$

where N_i and N_z are the total number of gradient orientations in $\boldsymbol{h}_i$ and $\boldsymbol{h}_z$, respectively.

4.5 Human Activity Recognition

We briefly describe our video representation and classification approaches, which are applied with our CoDe4D features to construct a complete system to recognize human activities from color-depth visual data. It is noteworthy that we are not



(a) Orientation computation **(b)** Orientation quantization

Figure 4.7: Illustration of 3D feature description based on spherical coordinates. Each 3D gradient is decomposed into two independent azimuth and elevation angles, as depicted in Figure 4.7a. Then, the angles are quantized via binning, as depicted by the example in Figure 4.7b, which subdivides azimuth and elevation angles into six bins each, leading to a histogram of 36 bins.

constructing new representation and classification algorithms; rather, we intentionally use existing benchmark representation and classifier in combination with our novel CoDe4D features to emphasize the performance gain resulting specifically from our features.

4.5.1 Representation

We apply the standard Bag-of-Features (BoF) representation to encode visual data as well as human activities, which is the most widely used representation based on LST features [Wang et al., 2009, Kläser et al., 2008]. An overview of our BoF encoding method is graphically presented in Figure 4.8.

The BoF representation requires a visual vocabulary. To this end, we construct our vocabulary using clustering, which is shown to be robust against scale changes and camera motions [Niebles et al., 2008]. We employ the standard k-means algorithm to cluster a subset of randomly selected CoDe4D features. Each cluster is indexed by a visual word. Then, each feature is assigned to its nearest visual word using Euclidean distance. During clustering, random feature selection is used to reduce computational complexity; we also execute the k-means algorithm multiple times using different initializations to obtain a vocabulary that has the lowest error (i.e., within-cluster sum of squares).

Vocabulary construction is a most important component in the BoF representation, since it can significantly reduce feature dimensions: each feature vector is encoded by a single visual word. Then, each instance of visual data (e.g., a sequence of 3D point clouds or color-depth frames) can be represented as a histogram of visual word occurrences.

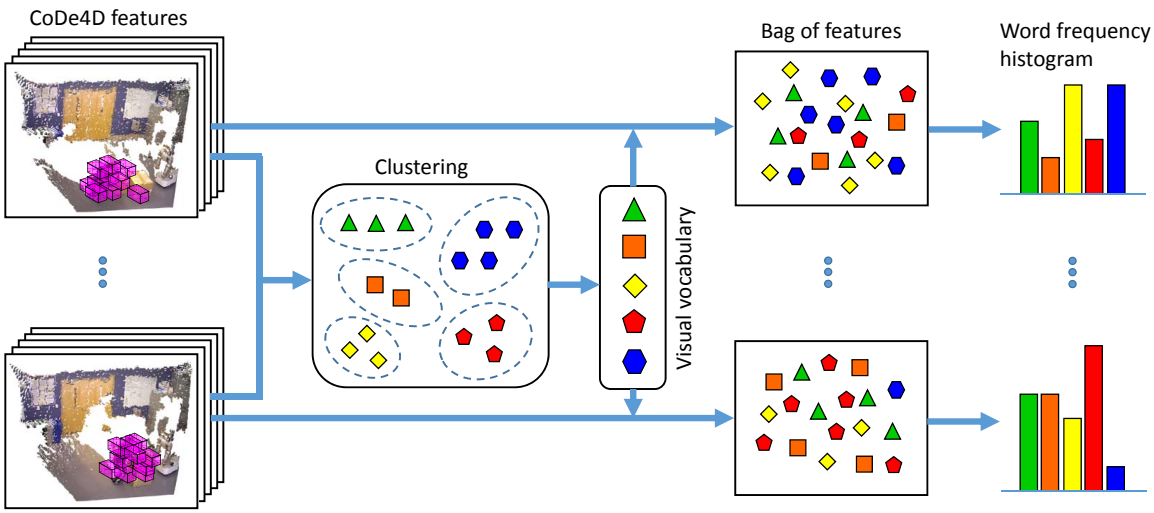


Figure 4.8: Illustration of the bag-of-features encoding for video representation. CoDe4D LST features extracted from color-depth visual data are clustered to construct a visual vocabulary, with each cluster representing a group of similar features. Then, each color-depth data instance is represented using the bag-of-features model, which is eventually encoded as a histogram of visual word frequency.

4.5.2 Classification

We apply Support Vector Machines (SVMs) as a benchmark classifier to perform activity recognition. In order to deal with the discrete BoF representation, which serves as the input to SVMs, we use the χ^2-kernel [Vedaldi and Zisserman, 2012]. Given two histograms $\boldsymbol{h}_a = \{h_{ak}\}$ and $\boldsymbol{h}_b = \{h_{bk}\}$, the kernel is computed by:

$$K(\boldsymbol{h}_a, \boldsymbol{h}_b) = \exp\left(-\frac{1}{A} D(\boldsymbol{h}_a, \boldsymbol{h}_b)\right), \tag{4.16}$$

where $D(\cdot)$ is the χ^2-distance defined as:

$$D(\boldsymbol{h}_a, \boldsymbol{h}_b) = \frac{1}{2} \sum_k \frac{(h_{ak} - h_{jk})^2}{h_{ak} + h_{jk}}, \tag{4.17}$$

and A is a constant denoting the average χ^2-distance between all pairs of N training instances [Zhang et al., 2007]:

$$A = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} D(\boldsymbol{h}_i, \boldsymbol{h}_j). \tag{4.18}$$

For multi-class activity classification, the standard one-against-all methodology is used [Chang and Lin, 2011].

4.6 Experiments

In this section, we detail the empirical study performed to evaluate our CoDe4D LST features' performance on recognizing human activities from color-depth visual data. We use four benchmark color-depth human activity datasets to evaluate our feature's performance on activity recognition: the UTK Action3D, Berkeley MHAD, ACT4[2] and MSR Daily Action 3D datasets.

4.6.1 Implementation

In our CoDe4D LST feature detector, we set the spatial scale along xy dimensions to $\sigma = 5$ pixels and $\delta = 0.3$ meters in the z dimension, and assign the temporal scale to $\tau = 3$ frames. We apply the color-depth mixture parameter $\alpha = 0.75$. We use 18-connected neighbors to select local maxima of the saliency map in xyt space. These parameter values can result in satisfactory activity recognition performance in general situations. Further explanations regarding the parameter selection process will be discussed in Section 4.6.3.

When implementing our adaptive feature support region, we set the relative spatial sizes to $\sigma_0 = \delta_0 = 5$ and the relative temporal size to $\tau_0 = 4$. In our MCOH descriptor, we divide elevation angle φ into 6 bins and azimuth angle θ into 12 cells, resulting in a final feature vector containing 72 elements.

In the BoF encoding, we randomly select 100,000 CoDe4D LST features extracted from the training set of a given dataset to construct a vocabulary containing 2000 visual words, which empirically shows promising activity recognition performance (as will be discussed in Section 4.6.3). The vocabulary construction process is repeated 8 times using different initializations; the result with the minimum clustering error is selected as our final vocabulary. A total number of 500 CoDe4D LST features that have the largest values in the saliency map are used to construct the histogram of word occurrences from each data instance. The histogram is used as input to SVMs.

4.6.2 Activity Recognition Evaluation

Using the above mentioned benchmark color-depth activity datasets, we evaluate the performance of our CoDe4D features, combined with the BoF representation and SVM classifier, on activity recognition. In addition, we compare our system with state-of-the-art activity recognition methods based on the BoF model using color-depth visual data.

UTK Action3D

In this experiment, the dataset is divided into training and testing sets: the training dataset contains 22 color-depth instances; the remaining 11 instances are used for

testing. We adopt accuracy as our measure to evaluate our system's recognition performance.

Table 4.1: Confusion matrix obtained by our CoDe4D features over the UTK Action3D dataset. Each column corresponds to the predicted category and each row corresponds to the ground truth category.

	Lifting	Removing	Waving	Pushing	Walking	Signaling
Lifting	88.1	11.9				
Removing	13.4	86.6				
Waving			100			
Pushing	2.1	0.9		97.0		
Walking				5.7	94.3	
Signaling			2.6			97.4

The confusion matrix obtained by our activity recognition system based on CoDe4D LST features is presented in Table 4.1. It can be observed that our algorithm is able to accurately recognize human activities from color-depth visual data. There are several important phenomena that are worth noting. First, our CoDe4D LST feature is capable of encoding time information, which is indicated by the successful separation between "lifting" and "removing" activities. Because the BoF encoding and the SVM classifier used in our approach are not capable of modeling time, we can infer that the separation between the sequential activities results from our CoDe4D feature. On the other hand, it can be also observed that there exists a large confusion between "lifting" and "removing". This is because local spatial-temporal features generally cannot capture long-term temporal dependencies, due to the fact that LST features only incorporate information contained in the support region, which contains only several frames. Second, human activities such as "pushing" and "walking", in which the subject crosses the entire horizontal view field of the color-depth camera, are often misclassified by several other activities. For instance, the activity "pushing" is misclassified as "lifting" and "removing". This phenomenon can be partially explained by the observation that these activities share similar atomic motions with other activities; that is, "pushing", "lifting" and "removing" contain similar box-holding and body-moving motions. These similar motions can cause overlaps between the feature sets generated by the activities, which often lead to classification errors.

The recognition system using our CoDe4D features obtains an accuracy of 93.9% over the UTK Action3D dataset. We compare the performance of our system with the baseline methods using the cuboid detector and descriptor [Dollár et al., 2005], which is applied on a sequence of either color or depth images. The comparison results are presented in Table 9.5. It is observed that our CoDe4D LST features improve recognition accuracy by around 1.3% over depth-cuboid features and around 2% over color-cuboid features. In addition, we compare our activity recognition system with the approaches reported in our previous work [Zhang and Parker, 2011], which employed cuboid descriptors (extended from [Dollár et al., 2005]) and the Latent

Table 4.2: Accuracy comparison of our approach with baseline and previous methods over the UTK Action3D dataset.

Feature detector + descriptor + classifier	Precision
CoDe4D + Color cuboid + LDA [Zhang and Parker, 2011]	77.7%
CoDe4D + Depth cuboid + LDA [Zhang and Parker, 2011]	85.5%
CoDe4D + Color-depth cuboid + LDA [Zhang and Parker, 2011]	91.5%
Color cuboid detector and descriptor [Dollár et al., 2005] + SVM	90.9%
Depth cuboid detector and descriptor [Dollár et al., 2005] + SVM	92.6%
Cuboid + Adaptive MCOH + SVM	93.2%
CoDe4D + HOG/HOF + SVM	92.8%
CoDe4D + Adaptive MCOH + SVM	**93.9%**

Dirichlet Allocation (LDA) [Blei et al., 2003] model to recognize human activities, as presented in Table 9.5. It is observed that by introducing the adaptive MHOH descriptor as well as applying supervised SVM classifiers, the activity recognition approach significantly outperforms our previous approaches on the UTK Action3D dataset. From Table 9.5, we also observe that features incorporating both color and depth information perform much better than methods based only on color or depth cues, which highlights the importance of encoding color and depth cues in LST feature design.

Berkeley MHAD

Following the experimental setup in [Ofli et al., 2013], the first seven subjects are adopted for training and the last five subjects for testing. Experiments and comparisons are conducted based on channel three (C-3) of the depth-layered multi-channel data, which generally results in superior performance, as demonstrated in the original work [Ofli et al., 2013]. Accuracy is used as evaluation metric to assess human activity recognition performance in this experiment.

Table 4.3: Comparison of average recognition accuracy over C-3 color-depth data from the Berkeley MHAD dataset.

Feature detector + descriptor + classifier	Precision
Depth Harris3D + HOG/HOF + 1-NN [Ofli et al., 2013]	77.4%
Depth Harris3D + Depth HOG/HOF + 3-NN [Ofli et al., 2013]	76.3%
Depth Harris3D + HOG/HOF + SVM [Ofli et al., 2013]	70.0%
Depth Harris3D + HOG/HOF + MKL-SVM [Ofli et al., 2013]	91.2%
Color cuboid detector and descriptor [Dollár et al., 2005] + SVM	90.5%
Depth cuboid detector and descriptor [Dollár et al., 2005] + SVM	88.7%
Cuboid + Adaptive MCOH + SVM	92.1%
CoDe4D + HOG/HOF + SVM	91.7%
CoDe4D + Adaptive MCOH + SVM	**92.4%**

We obtain an average activity recognition accuracy of 93.7% over C-3 color-depth data from the Berkeley MHAD dataset. We observe similar phenomena as what we obtained from the experiments using the UTK Action3D dataset, including the ability of our CoDe4D LST feature to capture short-term time dependencies. We compare our approach with several baseline approaches using the cuboid detector and descriptor [Dollár et al., 2005]. In addition, we compare with state-of-the-art approaches, such as SVMs with Multiple Kernel Learning (MKL) [Ofli et al., 2013], which are evaluated using the same color-depth data from the Berkeley MHAD dataset. We present our comparison results in Table 4.3. It can be observed that our system, based on the CoDe4D LST features, obtains state-of-the-art accuracy on the C-3 color-depth data from the Berkeley MHAD dataset and outperforms the baseline and previous methods.

ACT4^2

Following the experimental setup used in [Cheng et al., 2012], eight human subjects are used for training and the remaining for testing; precision is used as our evaluation metric to assess activity recognition performance. Using this experimental setting, we train our activity recognition system using the training set, and evaluate its performance over the testing set.

Table 4.4: Comparison of average recognition precision over the ACT4^2 dataset.

Feature detector + descriptor	Precision
Harris3D + Color-HOG/HOF [Cheng et al., 2012]	64.2 %
Depth layered multi channel STIPs + HOG/HOF [Ni et al., 2011]	66.3%
Harris3D + Depth-HOG/HOF [Cheng et al., 2012]	74.5%
Harris3D + Comparative coding descriptor [Cheng et al., 2012]	76.2%
Harris3D + Super feature representation [Cheng et al., 2012]	80.5%
Color cuboid detector and descriptor [Dollár et al., 2005]	70.9%
Depth cuboid detector and descriptor [Dollár et al., 2005]	78.8%
Cuboid + Adaptive MCOH	80.4%
CoDe4D + HOG/HOF	79.2%
Our CoDe4D + Adaptive MCOH	**81.9%**

An average human activity recognition precision of 81.9% is obtained over the ACT4^2 testing dataset, using the proposed CoDe4D LST features. Our activity recognition system is able to distinguish sequential activities including sitting down and standing up, due to our feature's capability of encoding short-term temporal dependencies. Table 9.4 presents comparisons of our complete recognition system with baseline algorithms and methods that obtain the previous state-of-the-art performance on the dataset. It is observed that the proposed CoDe4D LST features outperform previous LST features on human activity recognition over the ACT4^2 dataset.

MSR Daily Activity 3D

We follow the experiment setup used in [Xia and Aggarwal, 2013] in our experiments; accuracy is employed as the performance metric. The experimental results over this dataset is reported in Table 4.5. An average activity recognition accuracy of 86.0% is obtained, using our CoDe4D LST features. Comparisons of our approach with baseline algorithms and previous state-of-the-art methods are also presented in Table 4.5. In addition, to separately evaluate the performance of the Code4D detector and adaptive MCOH descriptor, we conduct the following experiments: (1) comparing CoDe4D detectors with benchmark Cuboid detectors, using the same adaptive MCOH descriptor; (2) comparing MCOH descriptors with benchmark HOG/HOF descriptors, using the same CoDe4D detector; and (3) comparing adaptive MCOH descriptors with MCOH descriptors, using the same CoDe4D detector. These comparisons are reported in Table 4.5, which demonstrate that either CoDe4D detectors or adaptive MCOH descriptors can improve activity recognition accuracy; best performance can be achieved by combining both algorithms.

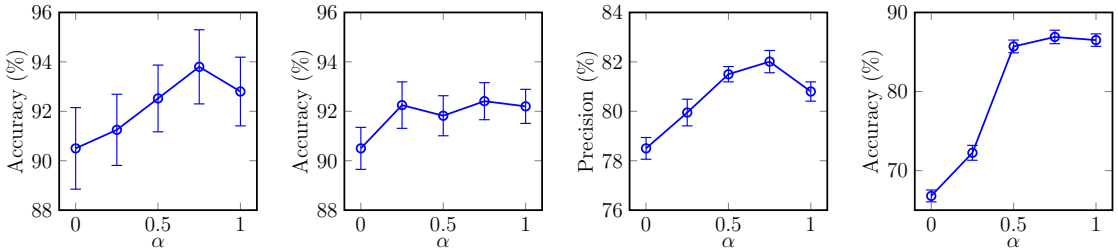
Table 4.5: Comparison of average human activity recognition accuracy over the MSR Daily Activity 3D dataset.

Features	Accuracy
Local occupancy pattern features [Wang et al., 2012b]	42.5%
Joint position features [Wang et al., 2012b]	68.0%
Harris3D + Depth-HOG/HOF [Laptev et al., 2008]	79.1%
Depth cuboid detector and descriptor [Dollár et al., 2005]	73.6%
Depth cuboid similarity features [Xia and Aggarwal, 2013]	83.6%
Actionlet Ensemble [Wang et al., 2012b]	85.8%
Cuboid + Adaptive MCOH	83.4%
CoDe4D detector + HOG/HOF	85.1%
CoDe4D + Adaptive MCOH	**86.0%**

4.6.3 Sensitivity Analysis

In this section, we focus on evaluating the sensitivity of our CoDe4D features to a variety of algorithm parameters that are critical for achieving satisfactory human activity performance. Specifically, we investigate our CoDe4D detector's parameters, including color-depth mixture weight, depth scale, and number of neighbors that define local maxima in the 3D saliency map. In addition, we analyze parameters of our MCOH descriptor, including number of cells used to divide elevation and azimuth angles. Finally, we investigate how human activity recognition performance is affected by vocabulary size and number of features per instance, when applying our CoDe4D LST features to form the BoF representation. We analyze our CoDe4D LST feature's sensitivity using 3-fold cross-validation over training sets. This learning-evaluation

procedure is performed three times, each applying a different subset for validation. When analyzing sensitivity to a specific parameter, other parameters are set to the values as listed in Section 4.6.1.



(a) UTK **(b)** MHAD **(c)** ACT4^2 **(d)** MSR

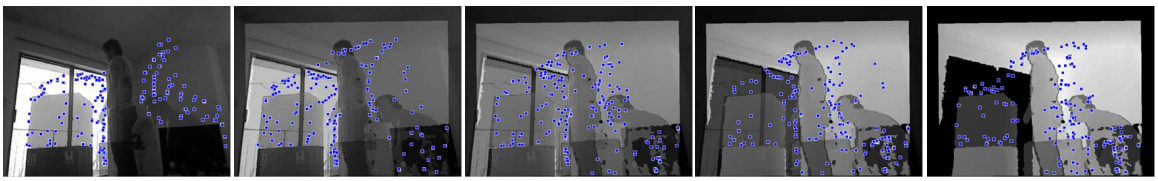
Figure 4.9: Sensitivity to the color-depth mixture weight α over different benchmark human activity datasets. Error bars denote deviations resulting from cross-variation.

Color-depth mixture weight

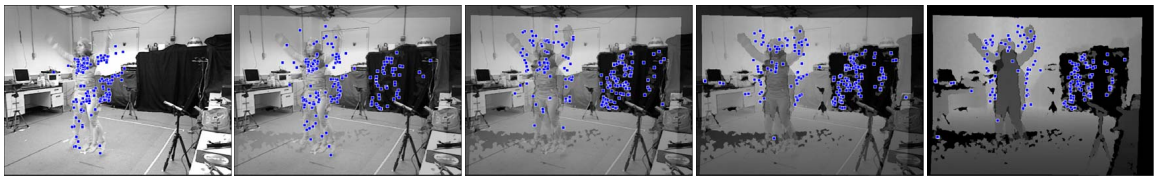
The parameter α is used to balance between color and depth information. Human activity recognition performance over different datasets using different α values is graphically presented in Figure 4.9. It is observed that in general, depth cues provide more helpful information than color cues. As shown in Figure 4.9, we obtain the best recognition performance when using $\alpha = 0.75$ for the UTK Action3D and ACT4^2 datasets; when $\alpha \in [0.25, 1]$, we obtain good activity recognition accuracy for the Berkeley MHAD dataset; when $\alpha \in [0.75, 1]$, we achieve satisfactory accuracy for MSR Daily Activity 3D dataset.

We observe in our experiments that the choice of the color-depth mixture weight value depends on characteristics of the application. For some datasets (e.g., the UTK Action3D) that have bad lighting conditions, significant illumination variations, and dynamic background resulting from screens, monitors or TVs, depth information is more important. For other datasets, color information can be weighted more than depth. When the scene is highly cluttered or there exist objects that can absorb infrared lights projected by the color-depth camera (as in the Berkeley MHAD dataset), depth images generally become very noisy and can contain a large number of black holes with missing depth values and temporally varying shapes. In these cases, using a smaller α to emphasize color information often leads to better recognition performance.

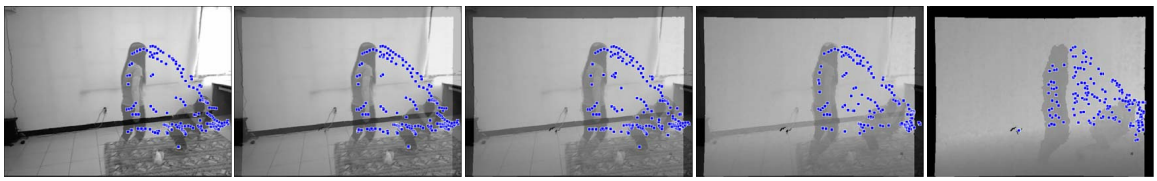
In order to provide an intuitive analysis of how the color-depth mixture weight affects our feature extraction algorithm, we extract CoDe4D LST features from an exemplary instance of the UTK Action3D, Berkeley MHAD, ACT4^2, and MSR Daily Activity 3D datasets. We draw CoDe4D features on an image that fuses two representative intensity frames with their respective depth frames, as shown in Figure



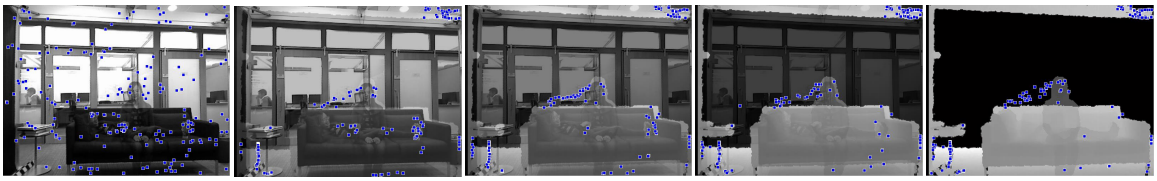
(a) Box-lifting activity from the UTK Action3D dataset



(b) Jumping-jacks activity from the Berkeley MHAD dataset (without using depth-layered multi-channel data).



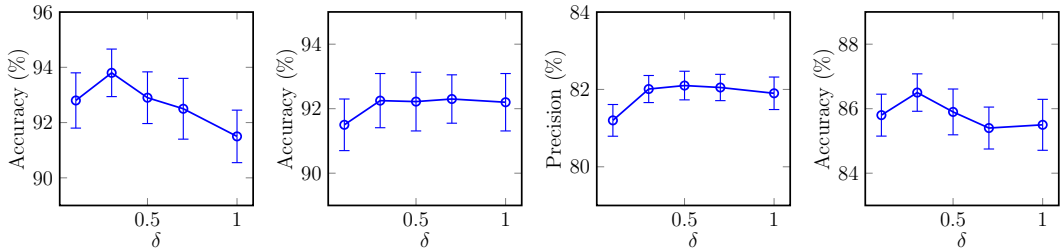
(c) Stumbling activity from the ACT4^2 dataset.



(d) Lying on sofa activity from the MSR Daily Activity 3D dataset.

Figure 4.10: Illustration of CoDe4D LST features extracted from an instance using different values of color-depth mixture weight, i.e., $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$. For a clear display, the detected features are projected onto a 2D image that is constructed through fusing two representative color and depth frames, using the mixture weight α. A total number of 200 CoDe4D features with largest values in the saliency map are drawn in the figure, using blue boxes with white boundaries.

4.10. It is observed that using different α values generally results in different sets of spatio-temporal interest points. Moreover, as observed from Figure 4.10b, objects that can absorb infrared lights, such as the black cloth in the background, often introduce a large amount of noisy features, which are not relevant to human activities and thus often not helpful to the recognition system. This observation intuitively illustrates why the depth-layered multi-channel approach, as used by the original work [Ofli et al., 2013] and this work, is a necessary component to recognize human activities from the color-depth Berkeley MHAD dataset.



(a) UTK **(b)** MHAD **(c)** ACT4^2 **(d)** MSR

Figure 4.11: Sensitivity to the depth scale δ over different benchmark datasets. Error bars denote deviations in cross-variation.

Depth scale

The parameter δ controls the spatial scale along the depth dimension of the cuboid used to detect interest points. Pixels falling outside of the cuboid are not used by our CoDe4D feature detector when building the saliency map. This parameter represents physical distance and is measured using meters. Human activity recognition performance over different datasets using different δ values is graphically presented in Figure 4.11. It is observed that when $\delta = 0.3$, our approach generally achieves the best performance over all used datasets. Another interesting observation is that when humans perform activities in open areas as in the Berkeley MHAD and ACT4^2 datasets, our approach is not very sensitive to the depth scale δ. On the other hand, when humans stay close to or interact with other objects as in the UTK Action3D and MSR Daily Activity 3D datasets, a smaller δ is preferred.

Neighborhood connectivity

This parameter defines how to select local maxima from our saliency map in xyt space, which can take values from a finite set $\{6, 18, 26\}$. The activity recognition performance is compared using a different number of connected neighbors over different datasets. We present our comparison results in Figure 4.12. From this figure, we can observe that the performance of our CoDe4D LST features is not very sensitive to the parameter of neighborhood connectivity. This can be partially explained as follows. Although the feature sets obtained using different numbers of connected neighbors can vary, given a fixed number of features to represent an instance, only features with largest values in the saliency map are used. This can lead to similar final feature sets for the data instance and thus result in similar activity recognition performance.

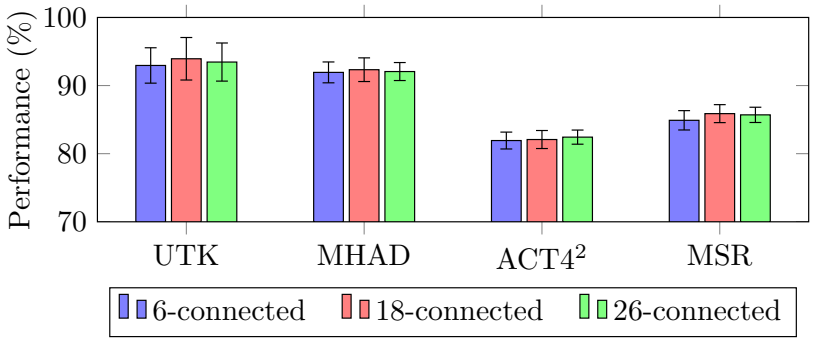
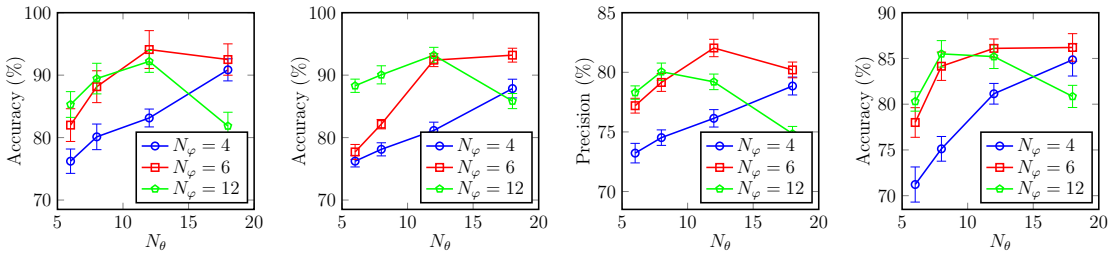


Figure 4.12: Sensitivity to the neighborhood connectivity parameter. While accuracy is used as our evaluation measure for UTK Action3D, Berkeley MHAD, and MSR Daily Activity 3D datasets, precision is used for the ACT4^2 dataset.

Numbers of angle bins

These parameters control the granularity of orientation histograms in our MCOH description algorithm. Applying different numbers of bins N_φ and N_θ on the elevation φ and azimuth θ angles respectively, we assess our system's recognition performance, as depicted in Figure 4.13. It can be observed that a moderate number of bins often leads to good activity recognition performance. A very coarse-grained subdivision often results in bad performance, because gradient orientations that are significantly different can be assigned to the same cell; consequently, the descriptor is not sufficiently discriminative. On the other hand, a very large number of bins can also reduce performance, since in this situation each cell is generally assigned with less gradient orientations; as a result, the formed feature vector is more sensitive to noise.

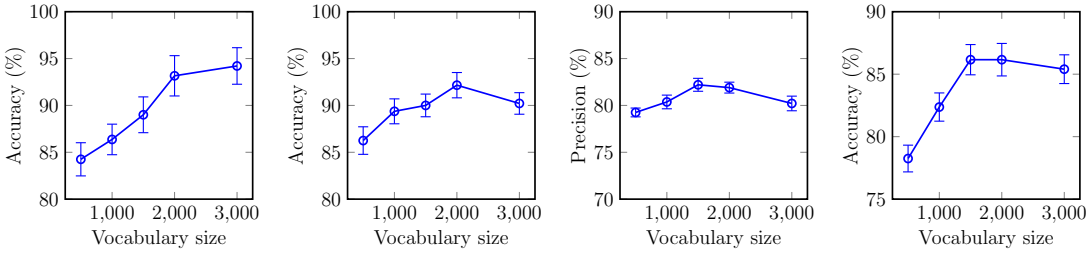


(a) UTK **(b)** MHAD **(c)** ACT4^2 **(d)** MSR

Figure 4.13: Sensitivity to the number of bins applied to subdivide the elevation angle φ and the azimuth angle θ.

Vocabulary size

This parameter controls the number of visual words obtained by clustering to encode the CoDe4D LST features for activity recognition. Variations of recognition performance using different vocabulary sizes are illustrated in Figure 9.12. It can be observed that a vocabulary that has a moderate size often leads to satisfactory recognition performance. This is because, when a small vocabulary size is adopted, features with different patterns can be incorrectly assigned to the same cluster (i.e., visual word); when a very large number of visual words are used, visual features with similar characteristics can be incorrectly assigned to different clusters.



(a) UTK **(b)** MHAD **(c)** ACT4^2 **(d)** MSR

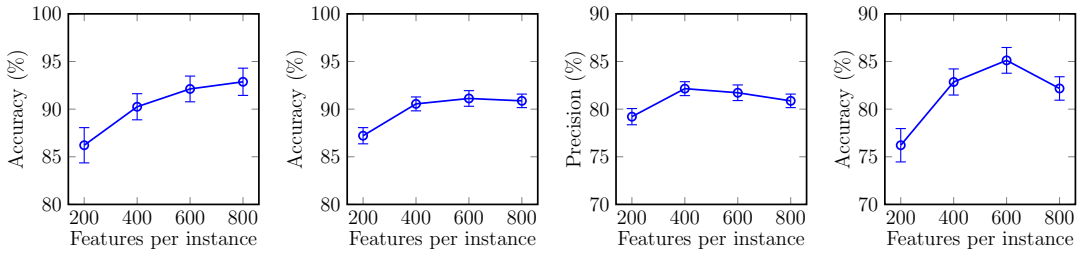
Figure 4.14: Variations of human activity recognition performance using different vocabulary sizes.

Number of features per instance

This parameter defines the total number of CoDe4D LST features to extract from each instance (e.g., 3D point cloud sequence or color-depth video). We plot variations of human activity recognition performance using different numbers of features per instance in Figure 4.15. It can be observed that extracting 400 to 600 CoDe4D features to represent each color-depth instance can generally result in satisfactory performance. While using a very small number of features can miss some important local visual cues contained in an instance, extracting a very large number of features can introduce noise, because the low-ranking features can be of poor quality (i.e., weak response in the saliency map).

4.7 Summary

We introduce a novel local spatio-temporal feature that is able to incorporate both color and depth information contained in a sequence of RGB-D frames. The features are extracted in 4-dimensional space (i.e., $xyzt$), which are able to capture 3D spatial and 1D temporal changes of human activities. To detect our 4-dimensional color-depth (CoDe4D) features, we apply a 2D Gaussian filter in the xy dimensions, a



(a) UTK **(b)** MHAD **(c)** ACT4^2 **(d)** MSR

Figure 4.15: Variations of human activity recognition performance using different numbers of features per instance.

pass-though filter in the z dimension, and a Gabor filter in the t dimension. The filtered color-depth information is used to construct a saliency map to encode changes of textures, shapes and poses. Then, local maxima of the saliency map are selected as our interest points. In order to form a feature vector for each interest point, we propose the multi-channel orientation histogram (MCOH) as our feature descriptor to encode spatio-temporal information in the neighborhood of each point. We place a support region, a hyper-cuboid in $xyzt$ space, at each interest point. Then, we compute gradients of the intensity and color patch sequences within the support region, and quantize their orientations using a spherical coordinate based approach. Our MCOH descriptor incorporates information from both color and depth channels and uses adaptive support region sizes to compensate for linear perspective view changes.

Combining our CoDe4D LST features with BoF models and SVM classifiers, we construct a complete system to recognize human activities from color-depth visual data. We evaluate the performance of the CoDe4D LST features as well as the complete system using the benchmark UTK Action3D, Berkeley MHAD, ACT4^2, and MSR Daily Activity 3D color-depth activity datasets. Experimental results demonstrate that the proposed CoDe4D LST features present satisfactory representation power and achieve the state-of-the-art activity recognition performance.

Chapter 5

Representation: SOD Descriptor

5.1 Introduction

Local spatio-temporal features have shown promising performance for human action representation and recognition in unconstrained scenarios [Dollár et al., 2005, Everts et al., 2013, Kläser et al., 2008, Marszalek et al., 2009, Scovanner et al., 2007, Wang et al., 2009, Zhang and Parker, 2011]. These features characterize local shape and motion variations, in space and time dimensions, and can provide robust representation of human actions against disturbing effects such as illumination, occlusions, and view variations, etc. Typically, local features are directly extracted from videos and thus avoid potential failures resulting from pre-processing steps, such as human segmentation. These desirable properties make these features the most popular method to recognize actions, and continue to attract increasing attention from the computer vision community [Everts et al., 2013, Xia and Aggarwal, 2013].

Feature description is a fundamental research problem in local feature extraction [Dalal and Triggs, 2005, Kläser et al., 2008, Lowe, 2004, Scovanner et al., 2007] aimed at construction of compact, descriptive representations of visual cues, including gradients and normals, computed within a feature's support region of a detected interest point. For example, the well-known scale-invariant feature transform (SIFT) [Lowe, 2004] and histograms of oriented gradients (HOG) [Dalal and Triggs, 2005] descriptors quantize 2D gradients in a support region by computing a histogram from their orientations. Because the orientation of a visual cue is independent of its magnitude, which is usually affected by image noise and illumination changes, orientation quantization has proven to be a powerful, robust approach for feature description [Dalal and Triggs, 2005, Lowe, 2004, Wang et al., 2009].

To recognize unconstrained human actions, a large number of 3D local spatio-temporal features have been recently introduced that are computed in xyt (i.e., 2D spatial and 1D temporal) space [Al Ghamdi et al., 2012, Derpanis et al., 2013, Everts et al., 2013, Kläser et al., 2008, Scovanner et al., 2007]. Although orientation description in 2D space is intuitive and well defined, description of 3D features is much more challenging. Previous methods to describe 3D feature orientations can

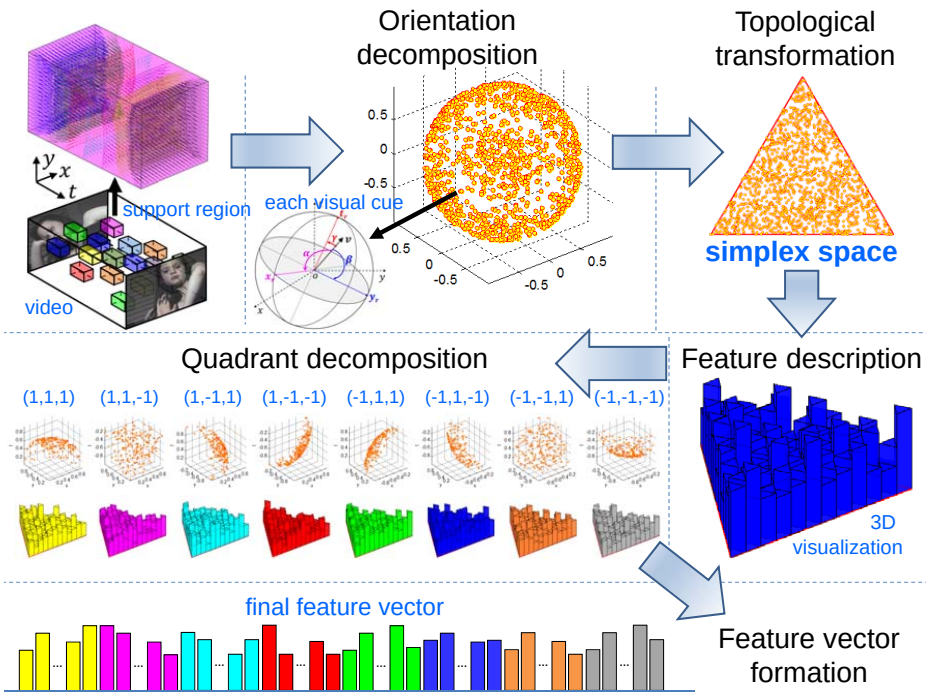
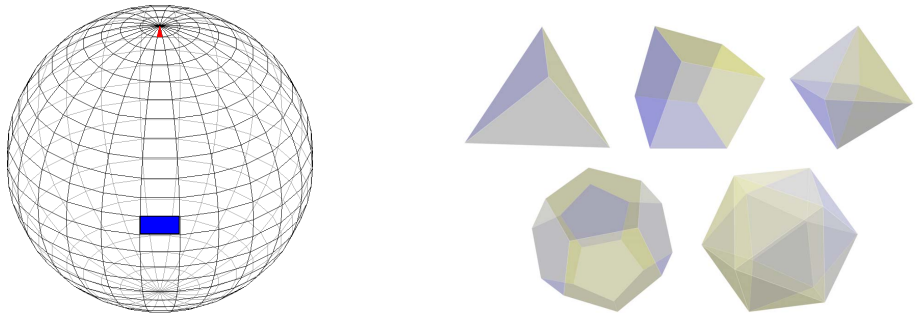


Figure 5.1: Overview of our novel *simplex-based orientation decomposition* feature descriptor to quantize and represent visual features in 3D space. Given a feature's support region containing a set of visual cues, our descriptor decomposes each cue's orientation into three angles. Then, the decomposed orientation vectors are transformed into the *simplex topological vector space*, and features are described in this space. After performing quadrant decomposition to further increase discrimination power, our SOD descriptor concatenates the histograms from all decomposed quadrants into a final feature vector.

be generally categorized into two groups: spherical coordinate-based description and regular polyhedron-based description. As shown in Figure 5.2, spherical coordinate description of 3D features suffers from the singularity issue at the poles, while regular polyhedron descriptors have limited discrimination power due to the limited number of regular polyhedrons (discussed further in Section 5.2.1).

In this chapter, we introduce a novel algorithm to describe visual features in 3D space, which addresses the singularity issue and provides a powerful description capability. The overview of our feature description algorithm is illustrated in Figure 5.1. Given the support region of a visual feature in 3D space (e.g., xyt spatio-temporal space), our description algorithm decomposes each 3D visual cue (e.g., gradients) into three dependent orientations. Then, all orientations are transformed into the standard 2-simplex topological vector space to deal with orientation dependency, and description is performed in the simplex topological vector space. Finally, to increase descriptive power, quadrant decomposition is performed to refine the quantization results. The final descriptor is a concatenated vector of the decomposed quantization



(a) Spherical coordinate based **(b)** Regular polyhedron based

Figure 5.2: Issues of previous 3D feature description methodologies: Spherical coordinate based approaches suffer from the singularity issue (Figure 5.2a): bins at the poles (red triangle) are significantly smaller than bins around the equator (blue rectangle). Regular polyhedron based approaches have limited discrimination power (Figure 5.2b), since only five regular polyhedrons exist.

results. Since our algorithm describes 3D features in the simplex topological vector space, we name it *Simplex-based Orientation Decomposition* (SOD) descriptor.

Our contributions are threefold. First, we introduce a novel simplex-based feature description algorithm to quantize and describe orientations of 3D visual features, which is an efficient, powerful, general algorithm to represent spatio-temporal (xyt) visual features in 3D space. Second, we develop visualization tools that can be applied to intuitively analyze feature characteristics in the abstract simplex topological space. Third, we empirically validate that visual features in 3D space, e.g., 3D local spatio-temporal features in xyt space, can greatly benefit from our descriptor, through demonstrating their state-of-the-art performance on unconstrained action recognition.

The remainder of the chapter is organized as follows. Section 5.2 discusses related existing studies. Then, Section 5.3 introduces our novel SOD algorithm for 3D feature description. Additional characteristics of our algorithm are discussed in Section 5.4. Experimental results are presented in Section 8.5. Finally, this work is concluded in Section 10.7.

5.2 Related Work

In this section, we discuss previous 3D visual feature description methods and briefly review existing 3D features with the focus on human action recognition applications.

5.2.1 Description of 3D Features

A naive method to describe visual features in 3D space is to directly concatenate 3D visual cues, such as 3D gradients, into a single vector [Zhang and Parker, 2011]. However, this method is not robust [Lowe, 2004, Wang et al., 2009], since the

magnitude of a visual cue is usually affected by image noise, illumination variations, etc. Because a visual cue's orientation is independent of its magnitude and is not similarly affected, orientation-based methodology dominates 3D feature description approaches.

A large number of 3D feature description methods are based on spherical coordinate systems [Flitton et al., 2013, Mattivi and Shao, 2011, Scovanner et al., 2007, Tang et al., 2012, Xia et al., 2012]. This description method applies polar angle θ and azimuthal angle ϕ in spherical coordinate systems to encode orientations and build orientation histograms. Then, θ and ϕ are divided into a set of bins, as illustrated in Figure 5.2a, which are used to construct a histogram of orientations of visual cues in a 3D feature's support region. However, as observed in [Flitton et al., 2013, Kläser et al., 2008], spherical coordinate based descriptors suffer from the *singularity issue* at the poles, as in Figure 5.2a, where the blue bin near the equator is significantly larger than the red bin at the north pole.

Another popular 3D feature description methodology is based on regular polyhedrons [Al Ghamdi et al., 2012, Everts et al., 2013, Ji et al., 2013, Kläser et al., 2008, Tian et al., 2013]. This technique approximates the orientation space by a regular polyhedron with congruent faces that are regular polygons, each of which serves as a bin. Tracing each 3D vector along its direction up to the intersection with a polyhedron face identifies the bin. Then, a feature is described using a histogram of visual cues' orientations. Since only five regular polyhedrons exist that support a maximum of 20 bins, as depicted in Figure 5.2b, this methodology has *limited discrimination power* when quantizing a large number of distinct features.

Because our SOD descriptor transforms 3D visual cues to the simplex topological vector space instead of describing them in original Euclidian space, we are able to appropriately subdivide the transformed feature space and avoid the singularity and limited discrimination power issues.

5.2.2 3D Features for Action Recognition

The large quantity of 3D spatio-temporal features proposed in recent years can be generally grouped based upon their information sources as follows:

- 3D spatio-temporal features computed in xyt spatio-temporal space using a temporal sequence of images, including 3D SIFT [Mattivi and Shao, 2011], ST-SIFT [Al Ghamdi et al., 2012], HOG3D [Kläser et al., 2008], 3D optical flow [Holte et al., 2010], CHOG3D [Ji et al., 2013], etc.

- Multi-channel 3D features, typically computed in xyt spatio-temporal space and from multiple information channels, such as RGB and depth channels, including Color-SIFT [Everts et al., 2013], 4D-LST [Zhang and Parker, 2011], etc.

The research problem we discuss in this chapter, i.e., 3D feature description, is an integral part of the methods to extract the above-mentioned features. Our SOD

Algorithm 2: Simplex-based 3D feature description

Input : $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ (3D support region),
$\mathcal{C} = \{\boldsymbol{v}_x^r, \boldsymbol{v}_y^r, \boldsymbol{v}_z^r\}$ (reference Cartesian coordinate),
k (parameter of edgewise simplex subdivision)

Output : $\boldsymbol{f}(\boldsymbol{S})$ (feature vector)

1: **for** $i \leftarrow 1$ **to** N **do**
2: Decompose the orientation of $\boldsymbol{v}_i$ by computing $\cos\alpha$, $\cos\beta$, and $\cos\gamma$ with respect to $\mathcal{C}$ acc. to Eq. (5.1);
3: Transform $\boldsymbol{v}_i$ into the standard 2-simplex topological vector space Δ^2: $\boldsymbol{\delta}_i = \{\cos^2\alpha, \cos^2\beta, \cos^2\gamma\}$;
4: Compute indices $\boldsymbol{i}(\boldsymbol{\delta_i}) = (r(\boldsymbol{\delta_i}), c(\boldsymbol{\delta_i}), l(\boldsymbol{\delta_i}))$, acc. to Eq.(5.5–5.7), of the sub-simplex in k edgewise subdivision;
5: Compute decomposed orientation quadrant assignment $\boldsymbol{q}(\boldsymbol{\delta_i})$ acc. to Eq. (5.8);
6: Increase the count of the sub-simplex indexed by $\boldsymbol{i}(\boldsymbol{\delta}_i)$ in quadrant $\boldsymbol{q}(\boldsymbol{\delta}_i)$ by one;
7: **end**
8: Form $\boldsymbol{f}(\boldsymbol{S})$ by concatenating counts of the sub-simplices in all eight quadrants;
9: **return** $\boldsymbol{f}(\boldsymbol{S})$

descriptor is mathematically proven to work with any 3D vector and can be directly applied to each of these 3D features. The universal applicability to a large number of 3D features highlights the significance of our SOD descriptor.

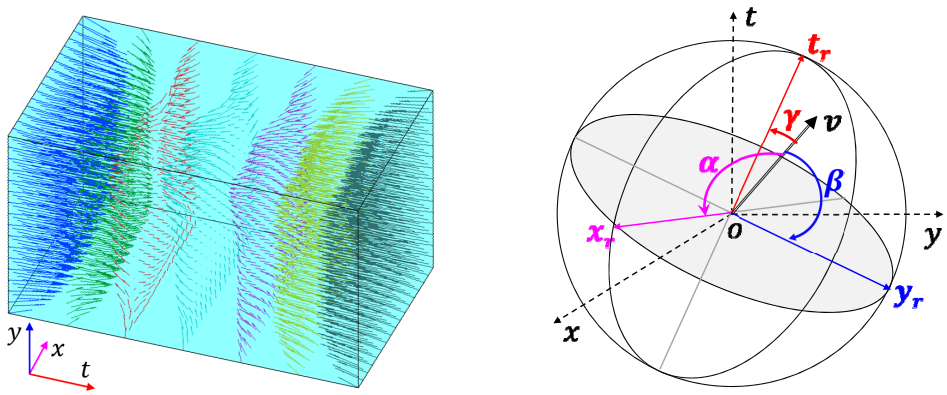
It is also worth noting that, unlike feature encoding approaches such as unsupervised k-means and supervised entropy optimization [Kuang et al., 2011], which aim to build a vocabulary of quantized features [Chatfield et al., 2011], our objective is to provide a description of each individual 3D visual feature.

5.3 The SOD Descriptor

In this section, we discuss our simplex-based orientation description algorithm. The goal is to construct a compact, representative description of 3D visual features. In particular, we describe 3D features in the simplex topological vector space to allow for appropriate subdivision of the 3D feature space. An overview of our SOD descriptor is depicted in Figure 5.1, and its algorithmic description is presented in Algorithm 2. Without loss of generality, we focus our discussion on describing 3D local spatio-temporal features that are extracted in xyt space.

5.3.1 Orientation Decomposition

The input to our SOD descriptor is the support region of a visual feature centered at a detected interest point in 3D space, which contains a set of 3D visual cues. An example of such a region containing 3D gradient cues in xyt space is visualized



(a) Input visual cues **(b)** Angle decomposition

Figure 5.3: Orientation decomposition: given a feature's support region, shown in Figure 5.3a computed from seven temporally adjacent frames, each 3D visual cue's orientation is decomposed into three angles (α, β and γ) with respect to a user-defined reference Cartesian coordinate system defined by axes x_r, y_r, and t_r (Figure 5.3b).

in Figure 5.3a. Given a support region, the goal of orientation decomposition is to decompose the orientation of each 3D visual cue into three angles.

Let $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$ denote a visual feature's support region that contains a set of 3D cues $\boldsymbol{v}_i = (x_i, y_i, t_i) \in \mathbb{R}^3$, $i = 1, \ldots, N$. Given a user-defined reference Cartesian coordinate system $\mathcal{C}$ defined by the unit vectors $\boldsymbol{v}_x^r$, $\boldsymbol{v}_y^r$ and $\boldsymbol{v}_t^r$ in the direction of x_r-axis, y_r-axis and t_r-axis, respectively, the orientation of $\boldsymbol{v}$ can be decomposed into three angles α, β, and γ with respect to the reference coordinates, which can be computed in constant time by:

$$\cos\alpha = \frac{\boldsymbol{v} \cdot \boldsymbol{v}_x^r}{\|\boldsymbol{v}\|}, \quad \cos\beta = \frac{\boldsymbol{v} \cdot \boldsymbol{v}_y^r}{\|\boldsymbol{v}\|}, \quad \cos\gamma = \frac{\boldsymbol{v} \cdot \boldsymbol{v}_t^r}{\|\boldsymbol{v}\|} \tag{5.1}$$

The definitions of the decomposed angles are illustrated in Figure 5.3b. To allow for flexible orientation decomposition, the reference coordinate system does not necessarily overlap the standard Cartesian coordinate system that is represented by the standard basis $\boldsymbol{i} = (1, 0, 0)$, $\boldsymbol{j} = (0, 1, 0)$, and $\boldsymbol{k} = (0, 0, 1)$ in the directions of x-axis, y-axis and t-axis, respectively, as shown in Figure 5.3b.

It is noteworthy that description through independently dividing α, β and γ into equally sized cells in 3D space is problematic, because the decomposed angles α, β and γ are not independent, as will be demonstrated by Eq. (5.3). For example, when α, β and γ are equally divided into six bins (a total number of 6^3 cells in 3D space), the 3D cell representing the angle range α, β, $\gamma \in [5\pi/6, \pi)$ can never be assigned by any cues, due to the constraints of the decomposed angles. We name this problem *constrained orientation quantization*, and for this reason it is not appropriate to independently discretize the angles into bins in 3D space.

5.3.2 Transformation to Simplex Space

We provide an elegant solution to the constrained orientation quantization problem to describe 3D visual features. Our novel visual feature description algorithm is based on the topological concept of simplex [Edelsbrunner and Grayson, 1999, Munkres, 1984, Rudin, 1964], which is a generalization of a tetrahedral region of space to arbitrary dimensions. Specifically, an n-simplex is the smallest closed convex set that contains $n+1$ vertices. For example, a 1-simplex is a line segment that contains two vertices, and a 2-simplex is a triangle that is specified by three vertices.

