A Generic Gesture Recognition Approach based on Visual Perception

by

Gang Hu

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
June 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "A Generic Gesture Recognition Approach based on Visual Perception" by Gang Hu in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Dated: June 22, 2012

External Examiner: _____

Research Supervisor: _____

Examining Committee: _____

_____

_____

Departmental Representative: _____

DALHOUSIE UNIVERSITY

DATE: June 22, 2012

AUTHOR: Gang Hu

TITLE: A Generic Gesture Recognition Approach based on Visual Perception

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: PhD CONVOCATION: October YEAR: 2012

_____

Signature of Author

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Current developments of hardware devices have allowed the computer vision technologies to analyze complex human activities in real time. High quality computer algorithms for human activity interpretation are required by many emerging applications, such as patient behavior analysis, surveillance, gesture control video games, and other human computer interface systems. Despite great efforts that have been made in the past decades, it is still a challenging task to provide a generic gesture recognition solution that can facilitate the developments of different gesture-based applications.

Human vision is able to perceive scenes continuously, recognize objects and grasp motion semantics effortlessly. Neuroscientists and psychologists have tried to understand and explain how exactly the visual system works. Some theories/hypotheses on visual perception such as the visual attention and the Gestalt Laws of perceptual organization (PO) have been established and shed some light on understanding fundamental mechanisms of human visual perception. In this dissertation, inspired by those visual attention models, we attempt to model and integrate important visual perception discoveries into a generic gesture recognition framework, which is the fundamental component of full-tier human activity understanding tasks.

Our approach handles challenging tasks by: (1) organizing the complex visual information into a hierarchical structure including low-level feature, object (human body), and 4D spatiotemporal layers; 2) extracting bottom-up shape-based visual salience entities at each layer according to PO grouping laws; 3) building shape-based hierarchical salience maps in favor of high-level tasks for visual feature selection by manipulating attention conditions of the top-down knowledge about gestures and body structures; and 4) modeling gesture representations by a set of perceptual gesture salience entities (PGSEs) that provide qualitative gesture descriptions in 4D space for recognition tasks. Unlike other existing approaches, our gesture representation method encodes both extrinsic and intrinsic properties and reflects the way humans perceive the visual world so as to reduce the semantic gaps. Experimental results show our approach outperforms the others and has great potential in real-time applications.

LIST OF ABBREVIATIONS USED

PGSE	Perceptual Gesture Salience Entity
PO	Perceptual Organization
HCI	Human-Computer Interface
STIP	Space-Time Interest Point
MEI	Motion Energy Image
MHI	Motion History Image
PCPG	Perceptual Curve Partition and Grouping
GET	Generic Edge Token
CPP	Curve Partition Point
TOF	Time-of-Flight
FOV	Field of View
SR	Spectral Residual
PQFT	Phase Quaternion Fourier Transform
MSER	Maximally Stable Extremal Region
SIFT	Scale-Invariant Feature Transform
SVD	Singular Value Decomposition
LTI	Linear Time Invariant
ST	Space-Time
MACH	Maximum Average Correlation Height
HOG	Histograms of Gradient
HOF	Histograms of Optic Flow
BOW	Bag of Words
FSM	Finite State Machine
TDNN	Time-Delay Neural Network
HMM	Hidden Markov Model
CRF	Conditional Random Fields
HCRF	Hidden-state Conditional Random Field
LDCRF	Latent-Dynamic Conditional Random Field
SVM	Support Vector Machine
OPABS	Order Preserving Arctangent Bin Sequence
TSW	Tangent Sliding Window
CVD	CPP Vertical Distribution
KNN	K-Nearest Neighbor
MST	Minimal Spanning Tree
PR	Precision-Recall
RANSAC	RANdom SAmple Consensus
MRF	Markov random Field
LOO	Leave-One-Out
RBF	Radial Basis Function

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Professor Dr. Qigang Gao, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for giving me opportunities to grow as a researcher. Your advice on research, career development, and life attitude has been invaluable.

I would also like to thank my external reader, Dr. Minglun Gong, for your commitment to my research. For my internal committee members, Dr. Norman Scrimger, Dr. Evangelos E. Milios, and Dr. Stephen Brooks, thanks for your support, not only in the final thesis defense, but also in the aptitude exam, thesis proposal, and my entire PhD study stage. I want to thank all of you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would especially like to thank the MITACS program, two industrial partners from the maritime region, and some colleagues for conducting experiments and collecting research data. This thesis could not be accomplished without the supports from you.

A special thanks to my family. Words cannot express how grateful I am to my beloved wife, Amber, for all of the sacrifices that you've made on my behalf. Thank you for supporting me for everything, and especially I can't thank you enough for encouraging me throughout this experience. Last but not least, I would like to thank my Mom for your constant support and patience, through my PhD study as well as through my life in general. I dedicate this thesis to them.

CHAPTER 1 INTRODUCTION

Research about human gesture and action recognition has been attracting increasing attention recently due to high demands from emerging applications. Despite the efforts that have been made in the past decades, it is still a very challenging task to build a generic gesture recognition solution that can support the developments of various gesture-based applications with robust performance. In this chapter, we briefly introduce the background of this research domain, several challenges, our approaches and broad application usages.

1.1 DOMAIN BACKGROUND

Visual content analysis involves large size datasets and high computational costs. Current developments of video capture technology, 3D imaging, computing power, storage capacity, and broadband networking have matured, and allowed the computer vision technologies to analyze complex human activities from image data in real time. Meanwhile the demands for human activity recognition technologies are increasing from many domains, such as surveillance systems, healthcare, sports training, video search and other systems that involve interactions between persons and electronic devices such as human-computer interfaces (HCI) and gesture-based video games. Therefore, both advanced hardware and high demanding applications are driving forces for the development of efficient computer vision algorithms for event and human activity recognition tasks.

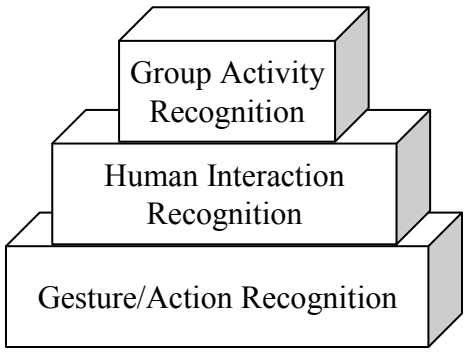


Figure 1 Hierarchy of Human Activity recognition.

Full-tier human activity interpretation is the ultimate goal, which cannot be fulfilled without conquering several underlying obstacles. Human activities are diverse, i.e. could be simple or very sophisticated depending on the motion intentions and environmental conditions and can be categorized into several levels: gestures, actions, human-human or human-object interactions, and group activities (see Figure 1). Gestures are basic movements of a person's body parts and are the atomic components of meaningful human activities. “Stretching an arm” and “raising a hand” are the typical examples. Actions are single person activities that may be composed of multiple gestures, such as “throwing”, “waving”, and “punching”. Interactions are human activities that involve two or more persons or other objects, such as two people “shaking hands with each other”. Group activities are performed by multiple persons, such as a marathon match. Among these categories, gestures and actions are at the basic layer of the human activity hierarchy. While the goal of understanding all types of human activities is ambitious and beyond the scope of this dissertation, we focus on the fundamental part and propose approaches to the gesture and action recognition tasks that are instrumental towards the ultimate goal. Without solid components for basic human gesture/action recognition, understanding sophisticated real-time human activities and events from visual data is impossible.

Machine-based human gesture/action understanding is a challenging task. In the past decades, there have been many approaches dealing with gesture/action recognition in many ways. However, most systems are too domain-specific and their performances still cannot compete with human vision; the semantic gaps between low-level visual features and human perception are not well bridged. For instance, some technologies are able to detect the object movements occurring in the recorded video or real-time image frames but fail to provide more intelligent interpretation. Figure 2 lists several human motion representation methods. Figure 2(a3) is the result of the frame difference between 2 consecutive frames Figure 2(a1) and (a2). Figure 2(b3) is a static image that records the history of several previous inter-frame differences. Many small yellow arrows in Figure 2(c2) indicate the optical flow vectors that occur during the motion (Figure 2(c1)). Figure 2(d6) is the 3D pattern of Space-Time Interest Points (STIP) on the legs detected

in 3D (XYT) space (Figure 2(d1)-(d6)). We can see that the human movements can be recognized in those representation methods, but they cannot directly provide qualitative descriptions about the movements and create barriers for deep understanding. Without specific domain knowledge, computer vision algorithms can only perceive that something is moving, but not its perceptual details with more semantics. As a consequence, robots/applications have difficulties answering some "simple" questions, such as "Is there a human in the scene?" or "What are the exact movements of the human in the scene?" or even "How do I (a robot or avatar) mimic and learn the human gesture/action?", or "Could you describe the motion qualitatively and quantitatively?"

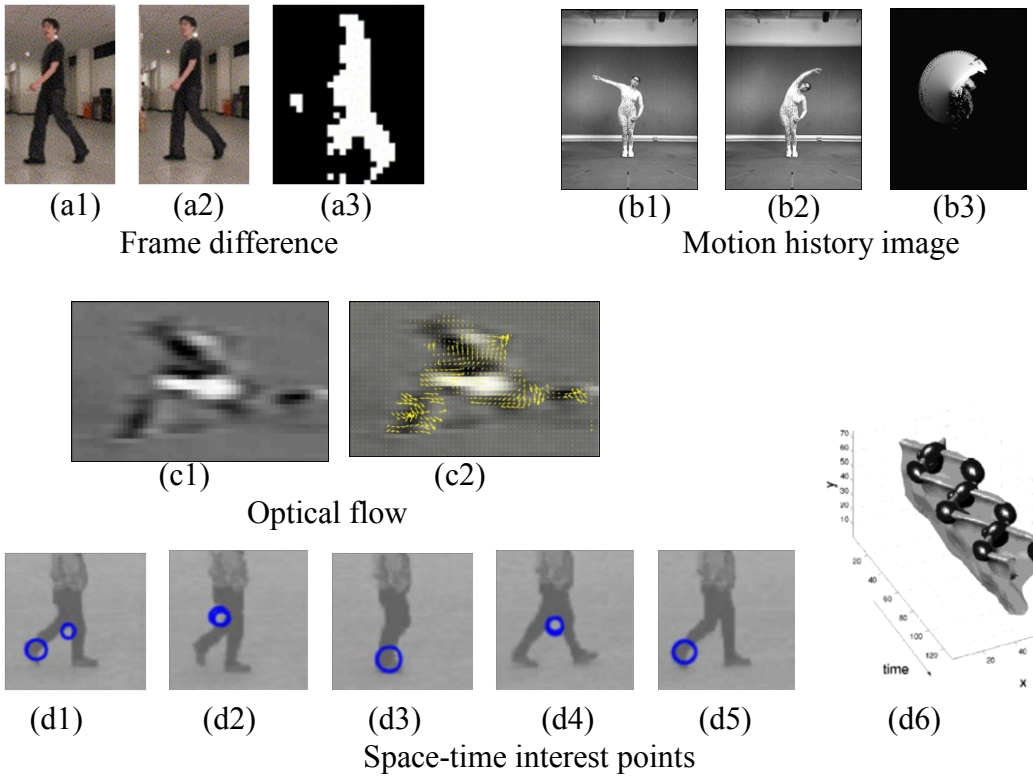


Figure 2 Without high-level semantic interpretation, low-level motion detection methods cannot answer even simple questions.

In contrast, humans can interpret body movements and actions and are nearly able to answer those "simple questions" subconsciously and effortlessly. Psychologists have revealed that human communications largely rely on body languages and other non-verbal cues [1]. In order to make machines gain the comparable performance of human visual perception, understanding the mechanisms of the human visual perception system

is essential. With countless efforts made by generations of researchers, several studies have shown that perception of human motion, under some circumstances, requires focused visual attention [2][3][4]; meanwhile, Perceptual Organization (PO) laws also play roles in the attention process [5]. These research findings shed light on understanding the mysteries of sophisticated biological neural systems. Several computational models of vision systems [6]-[11] based on the theories/hypotheses of both visual attention and perceptual organization laws have demonstrated the effectiveness in object recognition tasks. We believe that by applying computer vision algorithms based on those visual mechanisms, the semantic gaps between visual features and high-level perceptions can be reduced, and the performance of human action/gesture recognition systems can be improved.

The goal in this dissertation is to develop a generic framework that can answer those "simple" questions mentioned earlier, ultimately leading to machines understanding all kinds of human activities. The following section addresses the challenges we faced in achieving this goal.

1.2 CHALLENGES

We encountered several challenges when dealing with the computer vision-based human gesture/action recognition. One major challenge is perceptual feature extraction from visual data. Human visual pre-attention can be drawn to visual salience entities automatically. It is essential for a visual analysis system to detect and select relevant visual salience entities among numerous low-level features and build a semantic feature representation. According to the visual attention theories [12][13], features are components of target objects, and act as the visual saliency in visual attention. In the computer vision community, existing algorithms are able to accomplish some early-stage vision tasks with the same or even better performance than human vision. Those tasks include searching targets pre-defined by specific features within a cluttered image, counting colored elements, etc. However, in terms of high-level understanding, human

vision performs much better than any current computer algorithm does. One of many reasons is that the low-level features do not contain high-level semantics. Many vision cues have been explored as vision features, such as color, texture, shape and edge features. Obviously, color will play no role under some conditions, and texture patterns are not generic for different applications. Human vision largely relies on shapes, but many filtering-based shape features provide global measures without local details. What we need are the 3D perceptual shape features that are able to reflect the visual saliency and lead to target objects.

A second major challenge is the object recognition from the bottom-up pixel-level features. The target object in our case is the human body with the articulated structure (including head, torso and limbs) that has high degrees of freedom. To achieve the goal of human gesture/action recognition with robust performance, body parts segmentation, classification and pose estimation are required, i.e. a divide-and-conquer method is often applied to recognize individual body parts accurately first; and then, a robust pose estimation solution is used to provide seamless grouping for motion analysis. Previously, researchers have made efforts to develop approaches in this domain, and some of them have achieved encouraging performance for specific applications. However, there is no generic solution that satisfies all expectations, i.e. easily to be applied to different real-time applications under all circumstances. Since the configuration space of body poses is huge (i.e. exponential of the multiple body parts with various parameters), selective computing on most characteristic body features would not only reduce the ambiguities, but also improve the efficiency to make real-time gesture/action analysis possible. Therefore we need to capture the visual salient entities at the object level from the scene to facilitate the pose estimation.

A third major challenge is forming gesture/action representations in the 4D spatiotemporal space. Human gestures/actions are the patterns of body poses along the temporal dimension. How can we represent these patterns in the 4D spatiotemporal space in a way that can perceptually deliver the semantics? A wide variety of human action/gesture representations have been proposed, including Motion Energy Image

(MEI), Motion History Image (MHI), optical flow, Space-time Interest Points (STIP)-based descriptor etc. But they only work well in specific domains, and are unlikely to be able to act as the generic solutions. To reflect the visual attention conditions and deliver more semantics, we need a generic approach to provide qualitative descriptors for gestures/actions, which should be similar to how humans interpret the visual observations. Both external visual stimuli and internal correlation should be encoded within this generic representation, and then any human gesture/action can be described qualitatively and quantitatively.

A fourth challenge is dealing with the ambiguity in the recognition tasks. In real situations, even a simple human gesture/action with straightforward semantics contains many variations coming from different view-points, various environmental conditions (illumination, distance, clutter background etc.), and diverse human body shapes, behaviors and even cultural backgrounds. To handle the ambiguities, we must mimic the abilities of human vision and perception to take a statistical approach, learning the regularities of the human action properties and finding the most likely interpretation of a motion. We need to have a statistical approach to classify and recognize gestures.

1.3 OUR APPROACH

Great oaks from little acorns grow. It is ambitious that we provide solutions that attempt to make machines/robots understand all kinds of human activities semantically in real time. Therefore, we tackle the problems from the very fundamental stage (i.e. gesture and action recognition) in a generic way towards the ultimate goal. The proposed system is a 3D gesture recognition framework using hierarchical shape-based salience maps. Inspired by the biological mechanisms of human visual attention and perceptual organization, the models based on bottom-up and top-down visual salience map principles are employed within this hierarchical framework. Using our approach, we can go as far as to recognize complex human activities from the visual world.

As we mentioned earlier, research studies have shown that the visual perception of the human motion requires focused visual attention [2]-[4]. It has been largely agreed that perceptual organization (PO), the visual processes structuring pieces of visual information into coherent objects, exists in human visual systems. Both perceptual organization and visual attention are crucial for human visual perception [5]. These findings shed light on the fundamental mechanisms of human visual perception. Our approach will involve several terms related to human visual perception from the neuroscience and psychology areas, such as perceptual organization (PO), visual attention, visual salience, and salience maps. These definitions can be found in Chapter 2.

Visual attention and perceptual organization are indispensable perceptual processes for high-level visual perception tasks. According to most influential computational models of visual attention [12][13], both bottom-up and top-down information are responsible for the deployment of vision attention. The bottom-up stimuli-driven visual salient entities are weighted by the top-down task-specific knowledge and grouped into salience maps with selected visual information. The PO grouping laws [2] play vital roles during the salience map integration. Mechanisms from visual attention and perceptual organization work together for selecting and filtering visual information so as to make the human visual perception efficient and effective. These visual mechanism theories/hypotheses stem from plausible biological evidence. Studies on the neural systems of primates and humans suggest that the posterior parietal cortex may encode a visual salience map, while the pre-frontal cortex may encode a top-down prior knowledge. By integrating both, a formed attention guidance map, so-called salience map, is possibly stored in the superior colliculus [19]-[22]. Several computational models of vision systems [14]-[18] based on these visual attention theories have demonstrated the advances in object recognition and action interpretation tasks. Among them, hierarchical attention models provide a framework to formulate the salience maps at the low-level visual feature and the object level respectively. Figure 3 shows a simplified hierarchical structure of visual attention and perceptual organization models, summarized from existing research that encodes both attention and organization mechanisms at multi-levels.

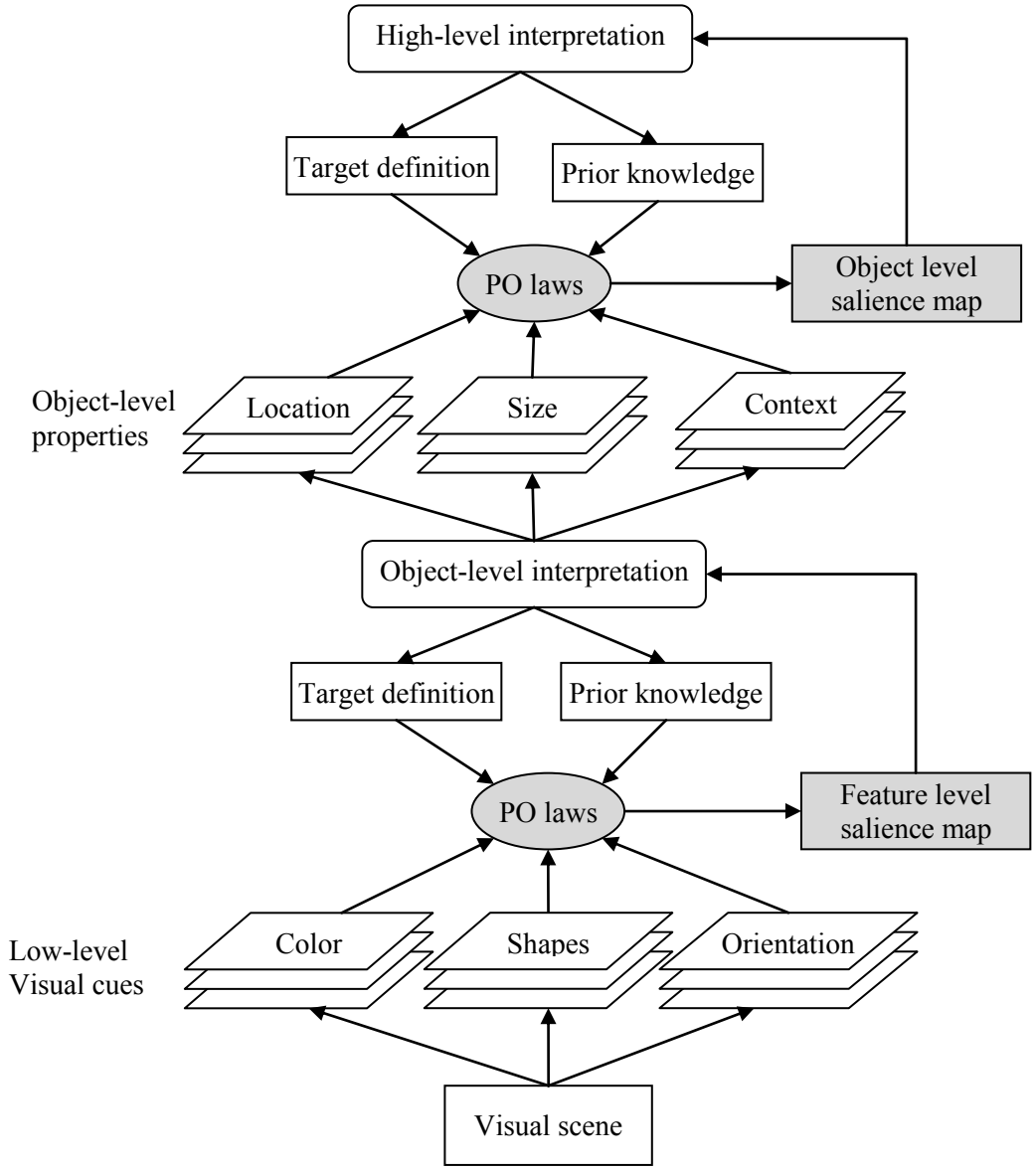


Figure 3 A hierarchical visual attention model incorporating perceptual organization.

Our approach is inspired by these research findings to handle the challenging gesture/action interpretation tasks in a systematic, coherent and biologically plausible manner. There are 3 levels of visual attention and perceptual organization processes jointed to select and organize the visual information for recognition tasks, feature level, object level and 4D spatiotemporal level. Shape feature is one of the most effective visual stimuli that human visual perception heavily relies on. At the feature level, perceptual shape features are extracted as visual stimuli for vision analysis. According to the

discoveries about generic criteria employed in human vision, a Perceptual Organization (PO)-based method is used to extract visual salient entities from edge pixels for object coding. A well-studied PO-based Perceptual Curve Partition and Grouping (PCPG) model is implemented and enhanced in this work. Based on the classical Gestalt Laws, image contents are perceived as a set of salience entities, generic shape tokens and critical structure points, which are grouped into a 2D feature-level salience map for each image. By taking advantage of 3D data, a 3D shape-based salience map is able to reflect the bottom-up attentional information, describe the properties or semantics of any visual object perceptually, and provide the selected features for further top-down processing. This 3D feature level shape-based salience map can also be extended to build a comprehensive visual descriptive language for shape-based content coding, pattern recognition and indexing for images.

The target object in a gesture/action recognition system is the human body, whose articulated structure has high degrees of freedom. Understanding gestures/actions of such a complicated object structure from a real-time visual stream is a challenging task. To achieve the goals of gesture/action recognition, a divide-and-conquer method is often applied to recognize individual body parts accurately first, and then estimate their spatial layouts to derive the pose status from every frame for motion analysis. By utilizing the prior kinematic knowledge about the body structure, the bottom-up 3D shape-based salience map is enriched in a top-down process by selecting, weighting and grouping 3D shape salience entities for individual body part recognition. Several grouping laws of Perceptual Organization are performed during the salience map generation.

Based on the classification results, a limb tree is built as an index for building the bottom-up salience map at the body object level. Meanwhile, prior knowledge about the body poses is encoded into the tree traversal criteria in a top-down process to weight the classified object salience entities. The updated body part results are the elements of the salience map at the object level, which contains selective salient visual information that provides a reduced search space to speed up the complex pose estimation process and benefit the gesture feature extraction/selection in 4D spatiotemporal space.

Finally, based on the results of body part classification and pose estimation processes, several dynamic properties of individual body parts are tracked in 4D spatiotemporal space. A set of Perceptual Gesture Salience Entities (PGSEs) are defined as the descriptors for the spatiotemporal patterns according to the law of continuity of perceptual organization. The combination of these gesture/action descriptors is the salience map at the 4D spatiotemporal level in which the bottom-up visual information for supporting high-level recognition tasks is selected under the influences of prior knowledge and target gesture definitions. The ultimate high-level interpretation tasks will be benefited by this novel PGSE-based gesture salience map. Figure 4 shows the overall structure of our proposed gesture/action recognition framework. A 3D camera is adopted to capture real-time depth image sequences that contain human gestures/actions. Feature, object and spatiotemporal level salience maps are the 3D Generic Edge Token (GET)/Curve Partition Point (CPP) maps, weighted limb regions and PGSE block patterns respectively. The bottom-up salience entities and the task-specific top-down knowledge are fused together by certain PO laws to provide selected attention-based gesture representation, which can reduce the search space and retain the discriminative power needed in the high-level recognition tasks.

Another important idea in our method is utilizing both internal and external properties of gesture/action patterns in the recognition process. We argue that any gesture contains both extrinsic and intrinsic patterns in 4D space. Extrinsic properties mainly describe the spatial temporal components during the dynamic, e.g. motion direction, velocity, shape changes, etc. Intrinsic patterns are not visually apparent, such as temporal and spatial ordering, and cannot be easily modeled; however, they play vital roles in classification tasks. Rather than spending time figuring out the temporal or spatial ordering among multiple body parts and dynamic components, several PGSE-based representations are able to encode both internal and external properties to provide a coherent interpretation about sophisticated human activities by employing a certain statistical method. This approach helps to provide robust recognition performance, as demonstrated in our experiments.

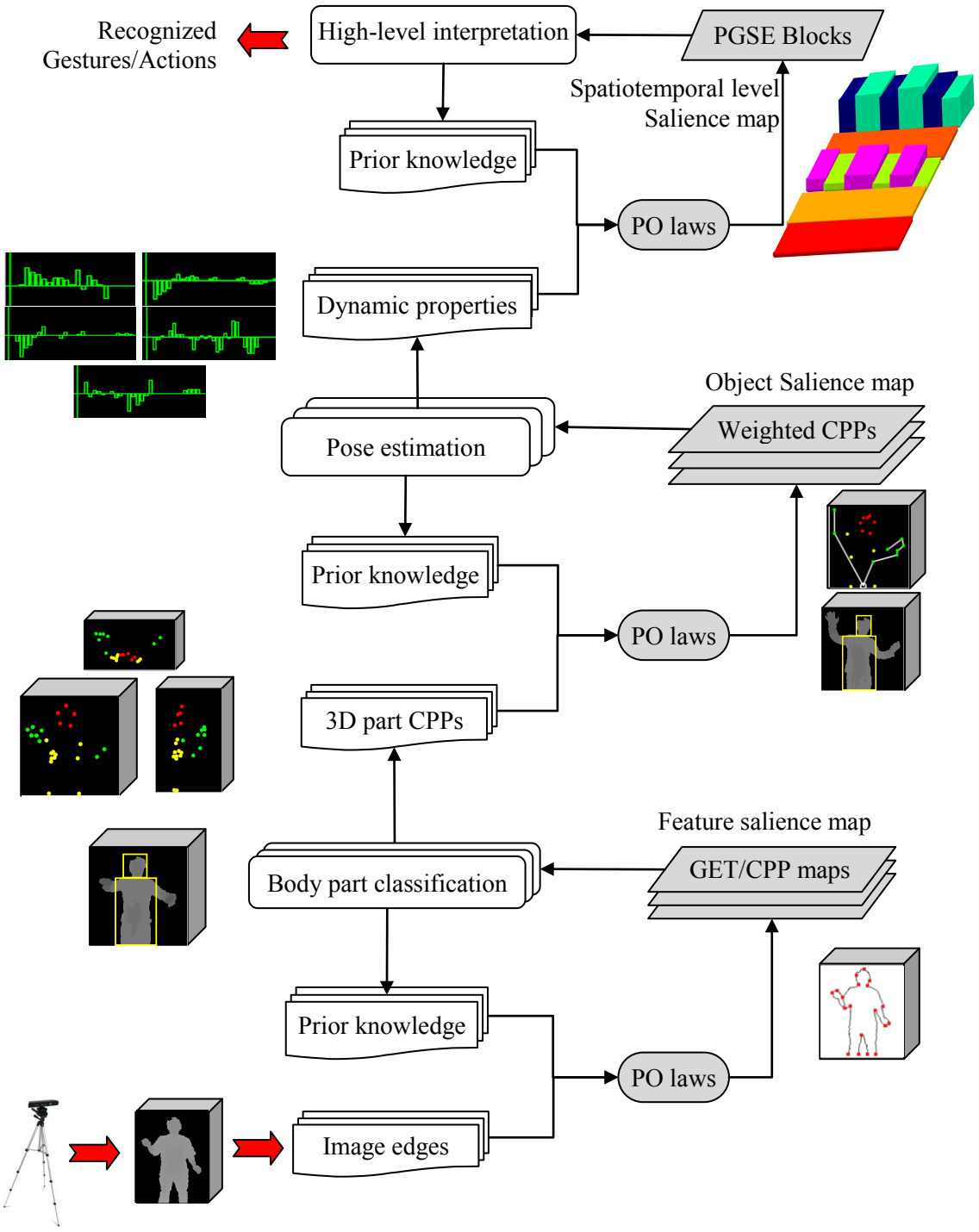


Figure 4 Multi-level visual salience model for human action/gesture interpretation.

1.4 APPLICATIONS

Human action/gesture recognition is a fundamental component of human activity analysis, and is useful in a wide range of applications. If we can recognize all of the human body parts and understand their dynamic extrinsic and intrinsic properties of the movements, any high-level complex activity can be interpreted semantically.

In the health care sector, it would be useful to understand the gesture languages of elderly or disability people who have difficulties expressing themselves normally. The life for deaf people would be a lot easier if there is a system that can automatically translate his/her sign language into scripts that other people can understand without hiring a special interpreter.

Gesture recognition can also be used for home entertainment. Users use their gestures to change TV channels or play video games. Gesture-based video game is currently a key area in the game industry. Three major companies have been making significant efforts to make the next-generation video games more attractive. Neither Nintendo's gesture game platform Wii [26] nor Sony PlayStation Move [27] fully relies on vision data to form gesture features and corresponding representations. Microsoft Kinect game platform uses a 3D camera to capture target gestures, and its recognition method is mainly built on model-based computer vision technology. According to a recent survey, Microsoft's Xbox 360 has gained great success in the video game market, and a majority of the increasing sales are from Kinect sensor-based gesture games [28].

Most current video surveillance systems need a human operator to constantly monitor them. Their effectiveness and response is largely determined by the due diligence of the person monitoring the camera system, instead of the technological capabilities. Some automated systems only detect general motions and location patterns without telling what the causes are. To overcome these limitations of traditional surveillance methods,

computer vision and artificial intelligence-based automated systems are required for the real-time monitoring of people, vehicles, and other objects. Automated surveillance systems in public places detect abnormal and suspicious activities as opposed to normal activities. For instance, an airport surveillance system must be able to automatically recognize suspicious activities like “a person leaving a bag”, “a person placing his/her bag in a trash bin” or “baggage theft”. The real-world scenarios are more complicated than the game environment, and involve not only multiple human activities, but also the a variety of object categories, and diverse backgrounds in indoor and outdoor environments etc. Gesture and action recognition is a significant step in the development of intelligent surveillance systems.

CHAPTER 2 RELATED WORK

3D Gesture modeling and recognition involve several research domains including image processing, 3D data estimation, pattern recognition and machine learning. There have been plenty of approaches for dealing with these challenging tasks in the past decades. Here we briefly discuss some related topics, including advanced 3D optical devices, low-level salient features, gesture representation, and gesture recognition methods.

2.1 3D CAMERAS

When humans perceive the world, they see not just a pattern of 2D color and texture, but the 3D visual objects. In the same way, computer vision algorithms must go beyond the pixels and reason about the 3D world. There are several ways of obtaining 3D data from images. Some techniques are capable of directly deriving 3D data from 2D single view images under certain constraints and assumptions. Since a single conventional 2D camera cannot directly provide 3D measurement for objects in the scene, 3D spatial data are estimated by a mapping method (map the 2D image into 3D space). It needs the support from the following processes: 1) domain knowledge collection, 2) 3D world coordinates setting and calibration, and 3) indexing generation and matching for 3D data estimation. Despite the availability of various feasible algorithms, the issues of reliability and efficiency of these methods still remain unsolved. Therefore deriving 3D data from 2D single view images is seldom used in real applications.

With the development of the optical device technologies, 3D cameras can efficiently provide accurate and reliable 3D images. On the current market, there are three types of 3D cameras: stereo camera, Time-of-Flight (TOF) camera, and speckle pattern camera that provide both reliable 3D spatial data and color/grayscale images simultaneously.

2.1.1 Stereo Camera

Humans understand depth based on the differences in appearance between the left and right eyes; so does the stereo camera. As we know, under some simple imaging configurations (both eyes looking straight ahead), the amount of horizontal disparity is inversely proportional to the distance from the observer. This basic physics and geometry relating visual disparity to scene structures are well understood and applied to the stereo cameras for estimating depth data. A stereo camera has two or more lenses with an image sensor for each. This configuration allows the camera to simulate human binocular vision, as can be seen by the differences between Figure 5(a) and (b), where the foreground objects shift left and right relative to the background. By using the stereo matching method [23] that finds matching pixels in two or more images captured from the left and right sensors in the same scene, the stereo camera is able to convert their 2D positions into 3D depths, and obtain the 3D model of the scene. Figure 5(c) is the estimated depth image derived from Figure 5(a) and (b). Stereo matching is one of the most widely studied and fundamental problems in the computer vision area [24]. Essentially the 3D data estimation methodology used by a stereo camera relies on well-defined image features or detailed textures and appropriate matching techniques. Such constraints often lead to large distance uncertainties if image pairs have the presence of non-textured areas or unmatched features.

(a) Image from the left sensor (b) Image from the right sensor (c) Derived depth image

Figure 5 Depth image derived from two images from a stereo camera.
The differences in the red circles in (a) and (b) show the horizontal disparity.

2.1.2 TOF Camera

A Time-Of-Flight camera (TOF camera) is a range imaging camera system that measures the depth based on the known speed of light. The entire scene is captured by a TOF camera using a set of laser or light pulses individually without scanning operations. TOF camera measures the round trip time of light from the light source to the objects in the field of view (FOV) and back to the sensor for calculating the depth data. The CCD/CMOS imaging sensor captures the returned signal from each light pulse for each pixel. The distance range is from a few meters up to about 60m, and the distance resolution is around 1 cm. The lateral resolution of time-of-flight cameras is currently low (320 × 240 pixels or less) compared to standard cameras. Although a TOF camera is sensitive to background light noise and multiple reflections, which may be controllable by some filtering methods, it provides a robust solution for obtaining 3D image data. Compared with stereo cameras, a TOF camera has the advantages of being simple, fast and with an efficient distance algorithm.

SR4000 (see Figure 6(a)), a primary TOF camera product from MESA Inc., was used for TOF performance evaluation. The depth data from SR4000 is measured by the round trip time of the infrared light (870nm wavelength), and the 2D grayscale image (176x144 pixels) is provided simultaneously. Within an evaluation system, every image captured by the SR4000 produces four types of outputs: distance image, grayscale image, segmented grayscale image and edge map of the segmented image, which allow us to examine the image qualities for gesture tracking and recognition. Figure 6(b) is the distance image. Every pixel of the distance image is a distance value from a surface point of an object within the FOV to the front face of the camera. The intensity of each pixel in the distance image is the value of Z coordinate in the Cartesian coordinate system where the origin is the center of the camera. A grayscale image is the result of the light saturation, which takes both background light and excessive reflected infrared light into consideration. Figure 6(c) shows a grayscale image which provides texture information of objects in the scene. Its quality is relatively poor compared with the images from conventional cameras. A background filtered grayscale image is shown in Figure 6(d). Rather than the entire scene, this image only presents the objects closer to the camera as

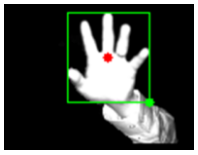
the original image (Figure 6(c)) has been filtered by a user-specified depth threshold. The shape salience features extracted from the filtered image are shown in Figure 6(e), and well reflect the object contour shapes. The efficiency of this TOF camera (SR4000) is high, 30+ fps is achieved for image edge extraction, background filtering, and depth data integration.