We start discussion of our novel *simplex-based orientation decomposition* descriptor by showing that each 3D cue can be transformed into a standard simplex topological vector space, where a standard n-simplex is a simplex whose edges have the same length. This is mathematically defined, in the context of a topological vector space, as follows:

Definition 1 (Standard n-simplex)**.** *The standard n-simplex is defined as a topological vector space that is the subspace of $\mathbb{R}^{n+1}$ satisfying:*

$$\Delta^n = \left\{ (\delta_0, \cdots, \delta_n) \in \mathbb{R}^{n+1} \mid \sum_{i=0}^{n} \delta_i = 1, \delta_i \geq 0, \forall i \right\} \tag{5.2}$$

Since we aim at describing visual features in 3D space, we are interested in the *standard 2-simplex* that is defined by three vertices $\Delta^2 = \{\boldsymbol{\delta}^r_\alpha, \boldsymbol{\delta}^r_\beta, \boldsymbol{\delta}^r_\gamma\}$, which can be used to represent feature vectors that take values in the space $\mathbb{R}^3$.

Given a feature's support region that contains a set of 3D visual cues (e.g., gradients), i.e., $S = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$, each 3D visual cue $\boldsymbol{v} \in S$ satisfies the following theorem:

Theorem 5.1. *Any visual cue in a 3D Cartesian space can be transformed into the standard 2-simplex topological vector space.*

Proof. For a given 3D visual cue $\boldsymbol{v} \in \mathbb{R}^3$, its orientation in 3D space can be decomposed into α, β and γ with respect to a given reference Cartesian space defined by the unit vectors $\boldsymbol{v}^r_x$, $\boldsymbol{v}^r_y$ and $\boldsymbol{v}^r_t$ (as shown in Eq. (5.1)). Assuming $\delta_\alpha = \cos^2\alpha$, $\delta_\beta = \cos^2\beta$, and $\delta_\gamma = \cos^2\gamma$, the vector representing the cue belongs to a standard simplex topological vector space, i.e., $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) \in \Delta^2$, because $\delta_\alpha \geq 0$, $\delta_\beta \geq 0$, $\delta_\gamma \geq 0$, and:

$$\begin{aligned} & \delta_\alpha + \delta_\beta + \delta_\gamma = \cos^2\alpha + \cos^2\beta + \cos^2\gamma \\ = \quad & \frac{(\boldsymbol{v}\cdot\boldsymbol{v}^r_x)^2 + (\boldsymbol{v}\cdot\boldsymbol{v}^r_y)^2 + (\boldsymbol{v}\cdot\boldsymbol{v}^r_t)^2}{\|\boldsymbol{v}\|^2} = 1 \end{aligned} \tag{5.3}$$

Thus, the 3D visual cue encoded by $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma)$ takes values in the standard 2-simplex vector space. □

The concept of simplex is rather abstract. To address this issue, we developed visualization tools for intuitive analysis of the visual cues' characteristics in the standard 2-simplex topological vector space. In the work, we also adopt these tools to intuitively explain the idea of our descriptor.

As shown in Figure 5.4a, the standard 2-simplex topological vector space can be graphically represented as an equilateral triangle on a plane. Using this representation, the element of a transformed visual cue vector $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma)$ represents the distance ratio of the projected point on the 2-simplex to its respective edge; that is:

$$\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) = \frac{1}{d_\alpha + d_\beta + d_\gamma}(d_\alpha, d_\beta, d_\gamma) \tag{5.4}$$

where $d_\alpha + d_\beta + d_\gamma = h$, and h is the height of the standard 2-simplex triangle that is computed by $h = \sqrt{3}b/2$, given the edge length b. For example, given the transformed vector $\boldsymbol{\delta} = (0.5, 0.3, 0.2)$ of the visual cue in Figure 5.3b, its projected data point on the simplex satisfies that $d_\alpha = 0.5h$, $d_\beta = 0.3h$, and $d_\gamma = 0.2h$, as illustrated in Figure 5.4a.

5.3.3 Description in Simplex Space

After projecting the 3D visual cues onto the standard 2-simplex, we discuss how to describe the transformed visual cue vectors in the standard 2-simplex topological space. In particular, we prove that the standard 2-simplex topological vector space can be subdivided into a large number of equally-sized cells, as stated by the following theorem:

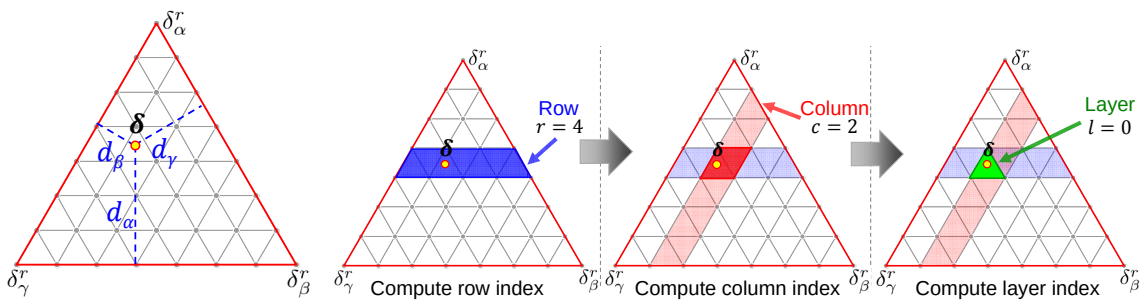
Theorem 5.2. *For every integer $k \geq 1$, there exists a subdivision of the standard 2-simplex topological vector space into k^2 standard sub-simplices that have the same size.*

Proof. Given a standard 2-simplex Δ^2 with edge length b and height h, we apply edgewise subdivision to divide Δ^2, which equally divides each edge into k segments and connects any pair of endpoints if the line segment represented by the endpoints is parallel to an edge. Then, the total number of sub-simplices is: $1+3+\cdots+(2k-1) = k^2$. Since all sub-simplices have the same edge length b/k and height h/k, they are thus standard and have the same size. □

From Theorem 5.2 arises the description power of our algorithm, which can scale without bound and therefore avoid the limited discrimination power issue of the regular polyhedron based approach. Theorem 5.2 also demonstrates that all bins (i.e., sub-simplices) have the same size, which addresses the singularity issue of the spheral coordinate based descriptor. Figure 5.4a depicts an example that subdivides the standard 2-simplex topological vector space into $k^2 = 49$ equally-sized sub-simplices.

To efficiently identify each individual sub-simplex in the standard 2-simplex topological vector space, we propose a new sub-simplex indexing method using three indices, i.e., *row*, *column*, and *layer*, which are defined as follows:

Definition 2 (Indices of sub-simplices)**.** *Given k edgewise subdivision of the standard 2-simplex $\Delta^2 = \{\boldsymbol{\delta}^r_\alpha, \boldsymbol{\delta}^r_\beta, \boldsymbol{\delta}^r_\gamma\}$, each height is divided into k intervals indexed by $1, \ldots, k$. Then,* row *and* column *are defined as the interval indices of the heights with respect to the edges opposite to $\boldsymbol{\delta}^r_\alpha$ and $\boldsymbol{\delta}^r_\beta$, respectively.* Layer *is a binary value that indicates whether a sub-simplex has a down-pointing triangular shape with respect to an edge.*



(a) Feature projection **(b)** Computation of subdivision indices

Figure 5.4: An illustrative example of our topological transformation and sub-simplex index computation in the standard 2-simplex topological vector space, when $k = 7$.

Using the row, column and layer definitions, we are able to efficiently assign each transformed visual cue vector to a sub-simplex in constant time. Given a transformed visual cue $\boldsymbol{\delta} = (\delta_\alpha, \delta_\beta, \delta_\gamma) \in \Delta^2$, our SOD algorithm computes its row r, column c and layer l indices as follows:

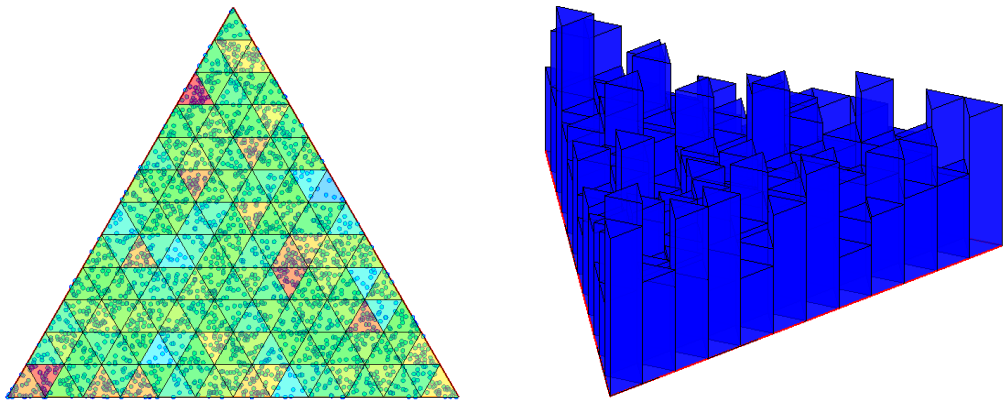
$$r(\boldsymbol{\delta}) = \lceil k\delta_\alpha \rceil + \mathbb{1}(\delta_\alpha = 0), \qquad r \in \{1, \ldots, k\} \tag{5.5}$$

$$c(\boldsymbol{\delta}) = \lceil k\delta_\beta \rceil + \mathbb{1}(\delta_\beta = 0), \qquad c \in \{1, \ldots, k\} \tag{5.6}$$

$$l(\boldsymbol{\delta}) = (r(\boldsymbol{\delta}) + c(\boldsymbol{\delta}) + \lfloor k\delta_\gamma \rfloor + \mathbb{1}(\delta_\beta \neq 1) + k) \bmod 2, \; l \in \{0, 1\} \tag{5.7}$$

where $\mathbb{1}(\cdot)$ is the indicator function that is used to deal with the special cases when $\boldsymbol{\delta}$ is projected onto the edges of the sub-simplices in the standard 2-simplex vector space.

Then, we can directly assign $\boldsymbol{\delta}$ to a sub-simplex indexed by r, c and l. An illustrative example is provided in Figure 5.4b to explain our index computation method. For the transformed 3D visual cue $\boldsymbol{\delta} = (0.5, 0.3, 0.2)$, after computing its row and column indices, i.e., $r(\boldsymbol{\delta}) = 4$ and $c(\boldsymbol{\delta}) = 2$, a diamond that contains a pair of sub-simplices is located. Then, the layer index is computed, i.e., $l(\boldsymbol{\delta}) = 0$ indicating that the sub-simplex is not upside-down, which determines the final sub-simplex assignment to the 3D visual cue.



(a) 2D view **(b)** 3D view

Figure 5.5: Visualization of the histogram of the visual cues contained in the support region of a 3D feature when $k = 12$. Figure 5.5a shows a 2D view with projection distribution of the cues, where a warmer color denotes a larger number of cues falling in the sub-simplex. A more intuitive 3D view is depicted Figure 5.5b.

After assigning all 3D visual cues in a feature's support region into their respective sub-simplices, each sub-simplex counts the number of cues assigned to it, and a histogram using these sub-simplices as bins is formed to describe the visual feature. An intuitive visualization tool is provided to investigate the histogram in the simplex topological vector space, as depicted in Figure 5.5. In particular, Figure 5.5a also visualizes the 3D visual cue's orientation distribution in the transformed simplex vector space.

5.3.4 Quadrant Decomposition

When the histogram of 3D visual cues is obtained in the simplex space, quadrant decomposition is performed to further improve the discriminative power of our SOD descriptor. Since the cosine-squared function maps all visual cues to the first quadrant and removes the signs of their orientations, the objective of quadrant decomposition is to describe the orientation signs of visual cues from different quadrants in the reference Cartesian coordinate system. There exist eight quadrants in a 3D Cartesian space that are represented by their signs $(\pm 1, \pm 1, \pm 1)$. Given the orientation of a 3D cue, its quadrant assignment is efficiently computed by:

$$\boldsymbol{q}(\boldsymbol{\delta}) = \left(\frac{\cos\alpha}{|\cos\alpha|}, \frac{\cos\beta}{|\cos\beta|}, \frac{\cos\gamma}{|\cos\gamma|} \right) \tag{5.8}$$

As a result, the orientation histogram obtained in the simplex vector space is decomposed into eight parts according to different orientation quadrants. It is noteworthy that quadrant assignments are computed with respect to a user-defined

coordinate system, which provides additional flexibility to our SOD descriptor. An example of quadrant decomposition is shown in Figure 5.1.

In order to construct a final vector to describe a 3D visual feature $\boldsymbol{S} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N\}$, all decomposed histograms in different quadrants are concatenated into a single vector $\boldsymbol{f}(\boldsymbol{S})$ that is of size $8k^2$, i.e., each of the eight orientation quadrants has a histogram formed by k^2 sub-simplices.

5.4 Discussion

5.4.1 Efficiency and Runtime

Our SOD descriptor employs cosine values to quantize feature orientations in 3D space, which are efficiently computed using the dot product. For each single 3D visual cue, orientation decomposition (Eq. (5.1)), topological space transformation (in Theorem 5.1), sub-simplex index computation (Eq.(5.5, 5.6, 5.7)), and quadrant decomposition (Eq. (5.8)) take constant time $O(1)$ to perform. Concatenation to form a feature vector takes $O(k^2)$ runtime, where k is the edgewise simplex subdivision parameter. Because typically $N \gg k^2$, i.e., the number of visual cues is much greater than the number of bins in a histogram, our SOD algorithm only takes $O(N)$ time to describe a 3D visual feature that contains N visual cues.

5.4.2 Multi-Channel 3D Features

Our SOD algorithm can be directly applied on visual features extracted from multiple channels in 3D space, which include color-depth spatio-temporal features [Zhang and Parker, 2011] that typically apply descriptors to intensity and depth image sequences in xyt space, and multi-color spatio-temporal features [Everts et al., 2013] that apply descriptors to multiple color channels of color image sequences. Following [Everts et al., 2013, Zhang and Parker, 2011], one can apply our descriptor over each channel to obtain a vector that describes 3D visual cues in that channel, and combine them together to form a final feature vector. In this scenario, the 3D visualization of our descriptor is a stacked bar plot on the standard 2-simplex.

5.4.3 High Dimensional Features

The SOD descriptor is not limited to describing features in 3D space; our methodology can be extended to quantize and describe high dimensional features. Given a d-dimensional visual cue $\boldsymbol{v} \in \mathbb{R}^d$ and a reference coordinate $\mathcal{C} = \{\boldsymbol{v}_1^r, \ldots, \boldsymbol{v}_d^r\}$, its orientation can be decomposed into d angles $(\alpha_1, \ldots, \alpha_d)$, in a manner similar to Eq. (5.1), which satisfies $\sum_{i=1}^{d} \cos^2 \alpha_i = 1$. Thus, $\boldsymbol{v}$ can be projected onto the standard $(d-1)$-simplex (i.e., an extension of Theorem 5.1). In addition, [Edelsbrunner and Grayson, 1999] showed that a $(d-1)$-simplex can be subdivided into k^{d-1} sub-simplices with the same $(d-1)$-dimensional volume using k edgewise

subdivision (i.e., an extension of Theorem 5.2). Thus, our fundamental theorems still hold, meaning the SOD descriptor can be applied to features in high dimensional space.

5.5 Empirical Study

Here we detail the experiments conducted to evaluate the performance of our SOD descriptor on action recognition. We would like to highlight that we are not constructing new classifiers and detectors; rather, we intentionally use existing benchmark classifiers and detectors in combination with our novel descriptor to emphasize the performance gain resulting specifically from our SOD descriptor.

5.5.1 Implementation and Experiment Setup

Detectors

Three detectors are adopted to detect spatio-temporal interest points from videos in xyt space. (1) **Harris3D** detector [Laptev et al., 2008] is a spatio-temporal extension of the Harris cornerness criterion that is based on the eigenvalues of a spatio-temporal second-moment matrix. We apply the original implementation [Laptev et al., 2008] and standard parameter setups $\sigma = \sqrt{2^i}$, $i = 2, \ldots, 7$ and $\tau = \{\sqrt{2}, \sqrt{4}\}$. (2) **Gabor** detector [Dollár et al., 2005] applies separable filters on spatial and temporal dimensions to select interest points in xyt space. We adopt the original implementation [Dollár et al., 2005] and standard parameter setups $\sigma = 2$, $\tau = 4$ in our experiments. (3) **Multi-channel Gabor** detector [Everts et al., 2013] detects spatial-temporal interest points using Gabor detectors to compute image responses based on intensity and normalized chromatic channels. We apply $\sigma = 2$, $\tau = 4$ as in the original work [Everts et al., 2013].

Descriptors

The size of support regions is set to $\Delta_x = \Delta_y = 8\sigma$, and $\Delta_t = 6\tau$, as in [Kläser et al., 2008, Wang et al., 2009]. The support region's size and cell layout may be optimized over a specific dataset [Kläser et al., 2008]. To maintain focus on the descriptors themselves, we refrain from such an optimization, following [Everts et al., 2013, Wang et al., 2009]. We use the standard Cartesian space as our reference coordinates. When using multi-channel detectors, the multi-channel description mechanism (discussed in Section 5.4) is applied.

Two 3D description methodologies based on **spherical coordinates**, such as 3D SIFT [Scovanner et al., 2007], and **regular polyhedrons**, such as HOG3D [Kläser et al., 2008] are used as our 3D description baselines (discussed in Section 5.2.1). Feature descriptors in previous works are also adopted as baselines to compare the feature discrimination's ability to recognize human actions.

Recognition

Following [Everts et al., 2013, Kläser et al., 2008, Wang et al., 2009], action recognition is performed in a standard bag-of-features learning framework and a codebook is created through clustering 200,000 randomly sampled features using k-means into 4000 codewords. For classification, we use non-linear SVMs with χ^2-kernels and the one-against-all approach [Everts et al., 2013, Kläser et al., 2008, Wang et al., 2009].

Experimental Setup

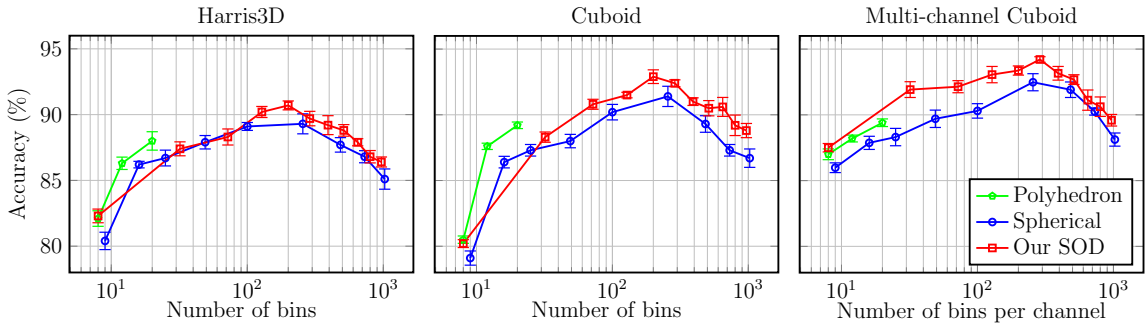
We perform experiments using three action datasets. Following [Schuldt et al., 2004], we apply the all-in-one experimental settings when using the KTH datasetand the accuracy metric as the performance measure. Following the standard settings [Rodriguez et al., 2008], performance is evaluated using accuracy in a leave-one-out cross validation framework over the UCF sport dataset. Following the standard setup of the Hollywood-2 dataset [Marszalek et al., 2009], the dataset is divided into 823 training and 884 testing examples; performance is evaluated using the precision measure.

5.5.2 Descriptor Evaluation

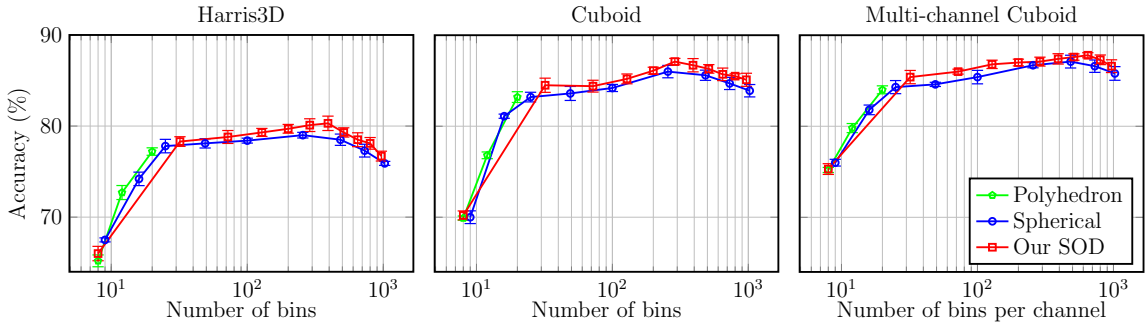
We show our SOD descriptor's superior performance by comparing it with the 3D baseline descriptors. We also investigate our descriptor's sensitivity with respect to the size of the final feature vector, which turns out to be very important but is rarely studied in previous descriptors. Sensitivity is empirically analyzed using five fold cross-validation over training sets. To focus on investigating characteristics of the descriptors themselves, no additional feature aggregation is applied, i.e., the support region is not divided into cells. Experimental results over three datasets are graphically shown in Figure 6.7. Because the baseline descriptor based on regular polyhedrons with four and six faces (i.e., bins) performs poorly, we only present the results using polyhedrons with 8, 12 and 20 faces. It is worth recalling that 20 is the maximum number of bins supported by this descriptor as it suffers from the limited discrimination power issue.

Table 5.1: Comparison of accuracy (%) on the KTH dataset.

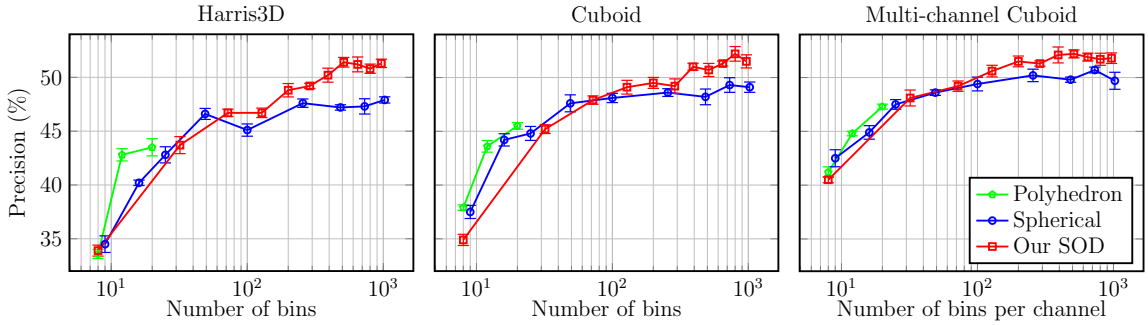
2D description methods	Acc.	3D description methods	Acc.
Harris3D + HOG [Wang et al., 2009]	80.9	Harris3D + 3D SIFT [Mattivi and Shao, 2011]	82.7
Gabor + HOG [Kläser, 2010]	82.3	Gabor + Cuboid [Dollár et al., 2005]	89.1
Gabor + HOF [Kläser, 2010]	88.2	ST-SIFT + HOG3D [Al Ghamdi et al., 2012]	90.7
Gabor + HOF/HOF [Kläser, 2010]	88.7	Gabor + HOG3D [Kläser et al., 2008]	91.4
Hessian3D + HOG/HOF [Wang et al., 2009]	88.7	Harris3D + HOG3D [Kläser, 2010]	92.4
Harris3D + HOG/HOF [Wang et al., 2009]	91.8	FAST + CHOG3D [Ji et al., 2013]	93.1
Harris3D + HOF [Wang et al., 2009]	92.1	Multi-ch. Gabor + Poly.	92.9
Oriented energy desc. [Derpanis et al., 2013]	93.2	Multi-ch. Gabor + Sphe.	93.8
Context + HOG/HOF [Han et al., 2009]	94.1	Multi-ch. Gabor + **SOD**	**94.8**



(a) KTH



(b) UCF Sport



(c) Hollywood-2

Figure 5.6: Sensitivity of our SOD descriptor and comparison with baseline 3D descriptors based on spherical coordinates or regular polyhedrons. Error bars denote standard deviations.

For all tested spatio-temporal feature detectors, our SOD descriptor significantly outperforms the 3D baseline descriptors, in general. The discrimination ability provided by the polyhedron baseline is not sufficient to represent complex actions in real-world scenarios. The performance improvement provided by our SOD descriptor over the spherical baseline highlights the advantages of quantizing and describing spatio-temporal features in the simplex topological space that can be equally subdivided into any large number of sub-simplices, thus addressing the singularity issue.

In addition, as Figure 6.7 illustrates, the descriptor's representation ability is greatly affected by the number of bins used to form the final feature vector. All descriptors generally produce poor recognition results when a small number of bins (e.g., less than 15) is used; in this case, the descriptors are not sufficiently discriminative. On the other hand, a very large number of bins (e.g., greater than 1000) also hurts recognition performance. This occurs because although the descriptors discriminate well between visual features, not enough cues fall into each bin. Another important observation is that the ideal number of bins depends on the dataset complexity; a more complex dataset usually requires a larger number of bins. For example, using around 300 bins for the KTH and UCF Sports datasets and around 600 bins for the more complex Hollywood-2 dataset generally leads to satisfactory recognition performance. In summary, our sensitivity analysis results demonstrate the importance of carefully selecting the number of bins, by considering both descriptor's discrimination ability and dataset complexity.

5.5.3 Comparison with the State of the Art

We compare our SOD descriptor with the state-of-the-art feature description methods, in terms of their performance on human action recognition. The compared methods generally follow similar experimental setups that are based on feature pooling, bag-of-features encoding and SVM-based classification. Following [Al Ghamdi et al., 2012, Derpanis et al., 2013, Everts et al., 2013, Ji et al., 2013, Wang et al., 2009], we adopt a spatio-temporal pooling scheme that divides each support region into $4 \times 4 \times 3$ cells to construct bag-of-features models.

Different descriptors are compared in Tables 5.1, 5.2 and 5.3, which show human action recognition performance over the KTH, UCF Sports and Hollywood-2 datasets, respectively. Our SOD descriptor achieves a 94.8% accuracy on KTH, a 87.5% accuracy on UCF Sport, and a 50.9% overall precision on Hollywood-2. Comparison shows that our SOD descriptor is the best-performing individual descriptor (i.e., *without* combining multiple descriptors, as in [Ullah et al., 2010]), which again shows the effectiveness of our SOD algorithm to describe local spatio-temporal features in xyt space.

Table 5.2: Comparison of accuracy (%) with state-of-the-art descriptors on the UCF Sports dataset.

2D description methods	Acc.	3D description methods	Acc.
Harris3D + HOG [Wang et al., 2009]	71.4	Gabor + Cuboids [Kläser, 2010]	76.6
Gabor + HOG [Kläser, 2010]	72.7	Harris3D + HOG3D [Wang et al., 2009]	79.7
Harris3D + HOF [Wang et al., 2009]	75.4	ST-SIFT + HOG3D [Al Ghamdi et al., 2012]	80.5
Gabor + HOF [Kläser, 2010]	76.7	Gabor + HOG3D [Kläser et al., 2008]	82.9
Gabor + HOG/HOF [Kläser, 2010]	77.7	Multi-ch. G. + HOG3D [Everts et al., 2013]	85.6
Harris3D + HOG/HOF [Wang et al., 2009]	78.1	Multi-ch. Gabor + Poly.	85.2
Hessian3D + HOG/HOF [Wang et al., 2009]	79.3	Multi-ch. Gabor + Sphe.	86.3
Oriented energy desc. [Derpanis et al., 2013]	81.5	Multi-ch. Gabor + **SOD**	**87.5**

Table 5.3: Descriptor comparison on Hollywood-2 using precision (%). '&f' denotes 'HOG/HOF combined with f features'. The compared existing descriptors include HOG3D [Kläser et al., 2008], HOG [Kläser, 2010], HOF [Kläser, 2010], HOG/HOF [Kläser, 2010], & SIFT [Marszalek et al., 2009], & context [Han et al., 2009], & global [Ullah et al., 2010].

Actions	Multi-ch. Cuboid +			Harris3D +	Harris3D + 2D descriptors					
	SOD	Poly.	Sphe.	HOG3D	HOG	HOF	HOG/HOF	& SIFT	& cont.	& glob.
AnswerPhone	18.1	15.9	17.1	16.3	11.8	11.6	15.3	13.1	15.57	**25.9**
DriveCar	**88.1**	85.8	87.2	86.3	79.0	84.8	85.8	81.0	87.0	85.9
Eat	61.6	57.8	60.7	55.8	43.4	58.6	**63.1**	30.6	50.9	56.4
FightPerson	76.2	74.5	75.8	**77.2**	60.4	72.1	71.3	62.5	73.1	74.9
GetOutCar	36.3	33.5	34.3	35.7	24.9	19.6	32.3	8.6	27.2	**44.0**
HandShake	**55.9**	51.3	53.5	55.7	36.3	50.2	49.5	19.1	17.2	29.7
HugPerson	**48.3**	46.5	47.2	47.9	29.6	30.9	38.6	17.0	27.2	46.1
Kiss	**58.4**	54.2	55.3	51.1	43.5	45.1	49.3	57.6	42.9	55.0
Run	**72.1**	67.3	69.7	71.7	62.1	68.5	67.2	55.5	66.9	69.4
SitDown	51.9	48.2	49.3	47.6	30.3	56.4	57.3	30.0	41.6	**58.9**
SitUp	22.4	18.5	20.3	22.2	16.1	8.5	**22.5**	17.8	7.2	18.4
StandUp	21.6	19.6	20.8	15.6	20.9	18.9	20.4	33.5	48.6	**57.4**
Overall	50.9	47.8	49.3	48.6	38.2	43.8	47.7	35.5	42.1	51.8

5.6 Summary

We introduce a novel *simplex-based orientation decomposition* descriptor to quantize and represent 3D visual features including local spatio-temporal features in xyt space. Our technique decomposes each 3D visual cue in a feature's support region into three angles and transforms the decomposed angles into the simplex topological vector space. Feature description is performed in the simplex space, which is able to deal with the singularity and limited discrimination power issues that were not addressed in previous works. Then, quadrant decomposition is performed to improve our descriptor's discrimination capability, and a final feature vector is formed by combining decomposed histograms from all quadrants. Extensive empirical study using three benchmark action datasets has been conducted, which shows that our descriptor significantly outperforms previous 3D feature descriptors based on spherical coordinates or regular polyhedrons and achieves state-of-the-art description power for recognition of human actions.

Chapter 6

Representation: AdHuC Features

6.1 Introduction

In this chapter, we discuss a new human representation to address the key 3D robotic vision problem of recognizing the actions of each individual in a group of humans in complex, dynamic 3D scenes. In a large number of real-world human-centered robotic applications, camera views usually contain multiple humans, and in many cases we care more about the actions of specific individuals in the group than the group as a whole. For example, robotic guards that observe multiple humans should be able to recognize each person's actions to detect abnormal activity; service robots need to be able to perceive the actions of each individual in a group in order to provide effectively for human needs; self-driving robotic cars need to be able to distinguish each pedestrian's behavior to better ensure safety.

Although a large number of approaches address single-person action recognition [Kläser et al., 2008, Xia and Aggarwal, 2013, Zhang and Parker, 2011] and group activity recognition [Choi et al., 2011b, Khamis et al., 2012, Lan et al., 2012, Ni et al., 2009], recognition of actions of a specific individual in a group has not been previously well studied. We name this essential problem *action recognition of multiple individuals* (ARMI), as demonstrated in Figure 6.1. Indoor environments, including business, hospitals, schools, etc., are the most common, real-world setting for human-centered robotic applications, in which human-robot interactions often occur. An autonomous robot in these settings will experience dynamic, cluttered environments with multiple humans present. The ability to perform ARMI in these settings, within complex 3D scenes with camera motions, background clutter, occlusions, and illumination variations is therefore extremely significant.

Among different approaches for representing human actions [Choi et al., 2011b, Khamis et al., 2012, Lan et al., 2012, Ni et al., 2009, Xia and Aggarwal, 2013], local spatio-temporal (LST) features are the most popular and promising representation. These features are generally invariant to geometric transformations; as a result, they are less affected by variations in scale, rotation and viewpoint. Because LST features are locally detected, they are inherently robust to occlusions. Use

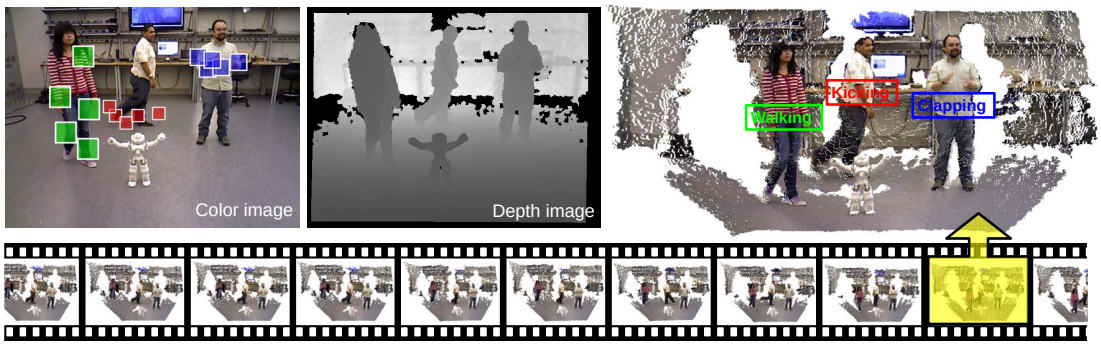


Figure 6.1: A motivating example of the ARMI task and the solution based on our novel AdHuC features. The goal of ARMI is to recognize the actions of multiple individuals in a group, such as the walking action performed by the female in the example. Our AdHuC features adopt the Depth of Interest (DOI) concept to coherently and efficiently localize humans and extract adaptive, human-centered local spatio-temporal features in $xyzt$ space. Our AdHuC is able to (1) identify *feature affiliations*, e.g., the green features are from the female, (2) avoid extracting irrelevant features from dynamic backgrounds or foreground obstacles (e.g., the humanoid robot), especially when the *camera is moving*, (3) address the *false descriptor size issue* by estimating the *true depth* of a feature's support region and adapting its size to the linear perspective view change, as shown by the larger supporting size of features from the closer female.

of orientation-based descriptors provides LST features with additional robustness to illumination variations. Recently, with the emergence of affordable commercial color-depth cameras, which have become a most popular sensor for 3D robotic vision, extraction of LST features incorporating valuable depth information continues to attract increasing attention from computer vision and robotics communities [Everts et al., 2013, Xia and Aggarwal, 2013, Zhang and Parker, 2011].

Despite their advantages, existing LST features have several shortcomings. First, because LST features encode local shape and motion variations, in complex scenes a large proportion of detected features often fall on cluttered backgrounds, especially when the camera is moving in robotic applications. Irrelevant features from backgrounds usually decrease the ability to represent actions themselves [Chakraborty et al., 2012]. Second, since local features ignore global spatial structure information and lack affiliation information, they are incapable of identifying the actions of multiple persons in the same scene. Representing the actions of each person in a group is considerably more challenging than representing the group as a whole, and requires modeling *feature affiliation*. Third, existing LST feature descriptors are not truly adaptive to linear perspective view changes, i.e., the size of the feature's support region does not adapt to its distance to the camera, resulting in decreased feature description ability. For example, the values of two features extracted from the same point on a human can vary significantly when the human is positioned

at different distances from the camera. We call this issue the *false descriptor size problem*.

In this chapter, we address the new problem of recognizing actions of each individual simultaneously in a group from 3D visual data, through proposing a novel *Ad*aptive *Hu*man-*C*entered (AdHuC) spatio-temporal feature that can address the above three shortcomings. The general idea of our work is depicted in Figure 6.1. More specifically, we construct affiliation regions of each human performing actions in $xyzt$ space (3D spatial and 1D temporal). Then, features are detected locally within a human's affiliation region, which are assumed to be affiliated with the individual. As a result, our human-centered feature detection method explicitly models feature affiliation to solve the ARMI task and avoids detecting irrelevant features from background clutter. For feature description, we define a new depth measure to represent the true depth of a feature support region; we adaptively resize the support region based on this depth to compensate for linear perspective view changes, and we propose a normalized multi-channel descriptor to quantize our features. Our feature extraction method is based on the new Depth of Interest (DOI) concept, which enables us to coherently localize humans, construct affiliation regions, and detect and describe our AdHuC features.

Our contributions are fourfold. First, we propose a novel multi-channel feature detector to detect human-centered features from color-depth visual data, which can represent the actions of an individual in a group of humans and deal with background clutter and camera movements. Second, we introduce a new multi-channel descriptor that is able to compensate for the linear perspective view change and solve the false descriptor size problem. Third, we introduce a coherent approach, based on the DOI concept, to simultaneously perform human localization and feature extraction in order to address ARMI at the feature level. Fourth, we identify the important but not well studied ARMI problem in human-centered robotic applications, with the objective to bridge the divide between single-person and group action recognition.

6.2 Related Work

6.2.1 Detector

LST features are detected by capturing local texture and motion changes. Laptev et al. [Laptev, 2005] detected LST features from color videos based on generalized Harris corner detectors with spatio-temporal Gaussian derivative filters. Dollar et al. [Dollár et al., 2005] detected such features using separable filters in space and time dimensions from color videos. Recently, Zhang and Parker [Zhang and Parker, 2011] extended [Dollár et al., 2005] to detect features in color-depth videos. These methods extract LST features from the entire frame; as a result, they detect a large portion of irrelevant features from background clutter and are incapable of distinguishing features from different individuals in a group.

Turcot and Lowe [Turcot and Lowe, 2009] reported that it is better to select a small subset of useful features for recognition problems. Chakraborty et al. [Chakraborty et al., 2012] proposed the selective LST feature, where interest points are extracted from the entire image and then features are pruned using surrounding suppression and space-time constraints. Our feature detector is inherently different from these feature selection methods; features irrelevant to humans are not detected (that is, no selection is performed), which significantly reduces the number of irrelevant features and increase computational efficiency, especially when the camera is in motion in robotics applications.

6.2.2 Descriptor

Nearly all LST feature descriptors used to represent human actions in videos are based on image gradients. Dollar et al. [Dollár et al., 2005] concatenated image gradients within a fixed support region into a single feature vector. Zhang and Parker [Zhang and Parker, 2011] extended [Dollár et al., 2005] to describe multi-channel features with a fixed support region in $xyzt$ space. Scovanner et al. introduced the SIFT3D [Scovanner et al., 2007] descriptor, an extension of the well-known SIFT [Lowe, 2004] descriptor to videos, to quantize gradients in space-time dimensions. Klaser et al. [Kläser et al., 2008] introduced the HOG3D descriptor, an extension of the well-known HOG descriptor [Dalal and Triggs, 2005], to describe xyt gradients in a fixed support region. The support regions of these LST descriptors have a fixed, nonadaptive size and are not capable of handling the linear view perspective changes.

Xia et al. [Xia and Aggarwal, 2013] introduced an approach to adapt support region size: detect features from entire frames and then assign each feature depth with the minimum depth value of the feature point within a time interval. This detect-assign approach suffers from the false descriptor size issue, because as most LST features are detected around the edges of moving body parts, a large proportion of features fall outside of the blob of human pixels with incorrect depth values within the time scale. This could improperly treat these features as belonging to either the incorrect background or foreground objects, which would yield improper support region sizes. Inherently different from [Xia and Aggarwal, 2013], we first analyze the depth to construct a feature affiliation region for each human, then detect features within each affiliation region. Since feature depth is constrained by affiliation regions, our descriptor is able to appropriately address the false descriptor size issue.

6.2.3 Human Localization for Action Recognition

Human localization is usually applied to modeling group activities [Choi et al., 2011b, Khamis et al., 2012, Lan et al., 2012, Ni et al., 2009]. Assuming people are localized by existing human detectors [Choi et al., 2011a, Felzenszwalb et al., 2008], Choi et al. [Choi et al., 2011b] built a log-linear model using context features to encode and represent human poses; Lan et al. [Lan et al., 2012] introduced discriminative

(a) Raw input color-depth data **(b)** Data with planes removed

Figure 6.2: 3D visual data represented by 3D point clouds or color-depth images.

latent models using similar context descriptors; Khamis et al. [Khamis et al., 2012] constructed a network flow-based model using context descriptors; and Ni et al. [Ni et al., 2009] introduced global causality features based on motion trajectories to recognize group activities. Different from previous works focusing on high-level models [Choi et al., 2011b, Khamis et al., 2012, Lan et al., 2012] and global features [Choi et al., 2011b, Ni et al., 2009], we aim to extract low-level local features, which is rarely discussed in previous studies. In addition, unlike previous works that treat human localization and action recognition as two independent tasks and assume human locations are known a priori to action recognition, we jointly address human localization and feature extraction using a coherent approach based on the DOI concept. Finally, inherently different from previous localization methods that typically follow a sliding window paradigm, our human localization algorithm avoids dense multi-scale scanning over the entire image, which greatly reduces computation costs.

6.3 Our AdHuC Features

The goal of introducing our AdHuC features is to affiliate LST features with the proper human and adapt to linear perspective view changes in order to efficiently address ARMI in practical human-centered robotic applications.

6.3.1 DOI Selection

We begin our discussion by defining *Depth of Interest*, which is the foundation of our coherent method for human localization and feature extraction. Just as a region of interest (ROI) is defined as a highly probable rectangular region of object instances [Kim and Torralba, 2009], we define a DOI as a highly probable interval of human or object instances in the depth distribution of color-depth data. Each instance in a DOI is referred to as a *candidate*.

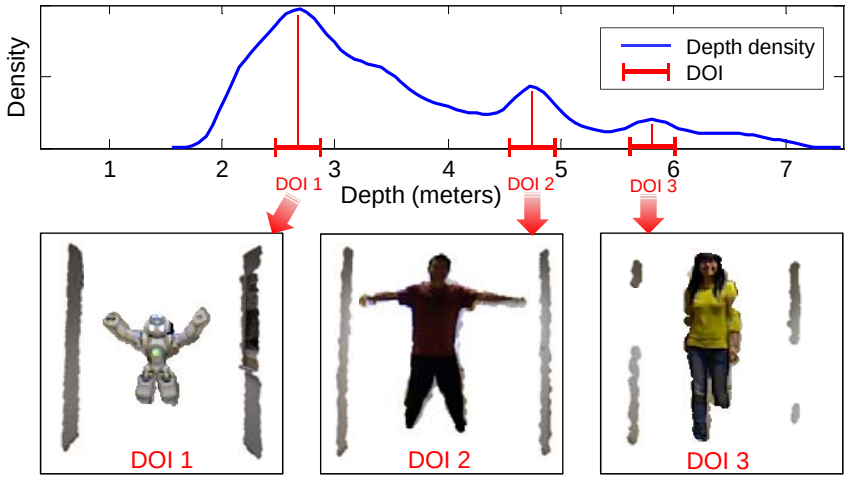


Figure 6.3: Depth distribution of the 3D visual data depicted in Figure 6.2b with three extracted DOIs, and the candidates contained in each DOI. Humans are well localized along with several other non-human objects (e.g., a humanoid robot) that will be rejected by our rejector cascade.

The input to our algorithm is a sequence of 3D point clouds or color-depth images acquired by a calibrated color-depth camera, as shown in Figure 6.2a. When the camera operates on the same ground plane as humans (e.g., installed on mobile robots), ground and ceiling planes are usually viewable. Because points on the ground always connect candidates that are located on the floor, it is important to eliminate this connection in order to robustly select DOIs that contain separate candidates of interest. In addition, since the ceiling plane usually consists of a significant amount of irrelevant points that gradually change depth, removing these points is desirable to select DOIs and increase processing speed, which is important for onboard robotic applications.

To remove ground and ceiling planes, we implement an incremental, prior-knowledge guided random sample consensus (RANSAC) approach. RANSAC [Fischler and Bolles, 1981] is an iterative data-driven method to estimate parameters of a mathematical model. To save computation, the following prior knowledge is applied: (1) the ground and ceiling planes are at the bottom and top, and (2) their surface norm is a vertical vector. Because there is only a slight change between adjacent frames with a moving camera on a robotic platform, we compute plane parameters in an incremental fashion, by using plane parameters in the previous frame to guide parameter estimation in the current frame. Results after removing the planes from the 3D visual data in Figure 6.2a are illustrated in Figure 6.2b.

With the ground and ceiling planes removed, a local maximum in the depth distribution represents a DOI, and a depth interval centered at that maximum generally has a high probability to contain candidates that we are interested in. We note that the correctness of the DOI concept is supported by the observation

that, in color-depth data, any candidate will include a set, or adjacent sets, of points with a similar depth. Since multiple candidates can be located at various depth ranges, the depth distribution usually has different shapes with a various number of local maximums, and the underlying density form is unknown. To estimate the depth distribution we therefore use the non-parametric Parzen window algorithm [Parzen, 1962]. Each frame also uses the DOI information from the previous frame to reduce false negatives; that is, if a depth does not exist in the previous DOIs, a new DOI is constructed. As an example, the DOIs selected on the depth distribution of the 3D visual data in Figure 6.2b and the candidates contained in each DOI are depicted in Figure 6.3.

6.3.2 Affiliation Region Construction

Affiliation region is defined in $xyzt$ space as a temporal sequence of cubes in 3D space, such that each affiliation region contains *one and only one* individual with the same identity, which is denoted as $\mathcal{A}_h = \{x, y, z, t, s_x, s_y, s_z\}$, with cube center (x, y, z) and size (s_x, s_y, s_z) at time point t, and human identity h. The goal of introducing affiliation regions is to set constraints on locations where features can be detected and to associate local features with the human in an affiliation region. To construct affiliation regions, humans are localized in 3D space at each frame, then the locations of the same human across frames are associated.

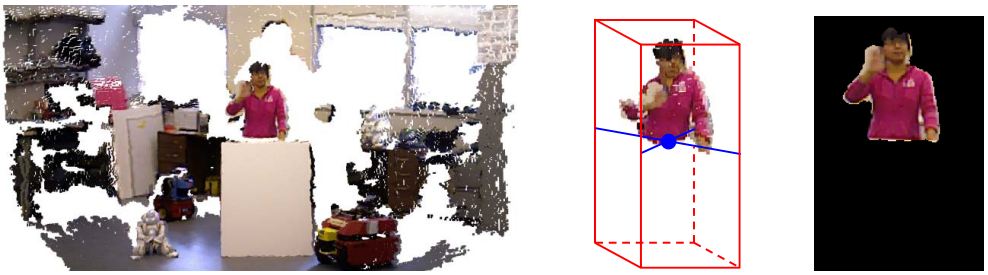
Human localization is performed based on DOIs. To preserve human candidates in each DOI, a cascade of rejectors is used to reject candidates that contain only non-human objects. In the rejector cascade framework, simple rejectors are first applied to reject the majority of candidates before more complex methods are employed, which has been shown to significantly increase detection accuracy while radically reducing computation costs [Viola et al., 2005]. To localize human candidates, we design a cascade with one HOG-based and three heuristic rejectors:

(1) Height-based rejector: After estimating a candidate's actual height, as depicted in Figure 6.4b, the candidate is rejected if its actual height is smaller than a min-height threshold (e.g., the humanoid), or larger than a max-height threshold.

(2) Size-based rejector: After estimating the actual size of a candidate, the candidate is rejected if its size is greater than a max-size threshold. However, in order to allow for occlusion, we do not reject small-sized candidates.

(3) Surface-normal-based rejector: This detector is applied to reject planes, such as walls and desk surfaces. If the surface normal of a candidate is vertical or within a horizontal plane, the candidate is rejected.

(4) HOG-based rejector: This rejector is based on a linear SVM and the HOG features, as proposed by Dalal and Triggs [Dalal and Triggs, 2005]. Their recommended settings are used for all parameters except that our window has a size of 96×64. The candidate is projected onto a color image of the same size to enable single-scale scanning to save computation. It is desirable to contain the whole candidate in the color image, including any occluded parts, which yields a height



(a) (b) (c)

Figure 6.4: Computation of the height and centroid of an occluded candidate. Figure 6.4a shows a raw color-depth frame. The actual height of a candidate is defined as the distance between its highest point to the ground, as shown in Figure 6.4b. The candidate centroid is drawn with a blue dot in the center of the 3D cube in Figure 6.4b. When the candidate is projected to a color image of size 96×64, it is placed in the center of the image according to its real size, instead of the blob size, as shown in Figure 6.4c.

closer to its actual height. Through using this height rather than the blob height, we obtain a more reliable rejection result.

To associate human localization results across frames, an efficient loose-tight association method is introduced. Loose association is based on localized human candidate positions: if the distance of a human candidate in the current frame to a human in the previous frame is smaller than a predetermined threshold, they are loosely matched. Then, tight association is performed to further match loosely-associated human localization results. We create a color-based appearance model for each localized human, which is learned and updated in an online fashion using an online AdaBoost algorithm as by Grabner et al. [Grabner and Bischof, 2006]. A color histogram is used as features and is computed from the human candidate's projected color image, which is highly accurate since background is naturally masked out by the projection, as demonstrated in Figure 6.4c.

The affiliation region in $xyzt$ space is the temporally associated human localization results in 3D spatial space. To make affiliation regions robust to noise, we apply an extended Kalman smoother [Briers et al., 2010] to smooth their location and size. Our affiliation region construction has several advantages: (1) Our color features are an accurate human representation, since the background is masked out in color images by applying DOIs. (2) Since our human appearance model is updated online, it adapts to appearance changes caused by occlusions and body configuration variations. (3) Human localization, based on the DOI concept and rejector cascade, avoids using computationally expensive window scanning over the entire frame and provides the ability to localize humans using a moving camera, which is critical for mobile robots with computational constraints.

6.3.3 Human-Centered Feature Detection

Given a sequence of color-depth frames containing depth $d(x, y, z, t) = z(x, y, t)$ and color $c(x, y, z, t)$ data in $xyzt$ space, and the affiliation regions $\{\mathcal{A}_1, \ldots, \mathcal{A}_H\}$ constructed based on the selected DOIs for H humans in the camera view, our goal is to detect multi-channel LST features that are affiliated with people, which are called multi-channel *human-centered features*. Different from previous feature detection methods that detect interest points from entire frames without extracting the affiliation information [Dollár et al., 2005, Xia and Aggarwal, 2013, Zhang and Parker, 2011], we detect our human-centered features within the affiliation region of each human, and associate the extracted feature's affiliations with the human.

In order to incorporate spatio-temporal and color-depth information in $xyzt$ space, we apply a cascade of three filters: a pass-through filter to encode cues along depth (z) dimension, a Gaussian filter to encode cues in xy space, and a Gabor filter to encode time (t) information; then, we fuse the color and depth cues. Formally, within the affiliation region $\mathcal{A}_h$ of individual h, we convert color into intensity $i(x, y, z, t)$ and compute a multi-channel saliency map by applying separable filters over depth $d(x, y, z, t)$ and color $i(x, y, z, t)$ channels in $\mathcal{A}_h$. Depth and intensity data are processed using the same procedure, as follows: First, the data are filtered in the 3D spatial space:

$$d_s(x, y, z, t) = \big(d(x, y, z, t) \circ f(z, t; \delta)\big) * p(x, y; \sigma) \tag{6.1}$$

where $*$ denotes convolution and $\circ$ represents entry-wise matrix multiplication. A pass-through filter $f(z, t; \delta)$ with parameter δ is applied along the z dimension:

$$f(z, t; \delta) = H(z + \delta) - H(z - \delta) \tag{6.2}$$

where $H(\cdot)$ denotes the Heaviside step function. A Gaussian filter $p(x, y, t; \sigma)$ is applied along the xy spatial dimensions:

$$p(x, y; \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6.3}$$

where σ controls the spatial scale along x and y dimensions. Then, a Gabor filter is used along the t dimension:

$$d_{st}(x, y, z, t) = d_s(x, y, z, t) * g(t; \tau, \omega) \tag{6.4}$$

where the Gabor filter $g(t; \tau, \omega)$ with parameter τ satisfies:

$$g(t; \tau, \omega) = \frac{1}{\sqrt{2\pi}\tau} \cdot e^{-\frac{t^2}{2\tau^2}} \cdot e^{i(2\pi\omega t)} \tag{6.5}$$

We use $\omega = 0.6/\tau$ throughout the work.