The accuracy of the distance data from SR4000 is affected by several factors, including lighting condition, speed of moving objects and the materials of nearby objects. The distance measurement will be distorted if there is other light with a similar wavelength in the scene. Some background objects with glass or mirror materials that produce reflections would also distort the distance measurement. Motion speed also affects the quality of the distance data. For a static or a slow moving object, the distance accuracy is high: its resolution is about one cm. If an object is moving fast, less accuracy is expected. The drawbacks of TOF camera (SR4000) are: 1) the image resolution is too low (176×144 pixels) to capture the necessary object details; 2) its angle of view is narrow for covering complete actions; 3) there is no color data that can be processed. Color information may be crucial for some cases; 4) TOF camera is expensive.

(a) SR4000 TOF Camera

(b) Depth image

(c) Grayscale Image



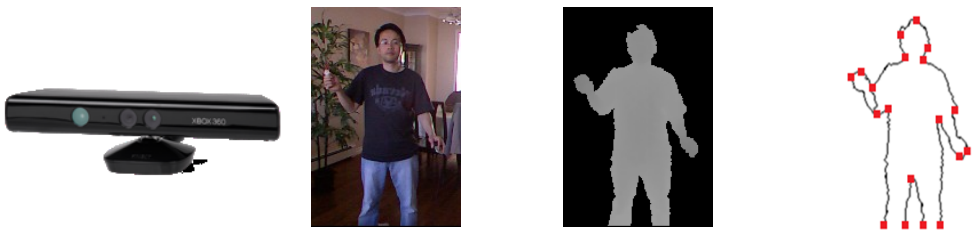
(d) Filtered image

(e) Salience features

Figure 6 Images and perceptual features from a TOF camera.

2.1.3 Kinect Camera

Another alternative solution for obtaining 3D data is the speckle pattern related method introduced in [25], where the camera system emits the infrared ray filtered by a diffuse object, ground glass, to generate random speckle patterns striking on the objects in FOV. The reflected speckle pattern contains inherent changes that uniquely characterize each location in 3D space. 3D spatial data are derived by analyzing the differences of the sensed speckle patterns. Microsoft Kinect's depth sensor (see Figure 7(a)) falls in this category. Besides the speckle pattern depth sensor, Kinect also uses another conversional 2D sensor to provide 2D color images with VGA (640×480) resolution. Both depth and color data are aligned accordingly. Currently Kinect is available on the market at low costs. We examined its performance by using the same evaluation system for the TOF camera. Figure 7(b) and (c) show the color image and its corresponding gray scaled depth image respectively. The depth image is derived by scaling each pixel's depth data into the intensity range [0-255], and is able to describe the object silhouette for gesture analysis. The accuracy of the depth data is affected by the lighting condition, speed of moving objects and the materials of nearby objects. However, the impact of the distorted data accuracy can be controlled by taking more local data into consideration, e.g. the depth value of a pixel is averaged from its neighbors. Thus the data uncertainty can be suppressed. Overall, the Kinect camera provides reasonable data quality with low economic cost for motion and gesture analysis, and is used in our research work.



(a) Kinect sensor (b) color image (c) depth image (d) edge map

Figure 7 Images from a Kinect camera.

2.2 SALIENCE-BASED LOW-LEVEL FEATURES

A human gesture is a spatial-temporal object pattern in 4D (XYZT) space. Low-level salient features from a single image reflect the gesture snapshot at a certain moment. The final gesture features are collectively parameterized by pixel-level salience from all images. Overall there are two types of image salient features, global and local. Global contrast-based image saliency is detected by evaluating the contrast against the entire image. By checking the pixel luminance contrast against others, Zhai and Shah [29] extracted the saliency efficiently. Achanta et al. [30] proposed a frequency-tuned method that calculates the pixel contrast against the average color and intensity of an entire image. Global contrast-based methods are simple and efficient, but fail to analyze complex variations due to the missing local details. Hou and Zhang's Spectral Residual (SR) approach [31] is based on the Fourier Transform. The SR difference between the original signal and a smoothed one in the log amplitude spectrum is calculated and transformed into the spatial domain, which acts as the salience map for visual analysis. Similarly a saliency map can also be calculated by the image's phase spectrum of the Fourier Transform [31]. Guo et al. [32] used a Phase Quaternion Fourier Transform (PQFT) method to calculate spatiotemporal saliency maps for both natural images and videos, considering color, orientation and motion.

Majority salience features are obtained by local-based methods, which use local contrast to find locations with higher saliency values near the boundaries of salient objects. Itti and Koch [33] proposed a biologically-plausible visual saliency model based on the center-surround contrast mechanism. Harel et al. [34] normalized the feature maps based on the Itti and Koch's approach [12] to highlight conspicuous parts and combine with other importance maps. Since Itti and Koch's linear model of the similarity measure on several cues is inconsistent with the properties of higher level human judgment, Gao et al. [35] proposed a decision-theoretic approach based on mutual information to measure the visual salience. Ma and Zhang [36] proposed a local contrast-based method using fuzzy growing to extract salient regions from images. More recently, Goferman et al. [37] simultaneously modeled local low-level clues, global considerations, visual organization

rules, and high-level features to highlight salient objects along with their contexts. Furthermore, some approaches compute saliency by applying machine learning techniques to fuse different visual feature channels. Alexe et al. combined multi-scale saliency, color contrast, edge density, and superpixels in a Bayesian framework [38]. In [39], the salient features are obtained by using conditional random fields to linearly combine multi-scale contrast in a Gaussian image pyramid, center-surround histograms and color spatial-distributions. However, the high computational cost is the weakness of these local-based models.

In the computer vision area, corner and point-based salient features from images have been widely studied and used in many image/video analysis applications. It is commonly agreed that an image's prominent points are salient to human vision, and can reveal more semantics of the image contents. Therefore, discovering salient points, extracting the properties of/around them, is a good strategy to bridge the semantic gaps between human perception and the image lower level features. The classical Harris detector [40] calculates the local auto-correlation function by measuring the local changes of the pixel values within a Gaussian window. The salient Harris corner points are invariant to rotation, scale, illumination variation and image noise. The Harris affine detector [41] is the extension of [40]. Having had initial Harris points detected, Gaussian-based multi-scale analysis is iteratively performed to obtain affine-invariant salient regions robust to shape deformation. The detected salience entities are invariant to scale, rotation and shearing. The Maximally Stable Extremal Region (MSER) detector [42] extracts sets of image elements invariant to affine transformation. The locations of MSERs are the salient regions where the transformations of coordinates and pixel intensity are continuous and monotonic respectively. David Lowe's approach [43] is another most influential solution using scale-invariant salient keypoints to extract visual features. By his method, keypoints are obtained from the pixel differences of image gradients, and then the gradient orientation histogram of the local area around each keypoint is built and normalized into a local descriptor - scale-invariant feature transform (SIFT), which outperforms other image descriptors reportedly in most cases. However, keypoints of SIFT are sensitive to noise, and its high dimension histogram representation is inefficient

when doing matching against large image data collection [44]. Since 2D features cannot be linked to the 3D object perceptually, several approaches extract 3D salient points for image analysis. Steder et al. [45], applied interest point detectors on depth images. Ruhnke et al. [46] applied the Harris detector [40] to the depth image in order to construct 3D object models in an unsupervised fashion from partial views. Plagemann et al. [47] built a 3D mesh to extract geodesic extrema interest points which are classified into 3 groups: head, hand and foot, with location and orientation estimation. But their approach cannot determine the left or right limbs.

We argue that human vision largely relies on the shape features and shape salience entities that attract visual attention. We will introduce an approach that extracts a local shape-based salience efficiently. The image is scanned by a pre-defined interval gird without heavy computational costs, and the extracted salient edges are described by a set of genetic shape tokens and structure critical points which are in the same spirit of point-based salient features, but with more semantics.

2.3 HUMAN GESTURE/ACTION REPRESENTATIONS

Various representation methodologies have been developed to enable computer vision systems to recognize human gestures/actions accurately from image sequences. They can be mainly categorized into four groups: sequence-based, trajectory-based, volume-based and local feature-based representations.

2.3.1 Sequence-based Representations

A video is a sequence of images. A sequence of feature vectors is a natural way to describe a human action video. Each vector contains location, color, orientation, size and shape features of one or more images. In [48], each image frame is divided into meshes, and each vector is simply an array of pixel numbers of the foreground objects within corresponding meshes. Some applications are more interested in specific body parts. In [49] and [50], the feature vector describes the shapes and locations of the tracked hand. The approach from [51] tracks the full body using a 3D skeleton model with 17 degrees-

of-freedom. The characterized joint angles of the skeleton model are the features. Thus, human movements are the sequences of angle values from all frames. Rather than using joint angles, in Park and Aggarwal's method [52], vector features from multiple body parts are object region-related. An action is expressed by a sequence of vectors including locations of skin regions, maximum curvature points, and the ratio and orientation of each body-part. In [54], optical flow-based features are wrapped into the sequential representations. It computes the space-time volume of each person being tracked, and then calculates 2D optical flows of the tracked humans at each frame. A video of a human action is interpreted as a sequence of motion descriptors which are a set of blurring motion channels converted from the optical flows.

Instead of using conventional features, Yacoob and Black [53] used singular value decompositions (SVD) to decompose the image data into eigen vectors. An activity is represented as a linear combination of eigen vectors from an image sequence. Furthermore, the motion scale and speed variations can be obtained by calculating the coefficients of the eigen vectors. Lublinerman et al. [55] presented a methodology that models human activities as the linear time invariant (LTI) systems. Two types of silhouette features are extracted from images: silhouette width and Fourier descriptors. An activity is represented as a LTI system capturing the dynamics of changes in silhouette features. Visual saliency is the visual stimuli which are different with their surroundings, sequential feature representations without further refinement only reflect the saliency at frame basis, but do not provide the spatiotemporal salience directly.

2.3.2 Volume-based Representation

For 2D images, a video can be represented as a 3D XYT Space-Time (ST) volume constructed by concatenating XY images along the time T. 3D ST volumes can be viewed as rigid objects, and the volume-based representations are constructed based on the salience features of the rigid objects. Bobick and Davis [56] proposed two 2D images: a binary motion energy image (MEI) and a scalar-valued motion history image (MHI) which are constructed from a sequence of foreground images. MEI and MHI essentially

are weighted XY projections of the original 3D ST volume. Shechtman and Irani [57] estimated motion flows from a 3D rigid object (ST volume). They extracted a small ST patch around every location of the volume, where the motion flow vectors of a particular local motion are calculated. A human action is represented as a set of flow vectors for all patches. Ke et al. [58] proposed a method that applies a hierarchical mean-shift algorithm to cluster similar colored voxels to obtain segmented sub-volumes. Each sub-volume contains the information about both flows and shapes. Rodriguez et al. [59] used the maximum average correlation height (MACH) filters to analyze 3D ST volumes. A synthesized MACH filter generated from example ST volumes is used to represent a human action. Later on, they extended the MACH filters to analyze vector data using the Clifford Fourier transform. The major issue of the volume-based representation is that they cannot describe the human motion details in terms of body parts correlation. The volume can be formed by shape contour along temporal order. A view-invariant representation, action sketches, was proposed in [69], where sparse features are extracted from a 3D contour concatenation.

2.3.3 Trajectory-based Representation

In the trajectory-based representation approaches, a person is usually represented as a set of 2D or 3D points corresponding to his/her joints. Human body part estimation is necessary for obtaining the joint positions. The action representation is a set of 4D or 3D ST trajectories that are the recorded joint position changes. Authors of [60] represented a human action as 2D curves in phase spaces. Based on the 3D body part models estimated from each frame, they converted the high-dimensional body systems into several low-dimensional phase spaces. The state of a person at each frame is a point and his/her action is a high-dimensional curve (a set of points). They projected this HD curve into multiple 2D subspaces, modeled into a cubic polynomial form, and maintained it as the action representation. Rao and Shah's method [61] tracks the positions of a hand in 2D images using the skin pixel detection, obtaining a 3D XYT space-time curve. Their system extracts the peaks of the trajectory curves. An action is represented as a set of peaks and intervals in-between which are view-invariant curvature-based patterns. In [62],

an action was represented by 13 joint trajectories in a 4D space (XYZT). An affine projection method was used to obtain the normalized XYT trajectories, where the view-invariant similarity measurement can be achieved. Sun et al.'s method [63] uses the trajectory-based hierarchical spatiotemporal features to model human actions. The trajectories are extracted by matching keypoints between two consecutive frames. The human action is modeled by the intra-trajectory transition and inter-trajectory neighborhood information. Recently Wang et al. [64] used dense trajectories to describe videos. Dense trajectories are constructed by matching dense points in the optical flow field between frames. After removing noise trajectories, the motion patterns are encoded by the trajectory shapes. The human motion is represented by a set of Histogram-based descriptors that describe the local and global properties of the dense trajectories.

2.3.4 Local Feature-based Representation

If a system is able to extract appropriate salient points or regions reflecting characteristics of action's 3D ST volume, the action can be described by the local feature-based representations around those salience entities within the ST volume. Some approaches extract local salience features at every frame and concatenate them temporally to describe the human movements. In [65], Motion energy receptive fields and Gabor filters were used to capture motion information from a sequence of images. Local spatial-temporal appearance features about motion orientations are detected per frame. An action is characterized by multidimensional histograms that are constructed based on the detected local features. The approach from [66] utilizes local spatial-temporal features at multiple temporal scales which are able to handle speed variations of an action. A normalized local intensity gradient is estimated for each point in a 3D ST volume. A histogram of these space-time gradient features is the action representation. Instead of utilizing optical flows for local feature calculation, Blank et al. [67] calculated appearance-based local features at each pixel in the ST volume. A wide variety of useful local shape properties including space-time saliency and space-time orientation are extracted from the ST volume by solving the Poisson equation. Meanwhile, the weighted moments of the local features are a set of global features of an action.

Rather than extract local features per frame in ST volume, Laptev and Lindeberg [68] extracted sparse salience entities, Spatial-Temporal Interest Points (STIP), from videos. They used the scale-invariant Harris3D detector to find salient points, spatial-temporal corners, in a 3D ST space. This detector is able to capture various types of non-constant motion patterns such as a direction change of an object, splitting and merging of an image structure, and/or collision and bouncing of objects. Dollar et al. [70] proposed a spatial-temporal feature detector which is to extract space-time salient points with local periodic motions, so as to obtain a sparse distribution of interest points from a video. Once detected, each salient point is associated with a so-called cuboid which captures the neighborhood appearance features of this point. The local descriptors can be a vector of brightness gradients of a cuboid. Similar to the features in [69], a 3D SIFT descriptor was proposed in [71]. In contrast to previous features only using intensities, Rapantzikos et al. [72] proposed a dense sampling method that divides a video into sub-volumes with multiple spatial and temporal scales. 5 parameters are used for dense sampling: (X,Y,T) location, spatial and temporal scales. Sampling is done with certain degree of overlap. Each sub-volume is the local feature data.

Klaser et al. proposed a HOG3D representation method that is based on the histograms of 3D gradient orientations [121]. It can be seen as an extension of the popular SIFT descriptor to video sequences, where gradients are computed using an integral video representation which is a set of regular polyhedrons, and the orientations of spatiotemporal gradients are uniformly quantized for histogram construction. The HOG3D descriptor combines shape and motion information at the same time. Savarese et al. [76] proposed a method to capture spatial-temporal proximity information among features. It measures feature co-occurrence patterns in a local 3D region, constructs histograms so-called ST-correlograms as the action representation. [77]'s approach provides spatial-temporal histograms, histograms of gradient orientations (HOG) and histograms of optic flow (HOF), by dividing an entire ST volume into several grids. According to the STIP distribution in the grid, this method provides coarse measurements about the distribution of local descriptors in the 3D ST space. Normalized HOG/HOF descriptor vectors are similar in spirit to the well known SIFT descriptor.

Since local descriptors lack the ability to convey motion/gesture semantics, bag-of-word (BoW) methods are used to cluster the local descriptors into groups, so-called visual words or codebooks, according to their spatial or temporal similarity. These intermediate level feature descriptors contain more semantics and have more discriminative powers in the recognition and classification tasks. Each gesture/action is modeled as a histogram of the visual words. Among many extracted BoW features, Liu et al. [73] presented a methodology to prune local features to find more important and meaningful features. Similarly, other methods from [64], [74], [75] also fall into this BoW category. BoW methods gain good performance on some human gesture/action datasets. However, they have the limitation of ignoring the temporal and spatial structural properties so as to cause failures for handling complex gestures/actions.

In sum, extracting local feature descriptors from ST volumes has several advantages. By its nature, background subtraction or other low-level components are generally not required, and the local features are invariant to scale, rotation, and translation in most cases. They are particularly suitable for recognizing simple periodic actions and big movements. The ST local feature-based BoW representation contains more semantics but it is weak in terms of describing complex motions in which the internal temporal and spatial relations matter. Instead, our approach will exploit both extrinsic and intrinsic properties of human gestures/actions and provide qualitative descriptions with more semantics.

2.4 GESTURE RECOGNITION

Recognizing complex human activities requires understanding both extrinsic and intrinsic dynamic properties. Several different approaches have been studied for gesture modeling tasks. In [78], a gesture model can be represented by a Finite State Machine (FSM) where the locations of the target object are the points spread in a Gaussian distribution. A K-means-based training process produces a state sequence on the Gaussian distributed sample data. A gesture is recognized if the input feature vectors match all the states along a sequence. Two hidden layers of Time-Delay Neural Network (TDNN) are employed to

classify the image sequence containing a motion into a particular gesture of the American Sign Language [79].

Some other gesture/action recognition approaches use visual word histogram [77], graph [80] [81], attribute list [73], or probabilistic models. Among them, graphical models have been used with great success to capture the structure of an activity in terms of the hierarchy and spatiotemporal arrangement of its components. Hidden Markov Model (HMM)-based models are capable of modeling spatial-temporal series of gestures effectively. Given the training data of a gesture, HMMs output the probability of the observation sequence. The maximum probability is compared with a threshold to determine if a gesture is recognized. Several HMM-based sentence-level American Sign Language recognition systems were presented in [82], [83] and [84]. More HMM-based recognition systems can be found in [85]. The gesture recognition task is tightly bundled with the gesture segmentation. Elmezain et al. [86] proposed a HMM system performing hand gesture segmentation and recognition tasks simultaneously.

Unlike the HMM, the Conditional Random Fields (CRF) method models the entire sequence, avoids the independence assumption between observations, and allows non-local dependencies between state and observations. Sminchisescu et al. [87] first applied the CRF to classify human walking and jumping actions. Yang et al. [88] introduced a method for designing threshold models in a CRF model, which performs an adaptive threshold for distinguishing signs and non-sign patterns. Since human actions are complex, some internal structure cannot be explicitly observed even by human vision. Wang et al. [89] introduced a hidden state conditional random field (HCRF) model as a gesture class detector, or as a multi-way gesture classifier, where discriminative models for multiple gestures are simultaneously trained. HCRF is a variant of CRF, where an additional state layer is put into the state graphical model to provide the context relations. From their results, HCRFs outperform both CRFs and HMMs for certain gesture recognition tasks. Morency et al. [90] provided Latent-Dynamic Conditional Random Field (LDCRF) model which is a discriminative approach for gesture recognition. LDCRF model combines the strengths of CRFs and HCRFs by capturing both extrinsic

dynamics and intrinsic sub-structures. But their HCRF/LDCRF models either just use the spatial feature ignoring temporal structures, or just use an oversimplified chain structure to model the complex dynamic properties. Recently context information has been explored for human action recognition. The recognition method not only focuses on the particular objects, but also the behaviors of other nearby objects which provide useful cues for recognition. Marszalek et al. [91] exploited scene-action context and demonstrated that recognizing the scene type of a video helps the recognition of human actions. Han et al. [92] used object-action context, where the context of an action is implicitly defined by the objects detected in the scene. Lan et al. [93] used the contextual feature representations to encode information about the action of an individual person in a video, as well as the behaviors of other people nearby, thus the human actions can be classified and recognized accordingly.

Support Vector Machines (SVMs) [142] are a useful technique for data classification, and have gained popularity for visual pattern recognition. The Bag-of-Word SVM framework has been adopted by many action recognition systems, and outperforms other methods on human activity, facial expression, and hand gesture datasets [143]. Wang et al. use this SVM framework to evaluate various local spatiotemporal features for action recognition [141].

2.5 GESTURE-BASED VIDEO GAMES

Gesture-based video games are now prevalent in the entertainment industry. Three major companies have been making great efforts to make the next-generation video games more attractive. Nintendo's system is a pioneer of gesture based video games. Its gesture game platform Wii [26] has been on the market since 2006. The players of Wii games are required to hold remote controllers, which have built-in accelerometers and infrared detectors, the hand gestures are captured by the non-vision sensor. Thus Wii's approach is different from that of the vision systems. Sony launched its gesture game platform, PlayStation Move [27], in September 2010. It is also based on a handheld controller, so-

called wand, which is equipped with more sensors, such as inertial sensor, linear accelerometer, angular rate sensor and a magnetometer. In addition, it uses a camera to track the colored light on the wand which is a strong tracking cue, so that the hand position can be tracked precisely. This system does not fully rely on the computer vision technologies; other sensors within the controller play roles in the detection of motions.

Microsoft released the Kinect gesture game platform in the November, 2010, which is fully controller-free. A Kinect camera is the only sensor capturing the full body movements, including jump, run, kick, waving hands, driving, boxing etc. The recognition method is purely based on the computer vision technology. There are two steps in the Microsoft Kinect system. First, with a 3D camera, a bottom-up human body detection approach [94][95] is used to match up the person in an image by recognizing 3D body skeleton joints based on the models trained from large datasets. The posture of a human body is represented by several metrics and parameters; its variant movements of a particular gesture are expressed by the distribution in a metric space. Secondly, the tracking process is either searching in the metric space by estimating the most likelihood [96], or finding a regression function from large training data by estimating the relationships among multiple variables [97][98]. Based on these advanced approaches, Kinect doesn't lose track of the human body easily, and is able to track multiple persons, making two-player games possible.

The main limitation of the Microsoft Kinect system is that it only captures significant motions containing large spatial changes in 3D space, and neglects local gesture details (e.g. finger movements, palm orientation etc.). This is because gentle and local motion gestures may not be easily modeled in their current systems due to the self-occlusion, interference, camera resolution, and other local uncertainties. Our approach attempts to overcome such limitations in a systematic, coherent and biologically plausible manner.

2.6 VISUAL ATTENTION MODELS

Visual attention refers to the processes by which some visual information in a scene is selected according to the high-level tasks. Two different attentions, bottom-up and top-down are responsible for performing perception tasks [12]. Bottom-up attention is driven by the low level image visual stimuli, and is automatic and task-independent. Top-down attention is guided by the tasks and human intention and requires more biological computation. The deployment of top-down attention is relatively slow and volition-controlled. Perceptual Organization (PO) laws play vital roles in the attention process.

2.6.1 Related Terminologies

Here we first briefly introduce several related terms about visual attention and perceptual organization.

— Perceptual organization

  Perceptual organization refers to the visual processes structuring the pieces of visual information into coherent units that we eventually experience as environmental objects. The Gestalt psychologists suggested that organization is composed of grouping and segmentation processes [99], and several stimulus factors determine organization. These include grouping factors such as proximity, similarity, good continuation, common fate, and closure [100], and factors that govern figure-ground organization, such as size, contrast, convexity, and symmetry [101]. Recently, researchers have identified additional factors that support grouping: common region and element connectedness [102], figure-ground assignment familiarity [103], lower region [104], spatial frequency [105], base width [106], and extremal edges [107].

— Visual stimuli

  Visual stimuli include basic low-level features, such as color, orientation, motion, depth, conjunctions of features. In computational models, visual stimuli can be specified with values, variables, or mathematical expressions according to their spatial, temporal, and chromatic properties.

— Visual salience

Visual salience [108] is related to the biological signals in the human neuro-system responding to various visual stimuli, whose local visual attributes significantly differ from the surrounding attributes. Visual salience associates the underlying neural mechanisms and the internal states of organisms. Rather than quantitative measurements, visual salience is the distinct subjective perceptual quality which makes some items stand out from their neighbors.

— Salience map

The Saliency map is a topographically arranged map that represents visual saliency of a corresponding visual scene. A saliency map is formed up by integrating low-level visual salience to provide a visual overview for the visual process [33]. The salience map is a symbolic representation of integrated visual and spatial information of visual salience, and acts as the visual evidence for the selective visual attention deployment.

— Visual attention

Visual attention refers to the processes of visual information selection, in particular, information that is most relevant to ongoing behavior. Two different attentions are responsible for performing perception tasks, bottom-up and top-down. Bottom-up attention is stimulus-driven in that the attention is drawn involuntarily by the bottom-up visual salience. Top-down attention is goal-directed, based on the human behavioral goals. If we know, for example, where is the most probable target location, we can direct our attention to this location voluntarily.

2.6.2 Computational Models

Several levels of computational models, including low-level feature and object-based models, of visual attention deployment have been proposed and implemented in the past decades. The object-based attention model is on the top of the hierarchy while low-level feature-based visual salient entities provide it with bottom-up visual foundations.

The attention process based on particular features is biased in a way that is optimal for detecting a known target. Attention is deployed either directly to the spatial locations, or object units by biasing the computation on a saliency map. Some models provide selected

attentional “spotlights” at some particular locations in the visual field [109][110]. Occasionally, attention is compared to a zoom lens [111], adapting the size of the spotlight to the attended area. Deco and Schurmann [9] modulate the spatial resolution of the image based on a top-down attentional control signal. The “Selective Tuning” model [8] deploys the visual attention for object recognition hierarchically [112]. In the first feed-forward pass through this hierarchical system, bottom-up visual stimuli are chosen in the Winner-Take-All networks. After forming up top-down selection criteria according to the particular tasks, visual stimuli satisfying the selection conditions are enhanced and processed in another feed-forward pass hierarchically for ultimate detection. The Selective Tuning model has been demonstrated successfully for motion-defined shapes [113]. Visual features are biased in the recognition hierarchy by using the mechanism of the feature modulation functions.

The attention is deployed on the saliency maps first ever proposed by [33]. The feature-based attention approach biases the particular features on the bottom-up saliency map. Most models of attention and object recognition follow this salience map-based approach. In the visual attention model given by Itti et al. [7], the salience map is first implemented in the attention process. Several low-level salience features are extracted from the input image at multiple scales in feature pyramids, including color channels, luminance, and orientations. Center-surround contrasts in these features are computed as differences between scales in the respective feature pyramids and, after normalization, stored in “feature maps”. A Winner-take-all mechanism is deployed to generate the final salience map. As an extended version of [7], Navalpakkam and Itti proposed a computational model [6] for the top-down task-specific guidance of visual attention in real-world scenes. They model the task influences on visual attention by adjusting the weights of salience entities on the salience map. The weights for particular targets are learned from training images. Its hierarchical matching recognition method is performed on the salience map (selective visual saliency information). Wolfe’s Guided Search model [114] provides more theories supporting the salience map concept to explain human behavior in visual search for targets, where visual salient features are computed at multiple scales in feature pyramids, and a final salience map is generated by a Winner-Take-All method too.

Experimental evidence suggests that attention can be also tied to objects, object parts, or groups of objects [115]. Object-based attention captures a variety of effects ranging from spatial limiting attention to the optimal feature biases for a particular search target. By treating attention as a by-product of a recognition model based on Kalman filtering, Rao's system [116] can attend to spatially overlapping (occluded) objects on a pixel basis. [117] modeled the object-based attention when objects are even not clearly separated. Their system can segment superimposed handwritten digits on the pixel level. Rolls and Deco [118] modeled object-based attention by shrinking the size of the receptive field of neurons to match the size of the attended object. Object-based attention is at the top of the attention hierarchy. By spreading of feature-level attention over a contiguous region of high activity in the salience map, a model from [12] thus obtains an estimate for the size and shape of the attended objects. Closely following and extending Duncan's Integrated Competition Hypothesis [10], Sun and Fisher [11] developed a framework for object-based and location-based visual attention using "groupings". Presented with a manually preprocessed input image, their model replicates human viewing behavior for artificial and natural scenes. Obviously grouping laws of perceptual organization play roles in object-level attention, such as the continuity connectedness, closure and so on.

2.6.3 Salience in Spatiotemporal Space

Recently attention model-related approaches have been applied to emerging applications such as video classification, event detection and activity recognition. The concept of saliency detection in spatiotemporal neighborhoods has been used for spatiotemporal analysis. Examples are the STIPs by Laptev [68], Cuboids [70], efficient space-time detector based on the determinant of the 3D Hessian matrix [121], dense spatiotemporal saliency [72], extended spatiotemporal salient point detector by Oikonomopoulos et al. [18], etc. Itti and Baldi [120] proposed a model of salient event detection on videos. They formulate the saliency as the KL divergence between the posterior and prior beliefs of an observer about the scene. In fact, this model extends the spatial center-surround contrast to the spatiotemporal domain. A saliency map of visual attention for hand spatiotemporal patterns was proposed in [16] for hand gestures recognition, where selective visual

features make the recognition efficient. A top-down visual attention model in an interactive gaming environment was recently presented to evaluate various task-relevant factors for attention deployment [17].

Inspired by models related to visual attention, salience map and perceptual organization, we study the hierarchical models of salience maps incorporating perceptual organization laws, and propose a 3D gesture/action recognition framework based on hierarchical visual attention and perceptual organization models.

2.7 SUMMARY

It is essential for a visual analysis system to detect and select effective low-level visual salience features for building semantic representations. Among a variety of approaches, global-based salience features can be obtained efficiently, but they are unable to describe complex details. Local salient features provide effective visual description but usually involve high computational costs for extraction. Based on the extracted local salience features, local spatiotemporal descriptors contain more semantics about gestures; and BoW-based representations reportedly outperform others in the recognition and classification tasks. But the local descriptor + BoW representation is still weak for describing human activities with complex internal temporal and spatial relations. Instead, our approach exploits both shape-based extrinsic and intrinsic properties of human gestures/actions. The local shape-based salience is extracted by scanning an image using a pre-defined interval grid without involving heavy computation. The extracted shape salient entities, i.e. a set of genetic shape tokens and structure critical points are in the same spirit of salient point-based features, with more semantics. Theories and hypotheses of visual attention, salience map and perceptual organization shed light on the mechanisms of human vision perception. Inspired by the existing visual attention models, our 3D gesture/action recognition framework provides a complete gesture recognition solution in a systematic, coherent and biologically plausible manner.

CHAPTER 3 3D SHAPE-BASED FEATURE SALIENCE

3.1 INTRODUCTION

A fundamental problem of image and video understanding is how to efficiently select effective visual salience among numerous low-level features and build an appropriate representation for semantic computing. Not every visual stimulus links to visual saliency attracting human visual attention, only the one whose local visual attributes significantly differ from the surrounding attributes will trigger the biological neuron signals in the brain system. To mimic the human visual mechanism, we need to extract perceptual salient features from images, which can effectively capture the visual attention to the target objects. Color information is relatively easy to obtain, and therefore, is popularly used in many vision systems. Obviously, the color-based methods will fail for the images whose colors are not available. Texture features provide more spatial or relational information than colors. Tamura et al. [122] proposed a set of texture patterns that contain six features selected by psychological experiments: coarseness, contrast, directionality, line-likeness, regularity and roughness. The limitations of texture features are that they are not generic for different applications and some methods involve high computation costs and implementation complexity [123].

Human visual perception largely relies on shapes, i.e. it can perceive a scene based on object contours alone without using color and texture information. Many shape-based image analysis techniques use Fourier description and moment invariant-based representations which contain a less semantic interpretation of human perception [124]. Most edge-based methods emphasize the global shape properties which often result in an inability to provide local details. Contour shape is one of the basic object features which can be discriminated effortlessly by human vision. Efficient shape feature extraction is crucial for visual semantic computing. The research discoveries from neuroscience provide us with useful insights on the generic human vision criteria. Perceptual Organization (PO)-based Perceptual Curve Partition and Grouping (PCPG) [125][126] is a shape perception model in which each contour is made up of Generic Edge Tokens

(GET) connected at Curve Partitioning Points (CPP). These perceptual features (GETs and CPPs) directly link to the visual salience entities, and their corresponding representation provides the bottom-up salience maps for high-level visual understanding. Zheng, et al. [127] presented a method that converts an image into an edge saliency map made up by Generic Edge Tokens (GET). In that approach, GET features encode both contour and texture content of an image according to the distribution of different GET types. Even with this oversimplified schema, GET salience-based representation is able to handle challenging image analysis tasks such as CBIR [127], visualization tool [128], and image segmentation [129]. Despite the proven power of the perceptual shape-based salience entities, issues of implementing the PCPG model remain unsolved, i.e., that conventional gradient-based measures often miss some CPPs if their gradient evidence is weak, therefore it will eventually affect the performance of classification [130] and recognition tasks.

To provide better feature level salience maps for high-level interpretation tasks, in this chapter, one of our goals is to improve the accuracy of CPP detection and GET classification by using a non-parametric statistical method. The approach includes: 1) Instead of using gradients, introduce an arctangent space to make the CPP evidence more measurable. 2) Utilize the pixel sequential order along a curve as the heuristic to improve the CPP detection accuracy. 3) Design a new CPP evaluation method including a Zero-Crossing scheme for locating strong CPPs, and an Order Preserving Arctangent Bin Sequence (OPABS) scheme for detecting weak CPPs. Since 2D data alone has less ability to describe human activities in 3D world, the second goal in this chapter is to provide the 3D perceptual shape-based salience entities by integrating both 2D CPPs/GETs and spatial data from a 3D camera.

The proposed OPABS scheme is a non-parametric statistical method [131] utilizing both local and global measures from the curve sequential data. Visual saliency based image representations, such as SIFT [132][133], have been widely recognized in the computer vision community. The extracted 3D shape salient points (CPPs) and generic shape segments (GETs) by our method are in the same spirit of the keypoints in SIFT, and

easily link to the 3D objects supporting top-down object level operations. GET/CPP based bottom-up image descriptors (salience maps) are able to bridge the semantic gap.

3.2 2D Perceptual Shape Salience

The human vision system is able to partition a perceived scene into perceptual elements then group them into meaningful clusters related to the known objects [134]. A Perceptual Curve Partitioning and Grouping (PCPG) model [125] was proposed to mimic the function of human perception in partitioning and grouping object contours from images. In this PCPG model, 2D curves are partitioned into a minimum set of Generic Edge Tokens (GET) types (Figure 8(a)) which are connected at Curve Partitioning Points (CPP) (Figure 8(b)). Each GET type satisfies a unique combination of monotone increasing or decreasing in x-y geometry [126].

CPPs are the salient points on edge curves. Some of them can be found where the monotone properties along an edge are broken. They are the so-called Strong CPPs, and their properties of x-y geometry and tangent values stand out from their neighbors (Figure 8(b)-(1~3)). In other cases, some shape critical points cannot be detected by only checking the monotone conditions. For instance, in Figure 8(b)-(4~6), the marked points are visually special to human vision perception, but the changes along these traces do not violate the monotonic properties. To distinguish the differences, we call the partitioning points in Figure 8(b)-(4-6) weak CPPs, which are still perceptually critical points, but without breaking the continuities of the monotonic properties.

In the previous CPP detection module [130], strong CPPs are detected by checking the gradient data, and weak CPPs are verified by a gradient-based curvature method. The challenge is that the gradients of the pixel intensity are unstable, and the local curvature method is sensitive to the edge noise and lack of global views for making right decisions. Inevitably, the local curvature-based detector often misses weak CPPs and brings some

false positives due to the impacts of the local maximum and noise data, therefore causes misclassification on GET types.

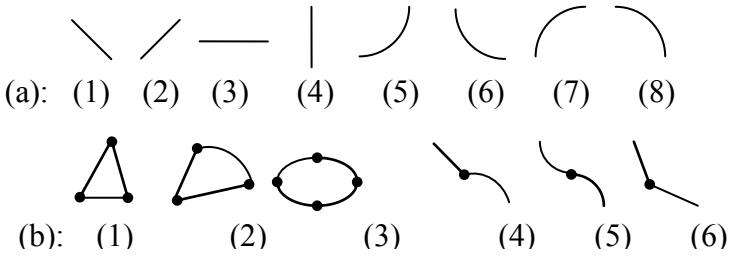


Figure 8 An illustration of GET and CPP types
(a) 8 type GETS; (b) Different types of CPPs, most CPPs are view and rotation invariant except type (3) floating if the curve is rotated.