After processing intensity data, we use the same procedure to obtain $i_{ds}(x, y, z, t)$. Then, we compute the spatio-temporal multi-channel saliency map as:

$$R(x, y, z, t) = (1 - \alpha) \cdot i_{st}^2(x, y, z, t) + \alpha \cdot d_{st}^2(x, y, z, t) \tag{6.6}$$

where α is a mixture weight to balance between intensity and depth cues. The saliency map generally represents variations of textures, shapes and motions, since any region undergoing such variations induces responses.

Then, our human-centered LST features are detected as local maximums of R on the surface $z = z(x, y, t)$ within $\mathcal{A}_h$ in $xyzt$ space, and each feature is affiliated with human h. Since $\mathcal{A}_h$ only contains a single individual, the detected features are affiliated with the human and are distinguishable from features belonging to other individuals. As a result, our human-centered features are able to address the ARMI task. In addition, since the region for detecting our features is bounded by $\mathcal{A}_h$ in a DOI, irrelevant features (e.g., from the background) are never detected. This characteristic is particularly impactful in robotic applications, since it provides (1) an increased ability for features to represent human actions, (2) an improvement in feature detection efficiency, and (3) the ability to handle the moving camera challenge in local feature detection.

6.3.4 Adaptive Feature Description

Here we introduce a new multi-channel feature descriptor with a support region that is adaptive to changing linear perspective views, and thereby addresses the false descriptor size issue. For each LST feature point (x, y, z, t, h), which falls in the affiliation region $\mathcal{A}_h = \{x_h, y_h, z_h, t_h, s_x, s_y, s_z\}$ and is detected with the scales $(\sigma, \sigma, \delta, \tau)$ in $xyzt$ space, we extract a support region $\boldsymbol{S} = (x, y, z_s, t, \sigma_s, \delta_s, \tau_s, h)$ of size $(\sigma_s, \sigma_s, \delta_s, \tau_s)$ along the x, y, z and t dimensions, respectively. To compensate for spatial linear perspective view changes, i.e., objects closer to the camera appearing larger, we propose adapting the spatial size of support regions to their true depth. Estimating the truth depth is a challenging task, since detected feature points can fall out of human blobs and consequently have incorrect depth values, resulting in the false descriptor size issue.

To address this issue, we propose a new approach to estimate the support region's *true depth*, based on $\mathcal{A}_h$ that is computed using the DOI concept. To this end, we formally define several important concepts and mathematically formulate our true depth statement as a proposition.

Definition 3 (Depth affiliation indicator)**.** *Given an individual's affiliation region* $\mathcal{A}_h = \{x_h, y_h, z_h, t_h, s_x, s_y, s_z\}$ *and a depth value* z*, the* depth affiliation indicator *is defined as a function such that:*

$$h(z) = \mathbb{1}(z \geq z_h - \frac{s_z}{2}) \cdot \mathbb{1}(z \leq z_h + \frac{s_z}{2}) \tag{6.7}$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Definition 4 (Support region's true depth)**.** *Given a feature point (x, y, z, t, h) detected using the scales $(\sigma, \sigma, \delta, \tau)$ in $\mathcal{A}_h = \{x_h, y_h, z_h, t_h, s_x, s_y, s_z\}$, the* true depth *of the feature's support region $\boldsymbol{S}$ is defined by:*

$$z_s(\boldsymbol{S}) = \frac{1}{\tau}\sum_{j=0}^{\tau-1} z(x, y, t-j) \cdot h(z(x, y, t-j)) + z_h \cdot (1 - h(z(x, y, t-j))) \tag{6.8}$$

Proposition 1. *Given the affiliation region of an individual $\mathcal{A}_h$, for all feature points detected in $\mathcal{A}_h$, the true depth of their support regions satisfies $h(z_s(\boldsymbol{S})) = 1$.*

Proof. Among τ depth values $z(x, y, t-j)$, $j = 0, \ldots, \tau-1$, assume $\tau_1 \in [0, \tau]$ out of τ depth values satisfy $h(z_i) = 1$, $i = 1, \ldots, \tau_1$; the remaining $\tau - \tau_1$ depth values satisfy $h(z_k) = 0$, $k = 1, \ldots, \tau - \tau_1$. Then, the support region's true depth $z_s(\boldsymbol{S})$ satisfies:

$$\begin{aligned} z_s(\boldsymbol{S}) =& \frac{1}{\tau}\left(\sum_{i=1}^{\tau_1} z_i + \sum_{k=1}^{\tau-\tau_1} z_h\right) \\ \leq& \frac{1}{\tau}\left(\tau_1\left(z_h + \frac{s_z}{2}\right) + (\tau - \tau_1) z_h\right) = \frac{s_z}{2}\frac{\tau_1}{\tau} + z_h \\ \leq& \frac{s_z}{2} + z_h \end{aligned}$$

Similarly, we can prove that $z_s(\boldsymbol{S})$ also satisfies:

$$\begin{aligned} z_s(\boldsymbol{S}) =& \frac{1}{\tau}\left(\sum_{i=1}^{\tau_1} z_i + \sum_{k=1}^{\tau-\tau_1} z_h\right) \\ \geq& \frac{1}{\tau}\left(\tau_1\left(z_h - \frac{s_z}{2}\right) + (\tau - \tau_1) z_h\right) = z_h - \frac{s_z}{2}\frac{\tau_1}{\tau} \\ \geq& z_h - \frac{s_z}{2} \end{aligned}$$

In summary, $z_h - s_z/2 \leq z_s(\boldsymbol{S}) \leq z_h + s_z/2$. Therefore $h(z_s(\boldsymbol{S})) = 1$ □

Proposition 1 indicates that the location of the support region $\boldsymbol{S}$ is bounded by $\mathcal{A}_h$. Thus, $z_s(\boldsymbol{S})$ encodes the true depth of the support region $\boldsymbol{S}$ in $\mathcal{A}_h$, in general. Based on z_s, we adapt the spatial support region size as follows:

$$\sigma_s = \frac{\sigma_0 \sigma}{z_s}, \qquad \delta_s = \frac{\sigma_0 \delta}{z_s} \tag{6.9}$$

where σ_0 characterizes the support region's relative spatial size. Since its temporal size is not affected by spatial linear perspective view changes, we define $\tau_s = \tau_0 \tau$, where τ_0 characterizes the relative temporal size. An example of our adaptive feature description is illustrated in Figure 6.5.

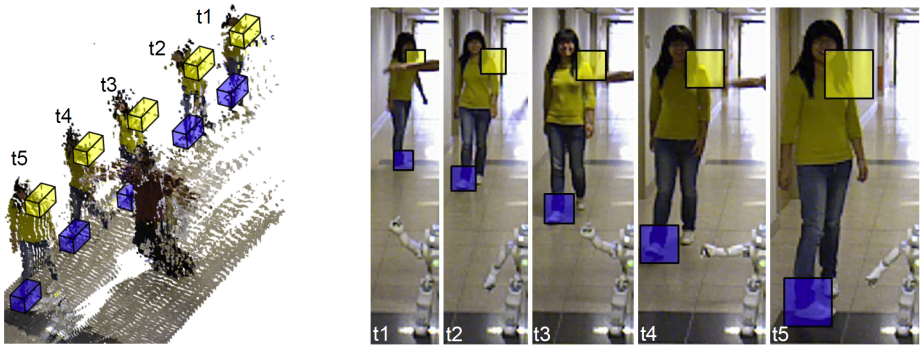


Figure 6.5: Feature support regions that have the same size in the 3D (xyz) physical space have different projected sizes when they are mapped onto 2D (xy) images, due to linear perspective view changes, as shown by the yellow and blue support regions. Accordingly, our adaptive multi-channel descriptor adapts the support regions to their *true depth* in order to compensate for this linear perspective view change.

We propose an extended HOG3D descriptor that slightly differs from the original [Kläser et al., 2008] in order to incorporate multi-channel information and deal with adaptive supporting size. HOG3D approximates orientations of 3D gradients in a feature's support region using a regular polyhedron with congruent faces that are regular polygons, each of which serves as a bin. Tracing each gradient along its direction up to the intersection with a face identifies the bin. Then, a feature is described by a histogram $\boldsymbol{h}$ that counts the number of gradients falling in the bins. Since the size of our feature's support region is adaptive, it can contain a different number of gradients; thus, histogram normalization is required. In addition, since gradients in our work are computed from both intensity and depth channels, we apply the standard practice of concatenation of the per-channel descriptors [Zhang and Parker, 2011, Everts et al., 2013], leading to our final descriptor:

$$\boldsymbol{h} = \left\{ \frac{\boldsymbol{h}_i}{M_i}, \frac{\boldsymbol{h}_d}{M_d} \right\} \tag{6.10}$$

where $\boldsymbol{h}_i$ is the histogram using M_i intensity gradients, and $\boldsymbol{h}_d$ is the histogram based on M_d depth gradients.

6.4 Experimental Validation

Here we detail our empirical study conducted to evaluate our AdHuC feature and its performance for human action recognition, especially on the ARMI task, using the Berkeley MHAD, ACT4^2 and UTK-ARMI dataset.

Table 6.1: Average accuracy on MHAD

Approach	Accuracy
Harris3D + HOG/HOF + 1-NN [Ofli et al., 2013]	77.37%
Harris3D + HOG/HOF + 3-NN [Ofli et al., 2013]	76.28%
Harris3D + HOG/HOF + SVM [Ofli et al., 2013]	70.07%
Harris3D + HOG/HOF + MKL [Ofli et al., 2013]	91.24%
Our AdHuC features + SVM	**97.81**

6.4.1 Experiment Setups

AdHuC Implementation For DOI-based affiliation region construction, the DOI width is set to 1.0 m; the min-height threshold is set to 0.4 m; the max-height threshold is set to 2.3 m; the max-size threshold is set to 4.0 m^2. Our generic HOG-based rejector is modified from the original work [Dalal and Triggs, 2005] and trained with the H3D dataset [Bourdev and Malik, 2009], using all of the positive and a subset of the negative samples; the loose association threshold is set to be 0.5 m. For human-centered feature detection, we assign scale parameters $\sigma = 5$, $\delta = 0.25$ m, and $\tau = 3$. For adaptive multi-channel feature description, we assign parameter values $\sigma_0 = 8$ and $\tau_0 = 5$. When a color-depth camera is employed (e.g., Kinect), the depth value is in [0.5, 8.0] m. A standard feature pooling scheme [Everts et al., 2013, Kläser et al., 2008, Ni et al., 2011, Xia and Aggarwal, 2013] is applied for human action recognition, which subdivides each support region into $N_x \times N_y \times N_t = 4 \times 4 \times 3$ cells.

Recognition Following the setups used in [Kläser et al., 2008, Everts et al., 2013, Wang et al., 2009], human action recognition is performed in a standard bag-of-features learning framework and a codebook is created through clustering 200,000 randomly sampled features using k-means into 1000 codewords. For classification, we use non-linear SVMs with χ^2-kernels and the one-against-all approach [Kläser et al., 2008, Everts et al., 2013, Wang et al., 2009]. The recognition method and our AdHuC features are implemented using a mixture of Matlab and C++ on a Linux machine that have an i7 3.0G CPU and 16Gb memory.

6.4.2 Single-Person Action Recognition

Here we evaluate the performance of our AdHuC features for recognizing single-person actions from color-depth videos using the benchmark MHAD and ACT4^2 datasets. Our results and comparisons with previous methods are presented in Tables 6.1 and 6.2. It can be observed that our AdHuC features achieve the state-of-the-art performance and significantly outperform previous methods on single-person action recognition from color-depth visual data. This highlights the importance of extracting human-centered features and avoiding noisy, irrelevant background local features.

Table 6.2: Average precision on ACT4[2]

Approach	Precision
Harris3D + Color-HOG/HOF [Cheng et al., 2012]	64.2 %
Depth layered multi channel STIPs + HOG/HOF [Ni et al., 2011]	66.3%
Harris3D + Depth-HOG/HOF [Cheng et al., 2012]	74.5%
Harris3D + Comparative coding descriptor [Cheng et al., 2012]	76.2%
Harris3D + Super feature representation [Cheng et al., 2012]	80.5%
Our AdHuC features	**85.7%**

6.4.3 Action Recognition of Multiple Individuals

In this section we provide extensive evaluation of our AdHuC features for the multi-individual action recognition task over the newly created ARMI dataset. The following all-in-one experimental setup is applied: we divide the instances in the dataset into 50% training and 50% testing, both containing actions from all individuals. Recognition is evaluated using the accuracy metric, computed over all actions performed by all individuals in the testing set.

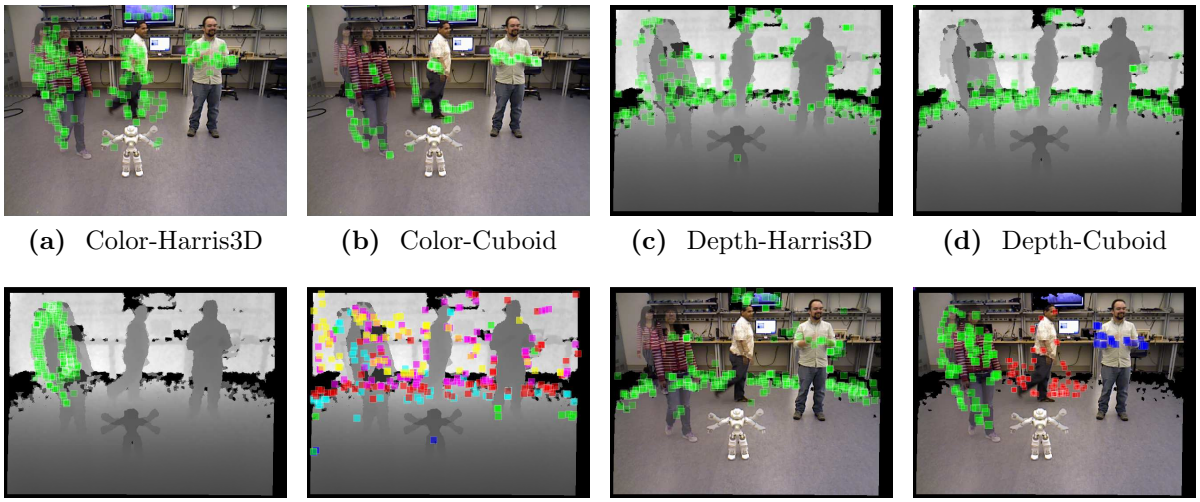
Qualitative evaluation

To perform a qualitative evaluation using the ARMI dataset we begin by providing an intuitive visualization of our AdHuC feature's performance, as depicted in in Figure 6.6h. To better illustrate our AdHuC feature's impact, we compare the introduced AdHuC feature with seven baseline features from previous studies, including **Harris3D** (color/depth) [Laptev, 2005], **Cuboid** (color/depth) [Dollár et al., 2005], **DSTIP** [Xia and Aggarwal, 2013], **DLMC-STIP** [Ni et al., 2011] and **CoDe4D** [Zhang and Parker, 2011] detectors and their descriptors, using their original implementation, as demonstrated in Figure 6.6.

We observe that all previous features are not capable of extracting feature affiliation information. In addition, previous methods based only upon color cues usually detect irrelevant features from the dynamic background (e.g., the TV) or foreground obstacles (e.g., the robot), while methods based on depth usually generate a large number of irrelevant features due to depth noise. Although the DSTIP method [Xia and Aggarwal, 2013] avoids extracting irrelevant background features, it also does not capture useful features from humans, especially in multiple individual scenarios. As illustrated in Figure 6.6h, our AdHuC algorithm is able to identify feature affiliation, avoid extracting irrelevant features, and adapt descriptor sizes to linear perspective view changes.

Quantitative evaluation

We also conduct empirical studies to quantitatively evaluate our AdHuC feature's performance (i.e., accuracy and efficiency). The recognition performance is presented in Table 6.3. It is observed that the proposed AdHuC features obtain promising



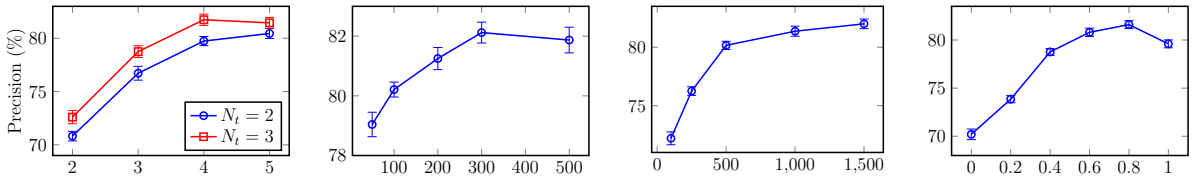
(a) Color-Harris3D **(b)** Color-Cuboid **(c)** Depth-Harris3D **(d)** Depth-Cuboid

(e) DSTIP **(f)** DLMC-STIP **(g)** CoDe4D **(h)** **AdHuC**

Figure 6.6: Qualitative comparison of the introduced AdHuC features with the state-of-the-art color/depth LST features, including Color-Harris3D [Laptev, 2005], Color-Cuboid [Dollár et al., 2005], Depth-Harris3D [Laptev, 2005], Depth-Cuboid [Dollár et al., 2005], DSTIP [Xia and Aggarwal, 2013], DLMC-STIP [Ni et al., 2011], CoDe4D [Zhang and Parker, 2011]. In Figure 6.6f, features with different colors are from different depth layers (eight layers in total). In Figure 6.6h, different feature colors denote different feature affiliations. The exemplary images are fused to clearly represent the start and end positions of the humans.

Table 6.3: Comparison of accuracy (%) and efficiency on the ARMI dataset

Approach	Bend	Clap	Flap	Kick	Walk	Wave	Overall	Rate
Color-Harris3D [Laptev, 2005]	74.2	73.1	73.4	71.8	78.2	76.6	74.5	$\sim$ 0.27
Color-Cuboid [Dollár et al., 2005]	78.4	74.9	79.2	76.5	79.6	76.4	77.5	$\sim$ 0.14
Depth-Harris3D [Laptev, 2005]	73.3	64.2	66.6	65.4	72.4	69.7	68.6	$\sim$ 0.21
Depth-Cuboid [Dollár et al., 2005]	74.5	66.4	63.2	65.7	73.7	72.4	69.3	$\sim$ 0.14
DSTIP [Xia and Aggarwal, 2013]	85.4	73.9	87.2	74.8	**85.2**	76.9	80.6	$\sim$ 0.04
DLMC-STIP [Ni et al., 2011]	75.3	67.2	69.9	70.8	75.2	70.5	71.5	$\sim$ 0.04
CoDe4D [Zhang and Parker, 2011]	84.0	75.6	87.3	76.4	80.3	79.6	80.5	$\sim$ 0.13
Our AdHuC	**86.7**	**78.4**	**89.4**	**80.8**	84.9	**82.3**	**83.8**	$\sim$ **3.27**



(a) Number of cells **(b)** Number of features **(c)** Codebook size **(d)** Mixture weight α

Figure 6.7: Sensitivity of our AdHuC algorithm on the ARMI dataset. Error bars are cross-validation standard deviations.

recognition accuracy of 82.0% with a frame rate of around 3.3 Hz on the ARMI dataset, which highlights our AdHuC feature's ability to accurately and efficiently distinguish different actions performed by multiple humans in the camera view, through estimating feature affiliations.

To better evaluate our AdHuC feature's performance, we compare our method with previous baseline feature extraction approaches. Because previous methods are not able to identify feature affiliations (and thus cannot address the ARMI task), we combine these feature extraction methods with a most commonly applied baseline HOG-based human localization method [Choi et al., 2011b, Lan et al., 2012, Ni et al., 2009, Dalal and Triggs, 2005, Choi et al., 2011a, Felzenszwalb et al., 2008], which employs a sliding window paradigm and a sparse scan with 800 windows. The experimental results are compared in Table 6.3. We observe that our AdHuC outperforms the tested baselines and obtains the best overall action recognition accuracy. In addition, our algorithm significantly improves feature computation efficiency, as it obtains the highest frame rate. The comparison results indicate the importance of avoiding extracting background features in improving feature discriminative power and computational efficiency.

Sensitivity evaluation

The sensitivity of our AdHuC algorithm to the parameters is evaluated using fivefold cross-validation on the training set; the results are illustrated in Figure 6.7. It is observed that incorporating color and depth cues can result in an improved performance (Figure 6.7d). We also observe that using a relatively small number of features per instance (e.g., 300) with a small codebook (e.g., of size 500) we can obtain promising ARMI performance. This highlights the superior description capability of AdHuC features: since our features are highly related to human actions, a small feature set is generally descriptive enough to distinguish human actions in 3D visual data.

6.5 Summary

In this chapter, we introduce the novel AdHuC features concept to enable human-centered robots to understand the actions of multiple individuals in a group from 3D visual data. We construct a feature affiliation region for each individual by applying the DOI concept and a cascade of rejectors to localize humans in 3D space. Then our human-centered features are detected within the affiliation region of each individual, which is able to recognize feature affiliations, avoid extracting irrelevant features and handle camera movements. An adaptive multi-channel feature descriptor is also introduced to deal with the linear perspective view changes and encode information from both color and depth channels into a final feature vector. Extensive experiments are performed using three benchmark color-depth datasets and a newly created ARMI dataset. Results show that our AdHuC features greatly improve single-person action recognition performance, and efficiently address the multiple individual action recognition problem.

Chapter 7

Recognition: RG-PLSA Model

7.1 Introduction

In this chapter, we focus on the human activity recognition task. Our objective is to automatically assign each video clip with an action out of a given number of predefined action categories. Latent variable topic models, including the probabilistic Latent Semantic Analysis (PLSA) model [Hofmann, 1999a], are extensively studied in the text mining community. PLSAs were originally proposed to categorize large collections of documents into a small set of pre-defined topics. In recent years, topic models have become increasingly popular in the machine vision community. These models have shown promising performance on image segmentation, scene understanding, and activity recognition.

The original topic models are based on the "bag-of-word" assumption, in which text words and visual features are assumed to be discrete. However, features that are used to classify human actions are usually continuously distributed in some high dimensional space. One approach to address this issue is discretizing all continuous features. However, discretization always introduces truncation errors. Another widely used method is quantization, which clusters the continuous features into discrete groups, or a fixed-size vocabulary. However, since quantization ignores the distance of features to their closest cluster center, this approach does not necessarily lead to optimal results.

Moreover, the PLSA model assumes that each observation has a different distribution over the pre-defined categories, even if the observations are generated from the same category. Since the number of model parameters increases linear with the number of observations, the PLSA model usually suffers from a severe overfitting problem. For example, in human action recognition, the PLSA model assigns a different distribution over the possible action categories to each observation. However, it is highly possible that a human performs the same action in a sequence of consecutive observations. In this case, PLSA models overemphasize the action variations within the same category, which can lead to overfitting.

Gaussian Mixture Models (GMMs) can model continuous features and provide a potential way to prevent overfitting. In GMMs, the observations within the same category are assumed to have the same per-observation category distribution. GMMs can achieve reasonable classification accuracy when modeling data that was generated from Gaussian distributions. However, GMMs are limited in their expressive power. These models can easily underfit if the true distributions are complex. For example, in the human action recognition, GMMs assume that actions within the same category have the same distribution. In this case, GMMs ignore variations within actions performed by different people or by the same person, but with different poses.

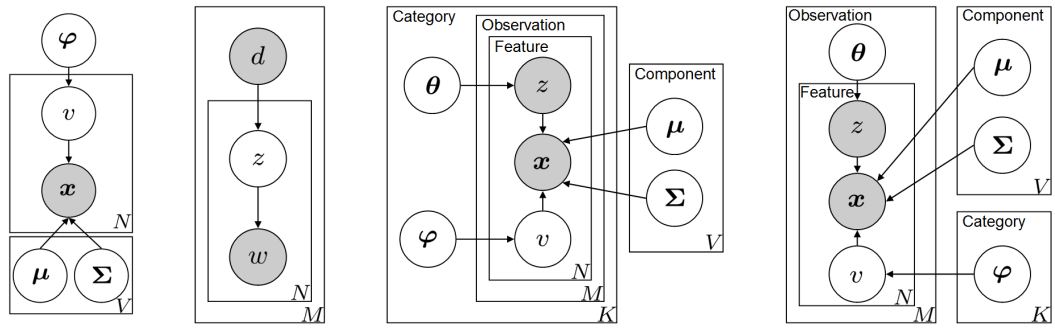
We propose a new latent variable graphical model, called the *Regularized Gaussian PLSA* (RG-PLSA) model, which uses a regularization term in the parameter learning process to combine the advantages of the GMM model and the PLSA model. The contributions of the proposed model are two-fold. First, the RG-PLSA model extends the original PLSA model to support continuous real-valued observations. Second, regularization is applied to reduce model complexity, which simultaneously prevents overfitting and provides the model with moderate flexibility. We illustrate these model capabilities by applying the RG-PLSA model to supervised action classification.

The remainder of the chapter is organized as follows. Section 10.2 reviews related work. Section 7.3 presents preliminary materials. Section 7.4 describes Hierarchical GMM models and the Gaussian PLSA model, followed by our novel RG-PLSA model. Section 7.5 presents our experimental results on two publicly available datasets, and a performance comparison against other approaches. Finally, we provide discussions and conclude our work in Section 10.7.

7.2 Related Work

Previous research conducted to improve the original PLSA model's limitations cover: 1) PLSA's inability to model continuous observations; and 2) PLSA's severe overfitting problems. To model continuous valued observations, Aspect GMMs [Ahrendt et al., 2005] extend the PLSA model by applying the Gaussian distributions to represent real-valued features in the music genre classification task. Another two Gaussian variations were introduced in [Hörster et al., 2008] to address scene classification tasks in an unsupervised fashion. Although these Gaussian PLSAs achieve satisfiable performance, they still suffer from overfitting, which is explicitly addressed in our work.

In the latent variable graphical models, two largely equivalent techniques are widely applied to prevent overfitting: Bayesian regularization via Maximum a posteriori (MAP) inference or penalty based regularization. The MAP extension of PLSAs [Asuncion et al., 2009] introduces a prior distribution over the parameters, and then maximizes the posterior log-likelihood via the expectation-maximization (EM) algorithm. However, the priors have to be manually selected, which usually requires significant domain knowledge. On the other hand, penalty regularization modifies



(a) GMM **(b)** PLSA **(c)** HGMM **(d)** Gaussian PLSA

Figure 7.1: Plate representation of the graphical models discussed in this chapter. The boxes are plates, representing replications.

ordinary likelihood maximization with a penalty on the magnitudes of the parameters, which was shown to greatly improve the discrete [Larsson and Ugander, 2011] and continuous [Si and Jin, 2005] PLSA model's performance.

PLSAs are widely used in human action recognition. This model was first used in [Niebles et al., 2008] to model an action category as a collection of spatio-temporal visual features in an unsupervised way, which demonstrates the PLSA model's capability to recognize simple actions from public datasets and some complicated actions, like skating. Another extension, the PLSA-ISM model, was proposed in [Wong et al., 2007] to capture the relative spatio-temporal location information of the features from the action center. For a comprehensive overview of the approaches to address human action recognition, we refer readers to [Aggarwal and Ryoo, 2011].

7.3 Preliminaries

7.3.1 Gaussian Mixture Models

Gaussian Mixture Models are a special instance of latent variable models, which provide a richer class of density modeling than a single Gaussian distribution over continuous variables. The graphical representation of GMMs is depicted in Figure 7.1a. The latent variable v is the index of a Gaussian component that generates the real-valued observation $\boldsymbol{x} \in \mathbb{R}^{\|\boldsymbol{x}\|}$. GMMs encode the distribution of the *i.i.d.* observations $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$:

$$P(\boldsymbol{X}|\boldsymbol{\varphi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{j=1}^{N} \sum_{v=1}^{V} P(v|\boldsymbol{\varphi}) P(\boldsymbol{x}_j|\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v) \tag{7.1}$$

where $P(v|\boldsymbol{\varphi})$ is the mixture coefficient, and $P(\boldsymbol{x}|\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$ is the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)$. We assume the elements in an observation vector are

independent. Accordingly, the covariance matrix $\boldsymbol{\Sigma}$ becomes a diagonal matrix: $\boldsymbol{\Sigma}_v = \sigma_v^2 \boldsymbol{I}$, where $\boldsymbol{I}$ is the unit matrix.

7.3.2 Probabilistic Latent Semantic Analysis

The Probabilistic Latent Semantic Analysis model [Hofmann, 1999a] was originally proposed for text mining, which provides a probabilistic formulation for modeling topics over a document and a corpus. The graphical representation of the PLSA model is depicted in Figure 7.1b. Given the hidden variable z, each discrete word w is assumed to be independent of the document d which contains it. The joint distribution of the observed variables (i.e., word and document variables) is obtained by marginalizing over the latent topic variable z:

$$P(d_i, w_j) = P(d_i) \sum_{z=1}^{K} P(z|d_i) P(w_j|z) \tag{7.2}$$

where $P(z|d_i)$ is the mixture weight, which is the probability that a topic z occurs in a document d, and $P(w_j|z)$ is the probability that a discrete word w occurs in a topic z. PLSAs treat each document as a convex mixture of topics, and treat each topic as a convex combination of discrete words.

7.4 Regularized Gaussian PLSAs

The original PLSA model has several shortcomings. First, PLSAs only deal with discrete observations. Moreover, the number of model parameters grows linearly with the number of documents, which causes the PLSA model to suffer severe overfitting. One the other hand, GMMs are able to deal with continuous observations, but have limited expressive power. This makes it difficult for GMMs to represent complicated distributions, and usually results in underfitting. Our proposed RG-PLSA model combines PLSA and GMM advantages, and minimizes or removes their disadvantages.

In the remainder of this section, we first extend standard GMMs to a Hierarchical GMM (HGMM) model that is able to simultaneously model the distribution of features and categories. Then, we discuss how to incorporate continuous features into standard PLSAs, and how to add penalty regularization, which allows a trade off between overfitting and underfitting and creates our RG-PLSA model. Lastly, we show how to learn the RG-PLSA model's parameters. For simplicity, we only focus on learning the mixture weights $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, and we assume that the Gaussian component parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are learned beforehand and remain unchanged during the learning and inference processes. For quick reference, all notations are listed in Table 7.1.

Table 7.1: Notations for our models

Notation	Meaning
$\boldsymbol{x}$	A feature vector with continous elements
$\boldsymbol{X}$	The set of features from all observations
d	A dummy variable indexing an observation
z	The category assignment to $\boldsymbol{x}$
v	The Gaussian component assignment to $\boldsymbol{x}$
K	The number of topics
M	The number of observations in a category
N	The number of features in an observation
V	The number of Gaussian components
$\boldsymbol{\theta}$	Parameters of per-observation category distribution
$\boldsymbol{\varphi}$	Parameters of per-category feature distribution
$\boldsymbol{\mu}$	Mean of a Gaussian component
$\boldsymbol{\Sigma}$	Variance of a Gaussian component
$\boldsymbol{\Psi}$	Model parameters to learn: $\boldsymbol{\Psi} = \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$
$\boldsymbol{\Phi}$	All model parameters: $\boldsymbol{\Phi} = \{\boldsymbol{\Psi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

7.4.1 Hierarchical GMMs

The HGMM model is graphically represented in Figure 7.1c. HGMMs explicitly model each continuous feature as a mixture of multivariate Gaussian components. Each category is also modeled as a mixture of the same Gaussian components. Thus, categories and features become dependent, and each category can be viewed as a mixture of the features. Moreover, each category is also modeled as a multinomial distribution over all categories, in which the correct category assignment has the highest probability. Formally, for each category, the HGMM model represents the following distribution:

$$\begin{aligned} P(\boldsymbol{X}|\boldsymbol{\Phi}) = \prod_{d=1}^{M}\prod_{i=1}^{N}\sum_{z=1}^{K}\sum_{v=1}^{V} & P(z^{(d,i)}|\boldsymbol{\theta}) \\ & P(v^{(d,i)}|\boldsymbol{\varphi}_z)P(\boldsymbol{x}^{(d,i)}|\boldsymbol{\mu}_{v,z}, \boldsymbol{\Sigma}_{v,z}) \end{aligned} \tag{7.3}$$

where the superscript (d, i) denotes the ith feature in the dth observation, the subscript indicates which parameter is used, and $\boldsymbol{\Sigma}$ is assumed to be a diagonal matrix. It should be noted that, in HGMMs, the parameter $\boldsymbol{\theta}$ is fixed for each category, which does *not* depend on observations.

7.4.2 Gaussian PLSA

The Gaussian PLSA model replaces the discrete words with continuous features that are modeled as a mixture of the multivariate Gaussian distributions. Gaussian PLSAs are a generative model. For supervised classification tasks, given a category, the generative process is explained as follows:

- For each observation d in a chosen category:
  - For each feature $\boldsymbol{x}^{(d,i)}$ in d:

    a. Choose a topic assignment according to the per-observation topic distribution: $z \sim \text{Mult}(\boldsymbol{\theta}_d)$;

    b. Select a Gaussian component: $v \sim \text{Mult}(\boldsymbol{\varphi}_z)$, given the category z;

    c. Pick a feature: $\boldsymbol{x}^{(d,i)} \sim \mathcal{N}(\boldsymbol{\mu}_{v,z}, \boldsymbol{\Sigma}_{v,z})$, given the topic z and the Gaussian component v.

The Gaussian PLSA model is depicted in Figure 7.1d. Each feature plate represents the distribution of the ith feature in an observation d along with its category assignment z and the Gaussian component assignment v. For the entire dataset $\boldsymbol{X}$, the joint distribution can be factorized as:

$$P(\boldsymbol{X}|\boldsymbol{\Phi}) = \prod_{d=1}^{M}\prod_{i=1}^{N}\sum_{z=1}^{K}\sum_{v=1}^{V} P(z^{(d,i)}|\boldsymbol{\theta}_d) \\ P(v^{(d,i)}|\boldsymbol{\varphi}_z) P(\boldsymbol{x}^{(d,i)}|\boldsymbol{\mu}_{v,z}, \boldsymbol{\Sigma}_{v,z}) \tag{7.4}$$

where $P(\boldsymbol{z}|\boldsymbol{\theta}_d)$ and $P(\boldsymbol{v}|\boldsymbol{\varphi}_z)$ are the multinomial distributions, and $P(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate Gaussian distribution with a diagonal covariance matrix. It should be noted that the only difference between Gaussian PLSAs and HGMMs is that the parameter $\boldsymbol{\theta}_d$ *does* depend on the observations in the Gaussian PLSA models.

7.4.3 Regulated Gaussian PLSAs

PLSAs overemphasize the variations within a category, while HGMMs underemphasize these variations. The RG-PLSA model balances both models to simultaneously prevent overfitting and provide moderate flexibility to model the variations variations within a category. The RG-PLSA model has the same graphical representation as the Gaussian PLSA model. However, in the parameter learning process, a regularization term is adopted to prevent overfitting.

The EM algorithm is used in this work to iteratively learn model parameters, which is the most widely used frequentist parameter estimation in the latent variable graphical models. For each category, the RG-PLSA model's parameters can be learned

by maximizing the regularized auxiliary function:

$$
\begin{aligned}
Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^t, \boldsymbol{\Psi}_G^t) &\triangleq P(\boldsymbol{X}|\boldsymbol{\Phi}) \\
&+ \sum_{d=1}^{M}\sum_{i=1}^{N}\sum_{z=1}^{K}\sum_{v=1}^{V} P(z,v|\boldsymbol{\Psi}^t) \log \frac{P(\boldsymbol{x}^{(d,i)}, z, v|\boldsymbol{\Phi})}{P(\boldsymbol{x}^{(d,i)}, z, v|\boldsymbol{\Phi}^t)} \\
&+ \lambda \sum_{d=1}^{M}\sum_{i=1}^{N}\sum_{z=1}^{K}\sum_{v=1}^{V} P(z,v|\boldsymbol{\Psi}_G^t) \log \frac{P(\boldsymbol{x}^{(d,i)}, z, v|\boldsymbol{\Phi})}{P(\boldsymbol{x}^{(d,i)}, z, v|\boldsymbol{\Phi}_G^t)} \\
&\propto \mathrm{E}_{P_R(\boldsymbol{z},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\Psi}^t,\boldsymbol{\Psi}_G^t)}[\log P(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{v}|\boldsymbol{\Phi})]
\end{aligned}
\tag{7.5}
$$

where $\lambda \in [0, \infty)$ is the regularization factor that controls model complexity, and $\boldsymbol{\Psi}_G = \{\boldsymbol{\theta}_G, \boldsymbol{\varphi}\}$, $\boldsymbol{\Phi}_G = \{\boldsymbol{\Psi}_G, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, i.e., only the per-observation category distribution is regulated. $P_R(z, v|\boldsymbol{X}, \boldsymbol{\Psi}^t, \boldsymbol{\Psi}_G^t)$ is the regularized distribution over the latent variables, which can be computed by:

$$
P_R(\boldsymbol{z},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\Psi}^t,\boldsymbol{\Psi}_G^t) = \frac{P(\boldsymbol{z},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\Psi}^t) + \lambda P(\boldsymbol{z},\boldsymbol{v}|\boldsymbol{X},\boldsymbol{\Psi}_G^t)}{1+\lambda} \tag{7.6}
$$

The regularized distribution demonstrates the importance of the regularization factor λ: a smaller λ makes the RG-PLSA model behave more similarly to the Gaussian PLSA model, which allows for more model complexity. When $\lambda = 0$, the RG-PLSA model has the same form as the Gaussian PLSA model. Similarly, a larger λ emphasizes more on preventing overfitting, and HGMMs are a special instance of the RG-PLSA model, as $\lambda \to \infty$.

In the E-step, given the data and the current parameter values, the posterior distributions over the latent variables are computed:

$$
w_{z,v}^{(d,i)} \triangleq P_R(z^{(d,i)}, v^{(d,i)}|\boldsymbol{x}^{(d,i)}, \boldsymbol{\Psi}^t, \boldsymbol{\Psi}_G^t) \tag{7.7}
$$

where we use $w_{z,v}^{(d,i)}$ as a simpler notation of this distribution.

In the M-step, new optimal parameter values are computed, given the re-estimated latent variables. Formally, the parameters are learned by:

$$
\begin{aligned}
\boldsymbol{\Psi}^{t+1} = \underset{\boldsymbol{\Psi}}{\operatorname{argmax}}(&Q(\boldsymbol{\Psi}|\boldsymbol{\Psi}^t, \boldsymbol{\Psi}_G^t) \\
&+ \sum_{d=1}^{M} \delta_d (1 - \sum_{z=1}^{K} \theta_{d,z}) + \sum_{z=1}^{K} \delta_z (1 - \sum_{v=1}^{V} \varphi_{z,v}))
\end{aligned}
\tag{7.8}
$$

where the second and third terms are the Lagrange multipliers. Solving Equation (7.8) results in the following parameter estimates:

$$
\theta_{d,z}^{t+1} = \frac{\sum_{i=1}^{N}\sum_{v=1}^{V} w_{z,v}^{(d,i)}}{\sum_{z=1}^{K}\sum_{i=1}^{N}\sum_{v=1}^{V} w_{z,v}^{(d,i)}} \tag{7.9}
$$

$$\varphi_{z,v}^{t+1} = \frac{\sum_{d=1}^{M}\sum_{i=1}^{N} w_{z,v}^{(d,i)}}{\sum_{v=1}^{V}\sum_{d=1}^{M}\sum_{i=1}^{N} w_{z,v}^{(d,i)}} \tag{7.10}$$

Finally, for each category, the regularized parameter estimate $\boldsymbol{\theta}_G$ is updated by:

$$\theta_{G_z}^{t+1} = \frac{\exp\left(\frac{1}{MN}\sum_{d=1}^{M}\sum_{i=1}^{N}\log\theta_{d,z}^{t+1}\right)}{\sum_{z'=1}^{K}\exp\left(\frac{1}{MN}\sum_{d=1}^{M}\sum_{i=1}^{N}\log\theta_{d,z'}^{t+1}\right)} \tag{7.11}$$

which is essentially the geometric mean of the observation-dependent $\theta_{d,z}$ in the same category. The log scale is applied to make the computation more manageable when $\theta_{d,z} \to 0$.

Given a new observation $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, the inference process selects the category that is most compatible with $\boldsymbol{Y}$. The first step is to estimate $\hat{\boldsymbol{\theta}}_d$ according to Equation (7.9), which depends on the observation. Then, with the estimated $\hat{\boldsymbol{\theta}}_d$, the RG-PLSA model chooses the category $C(\boldsymbol{Y})$ with the highest probability to generate the observation:

$$C(\boldsymbol{Y}) = \underset{c}{\operatorname{argmax}}\, P(\boldsymbol{Y}|\boldsymbol{\Phi}_c) \tag{7.12}$$

where $\boldsymbol{\Phi}_c = \{\hat{\boldsymbol{\theta}}_d, \boldsymbol{\varphi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}_c$ is the model parameters for the category c. It should be noted that the RG-PLSA model and the Gaussian PLSA model have the same inference process.

7.5 Experiments

We evaluate our RG-PLSA model on the task of human action recognition, using two publicly available benchmark datasets: Weizman [Gorelick et al., 2007] and KTH [Schuldt et al., 2004] datasets.

7.5.1 Methodology

We use the following pipeline to solve the task of human action recognition from a sequence of visual data: 1) video preprocessing, 2) feature extraction, 3) Gaussian component learning, and 4) action classification. In the preprocessing step, the foreground-background segmentation methods can be used to detect the regions of interest (ROIs) that contain the human subjects performing an action. However, in this work, we directly use the ROIs provided by the datasets*, since detecting ROIs is not our main concern.

* The bounding box data of the human subjects in the KTH Action Dataset are provided in [Lin et al., 2009], which are publicly available at: http://www.umiacs.umd.edu/~zhuolin/PrototypeTree/KTHBoundingBoxInfo.txt

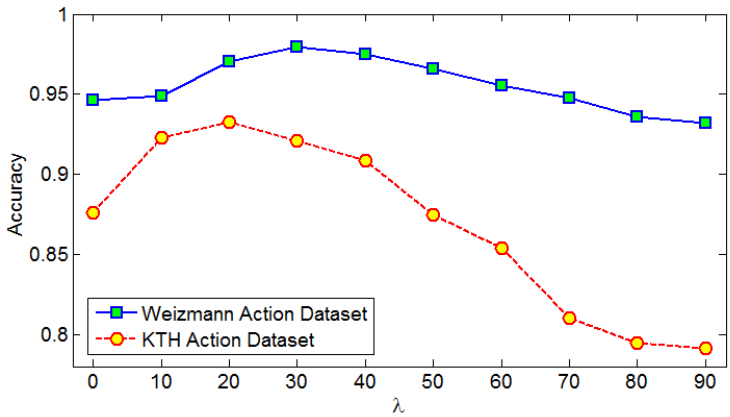


Figure 7.2: Average classification accuracy of the RG-PLSA model over the KTH and Weizmann datasets.

In the feature extraction step, the Scale Invariant Feature Transform (SIFT) [Lowe, 2004] feature detector and descriptor are used to represent local human appearance and shape. The SIFT features are reasonably invariant to changes in illumination, noise, rotation, scaling, and small changes in viewpoint. The feature detector detects the keypoints, i.e., the locations of features in an image, by locating the local extrema of the difference-of-Gaussian filters at different scales. Then, the feature descriptor is employed to compute the orientation histograms around each keypoint, which results in a feature vector with 128 elements. Finally, principal component analysis (PCA) is applied to project each feature vector to a lower dimensional space, leading to a 10 dimensional feature vector.

In the third step, the we fit the standard GMM model over the feature space. The number of Gaussian components is manually set to be 600 for both datasets in our experiments. The objective of this step is to obtain the Gaussian components, which are fixed during the RG-PLSA model's learning and inference processes. The Gaussian component learning step is similar to the vocabulary construction process in the original PLSA model with discrete features.

At last, the introduced RG-PLSA model is employed to perform human action categorization. Evaluation is done using the leave-one-person-out cross-validation technique, in which videos of one human subject are used as the validation data, and videos from the remaining subjects are used as the training data. This is repeated in such a way that videos from one subject are used exactly once as the validation data.

7.5.2 Results on Weizmann Dataset

Figure 7.2 illustrates the average classification accuracy of the proposed RG-PLSA model over the Weizmann dataset across different values of the regularization factor λ. When $\lambda = 30$, the RG-PLSA model achieves the best average accuracy of 97.96%,

Table 7.2: Comparison of action classification accuracy (%) over the Weizmann and KTH action datasets

The Weizmann Dataset		The KTH Dataset	
Methods	Accuracy	Methods	Accuracy
RG-PLSAs	**97.96**	**RG-PLSAs**	**93.27**
Gaussian PLSAs	94.65	Gaussian PLSAs	87.60
HGMMs	92.14	HGMMs	78.78
[Lin et al., 2009]	100	[Lin et al., 2009]	95.77
[Niebles et al., 2008]	90.00	[Niebles et al., 2008]	81.50

which outperforms the HGMM model and the Gaussian PLSA model, as shown in Table 7.2. When $\lambda \to 0$, the RG-PLSA model tends to overfit the data, and the average accuracy deceases. With $\lambda = 0$, the RG-PLSA model obtains the same average accuracy as the Gaussian PLSA model. When λ becomes too large, the model tends to underfit the data, and the average accuracy tends to slowly decrease. We also provide the results from other approaches, as listed in Table 7.2, to show that our method is comparable to the state of the art. We would like to emphasize that we are not making a direct comparison, because different approaches have variations in preprocessing, feature extraction, and experimental settings.

7.5.3 Results on KTH Dataset

Figure 7.2 depicts our model's average classification accuracy over the KTH Action Dataset across different values of λ. With $\lambda = 20$, the best average classification accuracy of 93.27% is achieved. This result empirically demonstrates the benefit of applying the regularization term to the Gaussian PLSA model to improve classification accuracy. It should be noted that comparing to the results in the Weizmann dataset, the best result with the KTH dataset is achieved with a smaller λ, which allows for more model complexity and emphasizes more on the variations in the data. This occurs because the KTH dataset contains more variations and is more complicated than the Weizmann dataset. On the other hand, similar to the results in the Weizmann dataset, very large λ values result in underfitting, and very small values lead to overfitting. We also show the comparison with other approaches over the KTH Action Dataset in Table 7.2, which empirically demonstrates that the RG-PLSA model is comparable to the state of the art.

7.6 Summary

7.6.1 Discussion

An alternative approach to control model complexity is to use Bayesian priors over the parameters as the regularization term in the parameter learning process, leading

to MAP parameter estimation. However, assigning initial values to the priors usually requires significant domain knowledge, which is not always available. On the other hand, the regularization term in the RG-PLSA model is automatically computed without any manual initialization.

However, the introduction of the regularization term also introduces an additional parameter λ, which significantly affects classification performance. This parameter needs to be manually selected in this work. Unfortunately, we do not know a priori which value is the best for a given problem. Consequently, we need to explore model selection algorithms for selecting the best λ. Heuristic search and grid-search with cross-validation can be good ways to achieve this objective.

The Gaussian component parameters are learned separately from the model learning process in our RG-PLSA model. However, it is straightforward to extend the proposed model to simultaneously update all parameters in $\mathbf{\Phi}$ with the EM algorithm, which directly follow the results in [Hörster et al., 2008]. Furthermore, the number of Gaussian components is usually manually selected. To further improve classification accuracy, it is necessary and important to automatically determine the proper number of Gaussian components with some model selection criteria, such as the Bayesian information criterion.

7.6.2 Conclusion

We introduce the novel RG-PLSA model that combines the Gaussian PLSA model and the HGMM model to simultaneously prevent overfitting and provide moderate flexibility to model observations with continuous values. The proposed model employs a regularization term in the standard PLSA learning process to control model complexity. The RG-PLSA model's parameters are learned with the EM algorithm. We use two publicly available benchmark dataset to evaluate the effectiveness of our model on the human action recognition tasks. We achieve the accuracy of 97.96% for the Weizmann Action Dataset, and the accuracy of 93.27% for the KTH Action Dataset. These experimental results demonstrate that the proposed RG-PLSA model outperforms Gaussian PLSAs and HGMMs, which are comparable to the state of the art.

Chapter 8

Recognition: MC-HCRF Model for Sequential Activity Understanding

8.1 Introduction

Human activity recognition is an important research topic that has a wide range of real-world applications in computer vision. However, human activity recognition from visual perception is a challenging problem due to illumination changes, diversities of human motions, variations of human appearances, etc. Most importantly, recognition of *sequential* human activities requires modeling their underlying temporal patterns. For example, it is generally impossible to separate "standing up" and "sitting down" from a single image, since humans may exhibit the same pose in the image for both activities; therefore, modeling their temporal patterns is necessary.

A most popular and powerful algorithm to encode human activity's temporal patterns is the Conditional Random Field (CRF) [Lafferty et al., 2001] model, which is a discriminative graphical model that avoids encoding the distribution over the input and is able to incorporate arbitrary overlapping features. However, CRFs are limited in the ability to combine latent variables that can capture underlying patterns within observations [Quattoni et al., 2007]. For example, a robot teacher may need to model a complex activity "tennis serving", where atomic temporal motion patterns, such as "ball tossing" and "racquet swinging", are unknown, and thus must be modeled using latent variables. To address this problem, Hidden Conditional Random Fields (HCRFs) [Quattoni et al., 2007] were introduced to combine CRF model's strengths with latent variables. Due to the latent variable's ability to model temporal patterns of a sequence, HCRFs are becoming increasingly popular in sequence labeling, including human activity recognition.

A key observation in previous HCRFs is that certainty in latent temporal patterns is never explicitly analyzed; the latent variables that encode temporal patterns are eliminated either through summation in the HCRF models based on maximum likelihood estimation (MLE) [Quattoni et al., 2007] or through maximization in max-margin (MM) HCRF models [Wang and Mori, 2011]. On the other hand, the latent

temporal pattern can provide useful information for improving prediction accuracy [Grandvalet and Bengio, 2004, Kumar et al., 2012]. In addition, there are many real-world scenarios in which we need a confident understanding of the latent temporal pattern itself. For example, the robot trying to teach a tennis serve needs to maximize its confidence on the chronological order of "ball tossing" and "racquet swinging" in the temporal pattern.

Our work in this chapter addresses this problem. We introduce a novel HCRF model to perform recognition of sequential activities by modeling their latent temporal structures. We call our new model *maximum-certainty HCRF* (MC-HCRF) in order to emphasize its capability of incorporating the information encoded in unobserved temporal structures. Our MC-HCRF is constructed as a multi-objective optimization problem that simultaneously maximizes the probability of the correct label and the certainty in the latent variables used to model temporal structures. We formulate our MC-HCRF under the energy-based learning framework [Lecun et al., 2006] and introduce a new energy function that incorporates both objectives to allow for efficient inference and learning. In addition, our MC-HCRF, as a dynamic graphical model, provides a sparse and factorized representation of the distribution over sequential data, leading to tractable inference.

The main contributions of this work are threefold. First, we introduce the novel MC-HCRF model, which explicitly models the certainty in latent underlying temporal patterns of sequential human activities, and we also design efficient inference and learning algorithms for our model. Second, we introduce an alternative perspective of HCRFs, i.e., inferring and learning HCRFs in the energy-based learning framework, which allows for a direct application of existing sophisticated energy-based learning methods to better estimate model parameters. Third, we empirically validate that HCRFs can benefit from energy-based formulation as well as temporal pattern certainty maximization, through showing that our MC-HCRF models obtain state-of-the-art performance on human activity recognition, especially over sequential activities, using four benchmark activity datasets.

The rest of this chapter is structured as follows. Section 10.2 provides a review of related work. In Section 8.3, the formulation of HCRFs under energy-based learning is discussed. Then, our MC-HCRF model is introduced in Section 8.4. Experimental results are presented in Section 8.5. Finally, we summarize this work in Section 10.7.

8.2 Related Work

General reviews of activity recognition are conducted in [Aggarwal and Ryoo, 2011]. In what follows, we discuss previous studies that focus on modeling temporal patterns for sequential activity recognition, which can be grouped into the following two categories.