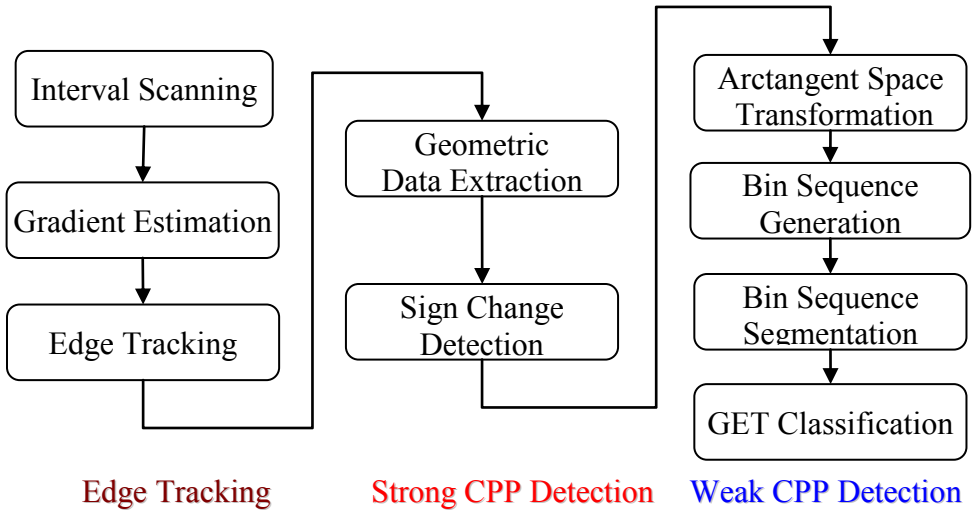


Figure 9 Workflow of the feature detection approach.

3.3 CPP Detection Method

In this section, we present a non-parametric statistical method for CPP detection and GET classification. Figure 9 illustrates the system workflow. The approach takes advantage of edge trace geometric properties to extract *dx* and *dy* values of each edge pixel. By using a local method to check the zero-crossings of *dx* and *dy*, strong CPPs are identified, and

then an edge trace is partitioned into raw segments. According to the application demands, raw segments can be further partitioned into finer segments by applying a weak CPP detection method. A proposed global approach, Order Preserving Arctangent Bin Sequence (OPABS) scheme, is used for the weak CPP detection. OPABS scheme is a distribution-free method which does not rely on assumptions that the data are drawn from a given probability distribution. It is a non-parametric statistics based method.

3.3.1 Geometric Properties of Edge Shapes: dx, dy

Instead of using unreliable pixel intensity-based gradients, geometric properties, *dx* and *dy* of each edge pixel are derived for CPP detection and GET classification. A *Tangent Sliding Window* (TSW) is used to determine each pixel's *dx* and *dy*. The TSW is a minimal rectangle box to enclose several edge pixels that are the neighbors of the pixel (in the middle) which *dx* and *dy* are calculated for. The window size of the TSW is the number of the pixels within the TSW, and must be an odd value. The TSW size is a granularity and scale related parameter, which can be set accordingly. The minimal size is 3. Here we use 5 as the size since it empirically gives better performance. Once a TSW is set, *dx* and *dy* of a pixel can be obtained as the differences along x and y within the TSW. Specifically, take two end pixels from the TSW, *dx* is the difference between both endpoints in X, and *dy* is the difference between both in Y. In Figure 10, the TSW size is 5, two endpoints of the small segment within the TSW are *N1* and *N2*, *dx* and *dy* of P are,

$$\text{P's dx} = \text{N2.x} - \text{N1.x}, \tag{3.1}$$

$$\text{P's dy} = \text{N2.y} - \text{N1.y}. \tag{3.2}$$

Since *dx dy* values indicate the pixel's slope along the trace, the changes of *dx* and *dy* along the curve reflect the curve's shape property.

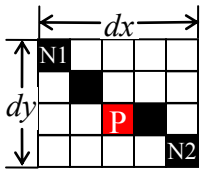
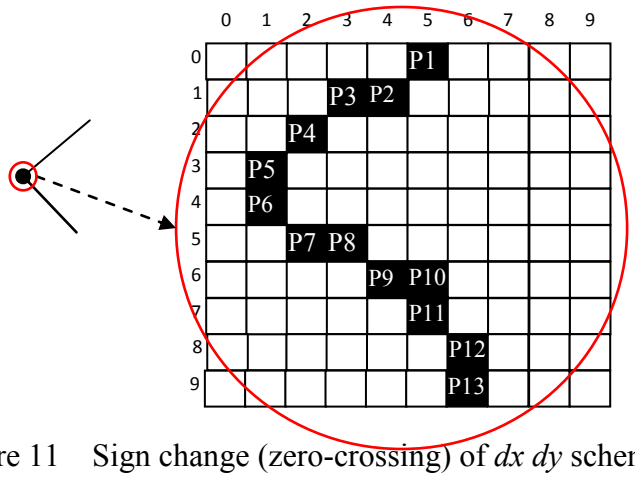


Figure 10 *dx dy* calculation for each edge pixel.
N1 and *N2* are 2 endpoints of the segment enclosed in the TSW whose size is 5. P's *dx*=4, P's *dy*=3.

3.3.2 Strong CPP Detection

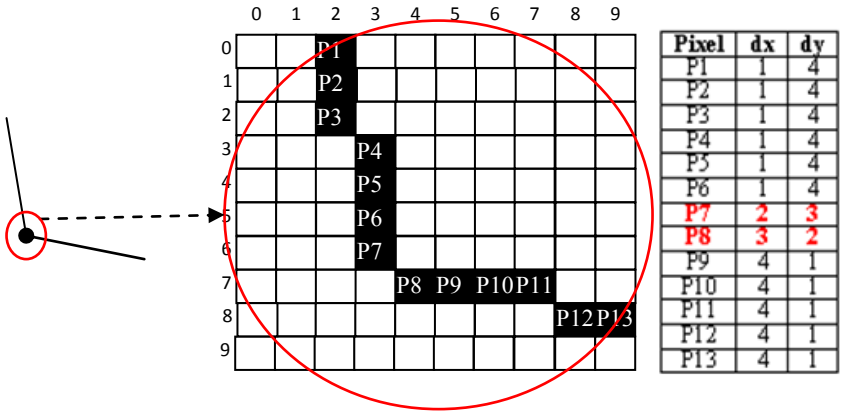
A strong CPP always corresponds to a GET junction along the curve where a generic criterion of a GET type is terminated, and its *dx* or *dy* value must cross a zero point. In Figure 11, the edge trace has two straight lines jointed at a strong CPP which is marked by a red circle. The grid shows pixels within the red circle in an enlarged view. Each cell of the grid is a pixel. The right side table in Figure 11 shows the *dx* and *dy* values of corresponding edge pixels in the grid. P1 locates at the starting end, while P13 is at the end. As we can see in the table, from the P1 to P13, the sign changing of the *dx* values occurs between P5 and P6. Consistently, P5 are P6 are visually the monotonic changing points of the original edge trace. Thus, a strong CPP can be detected by observing the ***dx-dy sign changing (zero-crossing)*** scheme.



Pixel	dx	dy
P1	-4	3
P2	-4	3
P3	-4	3
P4	-3	3
P5	-1	4
P6	1	3
P7	3	3
P8	4	2
P9	3	2
P10	3	3
P11	1	3
P12	1	3
P13	1	3

Figure 11 Sign change (zero-crossing) of *dx dy* scheme for strong CPP detection.

A weak CPP is also a shape salient point of an edge trace. Even though the differences from both sides of the weak CPP are not big enough to be treated as monotone changing, the saliency is visible and the continuity along the trace is broken. However, changes of weak CPPs cannot be detected by a Zero-Crossing scheme. In Figure 12, we see that P7 or P8 is a salience point perceptually, but the signs of their *dx* and *dy* values along the edge trace are unchanged. The detection method used for strong CPPs does not work for weak CPPs.



Pixel	dx	dy
P1	1	4
P2	1	4
P3	1	4
P4	1	4
P5	1	4
P6	1	4
P7	2	3
P8	3	2
P9	4	1
P10	4	1
P11	4	1
P12	4	1
P13	4	1

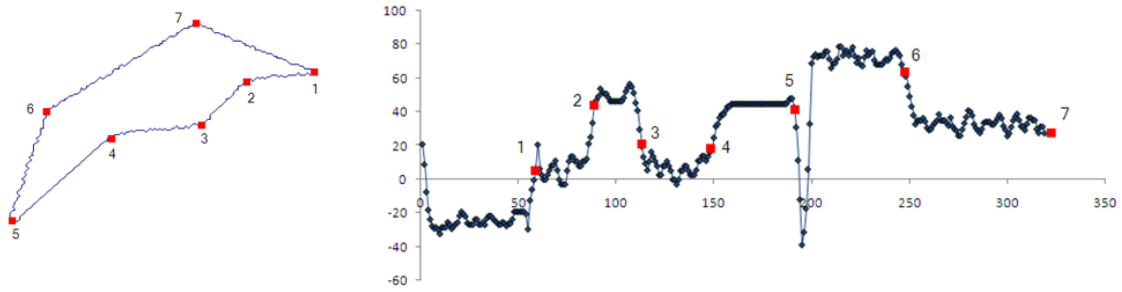
Figure 12 Weak CPP cannot be detected by zero-crossing schema.

3.3.3 Order Preserving Arctangent Bin Sequence

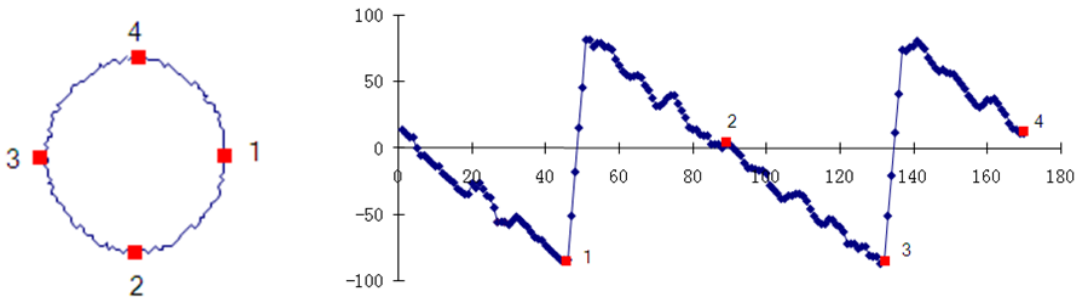
To detect weak CPPs, we introduce a new histogram representation which uses arctangents derived from *dx* and *dy*, and preserves the order of pixels on the curve. This representation can rearrange the data to signify the evidence of weak CPPs. The formula converting *dx* and *dy* into an arctangent value is:

$$\text{arctan\_degree} = \begin{cases} \arctan(\frac{dy}{dx}) \times \frac{180}{\pi}, & dx \neq 0 \\ 90 & dx = 0 \quad pre\_\arctan > 0 \\ -90 & dx = 0 \quad pre\_\arctan < 0 \end{cases} \tag{3.3}$$

By applying this formula (Eq. 3.3), edge pixels along a curve are converted into an arctangent degree sequence. For example, Figure 13(a) and (b) show a polygon and its corresponding arctangent degree sequence respectively. The Y axis in Figure 13(b) represents the value of a pixel's arctangent degree, and the X axis is the sequence order of edge pixels. The numbers labeled on the edge trace are the salient points that human vision can perceive, and their corresponding positions on the right side arctangent sequence are marked accordingly. As we can see, the arctangent degree values between every 2 marked salient points along the sequence fall into a limited range around a particular degree value because their corresponding edge pixels have similar slopes. For the circle edge in Figure 13(c), the majority of arctangent degree values in Figure 13(d) between 2 salient points are presented as a tilted line since the slopes of their corresponding edge pixels are gradually changing in a certain trend (decreasing). It is



(a) polygon edge trace (b) arctangent sequence of the polygon edge trace



(c) Circle edge trace (d) arctangent sequence of the circle edge trace

Figure 13 Edge traces and their corresponding arctangent sequences.

worth noting that, on the polygon edge trace (Figure 13(a)), point 2, 3, 4, and 6 are the weak CPPs which have evidence as obvious as the strong CPPs do in the arctangent space. The arctangent degree sequence well reflects the shape properties and provides a different view with less ambiguity for CPP detection.

Based on the properties of the arctangent degree sequence, we propose an Order Preserving Arctangent Bin Sequence (OPABS) histogram as the representation for better CPP detection. Arctangent degree values are evenly classified into eight discrete categories each of which covers a range of slopes (Table 1).

Table 1 Categories of Arctangent degree values.

Category	Cat-0	Cat-1	Cat-2	Cat-3	Cat-4	Cat-5	Cat-6	Cat-7
Degree Range	-90, -67.5	-67.5, -45	-45, -22.5	-22.5, 0	0 , 22.5	22.5, 45	45 , 67.5	67.5, 90

For any edge trace, arctangent degrees are calculated for all pixels, and then are mapped to corresponding bin categories according to Table 1; finally the OPABS histogram is aggregated accordingly. For example, the edge trace in Figure 12 contains one weak CPP. The mapping and aggregating results are illustrated in Table 2 and Table 3. There are three types of information in Table 3: bin order, arctangent category and bin size.

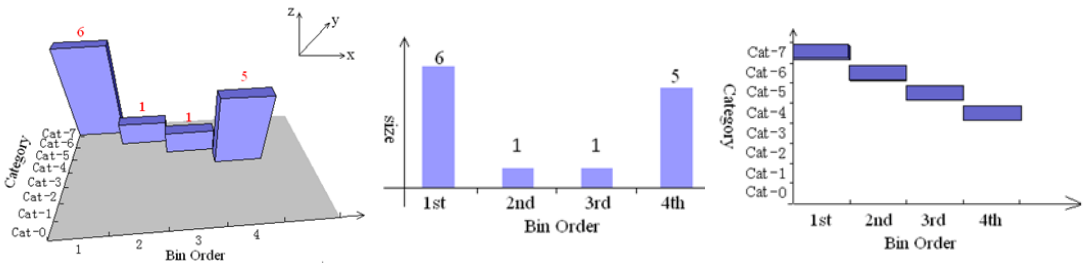
Table 2 Arctangent degrees and their categories of the example trace in Figure 12.

Pixel	dx	dy	Arctangent degree	Arctangent Category
P1	1	4	75.96389	Cat-7
P2	1	4	75.96389	Cat-7
P3	1	4	75.96389	Cat-7
P4	1	4	75.96389	Cat-7
P5	1	4	75.96389	Cat-7
P6	1	4	75.96389	Cat-7
P7	2	3	56.30995	Cat-6
P8	3	2	33.69009	Cat-5
P9	4	1	14.03626	Cat-4
P10	4	1	14.03626	Cat-4
P11	4	1	14.03626	Cat-4
P12	4	1	14.03626	Cat-4
P13	4	1	14.03626	Cat-4

Table 3 A bin sequence after aggregating based on Table 2.

Order	Category	Pixels	Size
1	Cat-7	P1-P6	6
2	Cat-6	P7	1
3	Cat-5	P8	1
4	Cat-4	P9-P13	5

There are 4 bins in Table 3, and their arctangent categories are Cat-7, Cat-6, Cat-5, and Cat-4 respectively. The first and the last bins contain more pixels, while the second and the third bins only have 1 pixel for each. In Figure 14(a), it is the 3D bin sequence derived from Table 3. The X axis is the bin order and the Y axis represents the arctangent categories from cat-0 to cat-7. The Z axis is the bin size. The shape along the bin sequence is like a valley, with 2 peaks on the both ends. If we project this 3D bar chart into the 2D order-size (x-z) view, we would have a clearer picture of this valley shape of the bin sequence with respect to the bin size (Figure 14(b)). During the OPABS construction, no pre-specified assumptions are required, the distribution of the pixel density is only determined by the arctangent data.



(a) 3D bin sequence (b) Size-bin view (c) Category-bin view

Figure 14 3D column bar sequence of the bins sequence and its projected 2D views for the edge in Figure 12.

In Figure 14(b), the 2nd and 3rd bins with smaller sizes in fact are the transition area of the edge trace. Project this 3D chart (Figure 14(a)) into a 2D order-category (x-y) view, the arctangent changing trend is clearly presented in Figure 14(c), the arctangent values are decreased from Cat-7 to Cat-4 along the bin order. This OPABS histogram representation has the following properties:

- data points within a bin share the same slope category.
- only the consecutive pixels sharing the same slope category will be grouped into a bin, and the pixel sequence order is still preserved within a bin.
- the *size* and the *arctangent category* along the bin sequence reflect the edge shape properties.

3.3.4 Weak CPP Detection

Our goal is to detect weak CPPs effectively, which are the breakpoints of the continuity of the shape property. From the previous bin sequence examples, we know that each bin is a segment of a trace that contains data sharing a same property, i.e. arctangent degree category. So each location between 2 neighbor bins is the transitioning spot between 2 sets of edge pixels with different properties, and is the possible CPP position. It is worth noting that not every bin is a valid segment according to our GET definition because the differences between neighboring bins do not mean the discontinuity happens. There are 2 types of continuity on the bin sequence. One is the bin sizes along a bin sequence. If some bin sizes are relatively similar, or, keep increasing or decreasing gradually, it means no changes, or, no sudden changes in terms of the sizes. The other type of continuity is regarding the arctangent categories. As the arctangent categories represent the slopes of the edge curve, the changing trend of the arctangent categories reflects the slope changes of an edge shape. If the changing trend of categories keeps consistent, either decreasing or increasing, or unchanged, the continuity is preserved. Otherwise, the edge should be partitioned because of the discontinuity.

Having had a bin sequence of an edge trace extracted, we detect the weak CPPs according to the properties of a bin sequence. There are two criteria to locate the CPPs.

Peak-Valley-Peak

The Peak and Valley are the terms to describe the relative sizes of neighboring bins. If there is a peak-valley-peak shape formed on the bin sequence, the CPP might locate within the valley bin, and the middle pixel of the valley bin is the CPP. Peak-valley-peak shape means that, at the beginning, many edge pixels' arctangent degree values belong to one category, thus the one side of the peak is formed. Later, the properties (slopes) of the following fewer pixels' have changed into other categories (bins). Since the number of pixels in this category(s) is much smaller, a cliff is formed. Then the slopes of following pixels are changed again, and all fall into another bin with large pixel population. Thus, along the bin sequence, another peak shows up to form a complete Peak-Valley-Peak shape. The pixels in valley bin(s) are leading the change. A metric Peak_valley_ratio is used for detection

$$Peak\_vally\_ratio = \frac{Peak\_bin\_size}{Valley\_bin\_size}, \tag{3.4}$$

where peak_bin_size and valley_bin_size are the bin pixel numbers. A trained threshold is used to determine the segmentation decision. Peak_valley_ratio should be used to check both sides of a valley. Once the peak-valley-peak shape has been confirmed, the CPP must be somewhere within the valley bin(s). Since in most cases, valley area has few pixels, whichever pixel within the valley bin is picked as the CPP will not affect the overall segmentation performance. For the sake of simplicity, we take the middle pixel of the valley bins as the CPP. Figure 15(a-d) show an example.

Changing Trend of Arctangent Categories

Besides the bin size changing, the arctangent category changing trend is used for CPP detection for the other scenario. If the arctangent category changing trend keeps unchanged, the continuity is preserved. Otherwise, the continuity is interrupted. More specifically, if the arctangent category changing trend is either increasing (Cat-1 to Cat-7), or decreasing (from Cat-7 to Cat-1), the continuity is preserved; otherwise, the continuity is broken and a CPP can be detected within the critical bin or its neighbor bins depending on the local extreme. See the example in Figure 15(e-h), a 3D size-order view of the bin sequence, from the left to the right, overall the size keeps decreasing until reaching the 7th

bin. Then the bin size has a big high jump on the 7th bin place. But the bin shape is not a peak-valley-peak since the size changing on the left side of the 6th bin is gradually decreasing and the size ratio (Eq. 3.4) probably less than a defined threshold. Thus, it does not meet the condition of the peak-valley-peak scheme. However the discontinuity can be found on the other 2D arctangent category view. The arctangent categories along the bin sequence are Cat-3, Cat-2, Cat-1, Cat-0, Cat-1, Cat-2, Cat-3, Cat-4 from the 2D category-bin view in Figure 15(h). The changing trend is broken at the 4th and 5th bin places. The arctangent bin categories are decreasing from Cat-3 to Cat-0, and starting to increase from Cat-0 to Cat-4, and then majority pixels fall into 7th bin with Cat-3. The continuity of the arctangent degree values' changing trend is broken at 4th and 5th bins. So the CPP should be located between 4th and 5th bins. Since the original pixel order is persevered within each bin, either the last pixel of the 4th bin, or the first pixel of the 5th bin, would be the exact CPP in this case.

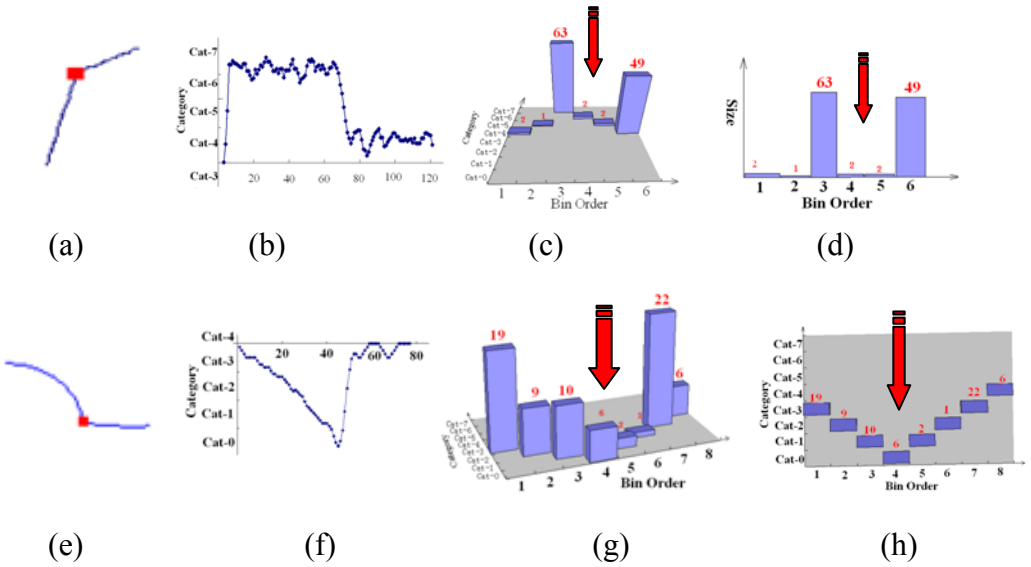


Figure 15 Two criteria for bin segmentation.

From this OPABS scheme, the location of weak CPPs can be determined based on the global statistics along the curve by analyzing its projected 2D views. The detection method takes advantage of pixel sequential order information, uses the trends of both bin sizes and arctangent categories, and utilizes both global and local views. Thus, they are

more generic for different types of images and robust to noise. Here is the algorithm of this detection method:

Algorithm 1. CPP detection and curve partitioning

```
1   Scan the image with interval and put all edge pixel candidates in a candidate_array
2   For Each candidate Edge pixel in the candidate_array  do
3       Find an edge trace, and store edge pixels in a trace_pixel_array
4       For all P(i) in trace_pixel_array do
5           Set up the initial signs of dx dy
6           check the signs of dx and dy for each P(i)
7           If sign changed then
8               Take this point as a strong CPP, put into a CPP_array
9               Construct OPABS for pixels btw 2 Strong CPPs
10              Check size-order space of the OPABS
11              If there is a size valley along the bin order, then
12                  partition the bin sequence, store  into a Bin_segment_array
13              End if
14              For each Bin_seg[j] in Bin_segment_array do
15                  Check the category-order of the (segmented) bin sequence
16                  If the category trend is changed then
17                      Bin_seg[i] is further partitioned
18                  End If
19                  Classification for each partitioned sub bin sequence
20              End For
21          End If
22      End For
23  End For
```

Figure 16 Algorithm of the CPP detection.

3.4 GET CLASSIFICATION

Generic Edge Token (GET) classification is another goal of the OPABS approach. After an edge trace has been partitioned into generic segments, we need to identify their types. According to the PCPG model, there are eight GET types which can be further generalized into 2 perceptual classes: curve and straight line GETs. So we can have a hierarchal classification structure of GETs (Figure 17). We focus on the first level classification, i.e. to identify curve and straight line GETs. Once the straight line and curve GET types have been classified, the further GET classification could be easily done by checking the topologic properties of a small number of pixels.

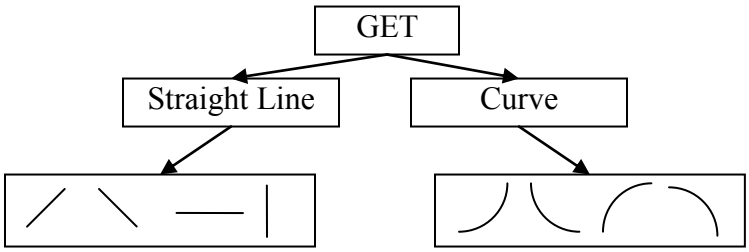


Figure 17 Hierarchal classification structure of Generic Edge Tokens (GET).

Straight Line

For a straight line GET, its pixels have the similar slope, and belong to a same arctangent category. So for its bin sequence, one bin must have a large size, and the rest of them are with much less pixel population, which are treated as noise data (Figure 19).

Curve

For a curve GET, the slopes of pixels are either increasing or decreasing monotonically, and the size changes of its arctangent degree bins are smoothly. For example, in Figure 20, the major bins' category changing trend is Cat-7, Cat-6, Cat-5, and Cat-4, in a decreasing trend (first 2 bins are ignored as their sizes are relatively small). There is no peak-valley-peak bins on the sequence. So it is a curve.

After segmentation, there are only 2 types of bin sequence segments, one is with a size

peak on its bin sequence, the other is without a peak bin but has the bins with same changing trend of arctangent categories along the bin sequence. Thus, decision making for the straight line/curve classification is in fact simply based on the peak's validation within a segmented bin sequence. If there is a peak within a bin sequence, it is a straight line; otherwise, it is a curve.

Classification Algorithm:

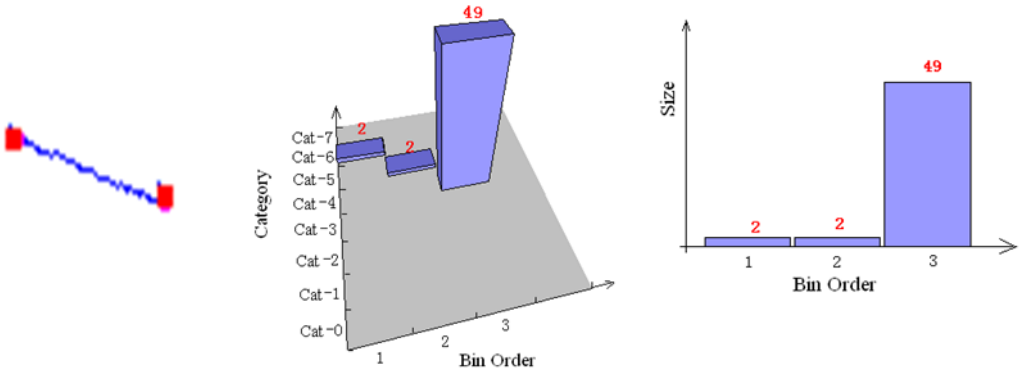
Algorithm 2. Segment_Classification

```
1   Find a bin with maximum size within the bin sequence
2   Calculate the average bin size of the rest bins
3   Calculate the bin size's derivates
4   If derivate<Threshold  then
5      It is a curve
6   Else if the max_size >threshold_1 then
7            If (Max_size / average_size) >  threshold_2 then
8               It is a straight line
9            End if
10        End if
11  End if
```

Figure 18 Algorithm of GET classification.

Time complexity is O(B), here B is the number of the bins.

For further classification of curve and straight line types, it could be done by simply analyzing the geometric property, i.e. spatial data of both endpoints of a segment.



(a) (b) (c)

Figure 19 Straight line classification.
(a) a straight line. (b) 3D view of the bin sequence. (c) 2D view of the bin.

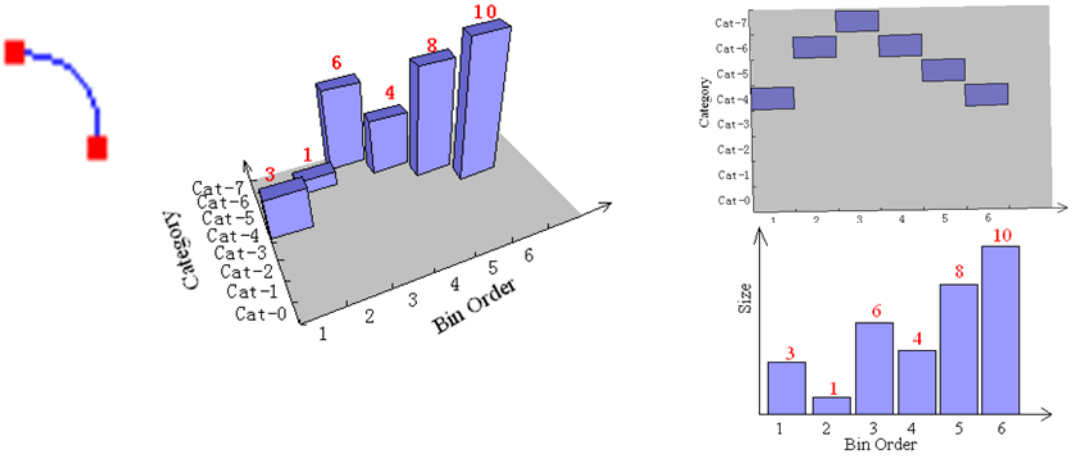
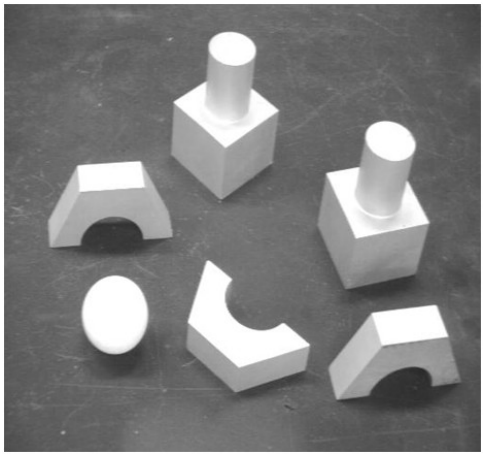
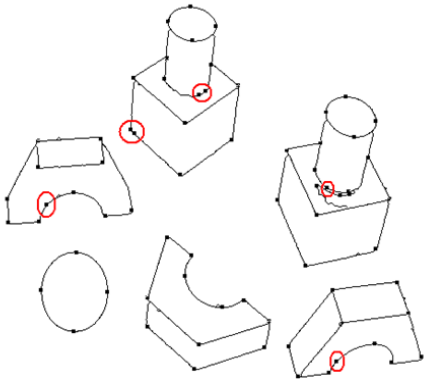


Figure 20 Curve Classification.

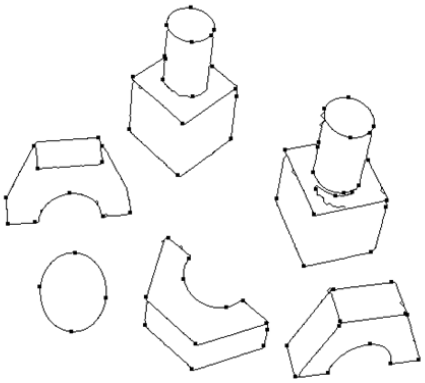
3.5 Experimental Results

Figure 21 provides brief comparison results. The top one is the original image, Figure 21(a1) shows the CPP detection results from the local gradient curvature-based method [130] which brings some noise (circled). Figure 21(a2) is from the new method which is not only able to detect both strong and weak CPPs, but also suppress the noise CPPs. Figure 21(a3) and (a4) demonstrate the classification of GETs into straight line and curve categories. Figure 22 and Figure 23 show two types of images (i.e. indoor and outdoor), their extracted edges, and detected CPPs. The proposed CPP detection method provides promising results.

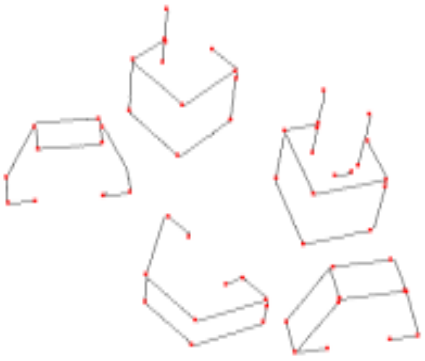




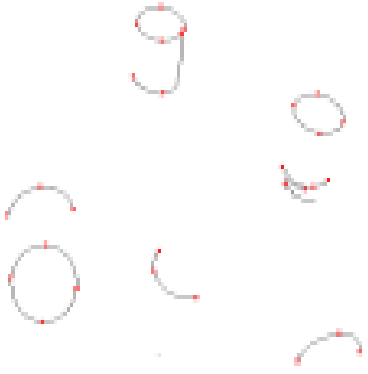
(a1)



(a2)



(a3)



(a4)

Figure 21 Comparison results with the previous method.

Figure 22 Example results from an indoor scene image.

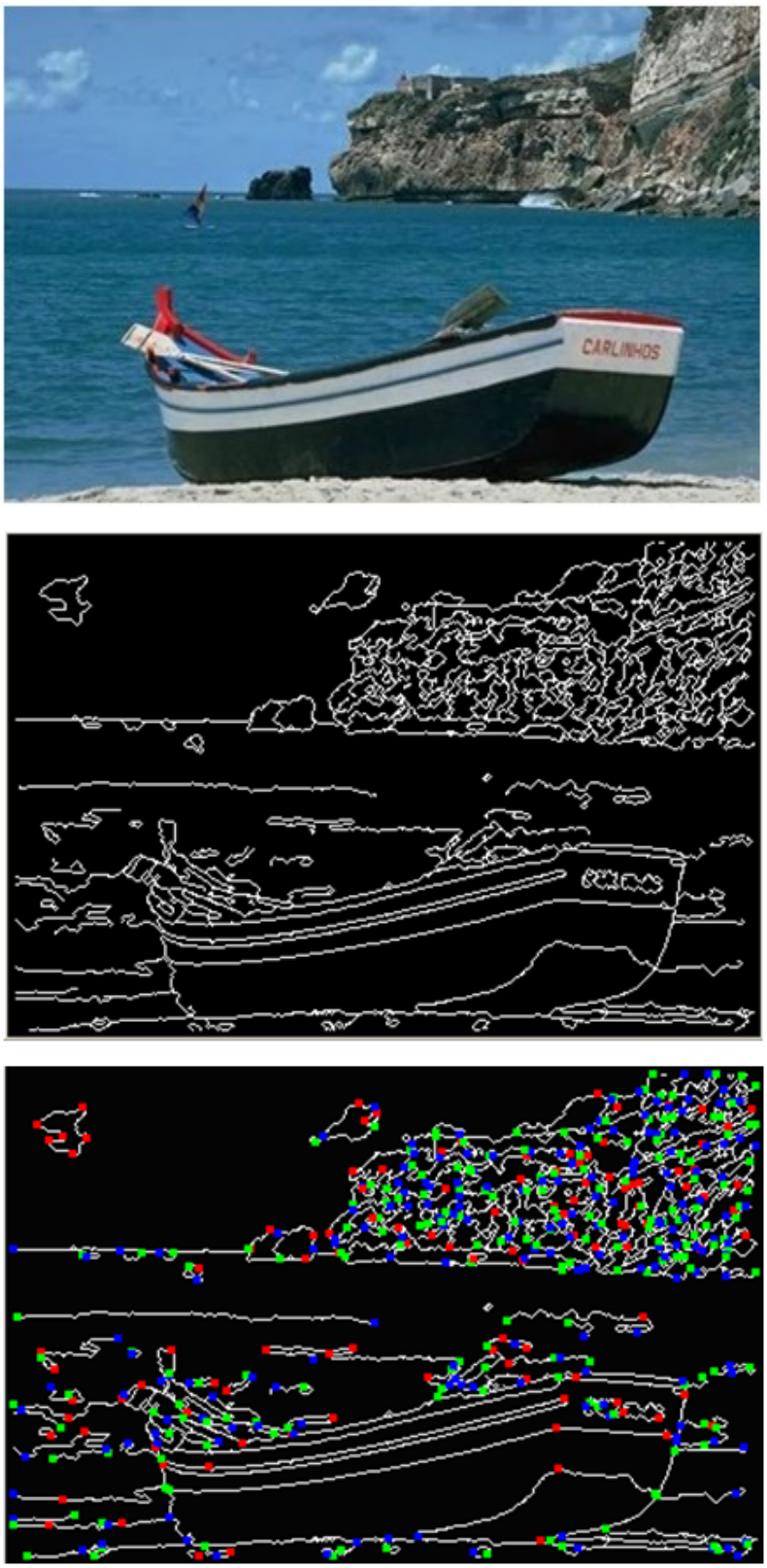


Figure 23 Example results from an outdoor scene image.