The first category uses space-time features to model temporal patterns of sequential activities. Laptev [Laptev, 2005] applied histogram of spatial gradient (HOG) and optic flow (HOF) to describe local motion and appearance patterns in space-time neighborhoods of the detected interest points. Dollar et al. [Dollár et al., 2005] described spatio-temporal features by concatenating the gradients around the interest point's space-time neighborhoods. Recently, Wang et al. [Wang et al., 2013a] introduced the motion boundary histograms (MBH) to describe temporal motion variations. Some other spatio-temporal features were also introduced in [Chakraborty et al., 2012, Oreifej and Liu, 2013, Wang et al., 2012a]. These features are often used to formulate a bag-of-words (BoW) model to represent human activities. Although satisfactory activity recognition performance has been reported, space-time features encode temporal patterns only within a short period of time, in general.

The second category to recognize sequential activities is based on dynamic graphical models, which are able to represent temporal structures that extend over long periods of time. In a generative setting, Dynamic Bayesian Networks (DBNs) [Zeng and Ji, 2010] are a popular method for sequence modeling, because they exploit structure in the problem to compactly represent distributions over multiple state variables. Hidden Markov Models (HMMs) [Brand, 1999] and their extensions [Piyathilaka and Kodagoda, 2013], a special case of DBNs, are a classical method for sequential activity recognition. In a discriminative setting, CRFs [Lafferty et al., 2001], HCRFs [Quattoni et al., 2007] and their extensions [Wang and Mori, 2011] are the most widely used approaches for modeling activity temporal structures. Although previous dynamic models use hidden variables to model the latent temporal pattern, the certainty in this pattern is not well studied. We address this issue for HCRFs.

Our work is different from the previous HCRFs in that, besides maximizing the correct assignment's probability as in [Quattoni et al., 2007, Wang and Mori, 2011], our model also aims to maximize latent temporal pattern certainty. Instead of making the partially observable assumption for the latent variables and modeling their uncertainty to incorporate missing data in semi-supervised structured learning scenarios as in [Grandvalet and Bengio, 2004, Kumar et al., 2012], we focus on supervised learning with no observed latent variables, and we assume that the latent temporal structure satisfies the first-order Markov property that is used to efficiently model the temporal pattern of a sequence. We also focus on analyzing how HCRFs benefit from the combination of energy-based learning and temporal pattern certainty maximization for sequential activity classification, as compared to improving different models in other applications [Brand, 1999, Grandvalet and Bengio, 2004, Kumar et al., 2012, Li et al., 2004, Miller et al., 2012].

8.3 HCRFs under Energy-Based Learning

We propose to formulate HCRFs under the energy-based learning framework. This is of essential importance since it allows for a direct application of sophisticated energy-based optimization algorithms to better estimate HCRF's parameters, including Bundle Cutting Plane (BCP) [Teo et al., 2010] and Non-convex Regularized Bundle Method (NRBM) [Do and Artières, 2012].

8.3.1 Energy-Based Learning w/o Latent Variables

Energy-based model (EBM) [Lecun et al., 2006] captures dependencies by associating a scalar energy function $e(\boldsymbol{x}, y; \boldsymbol{\theta})$ with each configuration of the inputs and the output. The value of $e(\boldsymbol{x}, y; \boldsymbol{\theta})$ is interpreted as the degree of compatibility between the variable values. By convention, a smaller value indicates higher compatibility. Inference of EBMs is conducted by picking the class label that minimizes the energy function, i.e., $y^* = \text{argmin}_{y \in \mathcal{Y}}\, e(\boldsymbol{x}, y; \boldsymbol{\theta})$. Learning of model parameters $\boldsymbol{\theta}$ is performed by minimizing a regularized loss function, i.e., $\boldsymbol{\theta}^* = \text{argmin}_{\boldsymbol{\theta}}\, l(\boldsymbol{x}, y; \boldsymbol{\theta})$, in which $l(\boldsymbol{x}, y; \boldsymbol{\theta})$ is typically defined as:

$$\begin{aligned} l(\boldsymbol{x}, y; \boldsymbol{\theta}) &= \lambda \cdot l_r(\boldsymbol{\theta}) + l_{emp}(\boldsymbol{x}, y; \boldsymbol{\theta}) \\ &= \lambda \cdot l_r(\boldsymbol{\theta}) + \frac{1}{N} \sum_{i=1}^{N} l_p(e(\boldsymbol{x}_i, y_i; \boldsymbol{\theta})) \end{aligned} \tag{8.1}$$

where $l_r(\boldsymbol{\theta})$ is the regularizer, $l_{emp}(\boldsymbol{x}, y; \boldsymbol{\theta})$ is the empirical loss function, and $\lambda > 0$ is the regularization constant, which defines a trade-off between model complexity and fitting. The empirical loss $l_{emp}(\boldsymbol{x}, y; \boldsymbol{\theta})$ is defined as the arithmetic mean of the per-sample loss function $l_p(e(\boldsymbol{x}, y; \boldsymbol{\theta}))$, which is a function of $e(\boldsymbol{x}, y; \boldsymbol{\theta})$. The main advantage provided by EBMs is that they place little restriction on the energy function, which is not required to be a probability distribution or a convex function. However, traditional EBMs do not deal with latent vairables.

8.3.2 Modeling HCRFs as EBMs

When using HCRFs for supervised multi-class classification, the goal is to learn a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ from a set of i.i.d. training data $\mathcal{D} = \{(\boldsymbol{x}^i, y^i), i = 1, \ldots, N\}$, to predict a class label $y \in \mathcal{Y}$ for an observation $\boldsymbol{x} \in \mathcal{X}$. Each observation is a vector of M attributes $\boldsymbol{x} = \{x_1, \ldots, x_M\}$, where $x_j \in \mathbb{R}$, $j = 1 \ldots M$, is an attribute extracted from visual data. The HCRF model also defines a vector of latent variables $\boldsymbol{h} = \{h^1, \ldots, h^N\}$, where $h^i \in \mathcal{H}$, $h = 1, \ldots, N$, corresponds to a hidden label associated with the observation $\boldsymbol{x}^i$. In the previous tennis serve example, h may correspond to the atomic "ball tossing" motion.

The HCRF is defined on an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose nodes satisfy $\mathcal{V} = \{\boldsymbol{x} \cup \boldsymbol{h} \cup y\}$. The HCRF graph is annotated with a set of real-valued potentials

$\boldsymbol{\psi}(\boldsymbol{D};\boldsymbol{\theta})=\{\psi_1(\boldsymbol{D}_1;\theta_1),\ldots,\psi_P(\boldsymbol{D}_P;\theta_P)\}$, where $\boldsymbol{D}$ is the scope of the potential ψ that satisfies $\boldsymbol{D}\subseteq\mathcal{V}$ and $\boldsymbol{D}\not\subseteq\boldsymbol{x}$, θ is the parameter, and P is the number of potentials. The HCRF network is connected with undirected edges $\mathcal{E}=\{v_i-v_j:\{v_i,v_j\}\subseteq\boldsymbol{D}_k;\forall i\neq j,k=1,\ldots,P\}$. HCRFs encode the following conditional distribution:

$$P(y,\boldsymbol{h}|\boldsymbol{x};\theta)=\frac{1}{Z(\boldsymbol{x};\theta)}\tilde{P}(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta}) \tag{8.2}$$

where $\tilde{P}(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta})$ is the unnormalized measure that can be represented by a product of potentials, i.e., $\tilde{P}(y,\boldsymbol{h}|\boldsymbol{x};\theta)=\prod_i\psi_i(\boldsymbol{D}_i;\theta_i)$, and each potential $\psi_i(\boldsymbol{D}_i;\theta_i)$ must capture some domain knowledge about the structure of the latent variables; $Z(\boldsymbol{x};\boldsymbol{\theta})$ is the partition function that is computed by $Z(\boldsymbol{x};\boldsymbol{\theta})=\sum_{y\in\mathcal{Y},\boldsymbol{h}\in\mathcal{H}}\tilde{P}(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta})$.

We propose to infer and learn HCRFs under the energy-based learning framework, with additional considerations to deal with latent variables that are not handled by traditional EBMs. From this perspective, MLE-HCRFs can be treated as using the l_2-regularization with the following energy and per-sample loss functions:

$$\begin{aligned}&e(y,\boldsymbol{x};\boldsymbol{\theta})=-\log P(y|\boldsymbol{x};\boldsymbol{\theta})=-\log\sum_{\boldsymbol{h}}P(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta})\\&l_p(y,\boldsymbol{x};\boldsymbol{\theta})=e(y,\boldsymbol{x};\boldsymbol{\theta})\end{aligned} \tag{8.3}$$

MM-HCRFs can be treated as using the l_2-regularization with the following energy function and the soft margin loss function:

$$\begin{aligned}&e(y,\boldsymbol{x};\boldsymbol{\theta})=-\max_{\boldsymbol{h}}\ \boldsymbol{\theta}^{\mathsf{T}}\boldsymbol{\phi}(y,\boldsymbol{x},\boldsymbol{h})\\&l_p(y,\boldsymbol{x};\boldsymbol{\theta})=\max_{y'}\big(\Delta(y,y')+e(y,\boldsymbol{x};\boldsymbol{\theta})-e(y',\boldsymbol{x};\boldsymbol{\theta})\big)\end{aligned} \tag{8.4}$$

where $\boldsymbol{\phi}(\cdot)$ denotes a set of basis functions, and $\Delta(y,y')$ is the misclassification loss.

It can be observed that this formulation can deal with the latent variables of previous HCRFs by either summation or maximization. However, a drawback of these models is that latent variable certainty is not explicitly analyzed.

8.4 Our Maximum Certainty HCRFs

In this section, we start by describing our metric to measure latent variable certainty. Then, we propose our energy-based MC-HCRFs, and detail how inference and learning are performed. At last, we discuss our model's relationships to previous HCRFs.

8.4.1 Certainty Measure

An objective of introducing our MC-HCRFs is to explicitly increase certainty in the latent variables that encode underlying temporal patterns. In information theory,

entropy is widely used to measure random variable certainty. In this work, we use the Karpur entropy [Kapur, 1967] as our certainty metric. Given discrete random variables $\boldsymbol{z}$, the Kapur entropy of order α and type β is defined as:

$$H_{\alpha,\beta}(P(\boldsymbol{z})) = \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{z}} P(\boldsymbol{z})^{\alpha+\beta-1}}{\sum_{\boldsymbol{z}} P(\boldsymbol{z})^{\beta}} \tag{8.5}$$

where $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$, and $\alpha+\beta-1 > 0$. If $\alpha \to 0$, $\beta = 1$, the Kapur entropy becomes the Hartley function [Klir, 2005], i.e., $H_{0,1}(P(\boldsymbol{z})) = \log K$, where K is the number of variables in $\boldsymbol{z}$ with a positive probability. In the limit $\alpha \to 1$ and $\beta = 1$, the Kapur entropy converges to the Shannon entropy. When $\alpha \to \infty$ and $\beta = 1$, we can obtain a quantity analogous to the Chebyshev norm, i.e., $H_{\infty,1}(P(\boldsymbol{z})) = -\log \max_{\boldsymbol{z}} P(\boldsymbol{z})$. The Kapur entropy is not convex, in general.

8.4.2 Model Formulation

Besides the objective of maximizing the correct label's conditional probability (i.e., same as the MLE-HCRF's objective in Eq. (8.3)), our MC-HCRFs also aim to achieve another objective, i.e., to maximize certainty in the latent variables that are used to model temporal patterns. As a result, our model forms a multi-objective optimization problem:

$$\text{maximum}\,(\, P(y|\boldsymbol{x};\boldsymbol{\theta}),\ -H_{\alpha,\beta}(P(\boldsymbol{h}|y,\boldsymbol{x};\boldsymbol{\theta}))\,) \tag{8.6}$$

A commonly used methodology to solve multi-objective optimization is to incorporate multiple objectives into a single objective function [Marler and Arora, 2004]. We propose to solve our multi-objective optimization problem through representing MC-HCRFs as energy-based models and defining a new energy function $e(y,\boldsymbol{x};\boldsymbol{\theta})$ that satisfies the following theorem:

Theorem 8.1. *The MC-HCRF's energy function:*

$$e(y,\boldsymbol{x};\boldsymbol{\theta}) = -\log P(y|\boldsymbol{x};\boldsymbol{\theta}) + H_{\alpha,\beta}(P(\boldsymbol{h}|y,\boldsymbol{x};\boldsymbol{\theta})) - \log Z(\boldsymbol{x};\boldsymbol{\theta}) \tag{8.7}$$

combines both objectives in the multi-objective optimization problem in Eq. (9.6).

Proof. in Appendix 11.2. □

Intuitively, $e(y,\boldsymbol{x};\boldsymbol{\theta})$ is encoded as a summation of the negative log-likelihood of $P(y|\boldsymbol{x};\boldsymbol{\theta})$, the entropy of $\boldsymbol{h}$ (i.e., $H_{\alpha,\beta}(P(\boldsymbol{h}|y,\boldsymbol{x};\boldsymbol{\theta}))$ that encodes latent variable certainty, and a constant that is independent of the output y. Accordingly, minimizing $e(y,\boldsymbol{x};\boldsymbol{\theta})$ is equivalent to simultaneously maximizing $P(y|\boldsymbol{x};\boldsymbol{\theta})$ and minimizing $H_{\alpha,\beta}(P(\boldsymbol{h}|y,\boldsymbol{x};\boldsymbol{\theta})$, which is the exact multi-objective function in Eq. (9.6).

We use the soft margin loss as the per-sample loss in our energy-based MC-HCRF model, leading to:

$$l_{emp}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\max_{y'^i\in\mathcal{Y}}\big(\Delta(y^i, y'^i)+e(y^i, \boldsymbol{x}^i;\boldsymbol{\theta}) - e(y'^i, \boldsymbol{x}^i;\boldsymbol{\theta})\big) \tag{8.8}$$

where $\Delta(y, y')$ is the 0-1 loss. The soft margin loss encodes the separation between correct and incorrect assignments. As a result, maximizing $l_{emp}(\boldsymbol{\theta})$ is equivalent to increasing the margin between the correct and incorrect classes, which thus incorporates the max-margin criterion.

8.4.3 Inference

Given an observation $\boldsymbol{x}$, the output label y is inferred by selecting the class that minimizes the energy function:

$$y^* = \operatorname*{argmin}_{y\in\mathcal{Y}} e(y, \boldsymbol{x};\boldsymbol{\theta}) \tag{8.9}$$

which leads to not only a high probability of the output, but also a high certainty in the latent variables.

We use three types of potentials in our MC-HCRFs: the pairwise potential $\psi(h_i, h_j, y)$ and the singleton potentials $\psi(h_i, y)$ and $\psi(h_i, \boldsymbol{x}_i)$. To model temporal structures of sequential human activities, we assume that the MC-HCRF's underlying graph satisfies the first-order Markov property, forming a tree-structured chain. Consequently, we can efficiently compute the MC-HCRF's energy function and solve the inference problem using a belief propagation algorithm. The tractability is formally presented in the following theorem and corollary:

Theorem 8.2. *If the latent variables $\boldsymbol{h}$ form a graph without loops, computation of the energy-based MC-HCRF's energy function $e(y, \boldsymbol{x};\boldsymbol{\theta})$, $\forall y$, is tractable, having the time complexity of $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$.*

Proof. in Appendix 11.2. □

Corollary 8.2.1. *Inference of our energy-based MC-HCRFs is performed in $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ time.*

Proof. After computing $e(x, \boldsymbol{y};\boldsymbol{\theta})$, $\forall y$, in an $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ runtime, $\operatorname{argmin}(\cdot)$ can be performed in $O(|\mathcal{Y}|)$ time. □

8.4.4 Learning

We now discuss how to train our MC-HCRFs from given i.i.d. training data $\mathcal{D} = \{(\boldsymbol{x}^i, y^i), i = 1, \ldots, N\}$. The MC-HCRF model's parameter $\boldsymbol{\theta}$ is estimated

by minimizing the l_2-regularized loss function $l(\boldsymbol{\theta}) = \lambda l_r(\boldsymbol{\theta}) + l_{emp}(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{\lambda}{2} \| \boldsymbol{\theta} \|^2 + \frac{1}{N} \sum_{i=1}^{N} \max_{y'^i \in \mathcal{Y}} \big(\Delta(y^i, y'^i) + e(y^i, \boldsymbol{x}^i; \boldsymbol{\theta}) - e(y'^i, \boldsymbol{x}^i; \boldsymbol{\theta}) \big) \tag{8.10}$$

A most popular approach to perform energy-based learning is the BCP method [Teo et al., 2010], based on the cutting plane technique [Franc and Sonnenburg, 2009]. A cutting plane of $l_{emp}(\boldsymbol{\theta}) = l_{emp}(\boldsymbol{x}, y; \boldsymbol{\theta})$ at $\boldsymbol{\theta}'$ is defined as:

$$\begin{array}{rcl} c_{\boldsymbol{\theta}'}(\boldsymbol{\theta}) & = & a_{\boldsymbol{\theta}'}^{\mathsf{T}} \boldsymbol{\theta} + b_{\boldsymbol{\theta}'} \\ \text{subject to} & & c_{\boldsymbol{\theta}'}(\boldsymbol{\theta}') = l_{emp}(\boldsymbol{\theta}') \\ & & \partial_{\boldsymbol{\theta}} c_{\boldsymbol{\theta}'}(\boldsymbol{\theta}') \in \partial_{\boldsymbol{\theta}} l_{emp}(\boldsymbol{\theta}') \end{array} \tag{8.11}$$

where $a_{\boldsymbol{\theta}'} = \partial_{\boldsymbol{\theta}} l_{emp}(\boldsymbol{\theta}')$, and $b_{\boldsymbol{\theta}'} = l_{emp}(\boldsymbol{\theta}') - a_{\boldsymbol{\theta}'}^{\mathsf{T}} \boldsymbol{\theta}'$. The cutting plane $c_{\boldsymbol{\theta}'}(\boldsymbol{\theta})$ is a linear lower bound of $l_{emp}(\boldsymbol{\theta})$. The BCP method iteratively builds an increasingly accurate piecewise quadratic lower bound of $l(\boldsymbol{\theta})$. Given an initial value, $\boldsymbol{\theta}$ is iteratively updated by:

$$\begin{array}{l} \boldsymbol{\theta}_{t+1} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, g_t(\boldsymbol{\theta}) \quad \text{and} \quad v_t = \min_{\boldsymbol{\theta}} g_t(\boldsymbol{\theta}) \\ \text{with} \quad g_t(\boldsymbol{\theta}) = \lambda \cdot l_r(\boldsymbol{\theta}) + \max_{j=1,\ldots,t} (c_j(\boldsymbol{\theta})) \end{array} \tag{8.12}$$

If $l_{emp}(\boldsymbol{\theta})$ is convex, $c_j(\boldsymbol{\theta}) \equiv c_{\boldsymbol{\theta}'}(\boldsymbol{\theta})$ as defined in Eq. (8.11).

However, similar to MLE and MM-HCRFs, the energy function of our MC-HCRFs is not convex in general, and the commonly used convex solvers, such as BCP, cannot solve Eq. (8.10). To solve this non-convex optimization problem, we adopt the NRBM algorithm [Do and Artières, 2012] that is described in Algorithm 3. Since a cutting plane of $l_{emp}(\boldsymbol{\theta})$ is not necessarily a lower bound, a *conflict* occurs if and only if the cutting plane does *not* satisfy:

$$c_{\boldsymbol{\theta}_t}(\boldsymbol{\theta}_t^*) = a_{\boldsymbol{\theta}_t}^{\mathsf{T}} \boldsymbol{\theta}_t^* + b_{\boldsymbol{\theta}_t} \leq l_{emp}(\boldsymbol{\theta}_t^*) \tag{8.13}$$

where $\boldsymbol{\theta}_t^*$ are the best observed parameters up to now (line 4 in Algorithm 3), in which case $l_{emp}(\boldsymbol{\theta})$ is overestimated at $\boldsymbol{\theta}_t^*$. The conflict is solved by tuning the parameters a_t and b_t to form an alternative cutting plane, $c_t(\boldsymbol{\theta}_t) = a_t^{\mathsf{T}} \boldsymbol{\theta}_t + b_t$, which satisfies Eq. (8.13) and the following condition:

$$\lambda l_r(\boldsymbol{\theta}_t) + c_t(\boldsymbol{\theta}_t) \geq l(\boldsymbol{\theta}_t^*) \tag{8.14}$$

The conflict resolution procedure is described between line 5 and line 11. Using the l_2-regularizer, the NRBM method is guaranteed to produce an approximation gap smaller than ϵ after T iterations and to converge with a convergence rate $O(1/(\lambda\epsilon))$ [Do and Artières, 2012], where $T \leq T_0 + 8C^2/(\lambda\epsilon) - 2$ with $T_0 = 2\log(\lambda \parallel \boldsymbol{\theta}_0 +$

Algorithm 3: NRBM for Learning MC-HCRFs

Input : $\mathcal{T}_c$, $\boldsymbol{\psi}(\boldsymbol{D})$, $\boldsymbol{\theta}_0$, λ, ϵ, $\mathcal{D}=\{(\boldsymbol{x}^i, y^i), i=1,\ldots,N\}$
Output : $\boldsymbol{\theta}^*$

for $t \leftarrow 0$ **to** ∞ **do**
  Compute $l_{emp}(\boldsymbol{\theta})$ over $\mathcal{D}$ acc. to Eq. (2) and (8.8);
  Define $c_{\boldsymbol{\theta}_t}$ with parameters $(a_{\boldsymbol{\theta}_t}, b_{\boldsymbol{\theta}_t})$ acc. to Eq. (8.11);
  Compute $\boldsymbol{\theta}_t^* = \text{argmin}_{\boldsymbol{\theta}_j \in \{\boldsymbol{\theta}_0,\ldots,\boldsymbol{\theta}_t\}} l(\boldsymbol{\theta}_j)$;
  if $c_{\boldsymbol{\theta}_t}(\boldsymbol{\theta}_t^*) = a_{\boldsymbol{\theta}_t}^\top \boldsymbol{\theta}_t^* + b_{\boldsymbol{\theta}_t} > l_{emp}(\boldsymbol{\theta}_t^*)$ **then** `/*There is a conflict*/`
    Compute upper bound U of b_t acc. to Eq. (8.13): $U = l_{emp}(\boldsymbol{\theta}_t^*) - a_{\boldsymbol{\theta}_t}^\top \boldsymbol{\theta}_t^* \geq b_t$;
    Compute lower bound L of b_t acc. to Eq. (8.14):
    $L = l(\boldsymbol{\theta}_t^*) - \lambda l_r(\boldsymbol{\theta}_t) - a_{\boldsymbol{\theta}_t}^\top \boldsymbol{\theta}_t \leq b_t$;
    if $L \leq U$ **then** Set $a_t = a_{\boldsymbol{\theta}_t}$ and $b_t = L$;
    else Assign $a_t = -\lambda \cdot \partial_{\boldsymbol{\theta}} l_r(\boldsymbol{\theta}_t^*)$ and $b_t = l(\boldsymbol{\theta}_t^*) - \lambda l_r(\boldsymbol{\theta}_t) - a_t^\top \boldsymbol{\theta}_t$;
    Define alternative cutting plane: $c_t(\boldsymbol{\theta}) = a_t^\top \boldsymbol{\theta} + b_t$;
  else Set $c_t(\boldsymbol{\theta}) = c_{\boldsymbol{\theta}_t}(\boldsymbol{\theta})$;
  Update $\boldsymbol{\theta}_{t+1}$ and compute v_t acc. to Eq. (8.12);
  Compute gap: $G_t = l(\boldsymbol{\theta}_t^*) - v_t$;
  if $G_t \leq \epsilon$ **then return** $\boldsymbol{\theta}_t^*$;
end

$a_0/\lambda \| /C) - 2$, and C is an upper bound on the norm of the cutting plane direction parameters.

8.4.5 Relationship to MLE and MM-HCRFs

We discuss our MC-HCRF model's relationship to previous HCRF models from the energy-based learning perspective. For MLE-HCRFs, we obtain the proposition:

Proposition 2. *MLE-HCRF's energy function has the same form as the MC-HCRF's energy when $\alpha \to 0$ and $\beta = 1$.*

Proof. in Appendix 10. □

Although MLE-HCRF's energy is a special case of our model's energy function, since MLE-HCRFs do not use the soft margin loss as its per-sample loss function, they do not incorporate the large-margin criterion. As for MM-HCRFs, we have the following proposition:

Proposition 3. *The MM-HCRF model is equivalent to the MC-HCRF model, when $\alpha \to \infty$, $\beta = 1$, and the potentials have the form $\boldsymbol{\psi}(\boldsymbol{D};\boldsymbol{\theta}) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\boldsymbol{D}))$.*

Proof. in Appendix 10. □

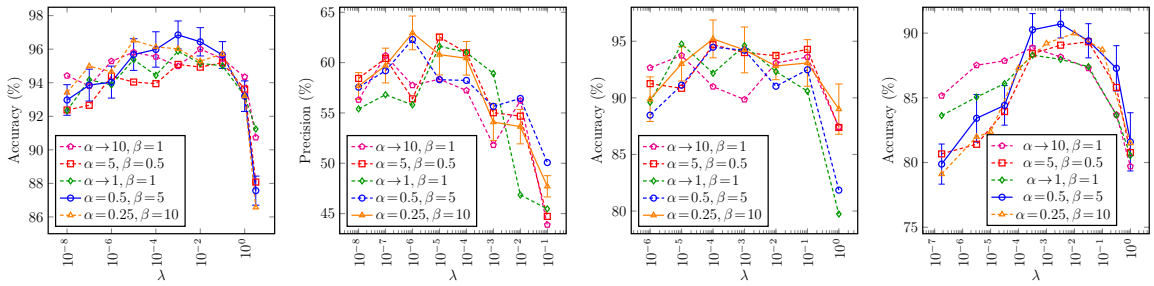
8.5 Experiments

Extensive experiments are performed to demonstrate our MC-HCRF's state-of-the-art performance on classifying sequential activities that are encoded by BoW or skeleton motion sequences (SMS). Following [Wang et al., 2006], we apply potentials with the form $\boldsymbol{\psi}(\boldsymbol{D};\boldsymbol{\theta}) = \exp(\boldsymbol{\theta}\cdot\boldsymbol{\phi}(\boldsymbol{D}))$ and set $|\mathcal{H}| = 10$. We split each dataset into disjoint training and testing sets. Fivefold cross-validation is employed over the training set to estimate model hyper-parameters. Then, the final model is trained using the selected hyper-parameters on the entire training set. Finally, we evaluate our model's performance over testing sets.

8.5.1 Results on KTH Dataset

Following [Schuldt et al., 2004], we adopt the all-in-one experimental setup, i.e., all scenarios are used in training and testing. We split the dataset by randomly selecting 16 subjects for learning and the remaining subjects for testing.

We use the standard BoW representation to evaluate our model. After applying cuboid detectors [Dollár et al., 2005], following [Wang et al., 2013a], we construct a codebook for the HOG, HOF, and MBH descriptors* via k-means quantization. We fix the number of visual words for each descriptor to 4000, which has empirically shown good results for a wide range of datasets. Then, a total number of 300 words are selected via a feature selection method [Brown et al., 2012] to reduce the complexity. The resulting histogram of visual word occurrences is computed from each frame in a video and used as our activity representation.

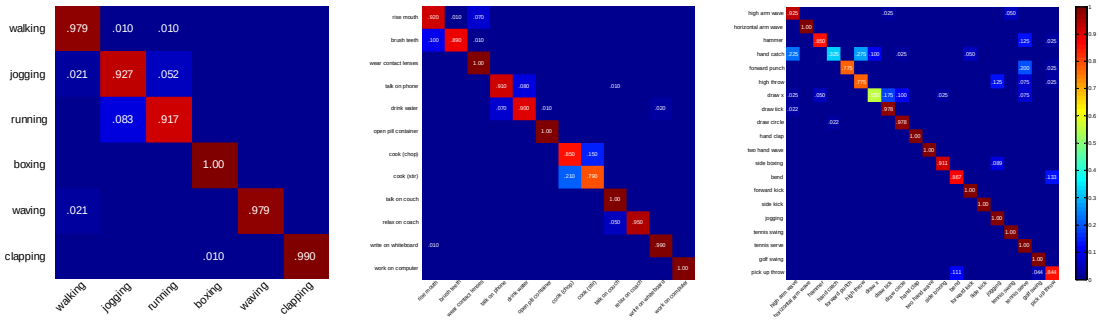


(a) KTH **(b)** Hollywood-2 **(c)** CAD-60 **(d)** MSR Action3D

Figure 8.1: Our MC-HCRF model's performance on training sets using different hyper-parameter settings. For a clear presentation, standard deviations are depicted only on the curves that contain the best results (depicted with solid lines).

Figure 8.1a demonstrates our MC-HCRF's accuracy over the training set. Using $\alpha = 0.25$, $\beta = 10$ and $\lambda = 10^{-3}$, the MC-HCRF model obtains the best cross-validation

* Code to compute HOG/HOF/MBH features is available at: http://lear.inrialpes.fr/people/wang/dense_trajectories.



(a) KTH **(b)** CAD-60 **(c)** MSR Action3D

Figure 8.2: Confusion matrix obtained by our MC-HCRF algorithms over testing sets. In the matrix, each column represents the instances in a predicted activity class, while each row represents the instances in an actual activity class. A warmer color denotes a higher recognition accuracy. (Precisions over the Hollywood-2 dataset is presented in Table 8.2.)

accuracy of $96.85 \pm 0.83\%$. Figure 8.1a also indicates that underfitting occurs when λ increases past one, implying that fairly complex models are required for recognizing human activities. Using these hyper-parameters, the MC-HCRF model achieves a 96.53% classification accuracy on the testing set. The confusion matrix obtained by our MC-HCRF model over the KTH dataset is presented in Figure 8.2a.

Table 8.1: Comparison of accuracy (%) over the KTH dataset using the all-in-one experimental setup.

Approach	Acc.	Approach	Acc.
MLE-HCRF [Wang and Mori, 2011]	90.29	Schuldt et al. [Schuldt et al., 2004]	71.72
MLE-HCRF (HOG/HOF/MBH)	92.15	Dollar et.al [Dollár et al., 2005]	81.17
MLE-HCRF (energy-based)	94.24	Wang et al. [Wang et al., 2013a]	95.30
MM-HCRF [Wang and Mori, 2011]	93.07	Gilbert et al. [Gilbert et al., 2009]	95.50
MM-HCRF (HOG/HOF/MBH)	94.50	Chakraborty et al. [Chakraborty et al., 2012]	96.35
MM-HCRF (energy-based)	95.29	**MC-HCRF**	**96.53**

We make comparisons between HCRF models and previous approaches over the KTH dataset, as shown in Table 8.1. For MLE or MM-HCRFs, we compare each energy-based HCRF model with two baselines: HCRFs from [Wang and Mori, 2011] and HCRFs using HOG/HOF/MBH features and the same experimental setups. Table 8.1 (left column) empirically validates that formulating HCRFs in the energy-based learning framework can increase classification accuracy. In addition, we compare our MC-HCRF model, which explicitly models certainty in the underlying temporal pattern, with other HCRFs. As shown in Table 8.1, the MC-HCRF model obtains better results than other energy-based HCRFs, demonstrating that explicitly modeling certainty in the latent temporal pattern can improve model performance.

At last, compared with previous approaches, our MC-HCRF model obtains the state-of-the-art result, as presented in Table 8.1 (right column). To summarize, the comparison in Table 8.1 highlights the benefit of formulating HCRFs under the energy-based learning framework and modeling certainty in the underlying temporal pattern, leading to state-of-the-art results.

Through investigation of our MC-HCRF's sensitivity to the entropy hyper-parameters (i.e., α and β), given a fixed regularization hyper-parameter $\lambda = 10^{-3}$, as shown in Figure 8.3a, we observe that, using a fixed λ, a careful selection of α and β does help improve action recognition accuracy.

Table 8.2: Average precision (%) over the Hollywood-2 dataset of our MC-HCRF model in comparison to previous approaches, which shows our model's state-of-the-art results, especially on classifying sequential activities such as SitDown and StandUp. Previous techniques used for comparison include [Marszalek et al., 2009], [Han et al., 2009], [Derpanis et al., 2013], [Gilbert et al., 2009], [Ullah et al., 2010], [Wang et al., 2013a], [Chakraborty et al., 2012].

Activity	Marszalek	Han	Derpanis	Gilbert	Ullah	Wang	Chakraborty	**MC-HCRF**
AnswerPhone	13.10	15.57	22.00	40.20	26.30	32.60	**41.60**	34.38
DriveCar	81.00	87.01	83.00	75.00	86.50	88.00	**88.49**	86.65
Eat	30.60	50.93	54.00	51.50	59.20	**65.20**	56.50	58.72
FightPerson	62.50	73.08	72.00	77.10	76.20	81.40	78.20	**82.12**
GetOutCar	8.6	27.19	32.00	45.60	45.70	52.70	47.37	**53.14**
HandShake	19.10	17.17	16.00	28.90	49.70	29.60	**52.50**	50.25
HugPerson	17.00	27.22	37.00	49.40	45.40	54.20	50.30	**54.34**
Kiss	57.60	42.91	59.00	56.60	59.00	**65.80**	57.35	57.20
Run	55.50	66.94	76.00	47.50	72.00	**82.10**	76.73	75.25
SitDown	30.00	41.61	56.00	62.00	62.40	62.50	62.50	**64.77**
SitUp	17.80	7.19	18.00	26.80	27.50	20.00	30.00	**33.65**
StandUp	33.50	48.61	56.00	50.70	58.80	65.20	60.00	**67.67**
Overall	35.50	42.12	48.00	50.90	55.70	58.30	58.46	**59.84**

8.5.2 Results on Hollywood-2 Dataset

We conduct experiments over realistic movie videos using the Hollywood-2 dataset [Marszalek et al., 2009]. Following the experimental setups [Chakraborty et al., 2012, Derpanis et al., 2013, Gilbert et al., 2009, Marszalek et al., 2009, Ullah et al., 2010, Wang et al., 2013a], the dataset is divided into 823 training and 884 testing instances; performance is evaluated using the precision metric[†]. The same HOG/HOF/MBH features described in our KTH experiment is applied on this dataset.

Figure 8.1b depicts our MC-HCRF model's precision over the training set across different hyper-parameter values. The best cross-validation precision, $62.95 \pm 1.67\%$, is obtained when $\alpha = 0.25$, $\beta = 10$, and $\lambda = 10^{-6}$. Using these hyper-parameters, our MC-HCRF model achieves a 59.84% overall perception on the testing set.

[†] Since precision is used as the performance metric on the Hollywood-2 dataset, no confusion matrix is produced, as in [Chakraborty et al., 2012, Derpanis et al., 2013, Gilbert et al., 2009, Han et al., 2009, Marszalek et al., 2009, Ullah et al., 2010, Wang et al., 2013a].

As compared in Table 8.2, the MC-HCRF model performs better than other state-of-the-art approaches, on average. Most importantly, our MC-HCRF model greatly increases classification precision over sequential activities, including SitDown, StandUp, GetOutCar, etc. The precision improvements demonstrate the importance of modeling latent temporal patterns of sequential activities, and highlight our MC-HCRF model's superiority on recognizing sequential human activities, such as "sitting down" and "standing up".

Our model's sensitivity to α and β given $\lambda = 10^{-6}$ over the Hollywood-2 dataset is shown in Figure 8.3b. Similar to what we observe in the experiment using the KTH dataset, carefully selecting entropy hyper-parameters increases the MC-HCRF model's classification performance.

8.5.3 Results on CAD-60

We use the SMS features that are provided by the dataset, which represent human activities using 15 skeleton joints in 3D space. Following [Sung et al., 2012], we adopt the "have seen" experimental setting, except that we randomly select 70% of each subject's available data for hyper-parameter selection and training. Same as [Sung et al., 2012], the performance is reported using precision and recall. In addition, we use accuracy as our performance metric for hyper-parameter selection and model evaluation.

Table 8.3: Performance comparison of our MC-HCRF model with the state-of-the-art on the CAD-60 dataset.

Approach	Accuracy (%)	Precision (%)	Recall (%)
SVM [Sung et al., 2012]	—	66.4	56.0
One-layer MEMM [Sung et al., 2012]	—	65.8	58.1
Piyathilaka et al. [Piyathilaka and Kodagoda, 2013]	—	84.0	73.0
Sung et al. [Sung et al., 2012]	—	84.7	83.2
MC-HCRF	**93.3**	**88.2**	**87.8**

Figure 8.1c shows our MC-HCRF model's accuracy variations over the training set using different hyper-parameter values. Using $\lambda = 10^{-4}$, $\alpha = 0.25$ and $\beta = 10$, our model obtains the best accuracy of $95.2 \pm 1.77\%$ over the training set. Using the same set of hyper-parameters, our MC-HCRF model achieves a 93.3% accuracy on the testing set, with a precision / recall of 88.2% / 87.8%, which are the state-of-the-art results as compared in Table 8.3. The confusion matrix obtained by our model is depicted in Figure 8.2b. Following [Sung et al., 2012], the matrix is constructed by aggregating the results in different scenarios that are tested separately. This empirical study highlights our model's ability to work with traditional global skeleton features that have become more accessible with the emergence of color-depth cameras.

Figure 8.3c illustrates the variations of our model's classification accuracy across different hyper-parameters α and β, given $\lambda = 10^{-4}$. Each hyper-parameter setting

produces large variations in the cross-validation accuracy. This is attributed to a disproportionate number of instances per individual human actor, which leads to under-representation in the learning set and over-representation in the validation set. However, this has little impact on the final model, since the final training set combines the validation and learning sets.

8.5.4 Results on MSR Action3D Dataset

The MSR Action3D dataset [Wang et al., 2012b] contains 567 sequences of skeleton motions and depth images, which are grouped into 20 human activity classes. Each activity is performed by ten subjects two or three times. This dataset reasonably covers various motions to interact with game consoles. We split the instances from each activity category according to 70% training and 30% testing.

Following [Oreifej and Liu, 2013], the HON4D features[‡] are applied to represent human activities in depth videos, which generate a 120-dimensional histogram. We extract HON4D features from each frame in a depth video to construct a temporal sequence of such histograms, which serves as the input to our model. Figure 8.1d depicts our MC-HCRF model's accuracy over the training data using different hyper-parameter values; our model obtains the best cross-validation accuracy of $90.79 \pm 1.07\%$ when $\lambda = 10^{-2}$, $\alpha = 0.5$ and $\beta = 5$. Using these hyper-parameters, our MC-HCRF model achieves an accuracy of 89.29% over the testing set. Comparisons with previous approaches in Table 8.4 indicate that our model achieves the state-of-the-art result. The confusion matrix is shown in Figure 8.2c.

We illustrate in Figure 8.3d our MC-HCRF model's recognition accuracy variations over the training set across different entropy hyper-parameters α and β, given a fixed regularization hyper-parameter $\lambda = 10^{-2}$, which again shows that carefully selecting hyper-parameters improves our model's performance on human activity recognition.

To show our MC-HCRF model's capability of modeling sequential activities that are represented by SMS features, we conduct an additional experiment using the skeleton features provided with the dataset. Each skeleton pose contains 20 joint positions with four values per joint. After selecting the hyper-parameter values using the training set, we evaluate our model over the testing set and obtain an accuracy of 88.17%. As compared in Table 8.4, using SMS features, our model still achieves good accuracy that is comparable to the state-of-the-art, although it does not perform as well as our MC-HCRF model using HON4D features.

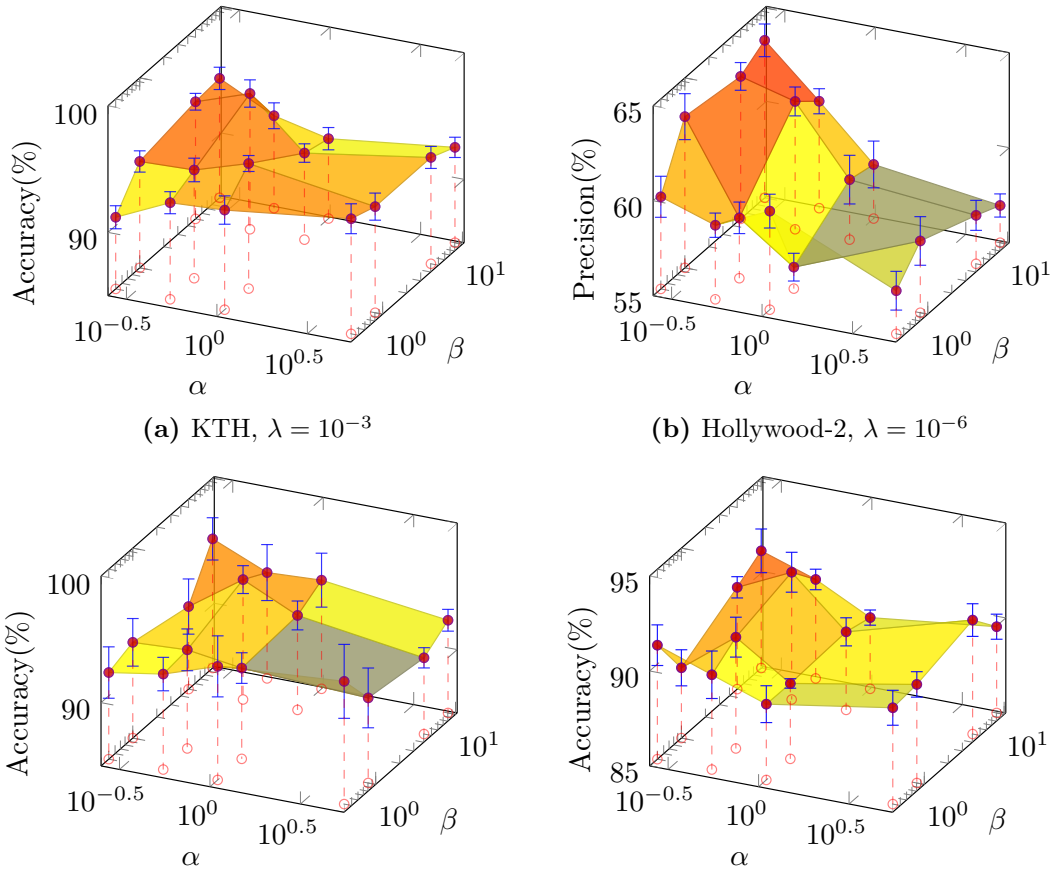
[‡]Code to extract HON4D features is available at: http://www.cs.ucf.edu/~oreifej/HON4D.html

Table 8.4: Performance comparison of our MC-HCRF model with the state-of-the-art on the MSR Action3D dataset.

Approach	Accuracy (%)
Vieira et al. [Vieira et al., 2012]	78.20
Yang et al. [Yang and Tian, 2012]	85.52
Wang et al. [Wang et al., 2012a]	86.50
Wang et al. [Wang et al., 2012b]	88.20
Oreifej et al. [Oreifej and Liu, 2013]	88.89
MC-HCRF (SMS)	88.17
MC-HCRF (HON4D)	**89.29**

8.6 Summary

We introduce the Maximum Certainty HCRF model to recognize sequential human activities. Our model aims to maximize both the probability of the correct class and the certainty in latent temporal patterns, which forms a multi-objective optimization problem. Through formulating our model in the energy-based learning framework, we introduce a new energy function to simultaneously incorporate both objectives. Inference is efficiently performed by maximizing the energy function, and learning is conducted using the NRBM method. Empirical studies on four real-world datasets show our MC-HCRF model achieves state-of-the-art performance for human activity recognition, especially on classifying sequential activities, which demonstrates the benefit of modeling certainty in latent temporal patterns and formulating HCRFs as energy-based models.



(a) KTH, $\lambda = 10^{-3}$

(b) Hollywood-2, $\lambda = 10^{-6}$

(c) CAD-60, $\lambda = 10^{-4}$

(d) MSR Action3D, $\lambda = 10^{-2}$

Figure 8.3: Classification performance across hyper-parameters α and β, given a fixed λ. The solid circles represent the cases we test in our experiments, with error bars indicating the classification errors obtained from cross-validation. The hollow circles are projections of our test cases onto the hyper-parameter plane to illustrate the hyper-parameter values that are used in corresponding test cases. The surface connecting the points is generated using interpolation for a clear representation.

Chapter 9

Recognition: FuzzySR Model for Continuous Action Understanding

9.1 Introduction

In this chapter, we address the problem of temporal segmentation and probabilistic recognition of continuous human activities in streaming visual data. The majority of previous studies [Aggarwal and Ryoo, 2011, Borges et al., 2013] in human activity recognition focus on classification of primitive activities contained in short, manually segmented clips, such as walking and hand-waving. However, in real-world scenarios, human activities always involve continuous, complicated temporal patterns, for example, grabbing a box then packing and delivering it; that is, human activities are always performed in a continuous fashion. Therefore, besides the capability of inferring human activities contained in the segmented events, the robots also need the crucial ability to identify the start and end time points of each activity, i.e., to partition continuous visual data into a sequence of activity events.

Not surprisingly, segmenting and recognizing a sequence of human activities from continuous, unsegmented visual data is considerably more challenging than the task of human activity recognition from a temporally partitioned event that contains a single human activity. Besides the well-known difficulties to categorize human activities in partitioned events, including variations of human appearances and movements, illumination changes and dynamic backgrounds, etc., recognizing human activities in continuous, unsegmented visual data introduces a few additional challenges. The biggest difficulty of continuous activity segmentation is to deal with the transition effect. Since transitions between temporally adjacent activities always occur gradually, their temporal boundaries are always vague and it is extremely challenging, even for people, to identify when one activity ends and another activity starts. In addition, generating ground truth to evaluate continuous human activity recognition systems is a challenging task [Hard et al., 2006]. Errors can arise due to the imprecise activity definition, clock synchronization issues, and the limited human reaction time [Minnen et al., 2006]. As a consequence, these challenges result

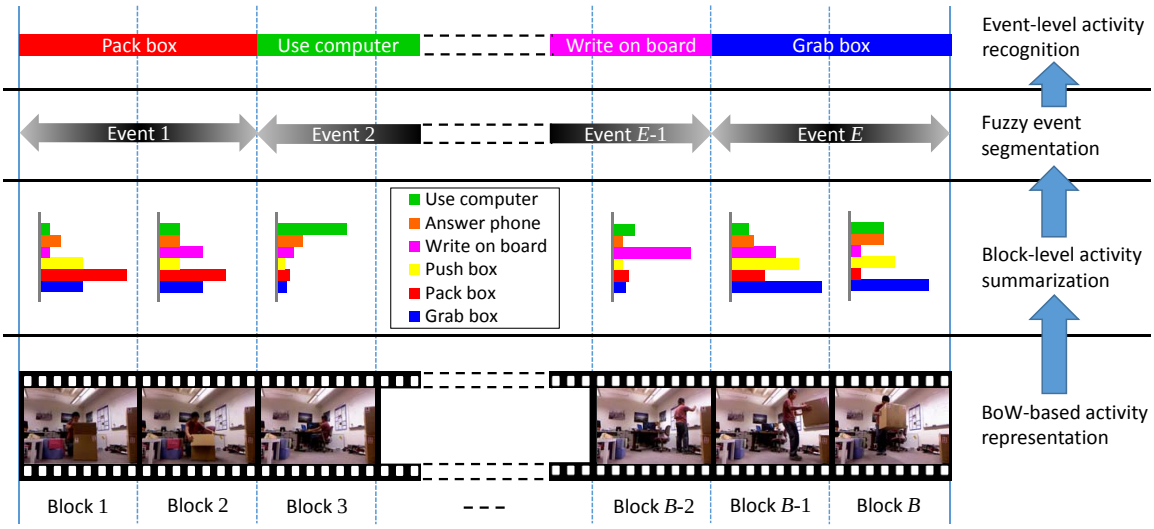


Figure 9.1: Illustration of our FuzzySR algorithm for continuous human activity segmentation and recognition. The block-level activity summarization module summarizes the activity distribution of each block by mapping high-dimensional discrete feature space to real-valued activity space. The fuzzy event segmentation module uses the activity summaries to form a multi-variable time series, and applies fuzzy temporal clustering to discover and segment events that are modeled as fuzzy sets. The event-level activity recognition module incorporates summaries of all blocks contained in an event to determine an activity label.

in significant difficulties in construction of a continuous activity segmentation and recognition system.

To address this important but challenging research problem, we introduce a novel algorithm, named *Fuzzy Segmentation and Recognition* (FuzzySR), to temporally partition continuous visual data into a sequence of coherent constituent segments in an unsupervised fashion and to recognize the human activity contained in each individual segment. The general idea of our new FuzzySR algorithm is graphically demonstrated in Figure 9.1, which contains three components: block-level activity summarization, fuzzy event segmentation, and event-level activity recognition. We advocate the use of unsupervised learning to address this problem, because it allows assistant robots, when deployed in a new human social environment, to discover new patterns of human activities and/or adapt to activity variations of different people. In addition, unsupervised learning opens the possibility to take advantage of the increasing amount of available data perceived by the robots, without the expense of human supervision and annotation [Niebles et al., 2008, Chua et al., 2011].

Our continuous human activity segmentation and recognition algorithm adopts the bag-of-words (BoW) representation based on local spatio-temporal features [Zhang and Parker, 2011] that are extracted from visual data. The BoW representation is a most popular model for human activity recognition due to its robustness in

real-world environments [Wang et al., 2009, Dollár et al., 2005, Zhang et al., 2012b]. Following the BoW representation, several approaches were proposed to construct a human activity recognition system. Although demonstrated to be effective to recognize primitive activities in segmented videos [Laptev et al., 2008, Dollár et al., 2005, Zhang et al., 2012b, Zhang and Parker, 2011], BoW models based on LST features ignore long-term temporal structures of the sequential data, which limits their applications on segmenting continuous visual data that can exhibit temporal patterns. In addition, since the BoW model represents videos as a histogram of visual words that are computed from local features, it takes discrete values generally in high dimensional space, which makes analysis directly using the BoW model very expensive and generally intractable [Blei et al., 2003]. Because of this high dimensionality, the BoW model generally cannot be directly used to form a time series to address the problem of temporal pattern analysis.

An important objective of this work is to bridge the divide between temporal human activity segmentation and the BoW representation based on LST features [Wang et al., 2009], which is not well studied in previous work. Our approach achieves this objective through applying the *block-level activity summarization*. A *block* is defined as a unit time interval of user-defined duration that contains a short sequence of consecutive video frames, in which the activity performed by a human subject is assumed consistent (i.e., the activities remain the same). As demonstrated in Figure 9.1, our block-level activity summarization partitions a continuous video into a sequence of non-overlapping blocks, and summarizes activity information of each block by mapping the high-dimensional discrete BoW representation in feature space to the real-valued distribution over activities in activity space. Then, the block-level activity distributions are used to form a multi-variable time series. It is noteworthy that the use of local spatio-temporal features also ensures that our FuzzySR algorithm captures the short-term temporal variation within each block.

Another important objective of this research is to discover and segment activity events from continuous visual data that can contain a sequence of human activities, and to infer an activity label for each individual event. An *event* is defined as a maximum continuous period of time during which the activity label is consistent. Through treating the block-level activity distribution as intermediate information to form a real-valued multi-variable time series, our FuzzySR algorithm follows a fuzzy temporal clustering method [Abonyi et al., 2005] to segment events. We use fuzzy sets to model events and employ fuzzy event boundaries to address gradual transition effects between continuous activities. This procedure is called *fuzzy event segmentation*, as illustrated in Figure 9.1. To determine the human activity category of a segmented event, we introduce a new, optimization-based approach that incorporates activity summaries of all blocks contained in the event to make the most appropriate decision. We name this procedure *event-level activity recognition*.

To validate the effectiveness of our FuzzySR algorithm, we conduct extensive empirical study using six different datasets. We first employ two simple and most commonly used human activity datasets (e.g., Weizmann and KTH) to demonstrate

how our FuzzySR algorithm segments continuous visual data into events and interprets the human activity in each individual event. Then, we focus on evaluating our FuzzySR algorithm's performance on continuous human activity segmentation and recognition in human social environments, using color-depth cameras (e.g., Kinect and PrimeSense cameras) that have currently become standard devices to construct 3D perception systems on autonomous robots. Experimental results on all used datasets shows promising performance of our FuzzySR algorithm for continuous activity understanding.