3.6 3D PERCEPTUAL SALIENCE

The shape-based salient features, CPPs and GETs, extracted from 2D images only contain 2D X-Y pixel values without 3D spatial information. To allow computers to truly understand the visual world, one key is to formulate the vision problem in terms of the underlying 3D scene, applying real-world knowledge to gain a 3D spatial understanding of the objects and the scene contents. As we stated earlier in Chapter 1.4, there are mainly 2 ways to obtain 3D data: estimate the depth data directly from 2D images according to some prior knowledge, and retrieve the depth data from advanced 3D cameras. Here we use a low-cost Kinect camera to get the 3D spatial data, and provide 3D GETs and CPPs.

A Kinect camera provides 2D color VGA (640×480) images with an aligned depth data array for each. Each pixel of a VGA image has a depth value from 0.10m up to 2m. A depth image can be derived by scaling each pixel's depth data into the intensity range [0-255]. Figure 7(b-c) show the color image and its corresponding gray scaled depth image, where the brighter intensity means closer distance to the camera, vice versa. Figure 7(d) shows the extracted GETs/CPPs. Besides the depth data (Z), each pixel of a depth image also contains horizontal (X) and vertical (Y) spatial values, which are obtained according to the basic geometry principle (see Figure 24) and camera parameters:

$$X_{3D} = \frac{Z \times X\_Image}{focal\_length}$$

$$Y_{3D} = \frac{Z \times Y\_Image}{focal\_length}$$

where *X_Image* and *Y_Image* are the object size in an image, and are calculated by multiplying the pixel number by the pixel *pitch size*.

$$X\_Image = \text{X\_pixel} \times pitch\_size$$

$$Y\_Image = \text{Y\_pixel} \times pitch\_size$$

Both *focal_length* and *pitch_size* are the camera intrinsic parameters. From the specification of the Kinect camera, the *focal_length* is 120 mm and the *pitch_size* is 0.2 mm.

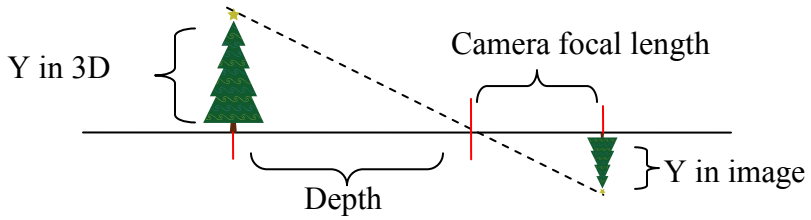


Figure 24 Object size calculation.

3D CPPs and GETs can be obtained by mapping the spatial data (XYZ) to the extracted 2D GETs/CPPs. An example illustrating the 3D features is given in Figure 25, in which the 2D edge pixels (Figure 25(b)) are extracted from a grayscale image. Having had the aligned depth image (Figure 25(a)), each edge pixel is mapped into a pixel in the depth image on a one-on-one basis. 3D data of each edge pixel needs to be denoised by using an n×n local window which covers all neighbors and itself (3×3 filter in Figure 25(c)). Depth value of each edge pixel is smoothed by a median filter [137],

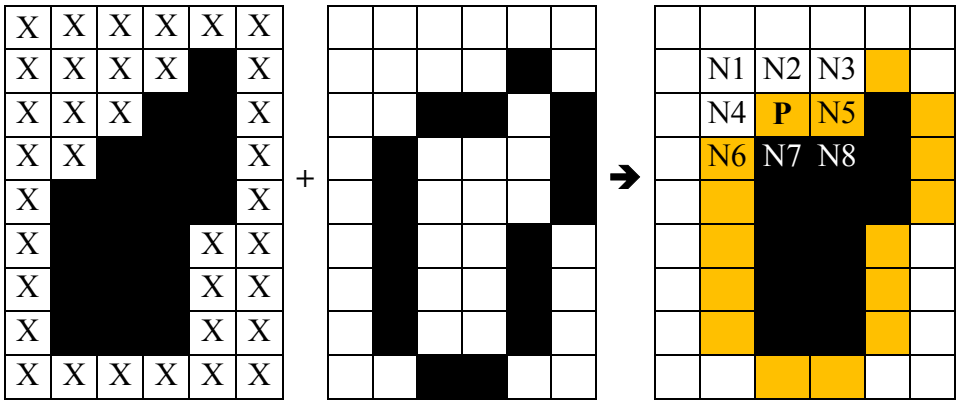
$$Z(P) = \arg median_{i=1}^{n^2} Depth(N_i), \tag{3.5}$$

where N_i is a pixel within the filter window, $Depth(N_i)$ is N_i's original depth value, n is the filter window size which can be specified by users. The 3D Xs and Ys for the edge pixels are obtained in a similar way,

$$X(P) = \arg median_{i=1}^{n^2} X_{3D}(N_i), \tag{3.6}$$

$$Y(P) = \arg median_{i=1}^{n^2} Y_{3D}(N_i), \tag{3.7}$$

We are living in 3D world; 2D visual data inevitably creates barriers for vision perception. Thus, we assume that 3D visual data contains more semantics and server better for recognition tasks. GETs and CPPs equipped with 3D data gain more ability to describe the 3D salience which directly links to the object in 3D space.



(a) Depth image (b) 2D Edge (c) 9-grid neighbors

Figure 25 3D Edge pixels of an object.

3.7 CONCLUSION

Extracting feature level visual saliency is crucial for building a representation attracting visual attention. In this chapter, we first introduced a non-parametric statistical-based method to implement and improve a PO-based curve partition and grouping model for 2D shape-based salience feature extraction. In this PCPG model, the generic perception criteria of curve partition and grouping are simulated by utilizing the patterns of sequentially ordered curve pixels and the bin statistics of the slope property. The partitioned segments are easily classified into straight line or curve tokens by using the same generic criteria generated during the curve partition. The experiments demonstrate its strength for handling complex images efficiently and effectively. The GET and CPP types are useful shape elements for supporting image semantic computing, and can be further extended by grouping GETs and CPPs to build a comprehensive image language for shape based content coding, pattern recognition and indexing. Secondly, by integrating XYZ data from a depth sensor, Kinect camera, 3D shape-based visual salience entities (3D GETs/CPPs) are able to describe the objects and the scene context in visual world. 3D GETs and CPPs-based image representation is the salience map at the feature level; it provides selective information reflecting the visual importance for high-level interpretation tasks.

CHAPTER 4 3D OBJECT SALIENCE FOR BODY PART CLASSIFICATION

4.1 INTRODUCTION

Human gestures and actions are the spatial-temporal patterns presented in frames from visual sensors. Human body poses from individual frames reflect the static status (snapshot) of a gesture or action at a certain moment. A set of ordered 3D body poses further provide context characteristics for inferring the high-level interpretation of human gestures/actions in 4D spatiotemporal space. Therefore, body pose estimation is crucial in human gesture/action recognition systems. In the previous chapter, we introduced a method extracting bottom-up feature level visual salience entities, 3D GETs/CPPs, from each 3D image. Now, we describe how to utilize top-down prior knowledge to enrich the feature level salience map for selecting and grouping effective visual salience at the object level; and how to build the object level salience map for pose estimation.

The articulated human body structure can be mainly divided into several parts including head, torso and four limbs, and has high degrees of freedom. Recognizing the complicated body structure from real-time visual stream data is a challenging object level operation task due to 1) wide variance range of human body shape; 2) motion variants; 3) self-occlusion and 4) impacts of the environmental condition (lighting, noise, view point etc.). To achieve reliable performance of body pose estimation, a divide-and-conquer method is often applied to accurately segment and label individual body parts first [145]. And then, a robust grouping operation is used to derive the body poses. Previously, researchers have made efforts to develop effective approaches for body part segmentation, classification and pose estimation, such as stick figure modeling, blob model [138], and some of them have achieved encouraging performance for specific applications. However, it seems that there is no existing generic solution that satisfies all expectations, i.e. to be easily applied to real-time applications under all circumstances. One of the major issues is that the gap between low-level features and the motion perceptual semantics could not be well bridged. We present a visual attention and

perceptual organization model-based approach which selectively extracts, classifies, and groups low-level features into object level visual salience for body pose estimation.

Body part classification is required prior to pose estimation. They both are top-down tasks that need the supports from the bottom-up information. In this chapter, we first introduce an approach that utilizes the top-down body structure prior knowledge to enrich the feature level salience map for body part classification. Here we assume that a single person's actions are captured by a 3D camera in a front view, and task-specific prior knowledge and definitions, such as human body structure, kinematic constraints are known. The Bottom-up salience entities (3D GETs/CPPs) combined with top-down prior knowledge provide enhanced visual evidence for information selection. Several laws of perceptual organization (PO) play roles during the visual feature grouping. The constructed feature-level salience maps provide reduced search space and strong heuristics for body part detection and tracking.

Secondly, the outputs of body part classification, classified 3D body CPPs, are the object feature data containing rich spatial distribution information and semantics for body pose estimation. According to prior knowledge, body poses of most human gestures and actions mainly are contributed by the limb states. The areas that limb CPPs spread out are salient region where a limb tree is built as the salience index for searching. Tree traversal criteria encoding prior pose knowledge are imposed to assign weights on individual salience entities (i.e. Limb CPPs). The updated object level salience map contains selected regions for pose estimation. Experimental results show the performance of the proposed approach. The overall architecture of our solution is illustrated in Figure 26.

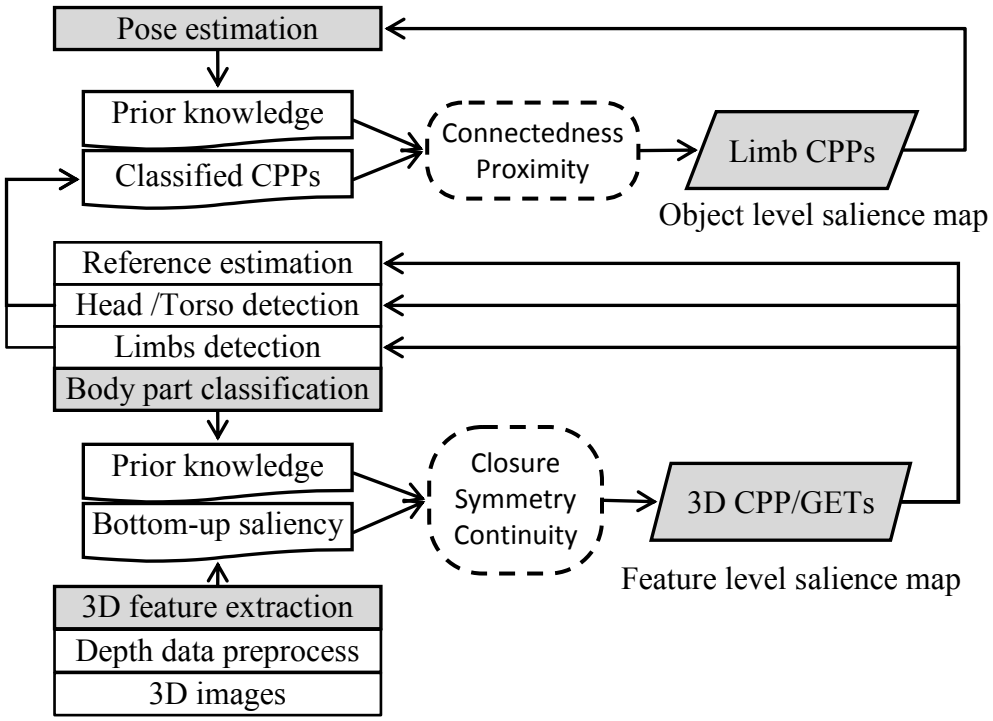


Figure 26 Architecture of body parts classification and pose estimation.

4.2 BODY REFERENCE ESTIMATION

Before detecting individual body parts, we first estimate the overall location of the human body. According to the environmental setting, suppose there is only one person in the FOV, all visual salience entities (3D CPPs/GETs) in the scene are related to the human body after removing noisy data. The human spatial location could be preliminary represented by a centroid of a largest pixel mess illustrated in Figure 27(b). Instead of considering all pixels in an image, we use a polygon as the representative of the pixel mess, and the center of polygon is taken as the mass centroid. The polygon is a hexagon (may not be a regular one) that consists of 2 the highest CPPs (at the top), 2 lowest CPPs (at the bottom), the most left and right CPPs. If the most left or right CPP sits on the same place of one of 4 top and bottom CPPs, the second left or right one will be chosen (Figure 27(a)). The centriod (C_x,C_y) is calculated from 6 vertices of the hexagon:

$$C_x = \frac{1}{6A}\sum_{i=0}^{5}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)\,, \qquad (4.1)$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{5}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)\,, \qquad (4.2)$$

where x_i, y_i are the coordinates of hexagon vertices, A is the area of this hexagon.

Based on the body centriod, a vertical line across the centroid is called central line of the body in Y direction, ending at the Y positions of both top and bottom boundaries of the

(a) (b)

Figure 27 Body reference position determination.

hexagon. Its length is approximately the height of the human if the full body appears in FOV. Similarly, a horizontal line across the centroid, ending at the X positions of the most left and right boundaries of the hexagon, is taken as a baseline. The length of the baseline approximately equals the width of the human body. Note that depth positions of both lines are derived from the 3D pixels of the region they pass through. The 3D location of a single human body can be roughly represented by both lines (Figure 27(b)), and acts as an important cue for individual part recognition.

4.3 CPP-BASED HEAD DETECTION

Compared with the other body parts, spatial and appearance properties of the human head are relatively unique and have fewer variants and ambiguities. The human head is easier to be identified than other parts, and can be used as the strong heuristic for inferring other body parts. The head can be found once the face has been detected. There are many existing libraries/algorithms for face detection [139]. However, those face detection algorithms require more computation costs to process image details. In additional, most face detection methods require desirable view angles, i.e. it will fail if a person's face is

not facing the camera directly, which would often happen during the motion of human activities. Therefore face recognition-based head detection is not suitable for real-time gesture recognition tasks. In contrast, the size and contour shape properties of human head have less variability, and are discriminative to the other body parts. Most importantly, they are view and rotation invariant. By utilizing 3D perceptual shape features, the head can be detected by our CPP-based method.

According to the prior knowledge, the head is a single object with a convex hull-like shape that can be described by a set of stable salience entities (3D GETs/CPPs) connecting with the torso via the neck. CPPs are the connecting points of GETs, and the distribution of the CPPs and GETs along the edge traces with the convex hull-like shape (Figure 29) can be used to detect the human head. Figure 28 shows some hull-like edge shapes with CPPs connecting 2 GETs. The shade areas are the foreground object. Figure 28(a) and (b) show the convex and concave shapes respectively. From feature-level salience entities, our head detection method first selects the convex hull shapes which have high probability of being the head contour, and then check their shape, spatial and size.

(a) Edge traces with convex hull-like shapes

(b) Edge traces with concave shapes

Figure 28 Several types of GET combination of edge traces.
Note the shade areas belong to the foreground object.

4.3.1 Convex Hull Shape Selection

The convex hull selection algorithm is based on some basic geometry principles. In Figure 29, there are 5 CPPs along an edge trace. First, the algorithm takes the first CPP

(CPP_1) of the edge trace as a start point to check the next 2 CPPs (CPP_2 and CPP_3) along the edge trace in turn. If the shape is a convex, CPP_3 would make a smaller angle than CPP_2 does if (and only if) CPP_3 lies on the right side of the connecting line between CPP_1 and CPP_2 (dashed line in Figure 29). The angle *a2* is smaller than *a1* in Figure 29. Thus these three CPPs form a convex hull-like shape. The convex hull shape continues if the next one (CPP_4) keeps on the right side of the line between its two previous neighbors CPP_2, CPP_3. Therefore just checking CPP locations relative to the corresponding lines is able to detect the convex hull-like shape on an edge trace. This method is accurate and efficient in that only 5 additions and 2 multiplications are needed for testing condition of each point. However, satisfying the above conditions only proves that the edge trace has a hull shape. We need to check whether the hull-like shape is convex or concave. It can be verified by checking the location of their center point. If the center point of CPPs forming the hull shape falls into the shade area, it is a convex shape; otherwise, it is a concave. Point A in Figure 29 is the center point of CPP_1, CPP_2 and CPP_3 and CPP_4, and is within the shade area. Point B is the center of CPP_3, CPP_4 and CPP_5, and is outside of the object region. CPP_1 to CPP_4 form a convex hull shape, CPP_3 to CPP_5 are from a concave hull shape.

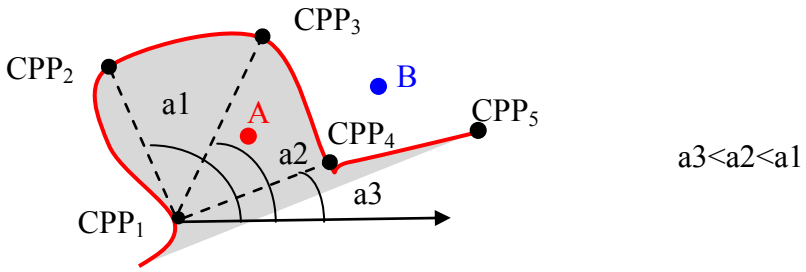


Figure 29 Convex hull shape detection.

There could be many hull-like shapes from the outputs of the convex hull selection algorithm, which need to be further filtered out since only one would belong to the valid head contour for a single person. Some convex hulls like the shapes 1, 3 and 4 in Figure 30(b) either do not have similar sizes to the human head, or, their relative positions to the body reference have less likelihood to be the head according to the prior knowledge. To

locate the head in the image accurately, the verification process checks three criteria: head shape, location and size.