Our main contributions are summarized as follows:

- We propose to use a temporal fuzzy clustering algorithm to explicitly model the gradual transition between temporally adjacent human activities, and thereby provide a novel solution to segment continuous visual data into a sequence of events.

- We introduce a new framework that uses the block-level human activity summarization to form a low-dimensional time series, and thereby provide a new solution to bridge the divide between the continuous activity understanding problem and the high-dimensional activity representation using LST features.

It is noteworthy that the focus of this work is to investigate temporal characteristics of continuous human activities, using temporal fuzzy clustering and unsupervised probabilistic recognition. In addition, we would like to mention that our method is a general framework that can work with both color videos and RGB-D visual data.

The remainder of the chapter is structured as follows. After reviewing related work in Section 9.2, we discuss our fuzzy continuous human activity segmentation and recognition method in Section 9.3. Then, experimental results are presented in Section 9.4. Finally, we summarize this work in Section 9.5.

9.2 Related Work

Segmentation and recognition of continuous activities from visual data is a research topic that involves several techniques, including human activity recognition from video segments and activity event partition from continuous visual data. Previous approaches related to these techniques are reviewed in detail in this section.

9.2.1 Human Activity Modeling

A large number of previous studies in human activity understanding have focused on the problem of recognizing repetitive or punctual activities from short, manually partitioned visual data, which can be acquired from color or color-depth cameras. Instead of discussing the supervised learning approaches used to classify human activities at the model level, such as Hidden Markov Models (HMMs) [Turaga et al., 2008] and

Conditional Random Fields (CRFs) [Lafferty et al., 2001], we focus our review on encoding spatio-temporal information at the feature level to distinguish temporal patterns of human activities.

A most popular space-time representation of human activities is to apply centroid trajectories to encode human location variations in visual data. This methodology, as in [Chowdhury and Chellappa, 2003b], encodes a human as a single point, which represents human locations in spatial dimensions. However, this trajectory-based human representation is only applicable in the situations when people occupy a small region in an image. Another most widely used human activity representation is based on articulated human body models, such as the skeleton model [Ben-Arie et al., 2002, Shotton et al., 2011, Luo et al., 2013]. The third category of space-time representations employ a sequence of human shapes, such as human contours and silhouettes [Singh et al., 2008, Freifeld et al., 2010], to model temporal patterns of human activities. Despite the satisfactory recognition performance of the methods based on body models and human shapes, they generally depend on human localization and body-part tracking, which are hard-to-solve problems due to the challenges such as camera motion, occlusion, dynamic background, etc.

Different from the aforementioned global human representations, local spatio-temporal features have recently attracted an increasing attention, due to the robustness to partial occlusion, slight illumination variation, and image rotation, scaling and translation [Lowe, 2004, Wang et al., 2009]. Furthermore, because LST features are directly computed from raw visual data, they can avoid potential failures of preprocessing steps such as human localization and tracking. Dollar et al.[Dollár et al., 2005] detected LST features using separable filters in both spatial and temporal dimensions and described the features using a concatenation-based approach. Laptev et al. [Laptev, 2005] detected LST features based on generalized Harris corner detectors, and described these features using a histogram-based method. Other approaches were also implemented, based on the extended Hessian saliency measure [Willems et al., 2008] and salient region detectors [Oikonomopoulos et al., 2005], to extract LST features from color videos. With the emergence of color-depth cameras, features that are able to incorporate both depth and color information have attracted an increasing attention. A recent study [Zhang and Parker, 2011] introduced the LST feature in 4-dimensional (i.e., $xyzt$) space, which is able to encode both color and depth cues in RGB-D visual data. Inspired by this work, Xia et al. [Xia and Aggarwal, 2013] implemented a feature descriptor based on cuboid similarity to increase the feature's discriminative power to recognize human activities from depth images.

In this work, we follow this local representation based on LST features to encode human activities. Different from previous studies that generally focused on single human activity recognition in segmented videos, we address the task of continuous human activity segmentation and recognition in unsegmented sequences. A direct application of LST features to form a time series generally make the segmentation problem intractable, because the raw LST features can contain a large number of

elements in high-dimensional space. We bridge the divide between the continuous activity segmentation problem and the local human representation using LST features through introducing a new layer (i.e., block-level activity summarization) that projects the high-dimensional feature space to the low-dimensional activity space.

9.2.2 Temporal Activity Segmentation

Automatic segmentation of complex, continuous activities is important, since when intelligent robots are deployed in human social environments in real-world applications, they always receive continuous visual data from their onboard perception systems. Without the capability of segmenting the continuous visual data into a temporal sequence of individual activities, it is impossible for the robots to understand human behaviors and effectively interact with people.

In recent years, a large number of studies have been conducted to address the continuous activity segmentation task. Previous approaches to address this problem can be generally grouped into three categories, which are based on heuristics, optimization, and change point detection, respectively.

The first methodology applies simple heuristics to segment human activities from continuous visual data. Fanello et al. [Fanello et al., 2013] first calculated a Support Vector Machine (SVM) score from each frame, and then selected the local minima of the score's standard deviation as break points to define the end of a human activity and the start of another activity. Another similar approach introduced by Kozina et al. [Simon Kozina, 2011] defined the break points as both local maxima and minima of a given time series. These heuristic segmentation approaches are very sensitive to noise in the time series. When a time series contains multiple variables that usually have a significant amount of noise (as shown by the example in Figure 9.4b), the heuristic approaches always over-segment the given continuous visual data, i.e., each activity event is always incorrectly partitioned into a large number of small pieces that may have inconsistent human activity labels.

Another widely applied framework uses optimization, typically based on discriminative learning, to segment continuous human activities. Shi et al. [Shi et al., 2008] addressed the activity segmentation task using a SVM-HMM model, which is formulated as a regularized optimization problem. A similar approach was introduced by Hoai et al. [Hoai et al., 2011] to jointly segment and classify continuous human activities, which is based on the multi-label SVM-based classification and the discriminative optimization. The third category of approaches to partition continuous human activities are based on change point detection, which has a long history in statistics and machine learning. The earliest and best-known technique is the CUSUM detector [Page, 1954], which represents a time series as piecewise segments of Gaussian means with noise. More recently, change point detection has drawn increasing attention to process visual data. For example, Zhai et al. [Zhai and Shah, 2005] proposed to apply change point detection to segment video scenes, using heuristic features that are manually defined. Change point detection was also used by

Ranganathan [Ranganathan, 2012] to perform place classification, using local features such as dense Scale-Invariant Feature Transform (SIFT). Given the satisfactory performance of the methods based on optimization or change point detection, they typically assume fixed boundaries of each activity event, and thus are incapable of modeling gradual transitions between continuous activities in real-world situations.

Different from previous continuous human activity segmentation methods that assume fixed event boundaries [Shi et al., 2008, Hoai et al., 2011, Zhai and Shah, 2005, Ranganathan, 2012, Zhou et al., 2013], our objective is to explicitly model gradual transitions between temporally adjacent activities. We propose to apply temporal clustering [Warren Liao, 2005] to achieve this objective, which encodes each activity event as a fuzzy set with non-fixed boundaries, instead of segmenting visual data into disjoint events. In addition, the time series used in our algorithm is formulated in a new way by concatenating block-level human activity distributions.

There exist two research problems that look similar but are completely different from temporal fuzzy segmentation. The first problem is *fuzzy recognition*, which applies fuzzy method to recognize activities, i.e., to assign an activity category to a data instance. For example, Banerjee et al. [Banerjee et al., 2014] applied the Gustafson-Kessel clustering to recognize activities of daily living. The second problem is the *background-foreground segmentation*, which aims at localizing humans in the scene and spatially segmenting people from background. For example, Anderson et al. [Anderson et al., 2010] used genetic algorithms to segment people and objects out of 3D scenes. Different from the problems of fuzzy recognition and background-foreground segmentation, temporal fuzzy segmentation, which is the focus of our work, aims at partitioning continuous visual data into events along time dimension using temporal fuzzy clustering.

9.3 FuzzySR Algorithm

We describe our FuzzySR algorithm for fuzzy continuous human activity segmentation and recognition in this section. FuzzySR provides a general framework to identify complex, continuous human activities from unsegmented visual data with gradual transitions between temporally adjacent human activities. Furthermore, our FuzzySR algorithm bridges the gap between the BoW model based on LST features and the continuous human activity segmentation task. The general idea of our algorithm is illustrated in Figure 9.1. Major notations used in this work are summarized in Table 9.1 for quick reference. In addition, in Algorithm 4 and Algorithm 5, we present two algorithms to describe how our algorithm is learned during the offline learning phase in an unsupervised fashion and how it is used during the online testing phase. Details of the procedures used in our approach are presented in the following subsections.

Algorithm 4: Offline unsupervised learning of FuzzySR

Input : K (number of activity clusters),
D (dictionary size),
$\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_B\}$ (a set of blocks)
Output: $\mathcal{M}$ (learned LDA model), $\boldsymbol{D}$ (dictionary)

1: Extract LST visual features from each block;
2: Apply k-means method to cluster features into D groups;
3: Encode each feature using its cluster index (i.e., visual word);
4: Construct dictionary $\boldsymbol{D}$ that contains all visual words;
5: Represent each block as a BoW model;
6: Learn the LDA model $\mathcal{M}$ using the BoW representation (given K) and compute block-level activity distribution;
7: **if** *block labels are available* **then**
8: | Perform semantic mapping using Hungarian method;
9: **end**
10: **return** $\boldsymbol{D}$ and $\mathcal{M}$

9.3.1 Block-Level Activity Summarization

The goal of block-level activity summarization is to reduce the input dimensionality in order to form a manageable time series, which is achieved by projecting the high-dimensional feature space to a low-dimensional activity distribution space. In Figure 9.2, we provide an intuitive illustration to overview the block-level activity summarization. Our approach is based on LST features (e.g., HOG features for color videos and 4D-LST features for RGB-D data, as specified in Section 9.4). To construct the dictionary, in the training phase, our approach uses the k-means algorithm to group the LST features (each is a vector containing real values) extracted from training blocks into a given number of clusters. Then each feature vector is encoded by the discrete index of the cluster (usually referred to as a dictionary word). The dictionary is defined as the collection of all the cluster indices. Given this dictionary, each block can be encoded by a bag-of-words representation, as illustrated in Figure 9.2.

Input to our FuzzySR algorithm is an unsegmented video with each frame encoded using the BoW representation based on LST features. This input video $\mathcal{W}$ is temporally partitioned into a sequence of disjoint blocks that have equal length: $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_B\}$, where B is the number of blocks. Each block $\boldsymbol{w}_j$, $j = 1 \ldots B$, is a short sequence of frames. Given a dictionary $\boldsymbol{D}$, which typically has a high dimensionality, each block is represented as a set of discrete visual words that are computed from the LST features using $\boldsymbol{D}$.

Our algorithm applies a statistical topic model, i.e., Latent Dirichlet Allocation (LDA) [Blei et al., 2003], to summarize human activity information that is contained in each block. Given a block $\boldsymbol{w}$, the LDA model represents each of K activities as

Algorithm 5: FuzzySR for online testing

Input : $\mathcal{W}$ (unsegmented visual data),
$\mathcal{M}$ (learned LDA model), $\boldsymbol{D}$ (dictionary)
Output: β (block fuzzy membership),
z (event activity category)

1: Represent $\mathcal{W}$ as a sequence of blocks;
2: Encode each block as a BoW model, given $\boldsymbol{D}$;
3: Apply $\mathcal{M}$ on each block to learn activity distribution $\boldsymbol{\theta}$;
4: Form a multivariate time series using $\boldsymbol{\theta}$ from all blocks;
5: Compute fuzzy membership β for each block according to Eq. (9.4);
6: Temporally segment $\mathcal{W}$ into a sequence of events using Eq. (9.3);
7: Compute event-level activity assignment z according to Eq. (9.11)
8: **return** β and z

the multinomial distribution of all possible visual words in the dictionary $\boldsymbol{D}$. This distribution is parameterized by $\boldsymbol{\varphi} = \{\varphi_{w_1}, \ldots, \varphi_{w_{|\boldsymbol{D}|}}\}$, where φ_w is the probability that the word w is generated by the activity. LDA also models each block $\boldsymbol{w} \subset \mathcal{W}$ as a collection of the visual words, and assumes that each word $w \in \boldsymbol{w}$ is associated with a latent activity assignment z_w. By using the visual words to associate blocks with activities, LDA models a block $\boldsymbol{w}$ as the multinomial distribution over the activities, which is parameterized by $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}$, where θ_k is the probability that $\boldsymbol{w}$ is generated by the kth activity. The LDA model is a Bayesian model, which places Dirichlet priors on the multinomial parameters: $\boldsymbol{\varphi} \sim \mathrm{Dir}(\boldsymbol{\beta})$ and $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, where $\boldsymbol{\beta} = \{\beta_{w_1}, \ldots, \beta_{w_{|\boldsymbol{D}|}}\}$ and $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$ are the concentration hyperparameters.

The objective in block-level activity summarization is to estimate $\boldsymbol{\theta}$, i.e., the per-block activity distribution. However, exact parameter estimation is generally intractable [Blei et al., 2003]. Gibbs sampling is a widely used technique to approximately estimate LDA's parameters, which is able to asymptotically approach the correct distribution [Porteous et al., 2008]. When Gibbs sampling converges, the probability of each human activity $\theta_k \in \boldsymbol{\theta}$, $k = 1, \ldots, K$, can be estimated by:

$$\theta_k = \frac{n_k + \alpha_k}{\sum_i (n_i + \alpha_i)}, \tag{9.1}$$

where n_k is the number of times that a word is assigned to the activity $z_w = k$ in the block.

After the per-block activity information is summarized for all blocks within the video, a real-valued multi-variable time-series can be formed: $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_B\}$, which contains B time-ordered summaries computed at time points $t_1, \ldots, t_B$, where $\boldsymbol{\theta}_j = \{\theta_{j,1}, \ldots, \theta_{j,K}\}^\top$, $j = 1, \ldots, B$, summarizes the activity information contained in the jth block at time t_j.

Table 9.1: Major notations used in our FuzzySR algorithm

Variable	Notation
$\mathcal{W}$	Input unsegmented visual data
$\mathcal{M}$	Learned LDA model
$\boldsymbol{w}$	Block (i.e., a short sequence of frames)
$\boldsymbol{\theta}$	Per-block activity distribution (Eq. (9.1))
$\boldsymbol{\Theta}$	Time series of $\boldsymbol{\theta}$
$\boldsymbol{l}$	Labels of training blocks
$\boldsymbol{c}$	Activity clusters
$\boldsymbol{D}$	Dictionary
$\boldsymbol{e}(t_s, t_e)$	Event that starts at t_s and ends at t_e
$A_i(t_j)$	Gaussian membership of $\boldsymbol{w}_j$ in $\boldsymbol{e}_i$ (Eq. (9.5))
$\beta_i(t_j)$	Fuzzy membership of $\boldsymbol{w}_j$ in $\boldsymbol{e}_i$ (Eq. (9.4))
y_j	Fixed membership of $\boldsymbol{w}_j$ (Eq. (9.10))
z_i	Activity label of $\boldsymbol{e}_i$ (Eq. (9.11))
B	Number of blocks in $\mathcal{W}$
E	Number of events in $\mathcal{W}$
K	Number of activity categories
D	Dictionary size
i, j, k	Index of event, block, and activity, respectively

It is noteworthy that since no ground truth labels are used in the learning process, our per-block activity summarization is performed in a complete unsupervised fashion, which has the potential to enable autonomous robotic systems to discover and adapt to unseen human activities in new environments.

On the other hand, when semantics (i.e., known activity labels) are available for a subset of blocks (e.g., ground truth of training blocks), the semantics $\boldsymbol{l}$ can be associated with the resulting clusters $\boldsymbol{c}$ that are obtained by the unsupervised LDA model. To address this semantic mapping problem, the Hungarian method [Kuhn, 1955] is used, which finds a bijective (i.e., one-to-one and onto) function $f : \boldsymbol{c} \rightarrow \boldsymbol{l}$ through solving the following optimization problem:

$$f^{\star} = \underset{f:\, \boldsymbol{c} \rightarrow \boldsymbol{l}}{\arg\max} \sum_{i=1}^{N} \mathbb{1}(\pi_i^l = f(\pi_i^c)). \tag{9.2}$$

The Hungarian approach formulates this optimization problem as a bipartite graph matching problem. The graph consists of two sets of nodes, corresponding to the recognized clusters and the semantic labels, and edge weights are defined as the number of matches. Please refer to [R.E. Burkard, 2012, Kuhn, 1955] for more details.

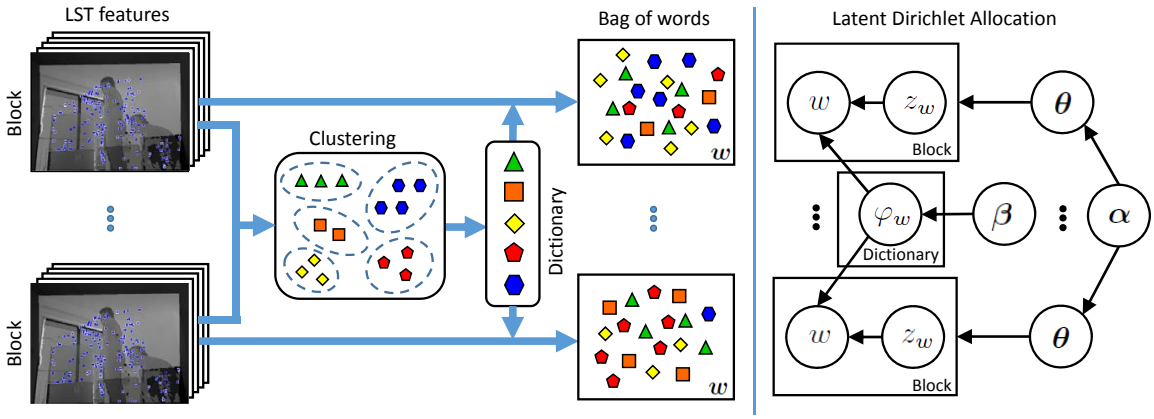


Figure 9.2: Illustration of block-level activity summarization. After the input continuous visual data is segmented into a sequence of disjoint blocks, the LST features extracted from the frames in each block are converted into discrete visual words using a dictionary. Then, each block is represented by a bag of words $\boldsymbol{w}$, which serves as the input to LDA. The LDA model is used to compute the human activity distribution $\boldsymbol{\theta}$ for each $\boldsymbol{w}$. Typically, $\boldsymbol{\theta}$ has a manageable dimensionality that is much lower than the dimensionality of $\boldsymbol{w}$. The human activity summaries from all blocks are applied to form a time series for continuous activity segmentation.

It is also noteworthy that, although our discussion is based on the benchmark LDA model, other sophisticated topic models are directly applicable in our framework.

9.3.2 Fuzzy Event Discovery and Segmentation

Given a time series of the block-level activity summaries, the task of continuous human activity segmentation is to seek a sequence of events $\boldsymbol{e}(t_{i-1}, t_i)$, $i = 1, \ldots, E$, where t_i is the temporal boundary of an event that satisfies $t_0 < t_1 < \ldots, < t_E$, and E is the number of events to segment. The segmentation task can be formulated as an optimization problem. Following [Abonyi et al., 2005], the optimal event boundaries can be determined through minimizing the sum of the individual event's cost:

$$cost(\boldsymbol{\Theta}) = \sum_{i=1}^{E} \boldsymbol{e}(t_{i-1}, t_i) = \sum_{i=1}^{E} \sum_{j=1}^{B} \beta_i(t_j) \cdot \mathrm{dis}_e(\boldsymbol{\theta}_j, \boldsymbol{v}_i^{\theta}), \tag{9.3}$$

where $\mathrm{dis}_e(\boldsymbol{\theta}_j, \boldsymbol{v}_i^{\theta})$ denotes the distance between the jth block summary $\boldsymbol{\theta}_j$ and the mean $\boldsymbol{v}_i^{\theta}$ of $\boldsymbol{\theta}$ in the ith event (i.e., center of the ith cluster), and $\beta_i(t_j)$ denotes the membership of the jth block in the ith event. In previous studies [Shi et al., 2008, Hoai et al., 2011, Page, 1954, Zhai and Shah, 2005, Zhou et al., 2013], a hard membership is typically used, which satisfies $\beta_i(t_j) = \mathbb{1}(t_i < t_j \leq t_{i+1})$, where $\mathbb{1}(\cdot)$ is the indicator function.

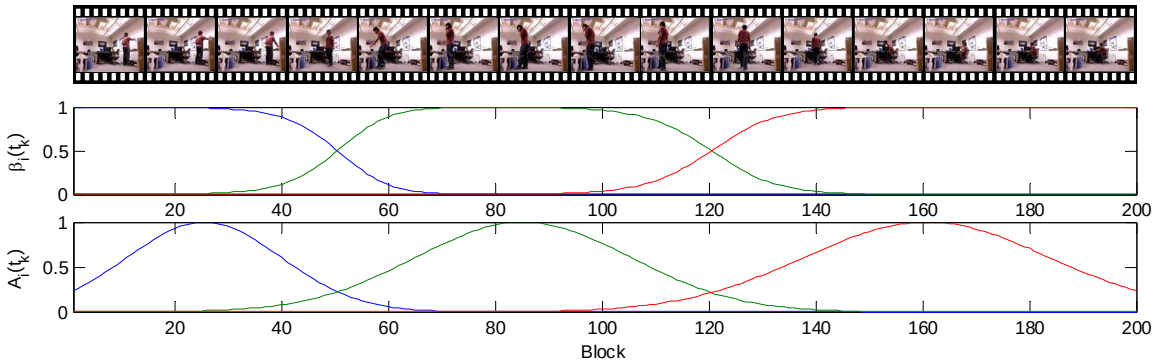


Figure 9.3: Illustration of modeling events using fuzzy sets that have fuzzy (not fixed) boundaries. A gradual transition always exists between continuous human activities in real-world scenarios. In this example, there exists a transition (block 40–60) between two adjacent activities; another transition (block 105–135) occurs later. Through solving the optimization problem in Eq. (9.6), we can obtain the fuzzy segmentation results, which are encoded by the fuzzy membership $\beta(t)$ that is computed using the Gaussian membership function $A(t)$.

However, transitions between temporally consecutive human activities are usually vague in the real-world scenario. Consequently, changes of the time series that is formed by the block summaries do not suddenly occur at any particular time point. Therefore, it is not practical to define hard boundaries of the events and not appropriate to model gradual activity transitions using hard memberships.

To address the gradual transition issue, instead of defining hard event boundaries, we represent each human activity event as a fuzzy set with fuzzy (not fixed) boundaries, and assign the jth block $\boldsymbol{w}_j$ with a fuzzy membership $\beta_i(t_j) \in [0, 1]$ to the ith event $\boldsymbol{e}_i$, as follows:

$$\beta_i(t_j) = \frac{A_i(t_j)}{\sum_{k=1}^{B} A_k(t_j)}, \tag{9.4}$$

where $A_i(t_j)$ is the Gaussian membership function that is computed by:

$$A_i(t_j) = \exp\left(-\frac{(t_j - v_i^t)}{2 \cdot (\sigma_i^t)^2}\right), \tag{9.5}$$

where v_i^t and $(\sigma_i^t)^2$ are the mean and variance of the ith block in time dimension, respectively. Figure 9.3 illustrates our idea of modeling events using fuzzy sets with fuzzy boundaries, which also visualizes the fuzzy segmentation results.

To divide a time series of block-level human activity summaries into a sequence of events with fuzzy boundaries, we need to estimate the parameters v^t and $(\sigma^t)^2$. In this work, a modified Gath-Geva (GG) clustering approach [Gath and Gev, 1989, Abonyi et al., 2005] is used to achieve this objective. Through adding time as a

variable to each block-level activity summary, i.e., $\boldsymbol{x}=[t,\boldsymbol{\theta}]$, the GG approach favors continuous clusters in time. Assuming that $\boldsymbol{x}$ conforms to the Gaussian distribution, our optimization problem can be defined as follows:

$$\begin{array}{lll} \underset{\boldsymbol{\eta}_i:i=1,\ldots,E}{\text{minimize}} & \sum_{i=1}^{E}\sum_{j=1}^{B}\mu_{i,j}^{m}\,\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_i) & \\ \text{subject to} & \sum_{i=1}^{E}\mu_{i,j}=1 & \forall j \\ & 0\leq\mu_{i,j}\leq 1 & \forall i,j \end{array} \tag{9.6}$$

where $\mu_{i,j} \in [0,1]$ denotes the membership degree of $\boldsymbol{x}_j$ to the ith cluster parameterized by $\boldsymbol{\eta}_i$, which is computed by:

$$\mu_{i,j}=\frac{1}{\sum_{k=1}^{E}\left(\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_i)/\,\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_k)\right)^{-(m-1)}}, \tag{9.7}$$

and $m \in (1,\infty)$ denotes the weighting exponent that encodes the fuzziness of the resulting clusters. A common choice of the weighting exponent [Gath and Gev, 1989, Abonyi et al., 2005] is $m=2$. This value will be used throughout this work.

The distance function $\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_i)$ used in Eq. (9.6) is defined inversely proportional to the probability that $\boldsymbol{x}_j$ belongs to the ith cluster that is parameterized by $\boldsymbol{\eta}_i$. Since the time variable t is independent of the block summary $\boldsymbol{\theta}$, $\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_i)$ can be factorized as follows:

$$\text{dis}(\boldsymbol{x}_j,\boldsymbol{\eta}_i)=\frac{1}{p(\boldsymbol{x}_j,\boldsymbol{\eta}_i)}=\frac{1}{\alpha_i p(t_j|v_i^t,(\sigma_i^t)^2)p(\boldsymbol{\theta}_j|\boldsymbol{v}_i^\theta,\boldsymbol{\Sigma}_i^\theta)}, \tag{9.8}$$

where $\alpha_i=p(\boldsymbol{\eta}_i)$ is the prior probability of the ith cluster, which satisfies $\sum_{i=1}^{E}\alpha_i=1$, and t_j and $\boldsymbol{\theta}_j$ in the jth block conform to the Gaussian distribution:

$$\begin{aligned} p(t_j|v_i^t,(\sigma_i^t)^2)&=\mathcal{N}(t_j|v_i^t,(\sigma_i^t)^2)\\ p(\boldsymbol{\theta}_j|\boldsymbol{v}_i^\theta,\boldsymbol{\Sigma}_i^\theta)&=\mathcal{N}(\boldsymbol{\theta}_j;\boldsymbol{v}_i^\theta,\boldsymbol{\Sigma}_i^\theta). \end{aligned}$$

In order to estimate the parameter of each cluster, that is $\boldsymbol{\eta}_i=\{\alpha_i,v_i^t,(\sigma_i^t)^2,\boldsymbol{v}_i^\theta,\boldsymbol{\Sigma}_i^\theta\}$, $i=1,\ldots,E$, the Expectation-Maximization approach is applied to solve the optimization problem in Eq. (9.6), resulting in the following model parameters along time dimension:

$$v_i^t=\frac{\sum_{j=1}^{B}\mu_{i,j}^m t_j}{\sum_{j=1}^{B}\mu_{i,j}^m},\quad (\sigma_i^t)^2=\frac{\sum_{j=1}^{B}\mu_{i,j}^m(t_j-v_i^t)^2}{\sum_{j=1}^{B}\mu_{i,j}^m}, \tag{9.9}$$

which can be used to compute the fuzzy membership $\beta_i(t_j)$ of the jth block $\boldsymbol{w}_j$ in the ith event $\boldsymbol{e}_i$, as defined in Eq. (9.4). As illustrated in Figure 9.3, $\beta_i(t_j)$ provides a fuzzy segmentation of the continuous visual data. Intuitively, $\beta_i(t_j)$ can be viewed as the probability that a block belongs to an event: at the gradual transition, the

probability of the old activity event decreases, and the probability of the new activity event increases.

9.3.3 Event-Level Activity Recognition

In this work, the continuous input visual data is uniformly divided into, as well as represented by, a sequence of disjoint blocks. Accordingly, an event can be defined as a maximum sequence of temporally distinct, contiguous blocks that have specific start time, end time, and a consistent human activity label. The objective of event-level activity recognition in our FuzzySR algorithm is to determine these parameters for each event that contains a consistent activity.

To determine the start time and end time of an activity event, which define the boundaries of an activity, the general computational principle "winner-take-all" is adopted to represent segmentation results corresponding to the fuzzy membership. Mathematically, given the fuzzy membership of the jth block, denoted by $\boldsymbol{\beta}_j = [\beta_i(t_j)]$, $i = 1, \ldots, E$, its corresponding hard segmentation result y_j can be computed as follows:

$$y_j = \underset{i=1,\ldots,E}{\arg\max}\, \beta_i(t_j) \tag{9.10}$$

After the hard segmentation result y_j is obtained for each block $\boldsymbol{w}_j$, the human activity label of an event is determined using summaries of all blocks that are contained in the event. Mathematically, given the sequence of block summaries $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_B\}$ and the segmentation results $\boldsymbol{y} = \{y_1, \ldots, y_B\}$, for each event $\boldsymbol{e}_i$, $i = 1, \ldots, E$, the activity category z_i can be determined by solving the following optimization problem:

$$z_i = \underset{k=1,\ldots,K}{\arg\max}\ \frac{1}{B} \cdot \sum_{j=1}^{B} \left(\mathbb{1}(y_j = i) \cdot \log \frac{\theta_{j,k}}{\sum_{s=1}^{K} \theta_{j,s}} \right). \tag{9.11}$$

By computing the probability that the jth block belongs to the kth activity, i.e., $\theta_{j,k} / \sum_{s=1}^{K} \theta_{j,s}$, our algorithm considers the importance of each block under a probabilistic framework to decide the final activity label of an event. In our case, since topic modeling is applied to summarize each block's activity information, $\sum_{s=1}^{K} \theta_{j,s} = 1$, $\forall j$ is satisfied. It is noteworthy that the proposed probabilistic framework has great potentials to recognize multiple concurrent human activities: when a activity probability threshold is used, activities whose probability are greater than the threshold can be retained (instead of using max function to select a single activity, as in Eq. (9.11)).

9.4 Empirical Studies

In this section, we evaluate our FuzzySR algorithm's performance on segmenting and recognizing continuous human activities. Six real-world human activity datasets are adopted in our experiments. We evaluate our FuzzySR algorithm on the widely used KTH and Weizmann activity datasets and the challenging Hollywood-2 dataset to demonstrate how to apply our algorithm to perform continuous activity segmentation and recognition. Then, we focus on applying RGB-D data to understand human activities that are performed by people in human social environments. Specifically, we adopt the CAD-60 and ACT4^2 datasets to evaluate our algorithm's performance on understanding human activities in typical home environments, and apply the UTK-CAP dataset to evaluate our algorithm in standard office environments. The CAD-60, ACT4^2, and UTK-CAP datasets are collected from RGB-D cameras. Finally, we investigate the performance sensitivity of our algorithm to its parameters, including block size and dictionary size. It is noteworthy that we intentionally choose the benchmark LST features (i.e., HOG features for color videos and 4D-LST features for RGB-D data) to emphasize the performance gain resulting specifically from our temporal fuzzy segmentation and probabilistic recognition approach. In addition, this feature choice allows for fair comparisons with previous works.

9.4.1 Results on KTH Dataset

Since the KTH dataset only contains manually segmented, single-activity videos, to evaluate our FuzzySR's performance on continuous human activity segmentation and recognition, we generate blocks from existing videos in the dataset, and then concatenate these blocks into long videos that contain continuous human activities, following [Hoai et al., 2011]. Specifically, we generate 500 blocks, each of which has a duration of five seconds and contains 75 frames. We apply 100 blocks (around 12–18 blocks for each activity) to construct an LDA model for block-level summarization, and the remaining 400 blocks for testing using the learned LDA model. In addition, since ground truth (i.e., activity label of each data instance) is available for KTH the dataset, we apply the Hungarian method to associate semantics, as discussed in Section 9.3.1.

Following [Laptev, 2005], we extract low-level LST features through detecting space-time interest points and describing them using histogram of oriented gradients (HOG). Features belonging to the same block are combined together. Then, a dictionary of local spatio-temporal words with 400 clusters are constructed using the k-means algorithm. Using this dictionary, features in each block can be converted to visual words. Accordingly, each block is represented by the BoW model, which serves as the input to our FuzzySR algorithm.

Experimental results over the KTH dataset are graphically presented in Figure 9.4. The time series of the block-level activity summarizations is illustrated in Figure 9.4a, which is obtained by applying the learned LDA model on the blocks in the test

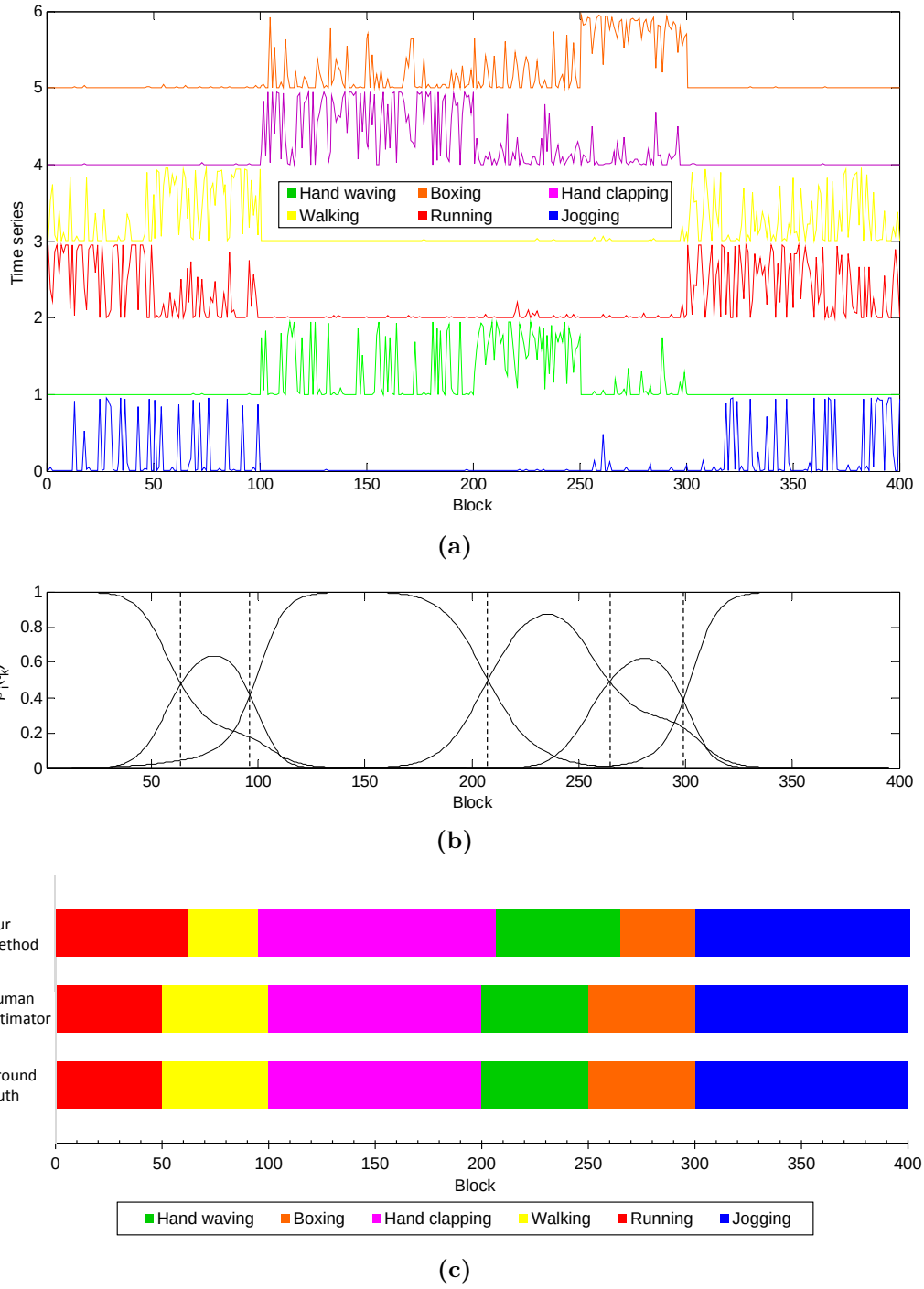


Figure 9.4: Experimental results of segmentation and recognition of continuous activities from the KTH dataset. The test video contains six events with instant transitions between human activities. (a) Time series of block-level activity summarizations. (b) Fuzzy segmentation (encoded by the fuzzy membership score $\beta(t)$). (c) Event-level activity recognition results and comparisons with ground truth and results provided by human estimators.

video. This observation demonstrates that the LDA model is capable of summarizing block-level activity information. In addition, it can be observed that activities with upper body motions (e.g., boxing, waving, and hand clapping) are easily confused with each other. Similarly, activities with lower body motions (e.g., walking, jogging, and running) are confused with each other. Especially, jogging and running are not well separated, because these two activities are extremely similar.

Based on the time series of block-level activity summarizations, the fuzzy segmentation result obtained by our method over the KTH dataset is graphically presented in Figure 9.4b. It can be observed that each activity event is encoded by a fuzzy set that has fuzzy boundaries. When a current activity is going to transfer to a new activity, the fuzzy membership score $\beta(t)$ of the current activity event decreases and simultaneously the new event's score increases. In addition, we observe that each activity event obtains its maximum fuzzy membership score at the center of a segment in time dimension, and an activity with a longer event duration generally obtains a more confident segmentation result with a greater fuzzy membership score. These observations indicate our algorithm's effectiveness to model activity transitions and segment continuous activities.

The event-level continuous activity recognition result that is obtained by our algorithm over the KTH dataset is shown in Figure 9.4c. In addition, our algorithm's performance is compared with ground truth and results that are manually estimated by human estimators, which are illustrated in Figure 9.4c. It can be observed that our FuzzySR algorithm well estimates the start and end time points of the events in the test video, and the activity contained in each event is correctly recognized. When the concatenated video is presented to human estimators, due to the clear, instant transitions between temporally adjacent activities, human estimators can perfectly identify the events and correctly recognize the activities contained in each event, as presented in Figure 9.4c.

9.4.2 Results on Weizmann Dataset

Similar to our previous experimental settings, we generate 227 blocks using the existing video clips contained in the Weizmann dataset. Each block has a duration of one second and contains 25 frames. Among the 227 blocks, we generate a test video through concatenating 100 blocks, which contains all ten activities. The test video contains twelve events and each event contains at least five blocks. The remaining blocks are employed to train the LDA model to summarize activity information in each block. We represent each block as a bag of visual words, which are computed by quantizing the HOG features [Laptev, 2005] extracted from the block using a dictionary of size 400.

Experimental results over the Weizmann dataset are graphically presented in Figure 9.5. It can be observed from this figure that our FuzzySR is very effective in segmenting a long video that contains continuous human activities into fuzzy events; the fuzzy boundaries can well estimate the instant transition between temporally

adjacent activities. Figure 9.5b presents our approach's event-level human activity recognition results and comparisons with ground truth and human estimations. Due to the instant transition between human activities in the test video, human estimators are able to accurately segment the test video and correctly label the activity contained in each event. In addition, it can be observed that, based on the fuzzy event membership score, our FuzzySR achieves comparable segmentation results, and the activity contained in each event is correctly recognized.

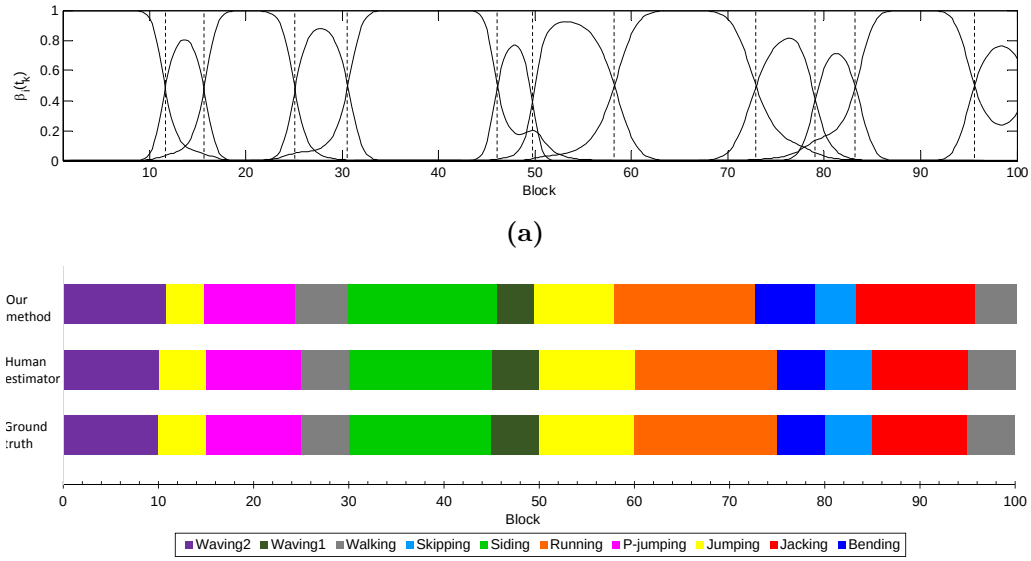
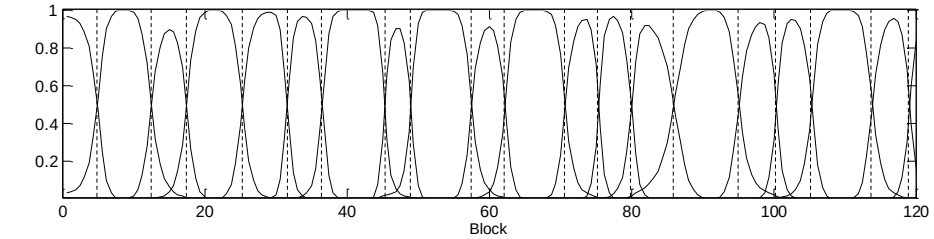


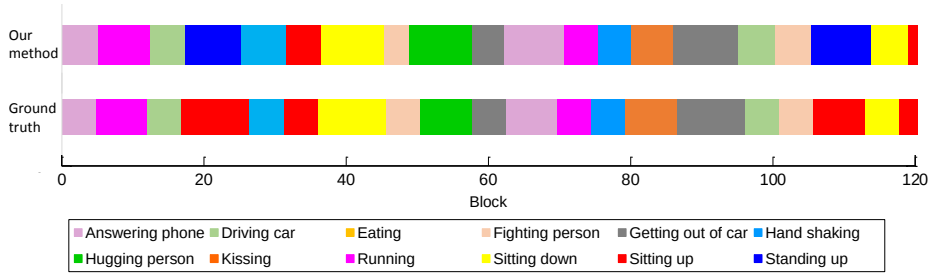
Figure 9.5: Experimental results of segmentation and recognition of continuous activities from the Weizmann dataset. The test video contains twelve events with instant transitions between temporally adjacent human activities. (a) Fuzzy segmentation. (b) Event-level activity recognition results and comparisons with ground truth and results provided by human estimators.

9.4.3 Results on Hollywood-2 Dataset

Following the setup suggested by the authors of the dataset [Marszalek et al., 2009], performance is evaluated using precision; 823 instances are used for training and 884 instances in testing. Following [Laptev, 2005], HOG features are applied in this experiment. We randomly generate 500 blocks from data instances in the training set, each training block containing 75 frames, which are applied to construct the LDA model and the dictionary that contains 600 visual words. In addition, we generate 120 testing blocks with the same duration from testing instance. These blocks are used to form a long video that is employed to evaluate our FuzzySR approach's temporal segmentation performance.



(a)



(b)

Figure 9.6: Experimental results of segmentation and recognition of continuous activities from the Hollywood-2 dataset. The test video contains twenty events with instant transitions between temporally adjacent activities. (a) Fuzzy segmentation. (b) Event-level activity recognition results and comparisons with ground truth and results provided by human estimators.

Experimental results of temporal fuzzy segmentation and event activity recognition over the Hollywood-2 dataset are reported in Figure 9.6. It is observed that our FuzzySR algorithm can well segment events out of continuous visual data. Recognition errors occur due the significant similarity between the sitting up and standing up activities.

In order to better evaluate our approach's performance, we compare our FuzzySR algorithm with unsupervised learning baselines [Jain et al., 1999], using the same LST features and experimental setups. The baselines unsupervised learning algorithms include partitioning unsupervised learning (e.g., k-means), hierarchical unsupervised learning (e.g., divisive analysis), artificial neural networks (e.g., self-organizing map), and model-based probabilistic unsupervised learning (e.g., mixture of Gaussian and probabilistic latent semantic analysis (PLSA) [Hofmann, 1999b]). The experimental results and comparisons are presented in Table 9.2. Our FuzzySR algorithm obtains an average precision of 42.6%. It is observed that our unsupervised FuzzySR method outperforms all used unsupervised learning baselines, which shows that our approach can well recognize event-level activities, even with the presence of occlusions in the dataset. We also compare our methods with previous works, which are based on supervised learning, as reported in Table 9.2. Supervised learning generally performs better than unsupervised learning since ground truth labels can be applied in the learning process to better estimate model parameters.

Table 9.2: Event-level average recognition precision (%) on the Hollywood-2 dataset.

Approach	Learning	Precision
Marszalek et al. [Marszalek et al., 2009]	Supervised	35.5
Derpanis et al. [Derpanis et al., 2013]	Supervised	48.0
Gilbert et al. [Gilbert et al., 2009]	Supervised	50.9
Wang et al. [Wang et al., 2013a]	Supervised	58.3
Chakraborty et al. [Chakraborty et al., 2012]	Supervised	58.5
K-means [Jain et al., 1999]	Unsupervised	29.9
Divisive analysis [Jain et al., 1999]	Unsupervised	34.8
Self-organizing map [Jain et al., 1999]	Unsupervised	31.6
Mixture of Gaussian [Jain et al., 1999]	Unsupervised	30.2
PLSA [Hofmann, 1999b]	Unsupervised	36.7
Our FuzzySR	Unsupervised	42.6

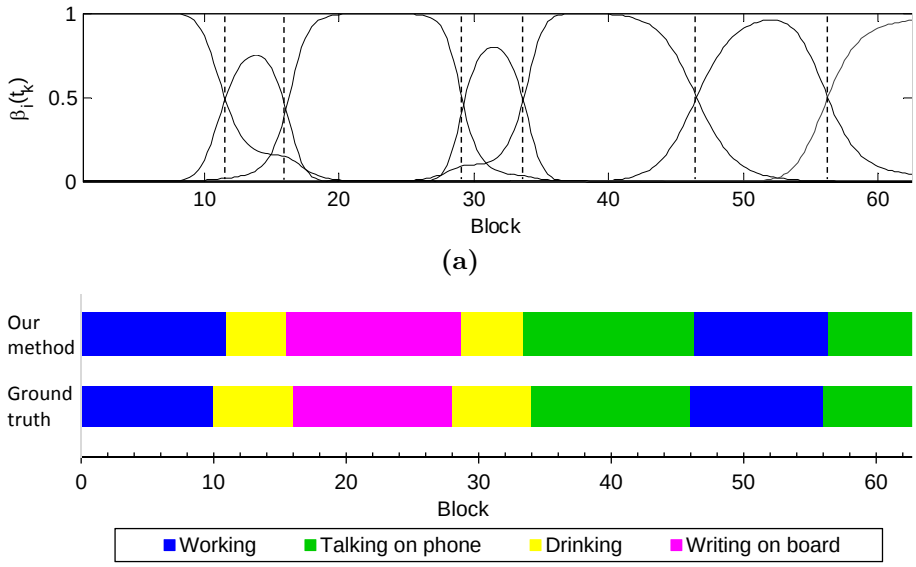


Figure 9.7: Experimental results of continuous human activity segmentation and recognition in the office scenario from the CAD-60 dataset. The test sequence contains seven events with instant transitions between temporally adjacent human activities. (a) Temporal Fuzzy segmentation. (b) Event-level human activity recognition results and comparison with ground truth.

Table 9.3: Average precision (%) and recall (%) of event-level average recognition over the CAD-60 dataset, and comparison with unsupervised baselines and existing supervised methods.

Approach	Learning	Precision	Recall
Sung et al. [Sung et al., 2012]	Supervised	67.9	55.5
Koppula et al. [Koppula et al., 2013]	Supervised	80.8	71.4
Ni et al. [Ni et al., 2013]	Supervised	75.9	69.5
Gupta et al. [Gupta et al., 2013]	Supervised	78.1	75.4
K-means [Jain et al., 1999]	Unsupervised	48.8	43.1
Divisive analysis [Jain et al., 1999]	Unsupervised	56.8	51.6
Self-organizing map [Jain et al., 1999]	Unsupervised	48.9	43.0
Mixture of Gaussian [Jain et al., 1999]	Unsupervised	51.7	46.2
PLSA [Hofmann, 1999b]	Unsupervised	57.7	55.2
Our FuzzySR	Unsupervised	60.4	55.8

9.4.4 Results on CAD-60 Dataset

Since the CAD-60 dataset only contains manually segmented color-depth videos, each only containing a single activity, we generate blocks from the dataset frames and then concatenate the blocks to form a long video that contains a sequence of continuous activities, following the typical protocol [Hoai et al., 2011]. As suggested by the authors of the dataset [Sung et al., 2012], the system's performance is evaluated according to different locations (e.g., kitchen); in addition, we apply the "new person" experimental setup and use precision and recall as our evaluation metrics. Specifically, we generate a number of 280 blocks, each of which contains 200 color-depth frames, using all instances in the dataset. Then, blocks from one person are used for testing, and blocks from the remaining three persons are used to learn a LDA model. Due to their ability to incorporate spatio-temporal color-depth information, we employ the 4D-LST features to encode the RGB-D frames in the CAD-60 dataset, following [Zhang and Parker, 2011]. A vocabulary that contains 1500 words is constructed and applied to convert a set of visual features from each block to a bag of words.

The continuous human activity segmentation and recognition results in the office scenario are graphically presented in Figure 9.7. This scenario includes four human activities: working on computer, talking on phone, writing on board, and drinking water. In this example, we observe that our FuzzySR obtains satisfactory activity segmentation performance. On the other hand, we notice that recognizing event-level activities from the CAD-60 dataset is very challenging, because several activities (e.g., stirring versus chopping, and relaxing on couch versus talking on couch) are almost identical. Accordingly, we also quantitatively evaluate our algorithm's performance on human activity recognition at the event level. The experimental results are presented in Table 9.3.

Qualitative evaluation with baseline unsupervised learning algorithms are conducted, using the same features and experimental setups. The comparison results are presented in Table 9.3. It is observed that our FuzzySR algorithm obtains superior performance over the baseline unsupervised learning methods. We also compare our unsupervised FuzzySR algorithm with existing supervised approaches, as demonstrated in Table 9.3. Although supervised learning often outperforms unsupervised learning in the event-level activity recognition task, supervised learning requires ground truth of all instances in the training set to learn model parameters. Since labeling instances is usually performed manually, it is very expensive to obtain ground truth and usually infeasible for a large amount of data in real-world situations. In addition, supervised learning approaches are not capable of discovering new activity patterns, which is critical for autonomous robotic systems to discover and adapt to unseen human behaviors.

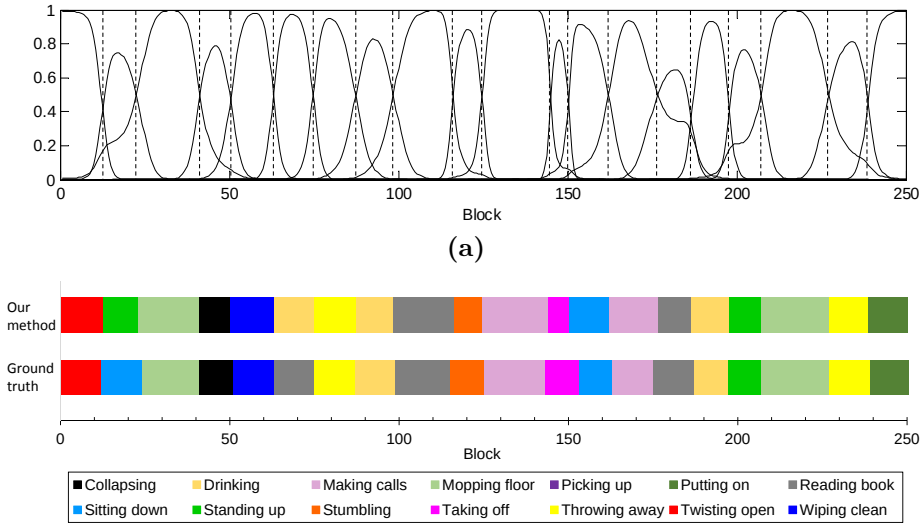
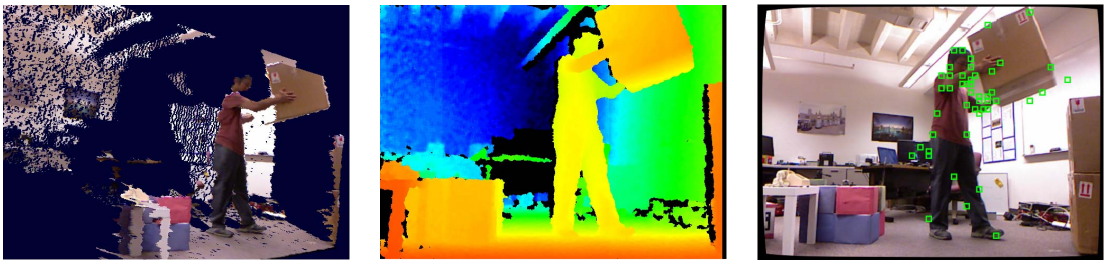


Figure 9.8: Experimental results of continuous activity segmentation and recognition in the office scenario from the ACT4^2 dataset. The test sequence contains twenty events with instant transitions between temporally adjacent human activities. (a) Temporal Fuzzy segmentation. (b) Event-level human activity recognition results and comparison with ground truth.

9.4.5 Results on ACT4^2 Dataset

Using similar settings as in our previous experiments, we generate 6000 blocks with each block containing around 100 color-depth frames; the 4D-LST features are applied to encode information from raw color-depth frames, and a vocabulary of size 1500 is used to construct the BoW representation from the 4D-LST features. Following evaluation setups in the original work [Cheng et al., 2012], we use blocks from eight

humans for learning our FuzzySR algorithm and the blocks from the remaining human subjects for testing; we apply average precision as the metric to evaluate our FuzzySR's performance on event-level human activity recognition.



(a) 3D view of an activity **(b)** Depth Frame **(c)** 4D-LST features

Figure 9.9: Setup of our experiments using the UTK-CAP dataset. The color-depth camera is installed on a Pioneer 3DX mobile robot. Our dataset represents continuous human activities in 3D space (Figure 9.9a), which contains both depth (Figure 9.9b) and color (Figure 9.9c) information. The extracted 4D local spatio-temporal features [Zhang and Parker, 2011] are also illustrated on the color image (Figure 9.9c).

Qualitative experimental results over the ACT4^2 dataset are shown in Figure 9.8. We can observe that our FuzzySR algorithm can well segment continuous human activities from the color-depth sequence. However, errors can occur when performing activity recognition from segmented events, due to the strong similarity of several human activities with small motions (e.g., drinking versus reading book). To better understand this error, we perform quantitative evaluation of our FuzzySR algorithm on event-level activity recognition, and present the results in Table 9.4. We also compare our algorithm against unsupervised learning baselines and existing supervised approaches. From Table 9.4, we can observe that our FuzzySR algorithm outperforms the unsupervised baselines, and can obtain comparable average event-level recognition precision to several supervised learning approaches (e.g., Color-HOGHOF [Cheng et al., 2012]).

9.4.6 Results on UTK-CAP Dataset

In real-world scenarios, gradual transitions always exist between temporally adjacent activities. Although the benchmark KTH, Weizmann, CAD-60 and ACT4^2 datasets can be used to generate long sequences, transitions between activities in the concatenated videos occur instantly, which is contradictory to the real-world situation. Accordingly, we employ a continuous activity dataset, i.e., UTK-CAP, to evaluate the effectiveness of our FuzzySR algorithm that explicitly models the gradual transition between adjacent activities in real-life situations.

We extract 600 blocks, that is 100 blocks for each activity, from the five color-depth videos to learn the LDA model for block-level activity summarization. We represent

Table 9.4: Event-level average recognition precision (%) over the ACT4^2 dataset.

Approach	Learning	Precision
Color-HOGHOF [Cheng et al., 2012]	Supervised	64.2
Depth-HOGHOF [Cheng et al., 2012]	Supervised	74.5
Depth-CCD [Cheng et al., 2012]	Supervised	76.2
DLMC-STIPs [Ni et al., 2011]	Supervised	66.3
SFR [Cheng et al., 2012]	Supervised	80.5
K-means [Jain et al., 1999]	Unsupervised	51.5
Divisive analysis [Jain et al., 1999]	Unsupervised	59.4
Self-organizing map [Jain et al., 1999]	Unsupervised	53.8
Mixture of Gaussian [Jain et al., 1999]	Unsupervised	50.9
PLSA [Hofmann, 1999b]	Unsupervised	62.7
Our FuzzySR	Unsupervised	65.2

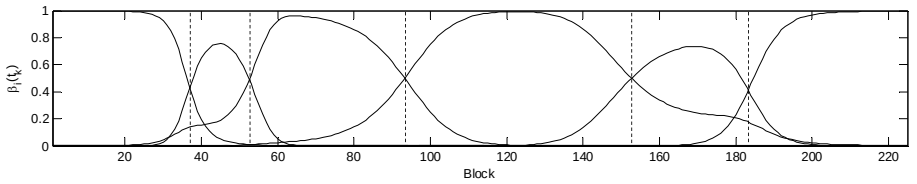
each block as a bag of visual words, which are computed through quantizing 4D-LST features [Zhang and Parker, 2011] that are extracted from the blocks using a dictionary of size 400. As an example, the extracted features for the grabbing box activity are shown in Figure 9.9c.

Experimental results over a color-depth video that contains six events are depicted in Figure 9.10. It can be observed from Figure 9.10a that the test color-depth video is well segmented by our algorithm, which is able to model gradual transitions between temporally adjacent activities. By representing events as fuzzy sets, our FuzzySR method well estimates the membership of each block. When a block appears in the center of an event, it has a high membership score. If a block approaches to the end of the current event, its membership score decreases. Blocks located in gradual transitions have low membership scores for the ongoing event and the new event.

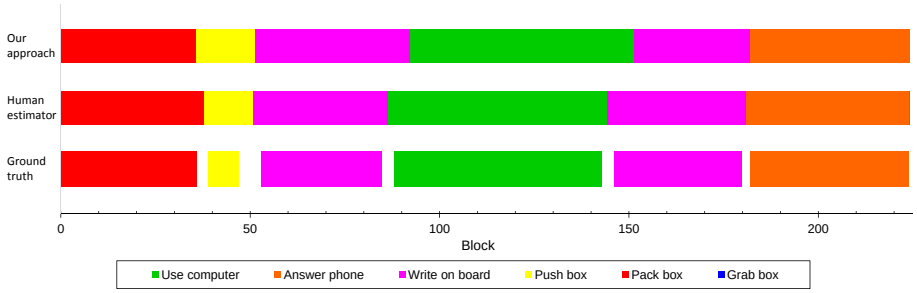
Table 9.5: Event-level average recognition precision (%) over the UTK-CAP dataset.

Approach	Learning	Precision
K-means [Jain et al., 1999]	Unsupervised	67.1
Divisive analysis [Jain et al., 1999]	Unsupervised	69.2
Self-organizing map [Jain et al., 1999]	Unsupervised	66.5
Mixture of Gaussian [Jain et al., 1999]	Unsupervised	68.9
PLSA [Hofmann, 1999b]	Unsupervised	76.2
FuzzySR (based on LDA)	Unsupervised	78.9

The continuous human activity recognition results over the UTK-CAP dataset are depicted in Figure 9.10b. It can be observed that, with the presence of gradual transitions between activities, our FuzzySR approach is still able to correctly recognize continuous activities and well estimate event boundaries. In this experiment, ground truth is provided by the human actor who performs these activities. Transitions



(a)



(b)

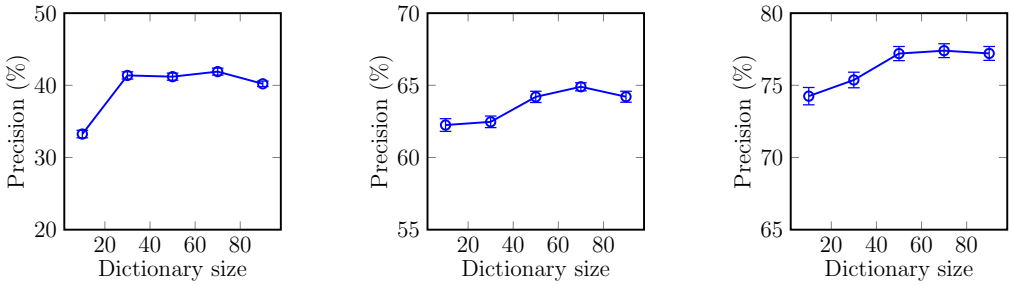
Figure 9.10: Experimental results of segmentation and recognition of continuous activities using our continuous activity dataset. The test color-depth sequence contains six events with gradual transitions between temporally adjacent activities. (a) Fuzzy segmentation. (b) Event-level activity recognition results and comparisons with ground truth and results provided by human estimators. The white spaces in ground truth denote transitions between activities.

between temporally adjacent activities are explicitly labeled in the ground truth, as denoted by the white spaces in Figure 9.10b. For comparison, we invited five human estimators to manually partition and recognize the continuous activities contained in the test video. Without knowing the number of activities, human estimators clustered the store owner's activities into 4, 4, 5, 6 and 44 categories, which indicates a strong ambiguity on the definition of the activities in this dataset. Given the number of human activities, human estimators correctly recognized the activities. On the other hand, with the presence of gradual transitions, human evaluators often have difficulty precisely labeling each event's boundaries. These phenomena can be seen in Figure 9.10b. Comparing with human estimations, our FuzzySR algorithm achieves comparable segmentation results over the UTK-CAP dataset, as depicted in Figure 9.10b.

In addition, we quantitatively evaluate our FuzzySR algorithm's average recognition precision and compare our result with the unsupervised learning baselines, as presented in Table 9.5. Similar to the phenomena observed in our previous experiments, the FuzzySR algorithm obtains better performance on recognizing event-level human activities.

9.4.7 Sensitivity Analysis

In this section, we focus on evaluating the sensitivity of our FuzzySR algorithm to algorithm parameters that are critical for achieving satisfactory activity segmentation and recognition performance. Specifically, the algorithm's parameters, including block size and dictionary size (i.e., number of visual words), are investigated. In addition, in order to analysis the effect caused by random initialization (as used by the k-means algorithm to construct the dictionary), each set of experiments are performed five times, and an error bar is used to represent the performance variation. Three datasets are employed to perform sensitivity analysis, including Hollywood-2, ACT4^2, and UTK-CAP datasets. When conducting sensitivity analysis to a specific parameter, other parameters are set to the values that are reported in Section 9.4.3, 9.4.5 and 9.4.6, for the three used datasets, respectively.

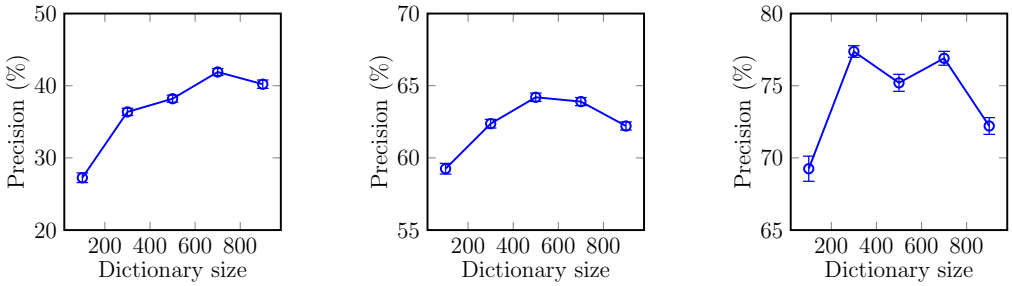


(a) Hollywood-2 **(b)** ACT4^2 **(c)** UTK-CAP

Figure 9.11: Our algorithm's sensitivity to the parameter of block size (i.e., number of frames contained in a block).

Block size

This parameter controls the temporal duration of each block (i.e., total number of frames in a block). Performance of event-level activity recognition over different datasets using different block sizes is graphically reported in Figure 9.11. It can be observed that a very small block size results in bad the event-level activity recognition performance. This is because in the case when less frames contained in the block, the number of extracted visual features is not large enough to represent the activities contained in the block. Intuitively, the block size cannot be assigned to a very large value, because otherwise the block may contain multiple human activities. As a general guideline, in real-world applications using cameras with 30Hz frame rate (e.g., Kinect), using the block size that is in the range between 30 and 60 frames (corresponding to 1–2 seconds) can usually result in satisfactory event-level activity recognition performance.



(a) Hollywood-2 **(b)** ACT4[2] **(c)** UTK-CAP

Figure 9.12: Our algorithm's sensitivity to the parameter of dictionary size (i.e., number of visual words).

Dictionary size

This parameter controls the number of visual words contained in the dictionary. Since the standard k-means algorithm is employed to construct the dictionary, this parameter also serves as the number of clusters that is provided as a prior to k-means. Event-level activity recognition performance using different dictionary sizes is reported in Figure 9.12. It is observed that the dictionary that has a moderate size usually results in satisfactory event-level activity recognition performance. This is because, when using a small dictionary size, LST features with different patterns can be incorrectly assigned to the same cluster (i.e., visual word). On the other hand, when a very large dictionary size is employed, visual features with similar characteristics can be incorrectly assigned to different clusters. In general, we observe that the dictionary size in the range between 300–800 can achieve good event-level activity recognition results. In addition, our approach is generally not sensitive to different initializations of k-means clustering, which is demonstrated by the small error bars that are computed using recognition results in different runs of the experiment.

9.5 Summary

We introduce the new FuzzySR algorithm to perform continuous human activity segmentation and recognition. Given a video containing continuous human activities, after uniformly partitioning the video into disjoint blocks, our algorithm computes the activity distribution of each block through mapping high-dimensional discrete feature space to real-valued activity space. Then, the summaries are used to form a multi-variable time series, and fuzzy temporal clustering is used to segment events. Lastly, our algorithm incorporates all block summaries contained in an event and solves an optimization problem to determine the most appropriate activity label for each event. Our main contributions include explicitly modeling the gradual transition between temporally adjacent human activities, and bridging the divide

between the bag-of-word model based on LST features and the continuous human activity segmentation problem. Extensive empirical studies are conducted using six real-world human activity datasets, with a focus on temporally segmenting and probabilistically recognizing continuous human daily activities from both color and RGB-D visual data in human social environments. Experimental results demonstrate our FuzzySR's satisfactory performance, which can allow an autonomous robot to interpret continuous human activities in real-world human social environments.

In many real-world situations, the number of activity clusters may not be available. Accordingly, an interesting future research direction is to automatically determine this parameter. Potential solutions to address this problem include using model selection or nonparametric extensions of LDA models, such as Hierarchical Dirichlet Processes (HDP)-LDA. In addition, online learning to construct the dictionary and LDA model is another interesting future direction, because it has the potential to improve our method's efficiency and adaptability, especially when the system is deployed in new, unknown environments.

Chapter 10

Recognition: Cognitive Model for Decision Making

10.1 Introduction

After perceiving human activities, appropriate decision-making is a crucial capability for an autonomous robot to interact with humans in daily life scenarios. Issues including the safety of humans and property, reliability, and general usefulness of an autonomous robot will all manifest in the absence of accurate human perception and appropriate decision making. In order to provide these important capabilities, intelligent systems, such as smart homes, intelligent vehicles, and human-assistance robots increasingly require a cognitive component that provides human perception, reasoning, and decision making capacities in order to interact with users in an intelligent way.

However, developing such an artificial cognitive system is a very challenging task, because where such a system requires high-level processes such as reasoning and decision making, these high-level processes also need to interact with more basic components such as perception [Schmid et al., 2011]. This combines the difficulty of developing accurate and reliable components with the complexity of combining them into a larger system. Moreover, the artificial cognitive model must be usable in a complex dynamic environment with great uncertainty [Knauff and Wolf, 2010].

This uncertainty arises mainly from the complexity of the human activity recognition (HAR) task itself. The scope of possible human activities can be very large, and cannot always be predefined. Substantial variations are inherently contained within an activity, especially when performed by different humans. Even the same activity conducted by the same human contains different speeds with different poses, giving rise to temporal variations. Uncertainty also exists in the vision-based perception system, which suffers from unpredictable changes (e.g., illumination changes) and dynamic environments due to camera motions or the involvement of other agents. In addition, humans might be only partially observable in the scene and, as a consequence, human activity prediction is even less certain.

The difficulties caused by uncertainty in perception systems have been studied and addressed in the machine vision community. It is well known in this community that well-designed local features provide valuable visual cues for human [Le et al., 2011] and object perception [Alahi et al., 2012]. These features are usually invariant to image rotation, scaling, and translation, partially invariant to illumination changes, and robust to partial occlusion [Lowe, 2004]. The emergence of these local features has greatly increased the popularity of the bag-of-visual-words (BoW) representation [Niebles et al., 2008]. In BoW, each visual feature in an observation is converted to a discrete visual word through vector quantization, and each observation is encoded as a histogram of word occurrences.

With the increase in popularity of the BoW representation, topic models that were originally introduced for document clustering, such as Latent Dirichlet Allocation (LDA) [Blei et al., 2003], have been adapted and extensively used in HAR tasks [Zhang and Parker, 2011]. In these tasks, observations encoded with the BoW representation are grouped into clusters using topic models. Then, the resulting clusters are mapped to known categories, and each observation is labeled with the category of highest probability [Niebles et al., 2008]. Since topic modeling is able to generate a distribution over activity categories, the risks of possible robot actions with regard to human activities can be incorporated in the decision making process to select the responsive action with the lowest overall risk. Where previous studies were only aimed at recognition of human activities, to the best of our knowledge, no work has attempted to build upon topic modeling by incorporating decision making in this way (i.e., selecting a responsive action) into a more complete artificial cognitive system.

To evaluate topic model's performance on HAR tasks, the accuracy metric, i.e., the rate of correctly clustered observations, is typically used [Niebles et al., 2008], which is an *extrinsic* performance metric that depends on the specific task and requires *a priori* ground truth. Accuracy is computed by comparing the ground truth label with the single estimated label. Because the accuracy metric ignores the distribution over activity categories, which is richer and more informative than a single label, it is therefore not appropriate for evaluating topic modeling. As an example, let us consider an HAR task with two activities and assume that two topic models obtain the distributions, $[0.8, 0.2]$ and $[0.55, 0.45]$, on a given observation, and ground truth indicates that the first activity is the correct assignment. Although both models are accurate, in that the activity with the highest probability matches the ground truth in both cases, the first model obviously performs better, since it better separates the correct assignment from incorrect assignments. In previous research this important observation has not been utilized.

In live, real-world HAR tasks, artificial cognitive systems are typically used in an online fashion. If the observations contain new activities that are not presented during the training phase, topic models become less practical, as the system is often likely to choose an incorrect or unsafe robot action response. For this reason, the ability to detect new activities in order to evolve and adapt to changing environments is essential

for the use of topic models in cognitive modeling. We propose an *intrinsic* metric that measures the generalization capacity of a topic model, which is an unsupervised, online indicator that is independent of any specific application. We demonstrate that this metric indicates the novelty of an observation, which shows strong potential to help make more appropriate decisions in online application scenarios. To the best of our knowledge, no previous research has developed intrinsic metrics to discover new knowledge in HAR tasks.

In this work, we focus on a new approach based on topic modeling to construct a cognitive model in an artificial intelligent system that is capable of making appropriate decisions in response to human activities. To accomplish this, we make the following contributions:

- We define a new, extrinsic metric, called the *interpretability indicator* (I_I), which measures how topic models are interpreted (i.e., how well human perception and topic modeling agree). We mathematically analyze its properties and prove that it generalizes the accuracy measure. We demonstrate that this indictor is an appropriate metric for selecting topic models that can better perceive human activities.

- We introduce an intrinsic metric, called the *generalizability indicator* (I_G), which measures the topic model's generalization capacity (i.e., how well an unseen observation is represented by the model learned using the training set). We demonstrate this indicator's effectiveness at discovering new knowledge in online scenarios.

- We perform extensive experiments to statistically investigate the relationship of I_I and I_G, and we make two very important observations: 1) If I_G is close to its maximum value, the training set is exhaustive (i.e., it well-represents the unseen observations). In this case, I_I is independent of I_G. 2) If I_G has a small value, the training set is non-exhaustive (i.e., new activity appears in the unseen data), and I_I and I_G are moderately to strongly correlated.

- We investigate topic modeling's advantages in constructing reliable artificial cognitive models in human-machine interaction applications. We demonstrate the benefits of applying topic models along with performance indicators for risk estimation and decision making. To the best of our knowledge, this is the first work that introduces topic modeling in an artificial cognitive model in this fashion.

The rest of the chapter is organized as follows. Related work is reviewed in Section 10.2. We describe the structure of our artificial cognitive model and its functional modules in Section 10.3. We also discuss topic modeling in this section. Then, Section 10.4 introduces the two new evaluation metrics. In Section 10.6, we present how to apply topic modeling to risk estimation and decision making. Experimental results are discussed in Section 10.6. Finally, we summarize this work in Section 10.7.

10.2 Related Work

10.2.1 Topic Models and Evaluation

Among other machine learning techniques, topic modeling has been widely applied to HAR tasks in previous research. [Niebles et al., 2008] used topic models to cluster human activities in videos encoded using the BoW representation. A semi-latent topic model trained in a supervised fashion was introduced in [Wang and Mori, 2009] and used to classify activities in videos. [Zhang and Parker, 2011] adopted topic models to classify activities in 3D point clouds obtained from color-depth cameras on mobile robots. Topic models were also widely used to discover human activities in streaming data. The use of topic models was explored in [Huynh et al., 2008] to discover daily activity patterns in wearable sensor data. An unsupervised topic model was introduced in [Farrahi and Gatica-Perez, 2011] to detect daily routines from streaming location and proximity data.

While there is a significant body of work introducing and developing sophisticated topic models and their applications, only a few efforts have been undertaken to evaluate topic model's performance. Existing evaluation methods are dominated by either intrinsic methods, (e.g., computing the probability of held-out documents to evaluate generalization ability [Wallach et al., 2009]) or extrinsic methods using external tasks, (e.g., information retrieval [Wei and Croft, 2006]). Recent work also focused on evaluation of topic modeling's interpretability as semantically coherent concepts. For example, [Chang et al., 2009a] demonstrated that the probability of held-out documents is not always a good indicator of human judgment. It was also shown by [Newman et al., 2010] that the metrics based on word co-occurrence statistics are able to predict human evaluations of topic quality.

As recently pointed out by Blei [Blei, 2012], topic model evaluation is an essential research topic. Despite this, in previous work, only the accuracy metric is used to evaluate topic modeling results in HAR tasks, and issues such as the model's interpretability and generalizability have not been studied. In this study, we analyze these two aspects of topic model evaluation in HAR tasks, explore their relationship, and show how they can be used to improve model selection and decision making.

10.2.2 Artificial Cognitive Modeling

Artificial cognition has its origins in cybernetics with the intention to create a science of mind based on logic [Varela and Dupuy, 1992]. Among other cognitive paradigms, cognitivism has undoubtedly been predominant to date [Vernon et al., 2007]. Within the cognitivism paradigm, several cognitive architectures were developed, including Soar [Laird et al., 1987], ACT-R [Anderson, 1996], C4 [Isla et al., 2001], and architectures for robotics [Burghart et al., 2005, Benjamin et al., 2004], which are relatively independent of the task [Gray et al., 1997]. Since architectures represent the fixed part of cognition, they cannot accomplish anything in their own right and need to be

provided with knowledge to conduct a specific task. The combination of a cognitive architecture and a particular knowledge set is generally referred to as a *cognitive model* [Vernon et al., 2007]. The knowledge incorporated in cognitive models is typically determined by human designers [Vernon et al., 2007]. The knowledge can be also learned and adapted using machine learning techniques.

Cognitive models have been widely used in human-machine interaction and robotic vision applications. For example, artificial cognitive modeling was adopted in [Duric et al., 2002] to construct intelligent human-machine interaction systems. Cognitive perception systems were also widely employed to recognize traffic signs [Yang et al., 2013], interpret traffic behaviors [Nagel, 2004], and recognize human activities [Crowley, 2006]. Over the last decade, probabilistic models of cognition, as an alternative of deterministic cognitive models, have attracted more attention in cognitive development [Xu and Griffiths, 2011]. For example, a cognitive vision system was designed in [Buxton, 2002] to use dynamic decision networks to interpret activities of expert human operators. Another cognitive model was introduced in [Town and Sinclair, 2003] to apply adaptive Bayesian networks for video analysis. Probabilistic models have also been used for learning and reasoning in cognitive modeling [Chater et al., 2006].

We believe we are the first to adopt topic models for the construction of reliable artificial cognitive models and show that they are particularly suited for this task. We demonstrate topic modeling's ability to combine risks in decision making. In addition, we develop two evaluation metrics and show their effectiveness in model selection and decision making. These aspects were not addressed in previous artificial cognitive modeling research.

10.3 Topic Modeling for Artificial Cognition

10.3.1 System Overview

Our artificial cognitive model is designed for human-robot interaction applications, where humans and robots are operating in the same workspace. The cognitive model is inspired by the C4 brain cognitive architecture [Isla et al., 2001]. As shown in Figure 10.1, our model is organized into four modules by their functionality:

- *Sensory and perception module:* The sensory module uses visual cameras to observe surrounding humans and sense the environment. Then, the perception system extracts local visual features from raw observations and encodes them to the BoW representation that can be 'perceived' (understood) by topic models.

- *Probabilistic reasoning module:* Topic models are applied to probabilistically reason about human activities, which are trained off-line and used online. The training dataset is provided as a prior, which encodes a history of sensory information. This module also uses evaluation indicators to select topic models

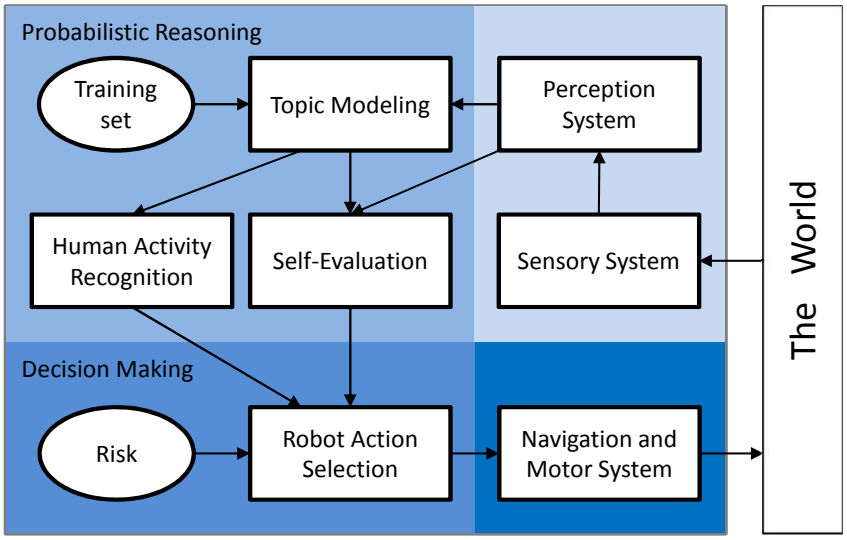


Figure 10.1: Overview of our artificial cognitive model that incorporates topic modeling and its evaluation. Information flows from modules with lighter colors to those with darker colors. Entities in ellipses are prior knowledge to our artificial cognitive model.

that better match human's perspective, and to discover new activities in an online fashion.

- *Decision making module:* This module estimates the overall risk based on topic modeling and evaluation results, and selects a response action for the robot that minimizes this risk. The risk is provided as a prior to the module.
- *Navigation and motor module:* The navigation module dynamically plans a smooth path to designated locations without collision with obstacles or other moving agents. The motor module generates motions of the robot action in response to human activities.

10.3.2 Topic Modeling

Prior to our work, topic models have not been applied to the construction of artificial cognitive systems capable of making reliable decisions. Although our discussion is based on the benchmark topic model, LDA [Blei et al., 2003], other sophisticated topic models are directly applicable to our cognitive model.

Given a set of observations $\mathcal{W}$, LDA models each of K activities as a multinomial distribution of all possible visual words in the dictionary $\boldsymbol{D}$. This distribution is parameterized by $\boldsymbol{\varphi} = \{\varphi_{w_1}, \ldots, \varphi_{w_{|\boldsymbol{D}|}}\}$, where φ_w is the probability that the word w is generated by the activity. LDA also represents each observation $\boldsymbol{w} \in \mathcal{W}$ as a collection of the visual words, and assumes that each word $w \in \boldsymbol{w}$ is associated with a

latent activity assignment z. By using these visual words to connect observations and activities, LDA models observation $\boldsymbol{w}$ as a multinomial distribution over the activities, which is parameterized by $\boldsymbol{\theta} = \{\theta_{z_1}, \ldots, \theta_{z_K}\}$, where θ_z is the probability that $\boldsymbol{w}$ is generated by the activity z. LDA is a Bayesian model, which places Dirichlet priors on the multinomial parameters: $\boldsymbol{\varphi} \sim \mathrm{Dir}(\boldsymbol{\beta})$ and $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, where $\boldsymbol{\beta} = \{\beta_{w_1}, \ldots, \beta_{w_{|\boldsymbol{D}|}}\}$ and $\boldsymbol{\alpha} = \{\alpha_{z_1}, \ldots, \alpha_{z_K}\}$ are the concentration hyperparameters.

One of the major objectives in HAR tasks to is to estimate the parameter $\boldsymbol{\theta}$, i.e., the per-observation activity proportion. However, exact parameter estimation is intractable in general [Blei et al., 2003]. We adopt Gibbs sampling [Griffiths and Steyvers, 2004] to approximately estimate the LDA model's parameters, based on two considerations: 1) This sampling-based method is generally accurate, since it asymptotically approaches the correct distribution [Porteous et al., 2008], and 2) This method can be used to intrinsically evaluate topic model's performance [Wallach et al., 2009], thereby providing a consistent method to infer, learn, and evaluate topic models. When Gibbs sampling converges, the element $\theta_{z_k} \in \boldsymbol{\theta}$, $k = 1, \ldots, K$, is estimated by:

$$\hat{\theta}_{z_k} = \frac{n_{z_k} + \alpha_{z_k}}{\sum_z (n_z + \alpha_z)}, \tag{10.1}$$

where n_z is the number of times that a visual word is assigned to activity z_k in the observation.

The incorporation of topic models into cognitive modeling has several important outcomes. First, as a probabilistic reasoning approach, it serves as a bridge to allow information to flow from the perception module to the decision making module. Second, the ability to model per-observation activity distribution allows topic models to take into account the risks of all robot actions in a probabilistic way and make an appropriate decision. Third, by introducing an extrinsic evaluation metric for topic model selection, the constructed cognitive system is able to accurately interpret human activities. Fourth, the unsupervised nature of topic modeling, which is explored using our new intrinsic metric, facilitates online discovery of new knowledge (e.g., human activities). Together, these outcomes allow us to apply topic models to construct an artificial cognitive system that is able to better interpret human activities, discover new knowledge and react more appropriately and safely to humans, which is highly desirable for real-world online human-robot interaction scenarios.

10.4 Cognition Improvement by Model Evaluation

In this section, in order to improve the performance of our artificial cognitive model, we introduce two novel evaluation indicators: the interpretability indicator, which is used to select a topic model that best matches human common sense, and the generalizability indicator, which provides the capability of online knowledge discovery. We also investigate the relationship between these indicators.

10.4.1 Interpretability Indicator

When an unsupervised model (e.g., LDA) is trained using labeled instances, the resulting clusters need to be mapped to the known categories by maximizing certain extrinsic performance metrics such as accuracy [Niebles et al., 2008]. However, the accuracy metric is not an appropriate performance measure in our applications, since it only represents whether the most probable category assignment matches the ground truth and ignores the category distribution, which contains much richer information.

To make use of the category distribution obtained by topic models, we introduce the new *interpretability indicator*, denoted by I_I, which is used to evaluate how well topic modeling matches human common sense, map discovered clusters to known categories, and select the best topic model to reason about human activities. Like the accuracy metric, I_I is an extrinsic metric, which depends on specific tasks and requires the ground truth to compute. Formally, we define I_I as follows:

Definition 5 (Interpretability indicator)**.** *Given an observation $\boldsymbol{w}$ with the ground truth g and the proportion $\boldsymbol{\theta}$ over $K \geq 2$ categories, let $\boldsymbol{\theta}_s = (\theta_1, \ldots, \theta_{k-1}, \theta_k, \theta_{k+1}, \ldots, \theta_K)$ be the sorted proportion satisfying $\theta_1 \geq \cdots \geq \theta_{k-1} \geq \theta_k \geq \theta_{k+1} \geq \cdots \geq \theta_K \geq 0$ and $\sum_{i=1}^{K} \theta_i = 1$, and let $k \in \{1, \cdots, K\}$ be the index of the assignment in $\boldsymbol{\theta}_s$ that matches g. We define the interpretability indicator $I_I(\boldsymbol{\theta}, g) = I_I(\boldsymbol{\theta}_s, k)$ satisfying*

$$I_I(\boldsymbol{\theta}_s, k) = \frac{1}{a}\left(\frac{K-k}{K-1} + \mathbb{1}(k=K)\right)\left(\frac{\theta_k}{\theta_1} - \frac{\theta_{k+\mathbb{1}(k\neq K)}}{\theta_k} + b\right), \tag{10.2}$$

where $\mathbb{1}(\cdot)$ is the indicator function, and $a = 2$ and $b = 1$ are normalizing constants.

The indicator I_I is defined over the per-observation category proportion $\boldsymbol{\theta}$, which takes values in the $(K-1)$-simplex [Blei et al., 2003]. The sorted proportion $\boldsymbol{\theta}_s$ is computed through sorting $\boldsymbol{\theta}$ that is inferred by topic models. In the definition, the ground truth is represented by its location in $\boldsymbol{\theta}_s$, i.e., the k-th most probable assignment in $\boldsymbol{\theta}_s$ matches the ground truth label. The indicator function $\mathbb{1}(\cdot)$ in Eq. (10.2) is used to deal with the special case when $k = K$.

For an observation in a classification task with K categories, given its ground truth index k and sorted category proportion $\boldsymbol{\theta}_s$, we summarize I_I's properties as follows:

Proposition 4 (I_I's properties)**.** *The interpretability indicator $I_I(\boldsymbol{\theta}, g) = I_I(\boldsymbol{\theta}_s, k)$ satisfies the following properties:*

1. If $k = 1$, $\forall \boldsymbol{\theta}_s$, $I_I(\boldsymbol{\theta}_s, k) \geq 0.5$.

2. If $k = K$, $\forall \boldsymbol{\theta}_s$, $I_I(\boldsymbol{\theta}_s, k) \leq 0.5$.

3. $\forall \boldsymbol{\theta}_s$, $I_I(\boldsymbol{\theta}_s, k) \in [0, 1]$.

4. $\forall k \in \{1, \ldots, K\}$ and $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ such that $\theta_1 \geq \theta'_1$, $\theta_k = \theta'_k$ and $\theta_{k+\mathbb{1}(k\neq K)} = \theta'_{k+\mathbb{1}(k\neq K)}$, $I_I(\boldsymbol{\theta}_s, k) \leq I_I(\boldsymbol{\theta}'_s, k)$ holds.

5. $\forall k \in \{1, \ldots, K\}$ *and* $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ *such that* $\theta_{k+\mathbb{1}(k\neq K)} \geq \theta'_{k+\mathbb{1}(k\neq K)}$, $\theta_1 = \theta'_1$ *and* $\theta_k = \theta'_k$, $I_I(\boldsymbol{\theta}_s, k) \leq I_I(\boldsymbol{\theta}'_s, k)$ *holds.*

6. $\forall k \in \{1, \ldots, K\}$ *and* $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ *such that* $\theta_k \geq \theta'_k$, $\theta_1 = \theta'_1$ *and* $\theta_{k+\mathbb{1}(k\neq K)} = \theta'_{k+\mathbb{1}(k\neq K)}$, $I_I(\boldsymbol{\theta}_s, k) \geq I_I(\boldsymbol{\theta}'_s, k)$ *holds.*

7. $\forall k, k' \in \{1, \ldots, K\}$ *such that* $k \leq k' < K$ *and* $\forall \boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ *such that* $\theta_k = \theta'_k$, $\theta_1 = \theta'_1$ *and* $\theta_{k+\mathbb{1}(k\neq K)} = \theta'_{k+\mathbb{1}(k\neq K)}$, $I_I(\boldsymbol{\theta}_s, k) \geq I_I(\boldsymbol{\theta}'_s, k')$ *holds.*

Proof. See Appendix 10. □

The indicator I_I can be used to quantitatively measure how well topic modeling matches human common sense because it captures three essential considerations to simulate the process of how humans evaluate the category proportion $\boldsymbol{\theta}$:

- A topic model performs better, in general, if it obtains a larger θ_k (Property 6). In addition, a larger θ_k generally indicates θ_k is closer to the beginning in $\boldsymbol{\theta}_s$ and further away from the end (Property 7).

  Example: A topic model obtaining the sorted proportion $[0.4, \boxed{0.35}, 0.15, 0.10]$ performs better than a model obtaining $[0.4, \boxed{0.30}, 0.15, 0.15]$, where the ground truth is marked with a box, i.e., $k = 2$ in this example.

- A smaller difference between θ_k and θ_1 generally indicates better modeling performance (Properties 4 and 5). Since the resulting category proportion is sorted, a small difference between θ_k and θ_1 guarantees θ_k has an even smaller difference from θ_2 to θ_{k-1}.

  Example: A topic model obtaining the sorted proportion $[0.4, \boxed{0.3}, 0.2, 0.1]$ performs better than the model with the proportion $[0.5, \boxed{0.3}, 0.2, 0]$.

- A larger distinction between θ_k and θ_{k+1} generally indicates better modeling performance (Properties 5 and 6), since it better separates the correct assignment from the incorrect assignments with lower probabilities.

  Example: A topic model obtaining the sorted proportion $[0.4, \boxed{0.4}, 0.1, 0.1]$ performs better than the topic model obtaining the proportion $[0.4, \boxed{0.4}, 0.2, 0]$.

We normalize I_I to the range $[0, 1]$ (Property 3), with a greater value indicating a better interpreted model. If an observation's most probable assignment matches the ground truth, I_I is guaranteed to be greater or equal to 0.5 (Property 1). Similarly, when the least probable assignment matches the ground truth (Property 2), I_I is no greater than 0.5.

During the training phase, topic modeling groups the training set $\mathcal{W}_{train}$ into clusters $\mathcal{W}_{c_1} \cup \mathcal{W}_{c_2} \cup \cdots \cup \mathcal{W}_{c_K}$. Because topic models are unsupervised, it is necessary to associate the resulting clusters with pre-defined categories. We introduce a procedure called *Topic Mapping*, based on I_I, to automatically perform topic association:

Definition 6 (Topic mapping)**.** *Let* $\boldsymbol{g} = \{g_1, \ldots, g_M\}$ *be the ground truth of a set of observations* $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$*. We denote the interpretability indicator over* $\mathcal{W}$ *to be:*

$$I_I(\mathcal{W}, \boldsymbol{g}) = \frac{1}{M} \sum_{m=1}^{M} I_I(\boldsymbol{w}_m, g_m). \tag{10.3}$$

Let $\boldsymbol{g} = \{g_1, \ldots, g_K\}$ *be the ground truth and* $\boldsymbol{c} = \{c_1, \ldots, c_K\}$ *be the cluster indices of* $\mathcal{W}_{c_1}, \ldots, \mathcal{W}_{c_K}$*. We define topic mapping to be a bijective function* $f : \boldsymbol{c} \rightarrow \boldsymbol{g}$ *such that*

$$f = \arg\max_{f'} \frac{1}{K} \sum_{k=1}^{K} I_I(\mathcal{W}_k, f'(c_k)). \tag{10.4}$$

Intuitively, topic mapping is a one-to-one and onto mapping of the cluster indices to the ground truth that maximizes topic modeling's average interpretability on the training set. Given the training set, the topic model whose hyper-parameters (e.g., $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in LDA) maximize this average interpretability is selected as our final model to reason about human activities. In this work, we employ the Hungarian algorithm [Kuhn, 1955] to solve this topic mapping problem.

It is noteworthy that I_I extends the most commonly applied *accuracy* metric I_A, which is defined as the rate of correctly classified instances, as described in Proposition 5:

Proposition 5 (I_I's relationship to I_A)**.** *The accuracy measure* I_A *is a special case of* $I_I(\boldsymbol{\theta}_s, k)$*, when* $\theta_1 = 1.0$*,* $\theta_2 = \ldots = \theta_K = 0$*, and* $k = 1$ *or* $k = K$*.*

Proof. See Appendix 10. □

10.4.2 Generalizability Indicator

An artificial cognitive model requires the crucial capability of discovering new knowledge in an online fashion in order to adapt to environment changes. To obtain this capability, we introduce a novel indicator, called *generalizability indicator* (I_G), to discover new knowledge online using topic modeling. This indicator is an intrinsic performance evaluation metric, which does not require ground truth to compute and consequently can be used in an online fashion.

The introduction of I_G is inspired by the perplexity evaluation metric (also called held-out likelihood), which evaluates the generalization capacity of topic models on a fraction of the held-out instances in a cross-validation manner [Musat et al., 2011], or on the unseen observations [Blei and Lafferty, 2006]. The perplexity of an observation is defined as the log-likelihood of the words in the observation [Wallach et al., 2009]. In our applications, different observations may contain a significantly different number

Algorithm 6: Left-to-right *Pvwp* estimation

Input : $\boldsymbol{w}$ (observation), $\mathcal{M}$ (trained topic model), and R (number of particles)

Output: $Pvwp(\boldsymbol{w}|\mathcal{M})$

```
Initialize l = 0 and N = |w|;
for each position n = 1 to N in w do
    Initialize p_n = 0;
    for each particle r = 1 to R do
        for n' < n do
            Sample z_{n'}^{(r)} ~ P(z_{n'}^{(r)} | w_{n'}, {z_{<n}^{(r)}}_{¬n'}, M);
        end
        Compute p_n = p_n + Σ_t P(w_n, z_n^{(r)} = t | z_{<n}^{(r)}, M);
        Sample z_n^{(r)} ~ P(z_n^{(r)} | w_n, z_{<n}^{(r)}, M);
    end
    Update p_n = p_n / R and l = l + log p_n;
end
return Pvwp(w|M) ≃ l / N.
```

of visual words. Given a trained topic model, an observation with a larger number of visual words generally has a smaller perplexity than an observation with fewer words. In this situation, it is reasonable to compute the *Per-Visual-Word Perplexity (Pvwp)*. Mathematically, given the trained topic model $\mathcal{M}$ and an observation $\boldsymbol{w}$, *Pvwp* is defined as follows:

$$Pvwp(\boldsymbol{w}|\mathcal{M}) = \frac{1}{N}\log P(\boldsymbol{w}|\mathcal{M}) = \frac{1}{N}\log\prod_{n=1}^{N} P(w_n|\boldsymbol{w}_{<n},\mathcal{M}), \tag{10.5}$$

where $N = |\boldsymbol{w}|$ denotes the number of visual words in $\boldsymbol{w}$, and the subscript $< n$ denotes positions before n. Since $P(\boldsymbol{w}|\mathcal{M})$ is a probability that satisfies $P(\boldsymbol{w}|\mathcal{M}) \leq 1$, it is guaranteed $Pvwp(\boldsymbol{w}|\mathcal{M}) \leq 0$. Since computing perplexity is intractable in general [Wallach et al., 2009], approximate estimation is needed to compute *Pvwp*. We adopt the left-to-right algorithm to estimate *Pvwp*, which is shown to be an accurate and efficient Gibbs sampling method to estimate perplexity [Wallach et al., 2009]. The left-to-right algorithm decomposes $P(\boldsymbol{w}|\mathcal{M})$ in an incremental, left-to-right fashion, as described in Algorithm 6, where the subscript $\neg n$ denotes a quantity that excludes data from the nth position. Given a set of observations $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$, $Pvwp(\mathcal{W}|\mathcal{M})$ is defined as the average of each observation's perplexity:

$$Pvwp(\mathcal{W}|\mathcal{M}) = \frac{1}{M}\sum_{m=1}^{M} Pvwp(\boldsymbol{w}_m|\mathcal{M}). \tag{10.6}$$

Based on $Pvwp$, we define the generalizability indicator I_G over previously unseen observations in the testing phase, using the held-out instances in cross-validation, as follows:

Definition 7 (Generalizability indicator)**.** *Let $\mathcal{M}$ be a trained topic model, $\mathcal{W}_{valid}$ be the validation dataset that is used in the training phase, and $\boldsymbol{w}$ be an previously unseen observation. We define the generalization indicator:*

$$I_G(\boldsymbol{w}) = \begin{cases} \dfrac{\exp(Pvwp(\boldsymbol{w}|\mathcal{M}))}{c \cdot \exp(Pvwp(\mathcal{W}_{valid}|\mathcal{M}))} \\ \quad \textit{if } \exp(Pvwp(\boldsymbol{w}|\mathcal{M})) < c \cdot \exp(Pvwp(\mathcal{W}_{valid}|\mathcal{M})) \\ 1 \quad \textit{if } \exp(Pvwp(\boldsymbol{w}|\mathcal{M})) \geq c \cdot \exp(Pvwp(\mathcal{W}_{valid}|\mathcal{M})) \end{cases} \tag{10.7}$$

where $c \in [1, \infty)$ is a constant representing novelty threshold.

Besides considering the topic model's generalization ability, I_G also evaluates whether previously unseen observations are well-represented by the training set, i.e., whether the training set used to train the topic model is exhaustive. The training set is defined as *exhaustive* when it contains instances from all categories that can possibly be observed in the testing phase [Dundar et al.,]. When some categories are missing and not represented by the training set, it is defined as *non-exhaustive*; in this case, novel categories emerge in the testing phase. Since it is impractical, often impossible, to define an exhaustive training set, mainly because some of the categories may not exist at the time of training, the ability to discover novelty is essential in cognitive modeling for HAR tasks. The indicator I_G provides this ability through evaluating how well new observations are represented by the validation set in the training phase. We constrain I_G's value in the range $(0, 1]$, with a greater value indicating less novelty, which means that the training set is more exhaustive and the topic model generalizes better on an observation. The constant c in Eq. (10.7) provides the flexibility to encode the degree to which we consider an observation to be novel. We set $c = 1$ in this work.

10.4.3 Relationship of I_I and I_G

In human-robot interaction applications, one major objective is to make the modeling recognition result match human common sense as closely as possible. This is captured by the I_I metric, i.e., a better evaluation result with a greater I_I value indicates better recognition performance. However, due to its extrinsic nature, the indicator I_I cannot be directly applied on previously unseen observations without knowledge of ground truth, i.e., a topic model's interpretability cannot be evaluated online during the testing phase.

On the other hand, as an intrinsic metric, the indicator I_G can be computed to evaluate the topic model's generalization ability over new observations. If we can understand the relationship between I_G and I_I, it should be possible to apply the

Table 10.1: Implication of I_I and I_G's values with respect to the novelty of the activity category and the model interpretability. The gray area denotes that the case is generally impossible, as a model will generally never be correct when presented with a novel activity.

	I_G: low	I_G: high
I_I: low	Category is novel Model is *not* applicable	Category is *not* novel Model is *not* well interpreted
I_I: high		Category is *not* novel Model is well interpreted

topic model's generalizability to indicate its interpretability in an online fashion. To empirically analyze this relationship, we conduct extensive experiments, as presented in Section 10.6; we summarize our findings as follows:

Observation 1 (Relationship of I_G and I_I)**.** *Let $\mathcal{W}_{train}$ be the training set used to train a topic model, and I_I and I_G be the model's interpretability and generalizability indicators.*

- *If $\mathcal{W}_{train}$ is exhaustive, then $I_G \to 1$ and I_I is generally independent of I_G.*
- *If $\mathcal{W}_{train}$ is non-exhaustive, then I_G takes values that are much smaller than 1; I_I also takes small values and is moderately to strongly correlated with I_G.*

Observation 1 answers the critical question of whether a more general topic model leads to better recognition performance. Intuitively, if $\mathcal{W}_{train}$ is non-exhaustive and a previously unseen observation $\boldsymbol{w}$ belongs to a novel category, which is indicated by a small I_G value, a topic model trained on $\mathcal{W}_{train}$ cannot accurately classify $\boldsymbol{w}$. On the other hand, if $\boldsymbol{w}$ belongs to a category that is known in $\mathcal{W}_{train}$, then $I_G \to 1$ and the recognition performance over $\boldsymbol{w}$ only depends on the topic model's performance on the validation set used in the learning phase. The implication of the indicators and their relationships are summarized in Table 10.1, where the gray area denotes that it's generally impossible for a topic model to obtain a low generalizability but a high interpretability, as a model will generally never be correct when presented with a novel activity. Given this relationship, we discuss the effectiveness of applying I_G for constructing risk-aware artificial cognitive models that are able to make more reliable decisions in the following section.

10.5 Risk-Aware Decision Making

After human activities are recognized and the recognition result is evaluated, the next task in our human-robot interaction applications is to select a responsive action for the robot to appropriately interact with humans. This task is addressed in the

Table 10.2: Risk levels

Levels	Values	Definition
Low risk	[1,30]	Human may feel unsatisfied with the robot's performance
Medium risk	[31,60]	Human may feel annoyed or upset by the robot's actions
High risk	[61,90]	Human may be interfered with, interrupted, or obstructed
Critical risk	[95,100]	Human may be injured or worse (i.e., a safety risk)

decision making module, which selects a robot action that minimizes the overall risk and sends the action information to the navigation and motor module for the robot to physically execute the action. Our risk-aware decision making algorithm is presented in Algorithm 7.

For each observation obtained from the perception module, the decision making module requires three types of information to select a robot action online: the per-observation activity proportion $\boldsymbol{\theta}$, I_G's value, and the risks of robot actions if taken in response to human activities.

Given the robot action set $\boldsymbol{a} = \{a_1, \ldots, a_S\}$ and the human activity set $\boldsymbol{z} = \{z_1, \ldots, z_K\}$, an action-activity risk r_{ij} is defined as the amount of discomfort, interference, or harm that can be expected to occur during the time period if the robot takes a specific action $a_i, \forall i \in \{1, \ldots, S\}$ in response to an observed human activity $z_j, \forall j \in \{1, \ldots, K\}$. While $\boldsymbol{\theta}$ and I_G are computed online, the risks $\boldsymbol{r} = \{r_{ij}\}_{S \times K}$, with each element $r_{ij} \in [0, 100]$, are manually estimated off-line by domain experts and are used as a prior in the decision making module. In practice, the amount of risk is categorized into a small number of risk levels for simplicity's sake. To assign a value to r_{ij}, a risk level is first selected. Then, a risk value is determined within that risk level. As listed in Table 10.2, we define four risk levels with different risk value ranges in our application. We intentionally leave a five-point gap between critical risk and high risk to increase the separation of critical risk from high risk actions.

A bipartite network $\mathcal{N} = \{\boldsymbol{a}, \boldsymbol{z}, \boldsymbol{r}\}$ can be used to graphically illustrate the risk matrix $\boldsymbol{r}$ of robot actions $\boldsymbol{a}$ associated with human activities $\boldsymbol{z}$. In this network, the vertices are divided into two disjoint sets $\boldsymbol{a}$ and $\boldsymbol{z}$, such that every edge with a weight r_{ij} connects a vertex $a_i \in \boldsymbol{a}$ to a vertex $z_j \in \boldsymbol{z}$. An example of such a bipartite network used in our work is illustrated in Figure 10.2. Given the bipartite network, for each new observation $\boldsymbol{w}$, after $\boldsymbol{\theta}$ and $I_G(\boldsymbol{w})$ are computed in the probabilistic reasoning module, the robot action $a^\star \in \boldsymbol{a}$ is selected in the decision making module according to:

$$a^\star = \underset{a_i : i=1,\ldots,S}{\arg\min} \left(\frac{1 - I_G(\boldsymbol{w})}{K} \cdot \sum_{j=1}^{K} r_{ij} + I_G(\boldsymbol{w}) \cdot \sum_{j=1}^{K} (\theta_j \cdot r_{ij}) \right). \tag{10.8}$$

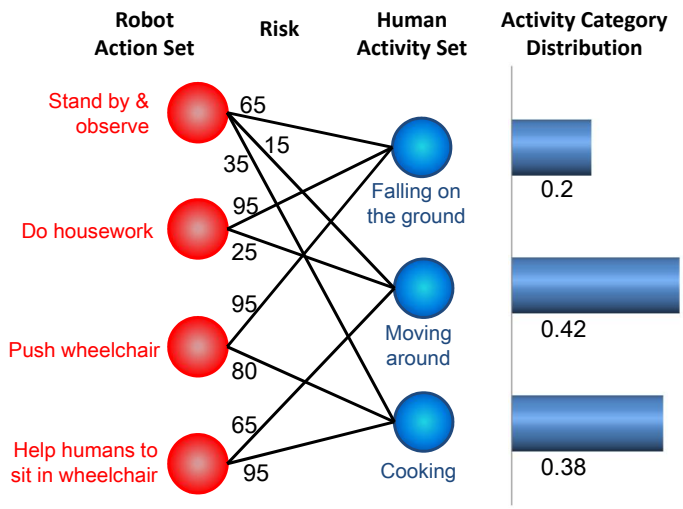


Figure 10.2: An illustrative example of a bipartite network (left) and the human activity distribution of an observation (right).

The risk of taking a specific robot action is determined by two separate risks: activity-independent and activity-dependent action risks. The activity-independent risk (i.e., $\frac{1}{K}\sum_{j=1}^{K} r_{ij}$) measures the inherent risk of an action, which is independent of the human activity context information, i.e., computing this risk does not require the category distribution. For example, the robot action "standing-by" has a smaller risk than "moving backward", in general. One the other hand, the activity-dependent risk (i.e., $\sum_{j=1}^{K}(\theta_j \cdot r_{ij})$) is the average risk weighted by the context-specific information (i.e., the activity distribution). The combination of these two risks is controlled by I_G, which intuitively encodes preference over robot actions. If the topic model well generalizes over $\boldsymbol{w}$, i.e., $I_G(\boldsymbol{w}) \to 1$, the decision making process prefers the robot action that is more appropriate to the recognized human activity. Otherwise, if the model generalizes poorly over $\boldsymbol{w}$, indicating $\boldsymbol{w}$ contains new human activities, our decision making module will ignore the recognition results and prefer the action with lower activity-independent risk.

This use of activity distribution, topic modeling evaluation and action-activity risks allows the robot to make more appropriate decisions, which is critical for constructing a cognitive model for human-robot interaction.

10.6 Empirical Study

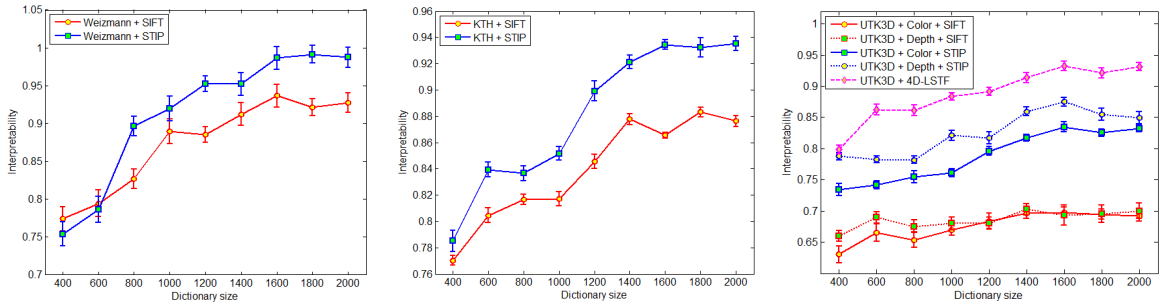
Extensive experiments demonstrate that our artificial cognitive model is capable of achieving promising human activity perception performance and discovering new activities that are not represented by the training set. In addition, we empirically investigate the relationship of the interpretability and generalizability indicators. Finally, we provide qualitative examples that demonstrate our cognitive model's

Algorithm 7: Risk-aware decision making

Input : $\boldsymbol{w}$ (observation), $\mathcal{M}$ (trained topic model), and $\mathcal{N}$ (decision making bipartite network)

Output: $a^\star$ (Selected robot action with minimum risk)

1: Estimate per-observation activity proportion $\boldsymbol{\theta}$ of $\boldsymbol{w}$;
2: Compute generalizability indicator $I_G(\boldsymbol{w})$;
3: **for** *each robot action* $i = 1$ *to* S **do**
4: Estimate activity-independent risk: $r_i^{in} = \frac{1}{K}\sum_{j=1}^{K} r_{ij}$;
5: Calculate activity-dependent risk: $r_i^{de} = \sum_{j=1}^{K}(\theta_j \cdot r_{ij})$;
6: Combine activity-independent and dependent risks, and assign to per-observation action risk vector: $\boldsymbol{r}^a(i) = (1 - I_G(\boldsymbol{w})) \cdot r_i^{in} + I_G(\boldsymbol{w}) \cdot r_i^{de}$;
7: **end**
8: Select optimal robot action $a^\star$ with minimum risk in $\boldsymbol{r}^a$;
9: **return** $a^\star$.



(a) Weizmann **(b)** KTH **(c)** UTK Action3D

Figure 10.3: Variations of our model's interpretability and its standard deviation versus dictionary size using different types of visual features over all of the datasets. Blue lines denote STIP features, and solid lines represent results over color videos.

effectiveness in selecting robot actions to safely and appropriately respond to human activities.

10.6.1 Experimental Setup

We employ three real-world benchmark datasets to evaluate our cognitive model on HAR tasks, which are widely used in the machine vision community: the Weizmann activity dataset [Gorelick et al., 2007], the KTH activity dataset [Laptev, 2005], and the UTK Action3D dataset [Zhang and Parker, 2011].

In our experiments, we apply different types of local visual features to encode these datasets. For 2D datasets that contain only color videos (i.e., the Weizmann and KTH datasets), we employ two different features: scale-invariant feature transform (SIFT)

features [Lowe, 2004] and space-time interest points (STIP) features [Laptev, 2005]. For 3D datasets that contain both color and depth videos (i.e., the UTK Action3D dataset), we adopt the 4-dimensional local spatio-temporal features (4D-LSTF) [Zhang and Parker, 2011].

SIFT features are the most commonly applied local visual features and have desirable characteristics including invariance to transformation, rotation and scale, and robustness to partial occlusion [Lowe, 2004]. We employ the algorithm and implementation in [Lowe, 2004] to detect and describe SIFT features. A disadvantage of SIFT features in HAR tasks is that these features are extracted in a frame-by-frame fashion, i.e., SIFT features do not capture any temporal information. To encode time information, we also apply STIP along with the histogram of oriented gradients (HOG) and histogram of optical flow (HOF) descriptors [Laptev, 2005]. These two types of features are extracted using only color or intensity information. Previous work has demonstrated that local features incorporating both depth and color information can greatly improve recognition accuracy [Zhang and Parker, 2011]. Therefore, for the UTK Action3D dataset we use 4D-LSTF [Zhang and Parker, 2011] features, which are highly robust and distinct and are generated using both color and depth videos. It is also noteworthy that SIFT and STIP features can be directly extracted from color or depth videos in the 3D dataset.

These feature extraction algorithms generate a collection of feature vectors for each visual observation. Then, the feature vectors are clustered into discrete visual words using the k-means algorithm, and the number of clusters is set equal to the dictionary size. Lastly, each feature vector is indexed by a discrete word that represents cluster assignment. At this point, each observation is encoded by a BoW representation, which can be perceived by topic modeling. Although we only test the most widely used features, one should note that our artificial cognitive model is capable of incorporating different types of local visual features, since our reasoning and decision making process is independent of the features given their BoW representation.

10.6.2 Activity Recognition

We empirically validate that our artificial cognitive system achieves promising human activity recognition performance in terms of interpretability. After using topic mapping to associate the recognized clusters with the known activity categories, we evaluate model interpretability using the proposed I_I indicator. We analyze variations of topic model's interpretability versus dictionary size, and also show that the model's interpretability varies, when the model is applied to recognize different human activities.

Exhaustive experimental setup: We employ the all-in-one experimental setup [Schuldt et al., 2004], in which models are trained and tested using data from all scenarios. Each dataset is split into disjoint training and testing sets. We randomly select 25% of the instances in each category as the testing set, and place the rest of the instances in the training set. Using this splitting method, we create an

exhaustive training set. In the learning process, the training set is further divided into training and validation sets, and four-fold cross-validation is used to estimate model parameters. Then, during the testing phase, the trained model's interpretability is computed using the testing set, which does not contain any new activities and is well represented by the training set. We repeat this learning-testing process five times to obtain reliable results.

Results: Experimental results of our model's interpretability and the standard deviation versus dictionary size are graphically presented in Figure 10.3, using different types of visual features over different datasets. From this figure, we can make several important observations. First, our model obtains promising recognition performance in terms of interpretability. For the Weizmann dataset, we obtain the best interpretability of 0.989 using STIP features and a dictionary of size 1800. For the KTH dataset, we obtain the best interpretability of 0.952 using STIP features and a dictionary of size 2000. For the UTK Action3D dataset, we obtain the best interpretability of 0.936 using 4D-LSTF features and a dictionary size of 1600. The high interpretability values indicate that our modeling process closely matches human common sense, which is a desirable characteristic for an artificial cognitive system in HRI applications. Second, STIP features perform better than SIFT features in most cases. STIP features over depth videos lead to better interpretability than STIP features over color videos. When depth information is available, 4D-LSTF features that incorporate both depth and color information result in the best interpretability. Third, in general, a large dictionary results in better interpretability, with diminishing returns once a dictionary size of 1500 is reached. Fourth, our model achieves very consistent recognition results, which can be seen by the small errors in each dataset. This also demonstrates our interpretability indicator's consistency. Last, one may note that SIFT features result in much worse interpretability in the UTK Action3D dataset, as shown in Figure 10.3c. This is because SIFT features are extracted in a frame-by-frame fashion without considering the relationship of temporally adjacent frames; and the UTK Action3D dataset contains sequential activities that are performed with a sequence of motions, including "lifting" and "removing", which cannot be recognized using only a single frame. For example, only from single frames of "lifting" and "removing", we cannot distinguish one activity from the other.

We also investigate our model's interpretability over different activities. The experimental results for the UTK Action3D dataset, which includes more complex activities (i.e., sequential activities) and contains more information (i.e., depth), are depicted in Figure 10.4. The dictionary size in this experiment is set to 1600. From this experiment, we conclude that topic model's interpretability varies over different activities. Using the other two datasets, we obtain similar conclusions. This interpretability difference is caused by three major factors: the topic model's modeling capability, each feature's representability to encode an observation, and the complexity and similarity of the activities. To demonstrate how well our framework generalizes to construct a reliable artificial cognitive system, we intentionally select

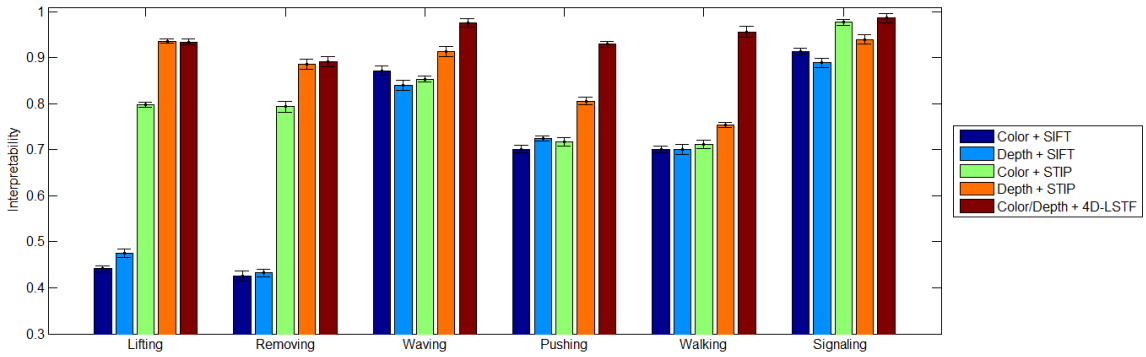


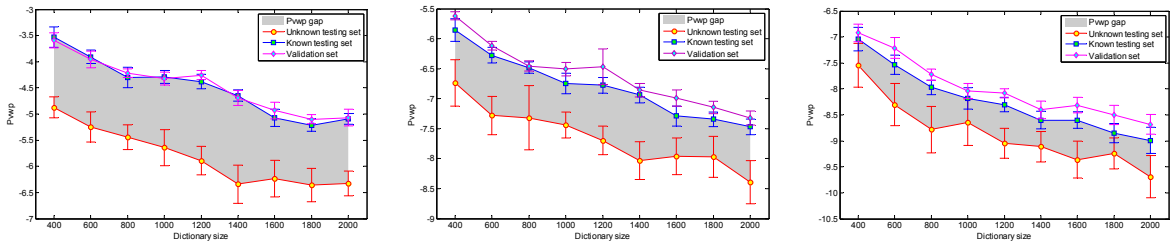
Figure 10.4: Interpretability of the activities in UTK Action3D dataset represented using different types of features. A higher interpretability indicates that the model more closely matches human common sense.

the most basic type of topic models (i.e., LDA), which is not capable of modeling time. This explains why the activities "lifting" and "removing" in Figure 10.4 result in much worse interpretability using SIFT features. In addition, Figure 10.4 indicates the importance of the feature's representability. For instance, while features ignoring time information (e.g., SIFT) lead to bad interpretability over the sequential activities, features representing additional depth information (e.g., 4D-LSTF) improve interpretability over most of the activities. In general, features capturing more information result in better interpretability. Lastly, our model's interpretability depends on the activity's complexity and its similarity to other activities. For example, since sequential activities are more complex than repetitive activities (e.g., "waving"), they generally result in lower interpretability. In another example, since "pushing" and "walking" are similar, which share similar motions such as moving forward, they generally lead to lower interpretability.

10.6.3 Knowledge Discovery

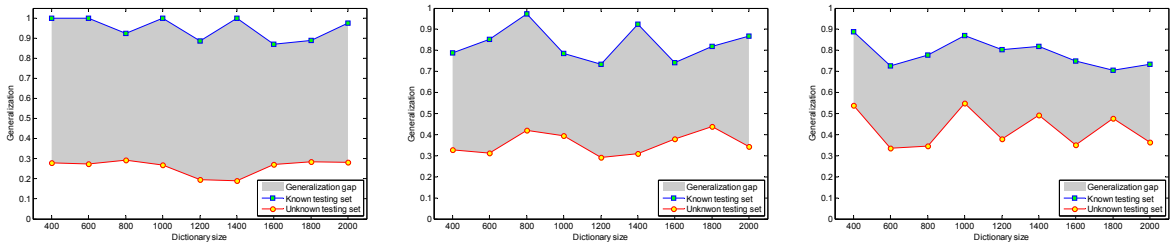
Further experiments show that our artificial cognitive model is capable of discovering new knowledge, i.e., new activities that are not considered in the training phase can be automatically detected. Knowledge discovery is achieved through the proposed I_G as discussed in Section 10.4.2. In this subsection, we analyze variations of $Pvwp$ and I_G versus dictionary size for each dataset. Then, we investigate variations of model generalizability versus percentage of overlapping features using a synthetic dataset.

Non-exhaustive experimental setup: Each dataset is divided into training and testing sets in a non-exhaustive fashion as follows. Each experiment is performed using F folds, where F is the number of activities in a dataset. In each fold, we place all instances of one activity in the *unknown testing set*. In addition, we randomly select 25% of the instances of the remaining activities in the *known testing set*. The rest of the instances are placed in the training set to estimate our model's parameters,



(a) Weizmann + STIP **(b)** KTH + STIP **(c)** UTK Action3D + 4D-LSTF

Figure 10.5: Variations of topic modeling's *Pvwp* versus dictionary size over validation set, known and unknown testing sets.



(a) Weizmann **(b)** KTH **(c)** UTK Action3D

Figure 10.6: Variations of our model's generalizability versus dictionary size over known and unknown testing sets for all datasets.

which is further divided into training and validation sets to perform four-fold cross-validation. This experimental setup is non-exhaustive, because it contains two testing sets: 1) the unknown testing set contains instances from a new activity that is not contained in the training set, and 2) the known testing set contains instances from activities fully represented by the training set. In this experimental setup, we use visual features that achieve the best model interpretability over each dataset as found in Section 10.6.2, i.e., we adopt STIP features for the Weizmann and KTH datasets, and 4D-LSTF features for the UTK Action3D dataset.

Results: We investigate the variation of *Pvwp* versus dictionary size for all of the datasets over the validation set, known testing set, and unknown testing set. Several important observations can be made for the results plotted in Figure 10.5. First, for a fixed dictionary size, there exists a large *Pvwp* gap between the known and unknown testing sets, as illustrated by the gray area in Figure 10.5. This indicates that topic models generate differently over instances from known or unknown activities. A better generalization result generally indicates a less novel instance, which is better represented by the training set. This also explains the small *Pvwp* gap between the known testing set and the validation set, since instances from both sets are well represented by the training set. It is noteworthy that the known testing set's *Pvwp* value can be greater than the *Pvwp* value of the validation set, if its instances are better represented by the training set, as shown in Figure 10.5a. Second, the

gap's width varies over different datasets. In our experiments, the Weizmann dataset generally has the largest $Pvwp$ gap, followed by the KTH dataset, and the UTK Action3D dataset has the smallest gap. The gap's width mainly depends on the instance's novelty in terms of the portion of overlapping features. A more novel activity is generally represented by a set of more distinct features with less overlapping with the features existing in the training set, and an instance of this activity generally leads to a larger gap. For example, activities in the Weizamann dataset share fewer motions and thus contain fewer overlapping features, which leads to a larger gap. Third, when the dictionary size increases, topic model's $Pvwp$ values decrease at a similar rate. At the same time, because the number of visual words that are used to represent each activity also increases, the probability that a word appears in each activity category (i.e., φ_w) decreases, which results in a decrease of $P(\boldsymbol{w}|\mathcal{M})$. Thus, $Pvwp$ also decreases, in general. Fourth, $Pvwp$ over unknown testing sets has larger standard deviations as shown by the error bars in Figure 10.5. This indicates that activities with different novelty result in different $Pvwp$ values. A more novel activity with less overlapping features generally results in a smaller $Pvwp$ value. For example, we observe "jogging" is less novel than "hand-clapping" in the KTH dataset, since "jogging" shares more features with other activities such as "running".

We also empirically analyze I_G's characteristics over known and unknown testing sets for all of the datasets using features that achieve the best interpretability. The results are graphically presented in Figure 10.6. An important characteristic of I_G is its invariance to dictionary size. Since $Pvwp$ over testing and validation sets has similar decreasing rate, the division operation in Eq. (10.7) removes the variance to dictionary size. In addition, an instance of a more novel activity generally leads to a smaller I_G value. For example, the Weizmann dataset has the smallest I_G values over the unknown testing set, since its activities are more novel in the sense that they share less overlapping features. In general, I_G is smaller than 0.5 for unknown activities and greater than 0.7 for activities that are included in training sets. Last but not least, similar to $Pvwp$, there exists a large gap between the I_G values over unknown and known testing sets, as indicated by the gray area in Figure 10.6. The average I_G gap across different dictionary sizes is 0.69 for the Weizmann dataset, 0.48 for the KTH dataset, and 0.36 for the UTK Action3D dataset. The well separated values demonstrate I_G's applicability and effectiveness to discover new knowledge that is not captured in the training set.

We have pointed out that the indicator I_G is heavily affected by the novelty of an activity in terms of its proportion of overlapping features. To validate this conclusion, we generate a synthetic dataset by manually controlling the proportion of overlapping visual words in the testing instances. In order to make the characteristics of the synthetic dataset as close as possible to real-world datasets, features used in the simulation are borrowed from the KTH dataset. Instances of two activities (i.e., "bending" and "waving2") are used to train a topic model, which is then applied as a classifier to perform recognition in this experiment. This topic model is also applied to generate overlapping visual words for a testing instance. Another topic model, whose

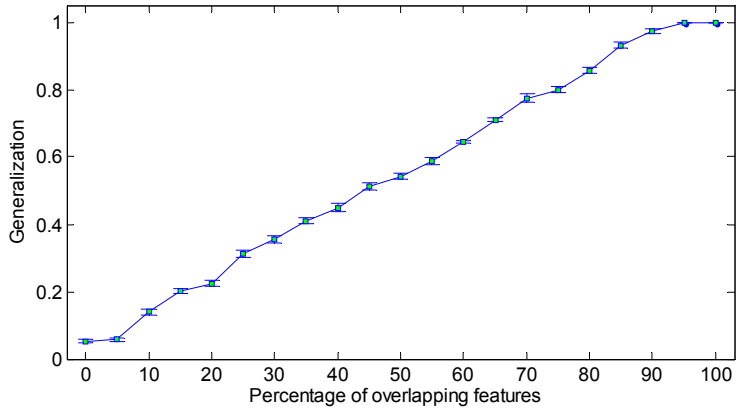


Figure 10.7: Model's generalizability variations versus percentage of overlapping features in synthetic data.

parameters are learned using activities "siding" and "jacking", is used to generate non-overlapping words in the testing instance. A dictionary of size 1800 is adopted, which is created using the visual words of the KTH dataset. The used four activities contain 906 unique visual words, with each pair of activities sharing less than 1% overlapping words. We generate 50 instances for each testing set, with the number of words in each instance set to 112, which is the average number of visual words in real-world instances. We present the results of five simulations in Figure 10.7, which clearly shows that, in general, I_G's value over testing instances increases linearly with the percentage of features that overlap with the features of known activities in the training set.

10.6.4 Relationship of I_G and I_I

Here, we empirically analyze the relationship between the interpretability and generalizablity indicators. We first validate the correlation of I_I and I_G, as introduced by Observation 1. In addition, we investigate additional relationships of I_I and I_G, such as the probability that $I_I \leq I_G$.

While we are able to employ the exhaustive experimental setup from Section 10.6.2 to analyze I_G and I_I's relationship when testing instances are fully represented by the training set, unfortunately, we cannot use the non-exhaustive setup in Section 10.6.3 to validate this relationship in cases where I_G takes small values. This is because ground truth cannot be assigned to instances belonging to novel activities to compute I_I, since these activities only exist in the testing set and are not presented to our model during the training phase. Inspired by the method used to generate synthetic data in Section 10.6.3, we adopt a semi-exhaustive experimental setup by replacing certain portions of words in each testing instance with visual words from novel activities. This experimental setup is used to validate the indicators' relationship when the training set cannot fully represent testing instances.

Table 10.3: Relationship between I_I and I_G over exhaustive and semi-exhaustive datasets.

Dataset + Features	Exhaustive		Semi-exhaustive	
	$\rho_{I,G}$	$P_{I_I \leq I_G}$	$\rho_{I,G}$	$P_{I_I \leq I_G}$
Weizmann + STIP	-0.065	0.456	0.664	0.912
KTH + STIP	0.036	0.324	0.685	0.853
UTK Action3D + 4D-LSTF	0.097	0.275	0.714	0.896

Semi-exhaustive experimental setup: Each experiment is performed using F folds, where F is the number of activities in a dataset. In each fold, we take all instances of one activity out from the dataset, which is treated as a novel activity that is not presented to the topic model in the learning phase. Then, we randomly select 75% of the instances of the remaining activities as training set, which is further divided into training and validation sets to perform four-fold cross-validation. The rest of the instances are used as an "initial" testing set. During the testing phase, the novel activity's word distribution is used to generate new visual words to replace a proportion of the words in each instance in the initial testing set. This testing is performed six times within each of the F folds using different replacement rates (i.e., 0.25, 0.35, ..., 0.75). Testing results from all F folds are used to investigate I_I and I_G's relationship. In this experimental setup, we use features that achieve the best interpretability over each dataset. In addition, we set the dictionary size to 1600, which achieves the best interpretability over all datasets in general.

This experimental setup is semi-exhaustive in the sense that, although training data cannot fully represent testing instances due to the replaced features that are generated from unknown activities, the remaining non-replaced features are presented to the model during the learning phase, and the ground truth assigned to each testing instance remains the same, which is also known to the model. It is noteworthy that we do not use very high or very low replacement rates. A very low replacement rate makes the experimental setup equivalent to the exhaustive setup. When using a very high replacement rate, testing instances can be viewed as being drawn from the novel activity; in this case the ground truth associated with a testing instance would be meaningless or incorrect.

Results: We empirically analyze the correlation between I_I and I_G, using both exhaustive and semi-exhaustive datasets, in order to determine whether better generalizability indicates better interpretability. The Pearson correlation coefficient is used to measure the strength and direction of the linear relationship between these two indicators. Given a dataset $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|\mathcal{W}|}\}$ and its ground truth $\boldsymbol{g} = \{g_1, \ldots, g_{|\mathcal{W}|}\}$, this correlation is mathematically defined as follows:

$$\rho_{I,G} = \frac{E[(\boldsymbol{I}_I - \mu_{I_I})(\boldsymbol{I}_G - \mu_{I_G})]}{\sigma_{I_I}\sigma_{I_G}}, \tag{10.9}$$

where $\boldsymbol{I}_G = \{I_G(\boldsymbol{w}_1), \ldots, I_G(\boldsymbol{w}_{|\mathcal{W}|})\}$ and $\boldsymbol{I}_I = \{I_I(\boldsymbol{w}_1, g_1), \ \ldots, I_I(\boldsymbol{w}_{|\mathcal{W}|}, g_{|\mathcal{W}|})\}$ are vectors of interpretability and generalizability indicators for all of the instances in the dataset, μ is the mean and σ is the standard deviation of the indicators in the vector. Our experimental results are listed in Table 10.3. For the exhaustive dataset, topic models which perform better on generalizability are not necessarily better interpreted, which is indicated by the weak linear correlation between the indicators. This is because, when testing on exhaustive datasets, I_G takes values closer to 1. But the model's interpretability takes a wide range of values, depending on the model's modeling capacity, feature representability and dataset complexity, as explained in Section 10.6.2. For semi-exhaustive datasets, I_I and I_G are moderately to strongly correlated, which indicates that a poor generalizability usually leads to a poor interpretability. Since I_G reflects the novelty of an instance as discussed in Section 10.6.3, a low I_G's value means the instance is badly represented by the training set. Therefore, the trained model cannot obtain a good interpretability over the instance of an activity that is not well represented during the training phase.

We also check an additional relationship, i.e., the probability that I_I is smaller than or equal to I_G. Given a labeled dataset $\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$ and its ground truth $\boldsymbol{g} = \{g_1, \ldots, g_M\}$, this probability is defined as follows:

$$P_{I_I \leq I_G} = \frac{1}{M} \sum_{m=1}^{M} \mathbb{1}(I_I(\boldsymbol{w}_m, g_m) \leq I_G(\boldsymbol{w}_m)). \tag{10.10}$$

The experimental results are presented in Table 10.3. One of the most important observations is that, for a majority of testing instances (more than 85%) in the semi-exhaustive experiment, I_G's value is greater than I_I's value. This again shows that a poor generalizability usually indicates a poor interpretability. Using the exhaustive experimental setup, it is more probable that I_G takes smaller values than I_I. This is because when the training set is exhaustive, the topic model is well trained and can well recognize testing instances, which leads to $I_I \rightarrow 1$ for most of testing instances. On the other hand, although I_G also takes a large value in general, it is usually slightly smaller than one, because features in testing instances usually do not completely overlap with features in training instances.

10.6.5 Case Study of Decision Making

We use several concrete examples to demonstrate the importance of computing the activity distribution and evaluating topic model's performance in order to make better decisions and select more appropriate robot actions. In these examples, we assume a mobile robot is helping the disabled in everyday life in the home environment. The bipartite network $\mathcal{N} = \{\boldsymbol{a}, \boldsymbol{z}, \boldsymbol{r}\}$ that is used in the examples is depicted in Figure 10.2. The number on an edge indicates the risk of a specific robot action (on the edge's left end) for a specific human activity (on the edge's right end), and no edge means the risk has a value of zero. For example, the risk of the robot action "push

wheelchair" for the human activity "cooking" is 80, indicating that this action may interrupt the human if the human is currently cooking. The distribution over human activities is also graphically presented in Figure 10.2. We discuss the following two case studies.

Demonstration 1 – *Distribution over activity categories is important for decision making*: Assume $\boldsymbol{\theta} = \{0.2, 0.42, 0.38\}$ and $I_G = 1$, that is, the model identifies two activities with similarly high probabilities and this model perfectly generalizes over an observation. Let's consider the situation that the final decision is made only based on a single activity without considering the activity distribution. Since the human activity "moving around" has the highest probability, the robot action "push wheelchair" will be selected, because it is the most appropriate action with no risk for this activity. However, this decision is not optimal. It is quite possible that the correct activity is "cooking", since it also has a high probability that is similar to the activity "moving around". In this case, "push wheelchair" will cause a high risk of value 80. On the other hand, when considering the distribution over all possible activities, the decision making process will select the action "do housework" according to Eq. (10.8) with an average risk of 29.5, which is much smaller than the average risk of 49.4 for the robot action "push wheelchair".

Demonstration 2 – *Model evaluation is important for decision making*: Let's assume $\boldsymbol{\theta} = \{0.1, 0.8, 0.1\}$, that is, the human activity "moving around" dominates the distribution. In the case of $I_G = 0.9$, "push wheelchair" has the lowest average risk of 21.58 among all possible robot actions. Since the model well generalizes over the observation, the decision making system is confident about the distribution. Thus, the best robot action "push wheelchair" can be safely selected, even if it has high risks for the other human activities. On the other hand, if $I_G = 0.1$, i.e., the model generates badly on the observation, the average risk of "push wheelchair" increases significantly to 54.25. Although this robot action has no risk to the most probable activity, it is still not selected by our decision making module, because "stand by & observe" leads to the lowest average risk of 36.7, which also has the lowest activity-independent risk (38.33) among all possible actions. This result indicates that our decision making module prefers robot actions with low risks when the model badly generalizes over an observation. This preference is achieved by using I_G to control activity-dependent and activity-independent risks, which demonstrates the importance of online model evaluation in the decision making process in the construction of reliable artificial cognitive models.

10.7 Summary

In this chapter, we construct an artificial cognitive model that provides these crucial capabilities: accurate perception and the ability to discover new information, in order to enable safe, reliable robot decision making for the HAR task in human-robot interaction applications.

We use topic models, specifically LDA, to create our cognitive model. Topic models are particularly suited to this purpose because they are unsupervised, and allow for the discovery of new knowledge not represented in training. Using topic modeling also allows us to treat activity estimation as a distribution and incorporate risk per action response, which allows our system to make better decisions. In order to provide the capability of accurately interpreting human activities, we define a new metric called the interpretability indicator (I_I) and demonstrate, as an extension of accuracy, its ability to measure how well a robot's interpretation matches a human's. The indicator I_I is applied to map detected clusters to known activity categories, and to select the best interpreted model. In addition, to provide for the ability of knowledge discovery, we introduce a novel metric, named the generalizability indicator (I_G), which we define as the model's generalization capacity, or how well a new observation is represented by a previously learned model. I_G can be constructively applied to estimate modeling performance online in order to make an appropriate decision instead of simply trusting the model.

We use our new artificial cognitive model in extensive experiments conducted to demonstrate our model's effectiveness using both synthetic and real-world datasets. We show that our model performs extremely well in terms of interpretability; that is, our model's recognition results closely and consistently match human common sense. We demonstrate that, using I_G, our cognitive model is capable of discovering new knowledge, i.e., observations from new activity categories that are not considered in the training phase can be automatically detected. We examine the relationship between I_I and I_G and show, both analytically and experimentally, that I_G can also be used as an indicator for I_I. We show that, with high confidence, scenarios with a low I_G score for an observation will equate to a low I_I score, i.e., a badly generalized model is likely to be inaccurate. We further demonstrate the advantage of using distributions over activity categories, as well as the importance of the evaluation metrics in order to create a system capable of safe, reliable decision making.

Our findings show that using topic modeling together with the incorporation of our new metrics allows us to construct more reliable artificial cognitive models for the HAR task in human-robot interaction applications. We plan to deploy our topic modeling based, risk-aware artificial cognitive model on the Meka Robotics M1 robot in our future work.

Chapter 11

Conclusions

This dissertation focuses on the research problem of robotic sensing of people, i.e., human perception, representation and activity recognition, mainly using 3D vision in human-centered robotics applications. To conclude this dissertation, I summarize the key contributions and also indicate several interesting directions for future research in this field.

11.1 Key Contributions

Real-Time Multiple Human Perception

Efficient and robust detection and tracking of people in complicated human social environments is critical in human-centered robotics to safe operation and effective robot interaction with humans. Previous studies generally follow a sliding window paradigm, which applies dense multi-scale scanning over the entire image to localize humans. This paradigm has a high computational complexity and is thus not suitable for the real-time requirement in human-centered robotics applications.

I introduced a real-time multiple human perception system in dynamic indoor environments [Zhang et al., 2013], using a mobile robot equipped with a color-depth camera (e.g., Kinect). The key component of this system is the novel concept of Depth of Interest, which is used to identify candidates for detection thereby avoiding the computationally expensive sliding window paradigm of previous approaches. The system achieves a processing rate of 7–15 frames per second and is able to address occlusion, robot movement, non-upright humans, humans leaving and re-entering the field of view (i.e., re-identification challenge), and human-object and human-human interaction.

Spatio-Temporal Features for Human Representation

One focus of my dissertation has been the development of discriminative local spatio-temporal (LST) features, which are currently the most popular and promising visual

representation. These features are generally invariant to geometric transformations; as a result, they are less affected by variations in scale, rotation and viewpoint. Although several color-based LST features have been developed, they do not make use of one important piece of information—depth. Because humans act in the 3D physical world, depth can be applied along with color information to develop more descriptive LST features. My dissertation focuses on designing and implementing LST features using both color and depth information, described as follows:

- 4-dimensional *Co*lor-*De*pth (CoDe4D) feature: I proposed the first LST feature that combines both color and depth information in a sequence of 3D point clouds [Zhang and Parker, 2011]. The feature is detected as the local maximum in the response image computed by applying separate filters along 3D spatial dimensions and 1D temporal dimension. A multi-channel feature descriptor was also introduced to incorporate both color and depth cues in the final feature vector. The CoDe4D feature demonstrates promising ability to represent not only repetitive but also sequential activities and activities with small motions.

- *Ad*aptive *Hu*man-*C*entered (AdHuC) feature: Previous features are generally incapable of representing activities of multiple individuals within a group, since such features ignore global spatial structure information and lack affiliation information. I introduced a novel algorithm to detect human-centered features through constructing an affiliation region for each human and detecting local features within the region, which also avoids extracting irrelevant features from backgrounds and deals with robot/camera movements. I also introduced a new descriptor that adapts its support region size to linear perspective view changes and incorporates color-depth information. The AdHuC feature addresses the challenging and previously not well studied task of *action recognition of multiple individuals* within a group.

- *S*implex-based *O*rientation *D*escription (SOD) of 3D features: Although a large number of 3D features have been developed that are computed in either xyt space (e.g., LST features) or xyz space, methods to describe 3D visual feature's orientations are very limited; they are based on either spherical coordinates or regular polyhedrons. Spherical coordinate descriptor suffers from the singularity issue at the poles, while regular polyhedron methods have limited discrimination power due to the limited number of regular polyhedrons. I introduced a novel, efficient feature descriptor that avoids both issues through decomposing 3D visual cues' orientations into three correlated angles and quantizing them in the 2-simplex topological vector space [Zhang et al., 2014b].

Graphical Models for Human Activity Interpretation

My Ph.D. research addresses a range of problems relating to the analysis of human activities applied in or arising from human-centered robotics, with a primary focus

on continuous, sequential activity analysis and decision making. Major contributions of my research in these areas are described below:

- Maximum temporal certainty models for sequential activity recognition: Predicting sequential human activities (e.g., sitting down and standing up) requires modeling their underlying temporal patterns. I introduced the novel Maximum-Certainty Hidden Conditional Random Field (MC-HCRF), which is an efficient discriminative graphical model that aims to maximize both the probability of the correct class and the certainty in latent temporal patterns. I mathematically proved that inference of our model is tractable and provided an efficient learning algorithm under the energy-based learning framework. This is the first to model certainty in the latent temporal pattern and formulate HCRFs as energy-based models.

- Fuzzy segmentation and recognition of continuous activities: Most previous work has focused on classifying single human activities contained in segmented videos. However, in real-world applications, activities are inherently continuous and gradual transitions always exist between temporally adjacent activities. I proposed the new fuzzy temporal segmentation and probabilistic recognition (FuzzySR) algorithm [Zhang et al., 2014a] that uses a topic model, i.e., Latent Dirichlet Allocation (LDA), to summarize the activity distribution at each time interval and applies temporal fuzzy clustering to discover and segment events. FuzzySR is the first work to explicitly model gradual transitions between human activities in continuous visual data.

- Unsupervised analysis of human activities and cognition for decision making: Although Bayesian topic models (e.g., LDA) can avoid overfitting, they have a high computational complexity that limits their application in time-constrained human-centered robotics. I proposed a topic model that combines Probabilistic Latent Semantic Analysis with Gaussian Mixture Model to address overfitting [Zhang et al., 2012a]. I also introduced an efficient expectation-maximization algorithm to learn model parameters in an incremental fashion. Besides using topic models to discover activity patterns directly from data, I applied these models to construct risk-aware artificial cognitive systems for decision making.

11.2 Future Directions

Active Semantic Perception

With the emergence of the inexpensive and accurate color-depth sensors, we have witnessed an explosion of new techniques using 3D information to construct robust robotic perception systems. However, a common assumption present in existing work is that the query object or human is located within the immediate sensory reach of a robot, which is not true in large-scale human social environments. In addition, since

3D perception in the physical world is by its nature a *big data* problem, where large streams of information must be acquired and managed, an exhaustive search over the entire environment becomes infeasible. To this end, an interesting future direction is to formulate robotic perception as an active process that seeks semantic cues actively beyond the sensory horizon. It has great potentials to expand the perception system discussed in this dissertation to extract sematic information, update the information during the whole lifetime of a robot, and deal with search queries in large, dynamic human social environments.

Robotics Learning of Social Norms

Social norms are unwritten rules that define appropriate behaviors for a social group. Human-centered robotic systems can significantly benefit from understanding social norms, since they provide a robotic system with an expectation of how humans behave in a particular social environment, and therefore can be applied to improve behavior recognition accuracy. For example, a person screaming alone in one's house is quite different than screaming while attending a football game. In addition, social norms provide instructions for developing a robot's behaviors in a way that conforms to people's expectations, e.g., a robot should not cut in line. To address this important, open problem, an interesting research direction is to expand my activity recognition algorithms to identify collective activities, and combine the results with perceived semantics that encode contextual information of human social environments, in order to reason about social norms for social groups.

Lifelong Adaptation

Because human social environments are highly dynamic (as are people), it is widely accepted that lifelong adaptation, to not only the environment but also the humans within it, is a critical capability for human-centered robotic systems. I would like to point out two open issues in lifelong adaptation, as follows:

- Transfer learning with social context awareness: Different from previous work that focuses on knowledge transfer between heterogeneous robots, an interesting topic is to investigate how social context affects knowledge transferability by incorporating perceived semantics into the knowledge transfer process.
- Self-motivated learning: rather than learning only when a human is around to teach, an intelligent robot should be self-motivated to learn general world knowledge from information stored in Internet databases, social networking services, and through *crowdsourcing*.

Bibliography

[Abonyi et al., 2005] Abonyi, J., Feil, B., Nemeth, S., and Arva, P. (2005). Modified Gath–Geva clustering for fuzzy segmentation of multivariate time-series. *Fuzzy Sets Systems*, 149(1):39–56. 127, 135, 136, 137

[Aggarwal and Ryoo, 2011] Aggarwal, J. and Ryoo, M. (2011). Human activity analysis: A review. *ACM Computing Surveys*, 43(3):16:1–16:43. 41, 100, 110, 125

[Ahrendt et al., 2005] Ahrendt, P., Goutte, C., and Larsen, J. (2005). Co-occurrence models in music genre classification. In *IEEE International Workshop on Machine Learning for Signal Processing*. 99

[Al Ghamdi et al., 2012] Al Ghamdi, M., Zhang, L., and Gotoh, Y. (2012). Spatio-temporal sift and its application to human action classification. In *European Conference on Computer Vision*. 65, 68, 77, 79, 80

[Alahi et al., 2012] Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. 154

[Anderson et al., 2010] Anderson, D., Luke, R., and Keller, J. (2010). Segmentation and linguistic summarization of voxel environments using stereo vision and genetic algorithms. In *IEEE International Conference on Fuzzy Systems*. 131

[Anderson, 1996] Anderson, J. R. (1996). ACT: A simple theory of complex cognition. *American Psychologist*, 51:355–365. 156

[Arulampalam et al., 2002] Arulampalam, M. S., Maskell, S., and Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188. 18

[Asuncion et al., 2009] Asuncion, A., Welling, M., Smyth, P., and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Conference on Uncertainty in Artificial Intelligence*. 99

[Banerjee et al., 2014] Banerjee, T., Keller, J., Skubic, M., and Stone, E. (2014). Day or night activity recognition from video using fuzzy clustering techniques. *IEEE Transactions on Fuzzy Systems*, 22(3):483–493. 131

[Ben-Arie et al., 2002] Ben-Arie, J., Wang, Z., Pandit, P., and Rajaram, S. (2002). Human activity recognition using multidimensional indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1091–1104. 42, 129

[Benjamin et al., 2004] Benjamin, D. P., Lyons, D., and Lonsdale, D. (2004). ADAPT: A cognitive architecture for robotics. In *International Conference of Cognitive Modeling*. 156

[Bernardin and Stiefelhagen, 2008] Bernardin, K. and Stiefelhagen, R. (2008). Evaluating multiple object tracking performance: the CLEAR MOT metrics. *Journal of Image Video Processing*, 2008:1:1–1:10. 35

[Blank et al., 2005] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *International Conference on Computer Vision*. 7

[Blei and Lafferty, 2006] Blei, D. and Lafferty, J. (2006). Correlated topic models. In *Neural Information Processing Systems*. 162

[Blei, 2012] Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84. 156

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022. 56, 127, 132, 133, 154, 158, 159, 160

[Bloom et al., 2012] Bloom, V., Makris, D., and Argyriou, V. (2012). G3D: A gaming action dataset and real time action recognition evaluation framework. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 49

[Borges et al., 2013] Borges, P., Conci, N., and Cavallaro, A. (2013). Video-based human behavior understanding: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(11):1993–2008. 41, 125

[Bourdev and Malik, 2009] Bourdev, L. and Malik, J. (2009). Poselets: body part detectors trained using 3D human pose annotations. In *International Conference on Computer Vision*. 18, 27, 30, 31, 93

[Brand, 1999] Brand, M. (1999). Structure and parameter learning via entropy minimization, with applications to mixture and hidden Markov models. In *International Conference on Acoustics, Speech and Signal Processing*. 111

[Brdiczka et al., 2009] Brdiczka, O., Langet, M., Maisonnasse, J., and Crowley, J. (2009). Detecting human behavior models from multimodal observation in a smart home. *IEEE Transactions on Automation Science and Engineering*, 6(4):588–597. 42

[Breitenstein et al., 2011] Breitenstein, M., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. (2011). Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833. 18

[Briers et al., 2010] Briers, M., Doucet, A., and Maskell, S. (2010). Smoothing algorithms for state-space models. *Annals of the Institute of Statistical Mathematics*, 62(1):61–89. 88

[Brown et al., 2012] Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(1):27–66. 118

[Burghart et al., 2005] Burghart, C., Mikut, R., Stiefelhagen, R., Asfour, T., Holzapfel, H., Steinhaus, P., and Dillmann, R. (2005). A cognitive architecture for a humanoid robot: a first approach. In *IEEE-RAS International Conference on Humanoid Robots*, pages 357–362. 156

[Buxton, 2002] Buxton, H. (2002). Generative models for learning and understanding dynamic scene activity. In *European Conference on Computer Vision Workshop*, pages 71–81. 157

[Cai and Cai, 2006] Cai, Y. and Cai, C. Y. (2006). Robust visual tracking for multiple targets. *European Conference on Computer Vision*. 18

[Chakraborty et al., 2012] Chakraborty, B., Holte, M. B., Moeslund, T. B., and González, J. (2012). Selective spatio-temporal interest points. *Computer Vision and Image Understanding*, 116(3):396–410. 39, 42, 82, 84, 111, 119, 120, 144

[Chang and Lin, 2011] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. 53

[Chang et al., 2009a] Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., and Blei, D. (2009a). Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*. 156

[Chang et al., 2009b] Chang, K.-Y., Liu, T.-L., and Lai, S.-H. (2009b). Learning partially-observed hidden conditional random fields for facial expression recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 206

[Chater et al., 2006] Chater, N., Tenenbaum, J. B., and Yuille, A. (2006). Probabilistic models of cognition: where next? *Trends in Cognitive Sciences*, 10(7):292–293. 157

[Chatfield et al., 2011] Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*. 69

[Cheng, 1995] Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799. 18

[Cheng et al., 2012] Cheng, Z., Qin, L., Ye, Y., Huang, Q., and Tian, Q. (2012). Human daily action analysis with multi-view and color-depth data. In *European Conference on Computer Vision Workshop*. 9, 42, 43, 49, 57, 94, 146, 147, 148

[Choi et al., 2011a] Choi, W., Pantofaru, C., and Savarese, S. (2011a). Detecting and tracking people using an RGB-D camera via multiple detector fusion. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 19, 84, 96

[Choi et al., 2011b] Choi, W., Shahid, K., and Savarese, S. (2011b). Learning context for collective activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 39, 81, 84, 85, 96

[Chowdhury and Chellappa, 2003a] Chowdhury, A. K. R. and Chellappa, R. (2003a). A factorization approach for activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, volume 4, page 41. 42

[Chowdhury and Chellappa, 2003b] Chowdhury, A. K. R. and Chellappa, R. (2003b). A factorization approach for activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 129

[Chua et al., 2011] Chua, S.-L., Marsland, S., and Guesgen, H. W. (2011). Unsupervised learning of human behaviours. In *AAAI Conference on Artificial Intelligence*. 126

[Crowley, 2006] Crowley, J. L. (2006). Things that see: Context-aware multi-modal interaction. In *Cognitive Vision Systems*, pages 183–198. 157

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 18, 26, 30, 36, 43, 50, 65, 84, 87, 93, 96

[Dalal et al., 2006] Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*. 18

[Derpanis et al., 2013] Derpanis, K. G., Sizintsev, M., Cannons, K. J., and Wildes, R. P. (2013). Action spotting and recognition based on a spatiotemporal orientation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):527540. 65, 77, 79, 80, 120, 144

[Do and Artières, 2012] Do, T.-M.-T. and Artières, T. (2012). Regularized bundle methods for convex and non-convex risks. *Journal of Machine Learning Research*, 13(1):3539–3583. 112, 116

[Dollár et al., 2005] Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*. 39, 42, 43, 55, 56, 57, 58, 65, 76, 77, 83, 84, 89, 94, 95, 111, 118, 119, 127, 129

[Dollár et al., 2012] Dollár, P., Wojek, C., Schiele, B., and Perona, P. (2012). Pedestrian detection: an evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761. 15, 18

[Dundar et al.,] Dundar, M., Akova, F., Qi, A., and Rajwa, B. Bayesian nonexhaustive learning for online discovery and modeling of emerging classes. In *International Conference on Machine Learning*. 164

[Duric et al., 2002] Duric, Z., Gray, W., Heishman, R., Li, F., Rosenfeld, A., Schoelles, M., Schunn, C., and Wechsler, H. (2002). Integrating perceptual and cognitive modeling for adaptive and intelligent human-computer interaction. *Proceedings of the IEEE*, 90(7):1272–1289. 157

[Edelsbrunner and Grayson, 1999] Edelsbrunner, H. and Grayson, D. R. (1999). Edgewise subdivision of a simplex. In *Symposium on Computational Geometry*. 71, 75

[Einicke and White, 1999] Einicke, G. and White, L. (1999). Robust extended Kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599. 30

[Ess et al., 2009] Ess, A., Leibe, B., Schindler, K., and van Gool, L. (2009). Robust multiperson tracking from a mobile platform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1831–1846. 19

[Everts et al., 2013] Everts, I., van Gemert, J. C., and Gevers, T. (2013). Evaluation of color stips for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 65, 68, 75, 76, 77, 79, 80, 82, 92, 93

[Fanello et al., 2013] Fanello, S. R., Gori, I., Metta, G., and Odone, F. (2013). Keep it simple and sparse: Real-time action recognition. *Journal of Machine Learning Research*, 14(1):2617–2640. 49, 130

[Farrahi and Gatica-Perez, 2011] Farrahi, K. and Gatica-Perez, D. (2011). Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology*, 2(1):3:1–3:27. 156

[Felzenszwalb et al., 2008] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*. 84, 96

[Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395. 22, 86

[Flitton et al., 2013] Flitton, G., Breckon, T. P., and Megherbi, N. (2013). A comparison of 3d interest point descriptors with application to airport baggage object detection in complex CT imagery. *Pattern Recognition*, 46(9):2420–2436. 68

[Fortmann et al., 1983] Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8(3):173–184. 19

[Franc and Sonnenburg, 2009] Franc, V. and Sonnenburg, S. (2009). Optimized cutting plane algorithm for large-scale risk minimization. *Journal of Machine Learning Research*, 10:2157–2192. 116

[Freifeld et al., 2010] Freifeld, O., Weiss, A., Zuffi, S., and Black, M. J. (2010). Contour people: A parameterized model of 2D articulated human shape. In *IEEE Conference on Computer Vision and Pattern Recognition*. 129

[Gath and Gev, 1989] Gath, I. and Gev, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):773–780. 136, 137

[Gilbert et al., 2009] Gilbert, A., Illingworth, J., and Bowden, R. (2009). Fast realistic multi-action recognition using mined dense spatio-temporal features. In *IEEE Conference on Computer Vision and Pattern Recognition*. 119, 120, 144

[Gonzalez and Woods, 2007] Gonzalez, R. C. and Woods, R. E. (2007). *Digital Image Processing*. Prentice Hall, 3 edition. 45

[Gorelick et al., 2007] Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. (2007). Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253. 105, 168

[Grabner and Bischof, 2006] Grabner, H. and Bischof, H. (2006). On-line boosting and vision. In *IEEE Conference on Computer Vision and Pattern Recognition*. 29, 88

[Grandvalet and Bengio, 2004] Grandvalet, Y. and Bengio, Y. (2004). Semi-supervised learning by entropy minimization. In *Neural Information Processing Systems*, pages 281–296. 110, 111

[Gray et al., 1997] Gray, W. D., Young, R. M., and Kirschenbaum, S. S. (1997). Introduction to this special issue on cognitive architectures and human-computer interaction. *Human-Computer Interaction*, 12(4):301–309. 156

[Griffiths and Steyvers, 2004] Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. In *National Academy of Sciences*. 159

[Gupta et al., 2013] Gupta, R., Chia, A. Y.-S., and Rajan, D. (2013). Human activities recognition using depth images. In *ACM International Conference on Multimedia*, pages 283–292. 145

[Han et al., 2009] Han, D., Bo, L., and Sminchisescu, C. (2009). Selection and context for action recognition. In *International Conference on Computer Vision*. 77, 80, 120

[Hard et al., 2006] Hard, B., Tversky, B., and Lang, D. (2006). Making sense of abstract events: Building event schemas. *Memory and Cognition*, 34(6):1221–1235. 125

[Hoai et al., 2011] Hoai, M., Lan, Z.-Z., and De la Torre, F. (2011). Joint segmentation and classification of human actions in video. In *IEEE Conference on Computer Vision and Pattern Recognition*. 130, 131, 135, 139, 145

[Hofmann, 1999a] Hofmann, T. (1999a). Probabilistic latent semantic analysis. In *Conference on Uncertainty in Artificial Intelligence*. 98, 101

[Hofmann, 1999b] Hofmann, T. (1999b). Probabilistic latent semantic indexing. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*. 143, 144, 145, 148

[Holte et al., 2010] Holte, M., Moeslund, T., and Fihl, P. (2010). View-invariant gesture recognition using 3d optical flow and harmonic motion context. *Computer Vision and Image Understanding*, 114(12):1353–1361. 68

[Hörster et al., 2008] Hörster, E., Lienhart, R., and Slaney, M. (2008). Continuous visual vocabulary modelsfor pLSA-based scene recognition. In *International Conference on Content-Based Image and Video Retrieval*. 99, 108

[Huynh et al., 2008] Huynh, T., Fritz, M., and Schiele, B. (2008). Discovery of activity patterns using topic models. In *International Conference on Ubiquitous Computing*. 156

[Ikemura and Fujiyoshi, 2011] Ikemura, S. and Fujiyoshi, H. (2011). Real-time human detection using relational depth similarity features. *Asian Conference on Computer Vision*. 19

[Isla et al., 2001] Isla, D., Burke, R., Downie, M., and Blumberg, B. (2001). A layered brain architecture for synthetic creatures. In *International Joint Conferences on Artificial Intelligence*. 156, 157

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323. 143, 144, 145, 148

[Ji et al., 2013] Ji, Y., Shimada, A., Nagahara, H., and ichiro Taniguchi, R. (2013). A compact descriptor chog3d and its application in human action recognition. *IEEJ Transactions on Electrical and Electronic Engineering*, 8(1):69–77. 68, 77, 79

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45. 18

[Kapur, 1967] Kapur, J. N. (1967). Generalized entropy of order α and type β. *Maths Seminar*, 4. 114

[Kaucic et al., 2005] Kaucic, R., Amitha Perera, A., Brooksby, G., Kaufhold, J., and Hoogs, A. (2005). A unified framework for tracking through occlusions and across sensor gaps. *IEEE Conference on Computer Vision and Pattern Recognition*. 19

[Keller et al., 2011] Keller, C., Enzweiler, M., Rohrbach, M., Llorca, D., Schnorr, C., and Gavrila, D. (2011). The benefits of dense stereo for pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(4):1096 –1106. 19

[Khamis et al., 2012] Khamis, S., Morariu, V., and Davis, L. (2012). A flow model for joint action recognition and identity maintenance. In *IEEE Conference on Computer Vision and Pattern Recognition*. 39, 81, 84, 85

[Kim and Torralba, 2009] Kim, G. and Torralba, A. (2009). Unsupervised detection of regions of interest using iterative link analysis. In *Neural Information Processing Systems*. 22, 85

[Kläser, 2010] Kläser, A. (2010). *Learning human actions in video*. PhD thesis, Université de Grenoble. 77, 80

[Kläser et al., 2008] Kläser, A., Marszalek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*. 39, 43, 52, 65, 68, 76, 77, 80, 81, 84, 92, 93

[Klir, 2005] Klir, G. J. (2005). *Uncertainty & Information: Foundations of Generalized Information Theory*. Wiley-IEEE Press. 114

[Knauff and Wolf, 2010] Knauff, M. and Wolf, A. G. (2010). Complex cognition: the science of human reasoning, problem-solving, and decision-making. *Cognitive Processing*, 11(2):99–102. 153

[Knoop et al., 2006] Knoop, S., Vacek, S., and Dillmann, R. (2006). Sensor fusion for 3D human body tracking with an articulated 3D body model. In *IEEE International Conference on Robotics and Automation*. 42

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. The MIT Press, 1 edition. 18, 205, 206

[Koppula et al., 2013] Koppula, H. S., Gupta, R., and Saxena, A. (2013). Learning human activities and object affordances from RGB-D videos. *International Journal of Robotic Research*, 32(8):951–970. 145

[Kuang et al., 2011] Kuang, Y., Byröd, M., and Åström, K. (2011). Supervised feature quantization with entropy optimization. In *International Conference on Computer Vision Workshop*. 69

[Kuhn, 1955] Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97. 134, 162

[Kumar et al., 2012] Kumar, M. P., Packer, B., and Koller, D. (2012). Modeling latent variable uncertainty for loss-based learning. In *International Conference on Machine Learning*. 110, 111

[Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*. 109, 111, 129

[Laird et al., 1987] Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: an architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64. 156

[Lan et al., 2012] Lan, T., Wang, Y., Yang, W., Robinovitch, S., and Mori, G. (2012). Discriminative latent models for recognizing contextual group activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1549–1562. 39, 81, 84, 85, 96

[Lanz, 2006] Lanz, O. (2006). Approximate Bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449. 18

[Laptev, 2005] Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123. 39, 42, 43, 83, 94, 95, 111, 129, 139, 141, 142, 168, 169

[Laptev et al., 2008] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition*. 43, 58, 76, 127

[Larsson and Ugander, 2011] Larsson, M. O. and Ugander, J. (2011). A concave regularization technique for sparse mixture models. In *Conference on Neural Information Processing Systems*. 100

[Le et al., 2011] Le, Q. V., Zou, W. Y., Yeung, S. Y., and Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3361–3368. 154

[Lecun et al., 2006] Lecun, Y., Chopra, S., Hadsell, R., Huang, F. J., and Ranzato, M. A. (2006). A tutorial on energy-based learning. *Predicting Structured Outputs*. 110, 112

[Leibe et al., 2005] Leibe, B., Seemann, E., and Schiele, B. (2005). Pedestrian detection in crowded scenes. *IEEE Conference on Computer Vision and Pattern Recognition*. 3

[Li et al., 2004] Li, T., Ma, S., and Ogihara, M. (2004). Entropy-based criterion in categorical clustering. In *International Conference on Machine Learning*. 111

[Lin et al., 2009] Lin, Z., Jiang, Z., and Davis, L. S. (2009). Recognizing actions by shape-motion prototype trees. In *IEEE International Conference on Computer Vision*. 105, 107

[Loper et al., 2009] Loper, M. M., Koenig, N. P., Chernova, S. H., Jones, C. V., and Jenkins, O. C. (2009). Mobile human-robot teaming with environmental tolerance. *ACM/IEEE International Conference on Human-Robot Interaction*. 15

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110. 18, 39, 43, 50, 65, 67, 84, 106, 129, 154, 169

[Luber et al., 2011] Luber, M., Spinello, L., and Arras, K. O. (2011). People tracking in RGB-D data with on-line boosted target models. *IEEE International Conference on Intelligent Robots and Systems*. 19

[Luo et al., 2013] Luo, J., Wang, W., and Qi, H. (2013). Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *IEEE International Conference on Computer Vision*. 129

[Marler and Arora, 2004] Marler, R. T. and Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26:369–395. 114

[Marszalek et al., 2009] Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8, 65, 77, 80, 120, 142, 144

[Mattivi and Shao, 2011] Mattivi, R. and Shao, L. (2011). Robust spatio-temporal features for human action recognition. In *Multimedia Analysis, Processing and Communications*. 68, 77

[Miller et al., 2012] Miller, K., Kumar, M. P., Packer, B., Goodman, D., and Koller, D. (2012). Max-margin min-entropy models. In *International Conference on Artificial Intelligence and Statistics*. 111, 205

[Minnen et al., 2006] Minnen, D., Westeyn, T., and Starner, T. (2006). Performance metrics and evaluation issues for continuous activity recognition. In *Performance Metrics for Intelligent Systems*. 125

[Morency et al., 2007] Morency, L.-P., Quattoni, A., and Darrell, T. (2007). Latent-dynamic discriminative models for continuous gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 206

[Muñoz Salinas et al., 2007] Muñoz Salinas, R., Aguirre, E., and García-Silvente, M. (2007). People detection and tracking using stereo vision and color. *Journal of Image Vision Computing*, 25(6):995–1007. 19

[Munkres, 1984] Munkres, J. (1984). *Elements of Algebraic Topology*. Advanced book classics. Perseus Books. 71

[Musat et al., 2011] Musat, C. C., Velcin, J., Trausan-Matu, S., and Rizoiu, M.-A. (2011). Improving topic evaluation using conceptual knowledge. In *International Joint Conference on Artificial Intelligence*. 162

[Nagel, 2004] Nagel, H.-H. (2004). Steps toward a cognitive vision system. *AI Magine*, 25(2):31–50. 157

[Newman et al., 2010] Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. 156

[Ni et al., 2013] Ni, B., Pei, Y., Moulin, P., and Yan, S. (2013). Multilevel depth and image fusion for human activity detection. *IEEE Transactions on Cybernetics*, 43(5):1383–1394. 145

[Ni et al., 2011] Ni, B., Wang, G., and Moulin, P. (2011). RGBD-HuDaAct: A color-depth video database for human daily activity recognition. In *International Conference on Computer Vision Workshop*. 42, 43, 57, 93, 94, 95, 148

[Ni et al., 2009] Ni, B., Yan, S., and Kassim, A. A. (2009). Recognizing human group activities with localized causalities. In *IEEE Conference on Computer Vision and Pattern Recognition*. 81, 84, 85, 96

[Niebles et al., 2008] Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79:299–318. 52, 100, 107, 126, 154, 156, 160

[Ofli et al., 2013] Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Winter Conference on Applications of Computer Vision*. 9, 43, 49, 56, 57, 60, 93

[Oikonomopoulos et al., 2005] Oikonomopoulos, A., Patras, I., and Pantic, M. (2005). Spatiotemporal salient points for visual recognition of human actions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(3):710–719. 42, 129

[Okuma et al., 2004] Okuma, K., Taleghani, A., de Freitas, N., Little, J. J., and Lowe, D. G. (2004). A boosted particle filter: multitarget detection and tracking. *European Conference on Computer Vision*. 18

[Oreifej and Liu, 2013] Oreifej, O. and Liu, Z. (2013). HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*. 111, 122, 123

[Page, 1954] Page, E. S. (1954). Continuous Inspection Schemes. *Biometrika*, 41:100–115. 130, 135

[Papadakis and Bugeau, 2011] Papadakis, N. and Bugeau, A. (2011). Tracking with occlusions via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):144–157. 19

[Parzen, 1962] Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076. 24, 87

[Piyathilaka and Kodagoda, 2013] Piyathilaka, L. and Kodagoda, S. (2013). Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features. In *IEEE Conference on Industrial Electronics and Applications*. 111, 121

[Porteous et al., 2008] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P., and Welling, M. (2008). Fast collapsed gibbs sampling for latent dirichlet allocation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 133, 159

[Quattoni et al., 2007] Quattoni, A., Wang, S., Morency, L.-P., Collins, M., and Darrell, T. (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1848–1852. 109, 111, 206

[Ranganathan, 2012] Ranganathan, A. (2012). PLISS: labeling places using online changepoint detection. *Autonomous Robots*, 32(4):351–368. 131

[R.E. Burkard, 2012] R.E. Burkard, M. Dell'Amico, S. M. (2012). *Assignment Problems* (Revised reprint). Society for Industrial and Applied Mathematics. 134

[Reid, 1979] Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854. 19

[Rodriguez et al., 2008] Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8, 77

[Rudin, 1964] Rudin, W. (1964). *Principles of mathematical analysis*. McGraw-Hill. 71

[Rusu and Cousins, 2011] Rusu, R. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation*. 13, 31, 32

[Salas and Tomasi, 2011] Salas, J. and Tomasi, C. (2011). People detection using color and depth images. *Mexican Conference on Pattern Recognition*. 19

[Schmid et al., 2011] Schmid, U., Ragni, M., Gonzalez, C., and Funke, J. (2011). The challenge of complexity for cognitive systems. *Cognitive Systems Research*. 153

[Schuldt et al., 2004] Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *IEEE Conference on Computer Vision and Pattern Recognition*. 7, 77, 105, 118, 119, 169

[Schwarz et al., 2010] Schwarz, L., Mateus, D., Castaneda, V., and Navab, N. (2010). Manifold learning for tof-based human body tracking and activity recognition. In *British Machine Vision Conference*. 42

[Scovanner et al., 2007] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional SIFT descriptor and its application to action recognition. In *IEEE International Conference on Multimedia and Expo*. 43, 50, 65, 68, 76, 84

[Shi et al., 2008] Shi, Q., Wang, L., Cheng, L., and Smola, A. (2008). Discriminative human action segmentation and recognition using semi-Markov model. In *IEEE Conference on Computer Vision and Pattern Recognition*. 130, 131, 135

[Shotton et al., 2011] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*. 129

[Shu et al., 2012] Shu, G., Dehghan, A., Oreifej, O., Hand, E., and Shah, M. (2012). Part-based multiple-person tracking with partial occlusion handling. *IEEE Conference on Computer Vision and Pattern Recognition*. 19

[Si and Jin, 2005] Si, L. and Jin, R. (2005). Adjusting mixture weights of gaussian mixture model via regularized probabilistic latent semantic analysis. In *Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*. 100

[Simon Kozina, 2011] Simon Kozina, Mitja Lustrek, M. G. (2011). Dynamic signal segmentation for activity recognition. In *International Joint Conference on Artificial Intelligence*. 130

[Singh et al., 2008] Singh, M., Basu, A., and Mandal, M. (2008). Human activity recognition based on silhouette directionality. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1280–1292. 42, 129

[Song et al., 2011] Song, Y., Demirdjian, D., and Davis, R. (2011). Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *IEEE International Conference on Automatic Face and Gesture Recognition*. 206

[Song et al., 2013] Song, Y., Morency, L.-P., and Davis, R. (2013). Action recognition by hierarchical sequence summarization. In *IEEE Conference on Computer Vision and Pattern Recognition*. 206

[Spinello et al., 2010] Spinello, L., Arras, K., Triebel, R., and Siegwart, R. (2010). A layered approach to people detection in 3D range data. *AAAI Conference on Artificial Intelligence*. 19

[Stauffer and Grimson, 1999] Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition*. 18

[Sung et al., 2012] Sung, J., Ponce, C., Selman, B., and Saxena, A. (2012). Unstructured human activity detection from RGBD images. In *IEEE International Conference on Intelligent Robots and Systems*. 121, 145

[Sung et al., 2011] Sung, J. Y., Ponce, C., Selman, B., and Saxena, A. (2011). Human activity detection from RGBD images. *AAAI Workshop on Pattern, Activity and Intent Recognition*. 42

[Tang et al., 2012] Tang, S., Wang, X., Lv, X., Han, T. X., Keller, J., He, Z., Skubic, M., and Lao, S. (2012). Histogram of oriented normal vectors for object recognition with a depth sensor. In *Asian Conference on Computer Vision*. 68

[Teo et al., 2010] Teo, C. H., Smola, A., Vishwanathan, S. V., and Le, Q. V. (2010). Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365. 112, 116

[Tian et al., 2013] Tian, Y., Sukthankar, R., and Shah, M. (2013). Spatiotemporal deformable part models for action detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. 68

[Town and Sinclair, 2003] Town, C. and Sinclair, D. (2003). A self-referential perceptual inference framework for video interpretation. In *International Conference on Vision Systems*. 157

[Treisman and Gelade, 1980] Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136. 39

[Turaga et al., 2008] Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488. 41, 128

[Turcot and Lowe, 2009] Turcot, P. and Lowe, D. G. (2009). Better matching with fewer features: The selection of useful features in large database recognition problems. In *International Conference on Computer Vision Workshop*. 84

[Ullah et al., 2010] Ullah, M. M., Parizi, S. N., and Laptev, I. (2010). Improving bag-of-features action recognition with non-local cues. In *British Machine Vision Conference*. 79, 80, 120

[Varela and Dupuy, 1992] Varela, F. J. and Dupuy, J. (1992). *Understanding Origins – Contemporary Views on the Origin of Life, Mind and Society*, chapter Whence perceptual meaning? A cartography of current ideas, pages 235–263. Kluwer Academic Publishers. 156

[Vedaldi and Zisserman, 2012] Vedaldi, A. and Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–492. 53

[Veeraraghavan et al., 2005] Veeraraghavan, A., Roy-Chowdhury, A., and Chellappa, R. (2005). Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909. 42

[Vernon et al., 2007] Vernon, D., Metta, G., and Sandini, G. (2007). A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. *IEEE Transactions on Evolutionary Computation*, 11(2):151–180. 156, 157

[Vieira et al., 2012] Vieira, A. W., Nascimento, E. R., Oliveira, G. L., Liu, Z., and Campos, M. F. M. (2012). Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences. In *Iberoamerican Congress on Pattern Recognition*. 123

[Viola et al., 2005] Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161. 3, 25, 26, 87

[Wallach et al., 2009] Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *International Conference on Machine Learning*. 156, 159, 162, 163

[Wang et al., 2013a] Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2013a). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79. 39, 42, 43, 111, 118, 119, 120, 144

[Wang and Schmid, 2013] Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision*. 43

[Wang et al., 2009] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Association*. 39, 50, 52, 65, 67, 76, 77, 79, 80, 93, 127, 129

[Wang et al., 2012a] Wang, J., Liu, Z., Chorowski, J., Chen, Z., and Wu, Y. (2012a). Robust 3D action recognition with random occupancy patterns. In *European Conference on Computer Vision*. 111, 123

[Wang et al., 2012b] Wang, J., Liu, Z., Wu, Y., and Yuan, J. (2012b). Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10, 49, 58, 122, 123

[Wang et al., 2013b] Wang, L., Qiao, Y., and Tang, X. (2013b). Motionlets: Mid-level 3D parts for human motion recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 43

[Wang et al., 2006] Wang, S. B., Quattoni, A., Morency, L.-P., and Demirdjian, D. (2006). Hidden conditional random fields for gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 118

[Wang and Mori, 2009] Wang, Y. and Mori, G. (2009). Human action recognition by semilatent topic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1762–1774. 156

[Wang and Mori, 2011] Wang, Y. and Mori, G. (2011). Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323. 109, 111, 119

[Warren Liao, 2005] Warren Liao, T. (2005). Clustering of time series data – a survey. *Pattern Recognition*, 38(11):1857–1874. 131

[Wei and Croft, 2006] Wei, X. and Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. In *Interational Conference on Research and Development in Information Retrieval*. 156

[Willems et al., 2008] Willems, G., Tuytelaars, T., and Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision*, pages 650–663. 42, 129

[Wong and Cipolla, 2007] Wong, S. F. and Cipolla, R. (2007). Extracting spatiotemporal interest points using global information. *IEEE International Conference on Computer Vision*, pages 1–8. 42

[Wong et al., 2007] Wong, S.-F., Kim, T.-K., and Cipolla, R. (2007). Learning motion categories using both semantic and structural information. In *IEEE Conference on Computer Vision and Pattern Recognition*. 100

[Wu and Nevatia, 2007] Wu, B. and Nevatia, R. (2007). Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266. 18

[Xia and Aggarwal, 2013] Xia, L. and Aggarwal, J. K. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*. 39, 43, 58, 65, 81, 82, 84, 89, 93, 94, 95, 129

[Xia et al., 2011] Xia, L., Chen, C.-C., and Aggarwal, J. (2011). Human detection using depth information by Kinect. *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 19

[Xia et al., 2012] Xia, L., Chen, C.-C., and Aggarwal, J. K. (2012). View invariant human action recognition using histograms of 3D joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 68

[Xu and Fujimura, 2003] Xu, F. and Fujimura, K. (2003). Human detection using depth and gray images. *IEEE Conference on Advanced Video and Signal Based Surveillance*. 19

[Xu and Griffiths, 2011] Xu, F. and Griffiths, T. L. (2011). Probabilistic models of cognitive development: Towards a rational constructivist approach to the study of learning and development. *Cognition*, 120:299–301. 157

[Yan et al., 2008] Yan, P., Khan, S., and Shah, M. (2008). Learning 4D action feature models for arbitrary view action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 42

[Yang et al., 2013] Yang, B., Zhou, L., and Deng, Z. (2013). C-HMAX: Artificial cognitive model inspired by the color vision mechanism of the human brain. *Tsinghua Science and Technology*, 18(1):51–56. 157

[Yang and Tian, 2012] Yang, X. and Tian, Y. (2012). EigenJoints-based action recognition using naive-Bayes-nearest-neighbor. *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 123

[Yu and Aggarwal, 2009] Yu, E. and Aggarwal, J. (2009). Human action recognition with extremities as semantic posture representation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. 42

[Zeng and Ji, 2010] Zeng, Z. and Ji, Q. (2010). Knowledge based activity recognition with dynamic bayesian network. In *European Conference on Computer Vision*. 111

[Zhai and Shah, 2005] Zhai, Y. and Shah, M. (2005). A general framework for temporal video scene segmentation. In *IEEE International Conference on Computer Vision*. 130, 131, 135

[Zhang et al., 2012a] Zhang, H., Edwards, R., and Parker, L. E. (2012a). Regularized probabilistic latent semantic analysis with continuous observations. In *IEEE International Conference on Machine Learning and Applications*. 181

[Zhang and Parker, 2011] Zhang, H. and Parker, L. E. (2011). 4-dimensional local spatio-temporal features for human activity recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2044–2049. 10, 39, 44, 49, 55, 56, 65, 67, 68, 75, 81, 82, 83, 84, 89, 92, 94, 95, 126, 127, 129, 145, 147, 148, 154, 156, 168, 169, 180

[Zhang et al., 2013] Zhang, H., Reardon, C. M., and Parker, L. E. (2013). Real-time multiple human perception with color-depth cameras on a mobile robot. *IEEE Transactions on Cybernetics*, 43(5):1429–1441. 42, 179

[Zhang et al., 2014a] Zhang, H., Zhou, W., and Parker, L. E. (2014a). Fuzzy segmentation and recognition of continuous human activities. In *IEEE International Conference on Robotics and Automation*. 12, 181

[Zhang et al., 2014b] Zhang, H., Zhou, W., Reardon, C., and Parker, L. (2014b). Simplex-based 3D spatio-temporal feature description for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 180

[Zhang et al., 2007] Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238. 53

[Zhang et al., 2008] Zhang, L., Li, Y., and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. *IEEE Conference on Computer Vision and Pattern Recognition*. 18, 19

[Zhang et al., 2012b] Zhang, Y., Liu, X., Chang, M.-C., Ge, W., and Chen, T. (2012b). Spatio-temporal phrases for activity recognition. In *European Conference on Computer Vision*. 127

[Zhou et al., 2013] Zhou, F., De la Torre, F., and Hodgins, J. K. (2013). Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596. 131, 135

Appendix

Proofs of MC-HCRF

Proof of Theorem 8.1

Theorem 8.1. *The MC-HCRF's energy function:*

$$e(y, \boldsymbol{x}; \boldsymbol{\theta}) = -\log P(y|\boldsymbol{x}; \boldsymbol{\theta}) + H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \tag{1}$$

combines both objectives in the multi-objective optimization problem in Eq. (9.6).

Proof. Because the partition function $Z(\boldsymbol{x}; \boldsymbol{\theta})$ is a constant that is independent of the output y, we can obtain:

$$\begin{aligned}
&\min\ e(y, \boldsymbol{x}; \boldsymbol{\theta}) \\
=&\min\left(-\log P(y|\boldsymbol{x}; \boldsymbol{\theta}) + H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta}))\right) \\
=&\min\left(-\log P(y|\boldsymbol{x}; \boldsymbol{\theta})\right) + \min H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})) \\
=&\max\left(\log P(y|\boldsymbol{x}; \boldsymbol{\theta})\right) + \max\left(-H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta}))\right) \\
&\text{Since } \log(\cdot) \text{ is monotically increasing:} \\
=&\max\left(P(y|\boldsymbol{x}; \boldsymbol{\theta})\right) + \max\left(-H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta}))\right)
\end{aligned}$$

This is the exact multi-objective optimization defined in Eq. (9.6). □

Proof of Theorem 8.2

Lemma .0.1. *For finite discrete random variable $\boldsymbol{z}$, the Kapur entropy satisfies: $H_{\alpha,\beta}(\tilde{P}(\boldsymbol{z})) = H_{\alpha,\beta}(P(\boldsymbol{z})) - \log Z$, where $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$, $\alpha + \beta - 1 > 0$, $\tilde{P}(\boldsymbol{z})$ is the unnormalized measure of $P(\boldsymbol{z})$, and $Z = \sum_{\boldsymbol{z}} \tilde{P}(\boldsymbol{z})$ is the partition function.*

Proof. Under the Kapur entropy constraints $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$ and $\alpha + \beta - 1 > 0$, we obtain the following:

$$\begin{aligned}
H_{\alpha,\beta}(\tilde{P}(\boldsymbol{z})) &= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{z}} \tilde{P}(\boldsymbol{z})^{\alpha+\beta-1}}{\sum_{\boldsymbol{z}} \tilde{P}(\boldsymbol{z})^{\beta}} \\
&= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{z}} (P(\boldsymbol{z}) \cdot Z)^{\alpha+\beta-1}}{\sum_{\boldsymbol{z}} (P(\boldsymbol{z}) \cdot Z)^{\beta}} \\
&= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{z}} P(\boldsymbol{z})^{\alpha+\beta-1}}{\sum_{\boldsymbol{z}} P(\boldsymbol{z})^{\beta}} + \frac{1}{1-\alpha} \log Z^{\alpha-1} \\
&= H_{\alpha,\beta}(P(\boldsymbol{z})) - \log Z
\end{aligned}$$

□

Using Lemma .0.1, we can prove the following lemma:

Lemma .0.2. *The MC-HCRF's energy function satisfies:*

$$e(y, \boldsymbol{x}; \boldsymbol{\theta}) = H_{\alpha,\beta}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \tag{2}$$

where $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$, $\alpha + \beta - 1 > 0$, *and* $Z(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{y \in \mathcal{Y}, \boldsymbol{h} \in \mathcal{H}} \tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})$ *is the partition function.*

Proof. Under the Kapur entropy constraints $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$ and $\alpha + \beta - 1 > 0$, we obtain the following:

$$
\begin{aligned}
& e(y, \boldsymbol{x}; \boldsymbol{\theta}) \\
&= H_{\alpha,\beta}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})) - \log P(y|\boldsymbol{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \\
&= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{h}} P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})^{\alpha+\beta-1}}{\sum_{\boldsymbol{h}} P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})^{\beta}} \\
&\quad - \log P(y|\boldsymbol{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \qquad \text{[via entropy definition]} \\
&= \frac{1}{1-\alpha} \log \left(\frac{\sum_{\boldsymbol{h}} \left(\frac{P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})}{P(y|\boldsymbol{x}; \boldsymbol{\theta})} \right)^{\alpha+\beta-1}}{\sum_{\boldsymbol{h}} \left(\frac{P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})}{P(y|\boldsymbol{x}; \boldsymbol{\theta})} \right)^{\beta}} \right) \\
&\quad - \log P(y|\boldsymbol{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \qquad \text{[via Bayes rule]} \\
&= \frac{1}{1-\alpha} \log \left(\frac{\sum_{\boldsymbol{h}} P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\alpha+\beta-1}}{\sum_{\boldsymbol{h}} P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\beta}} \cdot \frac{P(y|\boldsymbol{x}; \boldsymbol{\theta})^{1-\alpha-\beta}}{P(y|\boldsymbol{x}; \boldsymbol{\theta})^{-\beta}} \right) \\
&\quad - \log P(y|\boldsymbol{x}; \boldsymbol{\theta}) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \qquad \text{[via exponent manipulation]} \\
&= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{h}} P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\alpha+\beta-1}}{\sum_{\boldsymbol{h}} P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\beta}} \\
&\quad + \left(\frac{1}{1-\alpha} \log P(y|\boldsymbol{x}; \boldsymbol{\theta})^{1-\alpha} - \log P(y|\boldsymbol{x}; \boldsymbol{\theta}) \right) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \\
&= H_{\alpha,\beta}(P(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) - \log Z(\boldsymbol{x}; \boldsymbol{\theta}) \\
&= H_{\alpha,\beta}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \qquad \text{[via Lemma .0.1]}
\end{aligned}
$$

□

This proof also indicates the importance of incorporating the partition function $Z(\boldsymbol{x}; \boldsymbol{\theta})$, although it is a constant, in our MC-HCRF's energy. Lemma .0.2 is closely related to [Miller et al., 2012]. However, without explicitly handling the partition function, [Miller et al., 2012] is not directly applicable to our MC-HCRF model.

Lemma .0.3. *If HCRF's latent variables form a graph without loops, the cluster graph of the model with the singleton and pairwise potentials has a clique tree representation.*

Proof. If HCRF's latent variables $\boldsymbol{h}$ form an undirected tree $\mathcal{T}_h$, we can always construct a clique tree $\mathcal{T}_c$ by the following steps. First, we construct an undirected tree $\mathcal{T}_c$ that has the same topology as $\mathcal{T}_h$, and assign the singleton potentials $\psi(h_i, y)$ and $\psi(h_i, x_i)$ to the clique $\boldsymbol{C}_i$ with the scope $\{h_i, y, x_i\}$. Second, for each pair of the directly connected cliques $\boldsymbol{C}_i - \boldsymbol{C}_j$, we remove the edge between the cliques, add a new clique $\boldsymbol{C}_{ij}$ with the scope $\{h_i, h_j, y\}$ to form a chain $\boldsymbol{C}_i - \boldsymbol{C}_{ij} - \boldsymbol{C}_j$, and assign the pairwise potential $\psi(h_i, h_j, y)$ to the new clique $\boldsymbol{C}_{ij}$. It can be easily verified that the constructed tree $\mathcal{T}_c$ satisfies the family preservation property and the running intersection property [Koller and Friedman, 2009], and thus is a clique tree. □

Algorithm 8: Sum-Product Belief Propagation

Input : HCRF's graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, graph potentials $\boldsymbol{\psi}(\boldsymbol{D})$
Output: $\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta})$

1: Construct clique tree $\mathcal{T}_c = \{\mathcal{V}_c, \mathcal{E}_c\}$ from $\mathcal{G}=(\mathcal{V},\mathcal{E})$;
2: **foreach** node $i \in \mathcal{V}_c$ **do**
3: Initialize clique potentials: $\varphi_i(\boldsymbol{C}_i) = \prod_{\psi_j\,:\,\alpha(\psi_j)=i} \psi_j(\boldsymbol{D}_j)$;
4: **while** $\exists i, j$: $\boldsymbol{C}_i$ is ready to send $\delta_{i\to j}(\boldsymbol{S}_{i,j})$ **do**
5: Compute and send the message:
$\delta_{i\to j}(\boldsymbol{S}_{i,j}) = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \left(\varphi_i(\boldsymbol{C}_i) \cdot \prod_{k\in(\mathrm{Nb}_i - \{j\})} \delta_{k\to i}(\boldsymbol{S}_{k,i})\right)$
6: **end**
7: **foreach** node $i \in \mathcal{V}_c$ **do** Compute clique belief:
$\beta_i(\boldsymbol{C}_i) = \varphi_i(\boldsymbol{C}_i) \cdot \prod_{k\in\mathrm{Nb}_i} \delta_{k\to i}(\boldsymbol{S}_{k,i})$;
8: **foreach** edge $i-j \in \mathcal{E}_c$ **do** Compute sepset belief: $\mu_{i,j}(\boldsymbol{S}_{i,j}) = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{C}_i)$;
9: Compute $\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta}) = \frac{\prod_{i\in\mathcal{V}_\mathcal{T}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j)\in\mathcal{E}_\mathcal{T}} \mu_{i,j}(\boldsymbol{S}_{i,j})}$;
10: **return** $\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta})$

Belief propagation is the most widely applied method to perform inference in previous MLE-HCRF models [Chang et al., 2009b, Koller and Friedman, 2009, Morency et al., 2007, Quattoni et al., 2007, Song et al., 2011, Song et al., 2013]. We implement a belief propagation algorithm to compute $\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta})$, which is used to compute $e(y,\boldsymbol{x};\boldsymbol{\theta})$ and solve the inference problem. The algorithm is presented in Algorithm 8. Since we assume our model's underlying graph does not contain loops, the computational complexity of Algorithm 8 satisfies the following lemma (Similar time complexities were also observed in previous works [Quattoni et al., 2007, Song et al., 2013]):

Lemma .0.4. *Algorithm 8 requires $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ to compute the quantity $\tilde{P}(y|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{h}} \tilde{P}(y,\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta})$.*

Proof. The clique tree $\mathcal{T}_c$ is constructed in $O(|\mathcal{V}|)$ time (line 1) using the process described in Lemma .0.3. Each pairwise potential is assigned to its corresponding clique in $O(1)$ time. Each singleton potential requires $|\mathcal{H}|$ multiplication to be assigned, and the upper bound for the number of such cliques is $|\mathcal{V}|$. Therefore, the clique potentials are initialized (line 3) in an $O(|\mathcal{V}||\mathcal{H}|)$ runtime. Given a fixed value of $y \in \mathcal{Y}$, there are $2|\mathcal{E}|$ messages that are passed over $\mathcal{T}_c$, each of which requires $O(|\mathcal{H}|^2)$ time to compute, resulting in an $O(|\mathcal{E}||\mathcal{H}|^2)$ runtime. Accordingly, $\forall y$, the total runtime is $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ (line 5). Finally, $\forall y$, clique beliefs (line 7) and sepset beliefs (line 8) are computed in $O(|\mathcal{V}||\mathcal{Y}||\mathcal{H}|^2)$ and $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ time, respectively.

Thus, Algorithm 8 is performed at $O((|\mathcal{E}|+|\mathcal{V}|)|\mathcal{Y}||\mathcal{H}|^2)$. Since $|\mathcal{E}|=|\mathcal{V}|-1$ in a tree-structured graph*, the overall time complexity of Algorithm 8 is $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$. □

Using Lemma .0.2 and .0.4, we can prove Theorem 8.2.1:

Theorem 8.2. *If the latent variables $\boldsymbol{h}$ form a graph without loops, computation of the energy-based MC-HCRF's energy function $e(y, \boldsymbol{x}; \boldsymbol{\theta})$, $\forall y$, is tractable, which has the time complexity of $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$.*

Proof. Under the Kapur entropy constraints $\alpha \neq 1$, $\alpha > 0$, $\beta > 0$ and $\alpha + \beta - 1 > 0$, we obtain the following:

$$
\begin{aligned}
& e(y, \boldsymbol{x}; \boldsymbol{\theta}) \\
&= H_{\alpha,\beta}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) && \text{[via Lemma .0.2]} \\
&= \frac{1}{1-\alpha} \log \frac{\sum_{\boldsymbol{h}} \tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\alpha+\beta-1}}{\sum_{\boldsymbol{h}} \tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\beta}} \\
&= \frac{1}{1-\alpha} \left(\log \sum_{\boldsymbol{h}} \tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\alpha+\beta-1} - \log \sum_{\boldsymbol{h}} \tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})^{\beta} \right)
\end{aligned}
$$

Since $\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \theta) = \prod_i \psi_i(\boldsymbol{D}_i; \theta_i)$, we define the following quantities:

$$
\begin{aligned}
\tilde{P}_a(y, \boldsymbol{h}|\boldsymbol{x}; \theta) &= \textstyle\prod_i \psi_i^a(\boldsymbol{D}_i; \theta_i) \\
\tilde{P}_b(y, \boldsymbol{h}|\boldsymbol{x}; \theta) &= \textstyle\prod_i \psi_i^b(\boldsymbol{D}_i; \theta_i)
\end{aligned}
$$

where each potential has the same scope but new values:

$$
\begin{aligned}
\psi_i^a(\boldsymbol{D}_i; \theta_i) &= \psi_i(\boldsymbol{D}_i; \theta_i)^{\alpha+\beta-1}, \forall i \\
\psi_i^b(\boldsymbol{D}_i; \theta_i) &= \psi_i(\boldsymbol{D}_i; \theta_i)^{\beta}, \forall i
\end{aligned}
$$

As a result, we obtain:

$$
\begin{aligned}
& e(y, \boldsymbol{x}; \boldsymbol{\theta}) \\
&= \frac{1}{1-\alpha} \left(\log \sum_{\boldsymbol{h}} \tilde{P}_b(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta}) - \log \sum_{\boldsymbol{h}} \tilde{P}_b(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta}) \right) \\
&= \frac{1}{1-\alpha} \left(\log \tilde{P}_b(y, |\boldsymbol{x}; \boldsymbol{\theta}) - \log \tilde{P}_b(y, |\boldsymbol{x}; \boldsymbol{\theta}) \right)
\end{aligned}
$$

Lemma .0.4 demonstrates that $\tilde{P}_b(y, |\boldsymbol{x}; \boldsymbol{\theta})$ and $\tilde{P}_b(y, |\boldsymbol{x}; \boldsymbol{\theta})$ can be computed in an $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$ runtime using Algorithm 8. Therefore, computation of $e(y, \boldsymbol{x}; \boldsymbol{\theta})$ has the time complexity of $O(|\mathcal{E}||\mathcal{Y}||\mathcal{H}|^2)$. □

* Loopy graphs satisfy $|\mathcal{E}| \geq |\mathcal{V}|$.

Proof of Proposition 2 and 3

Proposition 2. *MLE-HCRF's energy function has the same form as the MC-HCRF's energy when $\alpha \to 0$ and $\beta = 1$.*

Proof. Because the Kapur entropy degrades to the Hartley function when $\alpha \to 0$ and $\beta = 1$, we obtain:

$$\begin{aligned} e(y, \boldsymbol{x}; \boldsymbol{\theta}) &= H_{0,1}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \\ &= -\log P(y|\boldsymbol{x}; \boldsymbol{\theta}) + H_{0,1}(P(\boldsymbol{h}|y, \boldsymbol{x}; \boldsymbol{\theta})) \\ &= -\log P(y|\boldsymbol{x}; \boldsymbol{\theta}) + \log M \end{aligned} \tag{3}$$

where M is the number of attributes in the inputs. Because $\log M$ is a constant, MLE and MC-HCRF models have the same form of energy function. □

Proposition 3. *The MM-HCRF model is equivalent to the MC-HCRF model, when $\alpha \to \infty$, $\beta = 1$, and the potentials have the form $\boldsymbol{\psi}(\boldsymbol{D}; \boldsymbol{\theta}) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\boldsymbol{D}))$.*

Proof. Since the Kapur entropy degrades to the Chebyshev norm when $\alpha \to \infty$ and $\beta = 1$, we obtain:

$$\begin{aligned} e(y, \boldsymbol{x}; \boldsymbol{\theta}) &= H_{\alpha,\beta}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \\ &\triangleq H_{\infty,1}(\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \\ &= -\log \max_{\boldsymbol{h}} (\tilde{P}(y, \boldsymbol{h}|\boldsymbol{x}; \boldsymbol{\theta})) \\ &= -\log \max_{\boldsymbol{h}} (\exp(\boldsymbol{\theta}^{\mathsf{T}} \cdot \boldsymbol{\phi}(\boldsymbol{h}, y, \boldsymbol{x}))) \end{aligned}$$

Since $\exp(\cdot)$ is monotonically increasing, we obtain:

$$\max_{\boldsymbol{h}} (\exp(\boldsymbol{\theta}^{\mathsf{T}} \cdot \boldsymbol{\phi}(\boldsymbol{h}, y, \boldsymbol{x}))) = \exp(\max_{\boldsymbol{h}} (\boldsymbol{\theta}^{\mathsf{T}} \cdot \boldsymbol{\phi}(\boldsymbol{h}, y, \boldsymbol{x})))$$

Then, the MC-HCRF's energy function satisfies:

$$\begin{aligned} e(y, \boldsymbol{x}; \boldsymbol{\theta}) &= -\log \exp(\max_{\boldsymbol{h}} (\boldsymbol{\theta}^{\mathsf{T}} \cdot \boldsymbol{\phi}(\boldsymbol{h}, y, \boldsymbol{x}))) \\ &= -\max_{\boldsymbol{h}} (\boldsymbol{\theta}^{\mathsf{T}} \cdot \boldsymbol{\phi}(\boldsymbol{h}, y, \boldsymbol{x})) \end{aligned} \tag{4}$$

Since MM and MC-HCRFs use the same per-sample loss function, i.e., the soft margin loss, the MM-HCRF model is a special instance of the MC-HCRF model, when $\alpha \to \infty$, $\beta = 1$, and $\boldsymbol{\psi}(\boldsymbol{D}; \boldsymbol{\theta}) = \exp(\boldsymbol{\theta} \cdot \boldsymbol{\phi}(\boldsymbol{D}))$. □

Proofs of Cognitive Model

Proof of Proposition 4 (I_I's properties).

Proof. If denominator in Definition 5 is 0, then limit is used. Given the normalizing constants $a = 2$ and $b = 1$:

1. If $k = 1$, $I_I(\boldsymbol{\theta}_s, k) = \frac{1}{a}\left(1 + b - \frac{\theta_2}{\theta_1}\right) = 1 - \frac{\theta_2}{2\theta_1}$. Since $\boldsymbol{\theta}_s$ is decreasingly sorted, satisfying $\theta_1 \geq \theta_2 \geq 0$, then $-\frac{\theta_2}{2\theta_1} \geq -0.5$. Thus, $I_I(\boldsymbol{\theta}_s, k) = 1 - \frac{\theta_2}{2\theta_1} \geq 0.5$

2. If $k = K$, $I_I(\boldsymbol{\theta}_s, k) = \frac{1}{a}\left(\frac{\theta_K}{\theta_1} - 1 + b\right) = \frac{\theta_K}{2\theta_1}$ Since $\boldsymbol{\theta}_s$ is decreasingly sorted, satisfying $\theta_1 \geq \theta_K \geq 0$, then $\frac{\theta_K}{\theta_1} \leq 1$. Thus, $I_I(\boldsymbol{\theta}_s, k) = \frac{\theta_K}{2\theta_1} \leq 0.5$.

3. First, we prove $I_I(\boldsymbol{\theta}_s, k) \geq 0$. Since $K \geq k > 0$ and $K \geq 2$, the second multiplier $F_2 = \frac{K-k}{K-1} + \mathbb{1}(k = K) > 0$. Given $b = 1$, the third multiplier satisfies $F_3 = \frac{\theta_k}{\theta_1} - \frac{\theta_{k+\mathbb{1}(k\neq K)}}{\theta_k} + b = \frac{\theta_k^2 + \theta_1(\theta_k - \theta_{k+\mathbb{1}(k\neq K)})}{\theta_1\theta_k}$. Since $\boldsymbol{\theta}_s$ is decreasingly sorted, then $\theta_1 \geq \theta_k \geq \theta_{k+\mathbb{1}(k=K)} \geq 0$. Thus, $F_3 \geq 0$. Equality is obtained when $\theta_k = \theta_{k+\mathbb{1}(k=K)} = 0$. Since $a > 0$, $F_2 > 0$ and $F_3 \geq 0$, then $I_I(\boldsymbol{\theta}_s, k) = \frac{1}{a} \cdot F_2 \cdot F_3 \geq 0$. Now, we prove $I_I(\boldsymbol{\theta}_s, k) \leq 1$. When $k = K$, by property 2, $I_I(\boldsymbol{\theta}_s, k) \leq 1$ directly holds. If $K > k \geq 1$, then $F_2 = \frac{K-k}{K-1} \leq 1$. Equality holds when $k = 1$. Since $\boldsymbol{\theta}_s$ is decreasingly sorted, satisfying $\theta_1 \geq \theta_k \geq \theta_{k+1} \geq 0$, then $\frac{\theta_k}{\theta_1} \leq 1$ and $\frac{\theta_{k+1}}{\theta_k} \geq 0$. Given $b = 1$, we have $F_3 = \frac{\theta_k}{\theta_1} - \frac{\theta_{k+1}}{\theta_k} + b \leq \frac{\theta_k}{\theta_1} + 1 \leq 2$. Equality holds when $\theta_k = \theta_1$ and $\theta_{k+1} = 0$. Thus, given $a = 2$, we obtain $I_I(\boldsymbol{\theta}_s, k) = \frac{1}{a} \cdot F_2 \cdot F_3 \leq 1$. Thus, $\forall \boldsymbol{\theta}$, $I_I(\boldsymbol{\theta}_s, k) \in [0, 1]$ holds.

4. Since $\forall k \in \{1, \cdots, K\}$, $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ satisfy $\theta_k = \theta'_k$ and $\theta_{k+\mathbb{1}(k=K)} = \theta'_{k+\mathbb{1}(k=K)}$, we obtain $I_I(\boldsymbol{\theta}_s, k) - I_I(\boldsymbol{\theta}'_s, k) = \frac{1}{a}\left(\frac{K-k}{K-1} + \mathbb{1}(k = K)\right)\left(\frac{\theta_k}{\theta_1\theta'_1}(\theta'_1 - \theta_1)\right)$. Since $\frac{K-k}{K-1} + \mathbb{1}(k = K) > 0$ and $\theta'_1 \geq \theta_1$, Then, $I_I(\boldsymbol{\theta}_s, k) - I_I(\boldsymbol{\theta}'_s, k) \leq 0$. Equality holds if $\theta'_1 = \theta_1$ or $\theta_k = 0$. Thus, $I_I(\boldsymbol{\theta}_s, k) \leq I_I(\boldsymbol{\theta}'_s, k)$.

5. Since $\forall k \in \{1, \cdots, K\}$, $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ satisfy $\theta_1 = \theta'_1$ and $\theta_k = \theta'_k$, we obtain $I_I(\boldsymbol{\theta}_s, k) - I_I(\boldsymbol{\theta}'_s, k) = \frac{1}{a}\left(\frac{K-k}{K-1} + \mathbb{1}(k = K)\right)\left(\frac{1}{\theta_k}(\theta'_{k+\mathbb{1}(k=K)} - \theta_{k+\mathbb{1}(k=K)})\right)$. Since $\frac{K-k}{K-1} + \mathbb{1}(k = K) > 0$ and $\theta_{k+\mathbb{1}(k=K)} \geq \theta'_{k+\mathbb{1}(k=K)}$, Then, $I_I(\boldsymbol{\theta}_s, k) - I_I(\boldsymbol{\theta}'_s, k) \leq 0$. Equality holds if $\theta_{k+\mathbb{1}(k=K)} = \theta'_{k+\mathbb{1}(k=K)}$. Thus, $I_I(\boldsymbol{\theta}_s, k) \leq I_I(\boldsymbol{\theta}'_s, k)$ holds.

6. Since $\forall k \in \{1, \cdots, K\}$, $\boldsymbol{\theta}_s$, $\boldsymbol{\theta}'_s$ satisfy $\theta_k = \theta'_k$ and $\theta_{k+\mathbb{1}(k=K)} = \theta'_{k+\mathbb{1}(k=K)}$, we obtain $I_I(\boldsymbol{\theta}_s, k) - I_I(\boldsymbol{\theta}'_s, k) = \frac{1}{a}\left(\frac{K-k}{K-1} + \mathbb{1}(k = K)\right)\left(\frac{1}{\theta_1} + \frac{\theta_{k+\mathbb{1}(k=K)}}{\theta_k\theta'_k}\right)(\theta_k - \theta'_k)$. Since

$\frac{K-k}{K-1}+\mathbb{1}(k=K)>0$ and $\frac{1}{\theta_1}+\frac{\theta_{k+\mathbb{1}(k=K)}}{\theta_k\theta'_k}\geq 0$, and $\theta_k>\theta'_k$, Then, $I_I(\boldsymbol{\theta}_s,k)-I_I(\boldsymbol{\theta}'_s,k)\geq 0$. Equality holds if $\theta_k=\theta'_k$. Thus, $I_I(\boldsymbol{\theta}_s,k)\geq I_I(\boldsymbol{\theta}'_s,k)$ holds.

7. $\forall k,\, k'\in\{1,\ldots,K\}$ satisfying $k\leq k'<K$, and $\forall\boldsymbol{\theta}_s,\, \boldsymbol{\theta}'_s$ satisfying $\theta_{k+1}=\theta'_{k'+1}$, $\theta_1=\theta'_1$ and $\theta_k=\theta'_{k'}$, we obtain $I_I(\boldsymbol{\theta}_s,k)-I_I(\boldsymbol{\theta}'_s,k')=\frac{1}{a(K-1)}\left(\frac{\theta_k}{\theta_1}-\frac{\theta_{k+1}}{\theta_k}+b\right)(k'-k)$. Since $K>1$, $\frac{\theta_k}{\theta_1}-\frac{\theta_{k+1}}{\theta_k}+b\geq 0$, and $k'\geq k$, $I_I(\boldsymbol{\theta}_s,k)-I_I(\boldsymbol{\theta}'_s,k')\geq 0$, with equality holding when $\theta_k=\theta_{k+1}=0$ or $k=k'$. Thus, $I_I(\boldsymbol{\theta}_s,k)\geq I_I(\boldsymbol{\theta}'_s,k')$ holds.

□

Proof of Proposition 5 (I_I's relationship to I_A).

Proof. Given an observation $\boldsymbol{w}$, the accuracy metric I_A indicates whether the recognition result $y(\boldsymbol{w})$ matches the ground truth g. Formally, I_A is defined as follows:

$$I_A(y(\boldsymbol{w}),g)=\mathbb{1}(y(\boldsymbol{w})=g). \tag{5}$$

With this definition, we prove that I_A is a special case of our I_I indicator in Definition 5 when $\theta_1=1.0$, $\theta_2=\ldots=\theta_K=0$, and $k=1$ or $k=K$.

Given the normalizing constants $a=2$ and $b=1$, when $\theta_1=1.0$, $\theta_2=\ldots=\theta_K=0$, and $k=1$ (i.e., the recognition result $y(\boldsymbol{w})$ matches the ground truth g), we obtain:

$$I_s(\boldsymbol{\theta}_s,1)=\frac{1}{a}\left(\frac{K-1}{K-1}+0\right)\left(\frac{\theta_1}{\theta_1}-\frac{\theta_2}{\theta_1}+b\right)=\frac{b+1}{a}=1.$$

When $k=K$ (i.e., $y(\boldsymbol{w})\neq g$), we obtain:

$$I_s(\boldsymbol{\theta}_s,K)=\frac{1}{a}\left(\frac{K-1}{K-1}+1\right)\left(\frac{\theta_K}{\theta_1}-\frac{\theta_K}{\theta_K}+b\right)=\frac{2(b-1)}{a}=0.$$

Combining both cases, we obtain:

$$\begin{aligned} I_s(\boldsymbol{\theta}_s,k) &= \begin{cases} 1 & \text{if } k=1 \ \ (\text{i.e.}, y(\boldsymbol{w})=g) \\ 0 & \text{if } k=K \ \ (\text{i.e.}, y(\boldsymbol{w})\neq g) \end{cases} \\ &= \mathbb{1}(y(\boldsymbol{w})=g). \end{aligned} \tag{6}$$

We observe Eq. (5) is equivalent to Eq. (6), and thereby prove that I_A is a special case of the I_I indicator in the cases when $\theta_1=1.0$, $\theta_2=\ldots=\theta_K=0$, and $k=1$ or $k=K$.

□

Vita

Hao Zhang was born in Anshan, Liaoning, P. R. China. He received his B.S. degree in Electrical Engineering from the University of Science and Technology of China in 2006, and earned his M.S. degree in Electrical Engineering from Chinese Academy of Sciences in 2009. In 2014, he completed his Ph.D. degree in Computer Science under the supervision of Dr. Lynne E. Parker in Distributed Intelligence Laboratory at the University of Tennessee. During summer 2011, he interned in Oak Range National Laboratory on Scientific Computing. He has been awarded the Chancellor's Honors Award for Extraordinary Professional Promise at the University of Tennessee.