4.3.2 Head Shape Verification

A metric, CPP Vertical Distribution (CVD), of a convex hull edge trace, is defined to evaluate the head shape likelihood. The CVD value is a ratio of the vertical distribution of CPPs along a convex hull edge trace:

- Set a middle point P_0 between the start and end CPPs,
- Sum up the Y distances between P_0 and all CPPs above P_0, and assign to S_1,

$$S_1 = \sum_{i=1}^{n} (y_0 - y_i), \quad \text{if } y_i < y_0 , \tag{4.3}$$

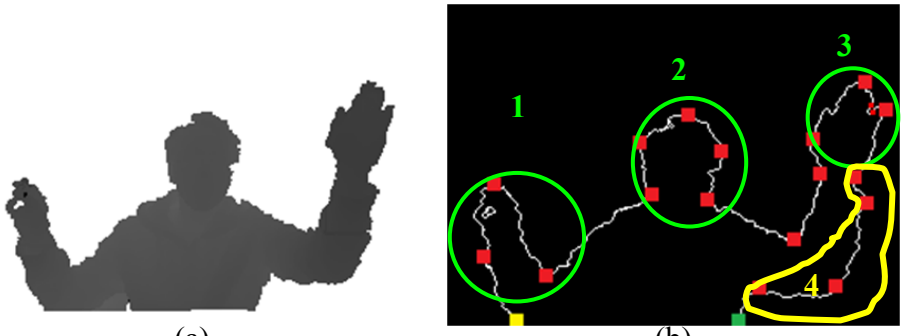
- Sum up the absolute Y distances between P_0 and all CPPs, and assign to S_2,

$$S_2 = \sum_{i=1}^{n} |y_i - y_0|, \tag{4.4}$$

- CVD is a ratio of S_1 and S_2,

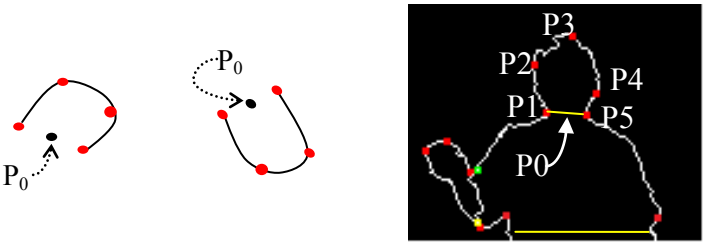
$$CVD = \frac{S_1}{S_2}. \tag{4.5}$$

The CVD value is in a range from (0, 1). The bigger the CVD value, the more likely the shape is the human head. In Figure 31 (c), P_1-P_5 are 5 CPPs, P_0 is in the middle between the start and end points (P_1 and P_5). S_1 is the sum of the Y distances between P_0 to P_1, P_2, P_3 and P_4. S_2 is the sum of the absolute Y distances between P_0 to all (P_1-P_5). The value of CVD (S_1/S_2) is 1 if all CPPs are above P_0. For a shape illustrated in Figure 31(a), its CVD is close to 1; for a shape of Figure 31(b), its CVD is close to 0.



(a) (b)

Figure 30 Multiple convex shapes detected from edge shapes.



(a) (b) (c)

Figure 31 CPP-based head detection method.

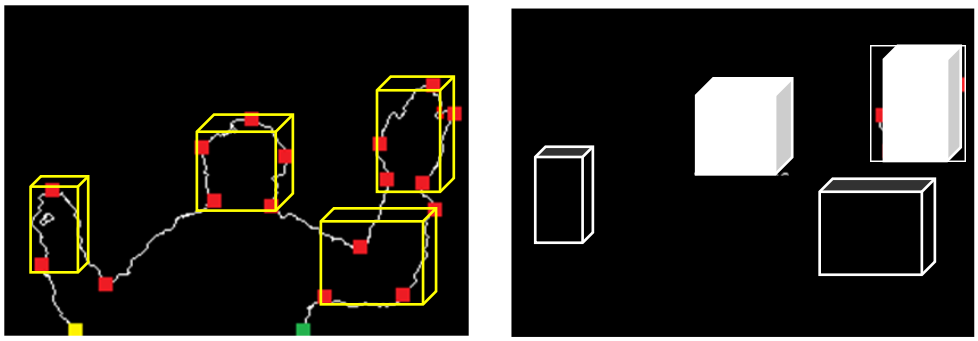
This CPP-based head shape evaluation is valid even when the head is tilted (Figure 31(a)). The head shape would not like the one in Figure 31(b) even when he/she is nodding. The head will be upside-down only if the human body is so, which is not under our system assumption. Therefore, the CVD-based head shape detection is robust.

4.3.3 Head Location Verification

As we introduced earlier, the body reference is represented by both the central line and the baseline. Since the intercrossed baseline and central line connect to the boundaries of the human body, the 3D spatial information of the human body can be easily derived. Meanwhile, the spatial data of the convex hull head contour (determined by its CPPs) is known. The depth (Z) positions of the head contour and the human body should be close. The farther they are apart on the Z direction, the less likely the hull-like shape contour is the head. The X position of the head is roughly close to the body's central line; otherwise the convex hull is less likely the head. The baseline is below the head area. If a convex hull is far away above the baseline, it is more likely the head.

4.3.4 Head Size Verification

From the CPPs of a convex hull shape, the top one (with the smallest Y value, e.g. P_3 in Figure 31(c)) is taken as the leading point of the head for building a 3D box. This 3D box is the shape enclosing the connected pixels below the top CPP within a certain XYZ range, and its size is analogous to the head size. Since the average head size is 25cm in height, 18cm in width, if the box size is far away from the average size, the corresponding convex hull is not on the head contour, vice versa.



(a) (b)

Figure 32 Result of head detection.

In summary, our method first selects the convex hulls consisting of feature salient entities (3D CPPs/GETs), and then builds a 3D box around them as the representatives for each, and finally checks 3 criteria to compute the head likelihood confidence value as the weight. Figure 32 shows 4 selected areas contain convex-like shapes from the feature-level salience entities as the candidates, and the brightness of right-side 3D boxes indicate the likelihood confidence value. Figure 32(b) is the salience map for the head detection. A straightforward method is applied to choose the final head location with the highest confidence value passing a pre-defined threshold.

4.4 Torso Detection

The torso occupies the major area of the human body, and its shape has left and right symmetric boundaries under the head, which can be grouped by a set of bottom-up salience entities (3D GETs/CPPs). Similar to the head detection, we first find out all long straight line GETs, and then makes all possible pairs according to their locations, i.e. each pair of 2 GETs should be on the both sides of the central line. Each pair is the left-right boundaries of a torso candidate. Three criteria are used to determine confidence value of the torso boundary: symmetry in length, symmetry in the distance to the central line, and the size. The final torso will be the pair with higher symmetric degrees and similarity in size (height and width) to the central line, baseline and head size. Once the torso boundary has been determined, the width and height of the torso can be derived from its boundary. The depth position of the torso can be determined according to the head location. Thus, the spatial range of the torso can be efficiently estimated. In sum, the

bottom-up salience entities, top-down prior knowledge about the spatial reference, and Gestalt Law of Symmetry are working together to provide selective visual information for the torso detection. Having had the head and the torso detected, other related body parts can be inferred accordingly:

- ***Neck***: it locates at the middle of the junction area between the head and torso.
- ***Shoulders***: They are joint parts connecting the torso and upper limbs, and locate at the top left and right of the torso.

4.5 LIMB DETECTION

Once we have the head and torso identified, the main structure of the body system is established. The limbs are a general term including arms, hands, legs and feet. Feature-level visual salience entities 3D GETs/CPPs that connect to the torso belong to the limbs. Here we introduce a CPP/GET-based method to further classify upper limb CPPs into the left and right ones. The Gestalt laws of connectedness and proximity play roles in this process, i.e., edge features closer or connected to the left and right shoulders belong to the left and right limbs respectively. There are 2 steps within this method.

- **Limb GET Definition**

  Each limb CPP has at least one connected GET, so-called a limb GET, which is a segment of an edge trace. In the spirit of Gestalt law of connectedness, the spatial distributions of limb GETs give us more information about the limb locations. Figure 33(a) is the front view of a human upper body; In Figure 33(b), the green, red and yellow points are the limb, head and torso CPPs respectively, and the white lines are the limb GETs connecting to the limb CPPs along the hands and arms. The left and right limb GETs are connected or closer to the 2 shoulder where two bigger solid yellow circles are.

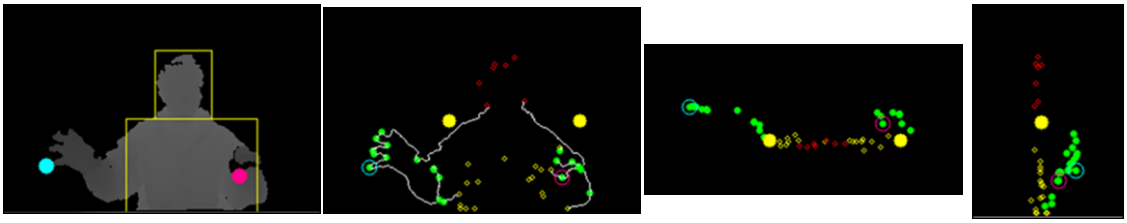
- **Limb GETs Grouping**

  Limb GET grouping is to cluster the limb GETs based on the K-Nearest Neighbor (KNN) method. For 2 upper limbs, K is 1 for assigning GETs to the closest part, and

both shoulders are the seed points for grouping. Each limb GET has a set of edge pixels, and the distance between 2 GETs is measured by the Hausdorff distance [140] which calculates how far two sets of 3D pixels are from each other. Thereafter, all upper limb GETs are clustered into 2 groups (left and right); and left and right limb CPPs are then determined accordingly. The spirit of Gestalt law of proximity is embedded in the Hausdorff distance measure.

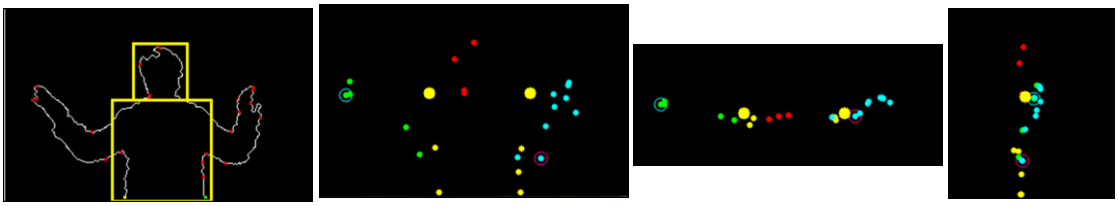
Visual salience entities CPPs reveal shape semantics. Classified CPPs including head, torso, and limb CPP clusters reflect the human body part salience at the object level.

Figure 34(a) shows the head and torso boundary boxes. Figure 34(b) shows groups of CPPs on XY plate without GETs, the red points are the head CPPs, yellow ones belong to the torso, green and blue ones are the left and right limbs respectively. In Figure 34(c) and (d) show the same CPP distribution presented in the YZ and XZ plates respectively. The spatial distributions of 4 groups of salience entities well reflect the body parts spatial status, and provide bottom-up object level visual salient evidence for pose estimation. The reduced search space and computational costs facilitate the object level operations.



(a) Front X-Y view (b) Edge features of (a) (c) XZ view (bird's eye) (d) YZ view

Figure 33 Limb CPPs and GETs of a upper human body.
The light blue point is the left hand; the pink point is the right hand.



(a) head & torso (b) XY view (c) XZ view (bird's eye) (d) YZ view

Figure 34 3D body part CPP classification.

4.6 BODY POSE ESTIMATION

From observations of the human motion, meaningful gestures and actions, such as, driving, throwing, grabbing, even kicking soccer, are mainly parameterized by the details of the limb configurations. Especially, the layout of the limb end parts (i.e. hands or feet) provides the overall limb structures. The spatial distribution of limb CPPs at the end areas are even more effective for delivering the attentional semantics. Meanwhile, the human head and torso have less spatial and shape variability while human actions/gestures are performed. So limb CPPs of the hands are the salience attracting attention for gesture/action interpretation. Based on this prior knowledge and classified body part CPPs, we take the head and torso as the main-board, and select and group the regions around the limb end CPPs to form the object-level salience map for pose estimation.

When performing gestures or actions, the hand locations are usually farther away from the main body than other limb parts. To locate them, for each limb CPP cluster (left or right), a minimal spanning tree (MST)-based saliency index is built for searching and weighting the limb CPPs according to the 3D body distances. The weighted object-level salient map is formed up for pose estimation by apply tree traversal criteria encoding both prior knowledge and perceptual organization laws. The process includes following steps:

- Limb graph construction

  An undirected graph *G(V,E)* is utilized for each limb CPP group. Each limb CPP is a vertex *V* of the graph *G*, and each edge *E* represents the direct 3D distance of the neighboring CPPs. Two limb CPPs are considered as neighbors if:

  - the CPPs are on the same edge trace; or
  - the 3D distance in-between does not exceed a threshold (e.g. 8cm).

  The graph *G* just provides the coarse spatial relations among CPPs.

- MST

  As we stated earlier, the hands and feet are at the end areas of the corresponding limb parts. The body torso acts as the baseboard of body pose, and the articulated body structure is mainly characterized by locations of the limb's end points which usually stick out and are apart from the main board. Based on the limb graph, a Minimal

Spanning Tree (MST) is built as the spatial index to locate the limb CPPs at the limb end area efficiently. We can take the body centroid (introduced early) as the reference point, and include it into the graph *G*. Build edges *E* from all related limb CPPs to this reference point to make the graph connected. Then use this reference point as the root in Prim's algorithm to build a MST [135]. Figure 36(b) and (d) show the MST index structures. It is worth noting that several other points can be also used as the MST root, such as the left or right shoulder, middle point of the image bottom. The white circles in Figure 36(e)-(g) are the tree roots (reference points). The spirit of Gestalt laws of proximate and continuity is embedded in the limb graph and limb MST structures.

- saliency measure (traversal criteria)

  An evaluation method is to weight every limb CPPs on the corresponding MST index. Several tree traversal criteria are employed as the saliency measure for ranking the salience entities (limb CPPs):

  - The path distance (*PathDis*) between a node (limb CPP) and the tree root;
  - The direct distance (*DirectDis*) between a node (limb CPP) and the tree root;
  - Body distance (*BodyDis*) of a node (limb CPP) reflects the closeness of a limb CPP to the head and torso areas.

  Based on these measurements, the saliency weight of each CPP_i can be evaluated as:

$$w_i = P_{DirectDis}(CPP_i)P_{PathDis}(CPP_i)P_{BodyDis}(CPP_i), \tag{4.6}$$

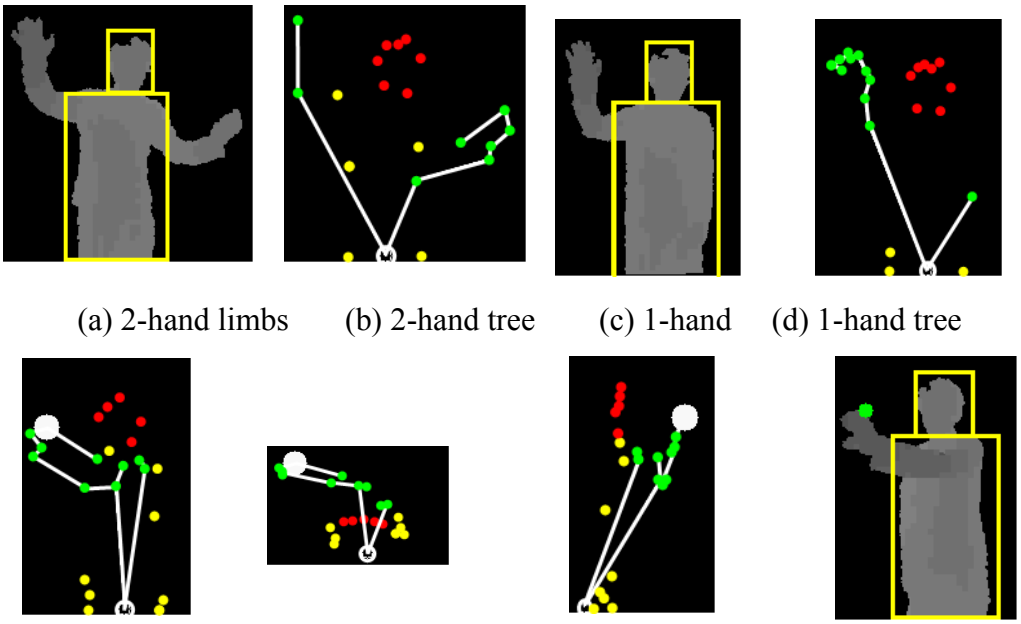
  The weight value reflects the top-down influences of object-level tasks. A weighted limb CPP is an object salience entity of the object level salience map which provides selected object information for locating the target object.

- End regions of limbs

  The region of the limb end part, e.g. a hand, would be the area surrounding the limb CPP with the largest weight value within a limb CPP group:

$$\arg\max_{i=1}^{k} w_i \tag{4.7}$$

  where *k* is the number of limb CPPs. and the selected limb CPP is the leading point of the target region (hand) (Figure 36(e-h)). After all the end regions of the limbs have been identified, the pose can be derived accordingly.

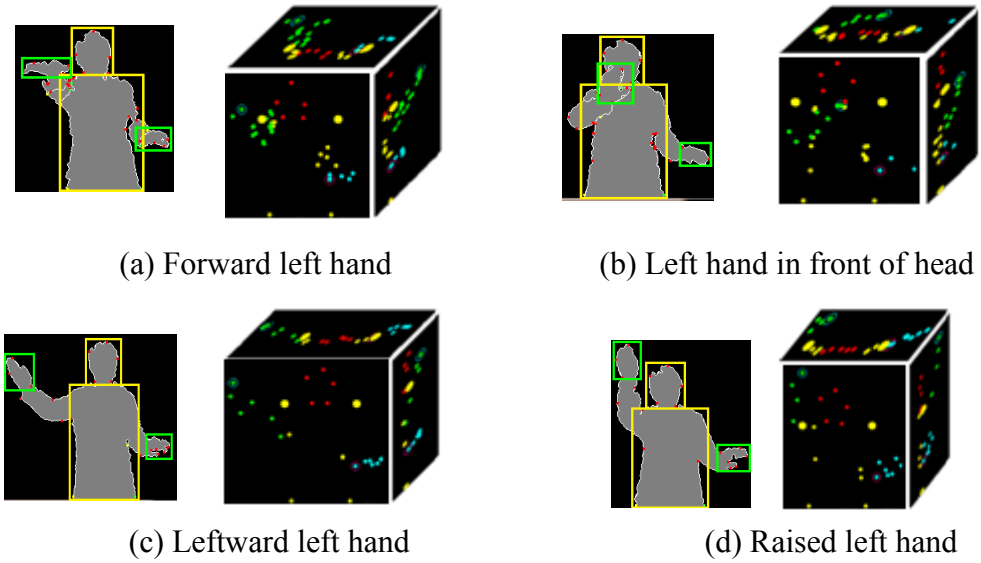


(a) 2-hand limbs (b) 2-hand tree (c) 1-hand (d) 1-hand tree

(e) salience in XY (f) salience in XZ (g) salience in YZ (h) Target

Figure 36 Limb tree structure for pose estimation.

Figure 35 shows 4 different body pose examples. The head and torso of each example are enclosed in the yellow boxes and the hands are in the green ones. The red, yellow, green and light blue points represent head, torso, left arm, right arm CPPs respectively.



(a) Forward left hand (b) Left hand in front of head

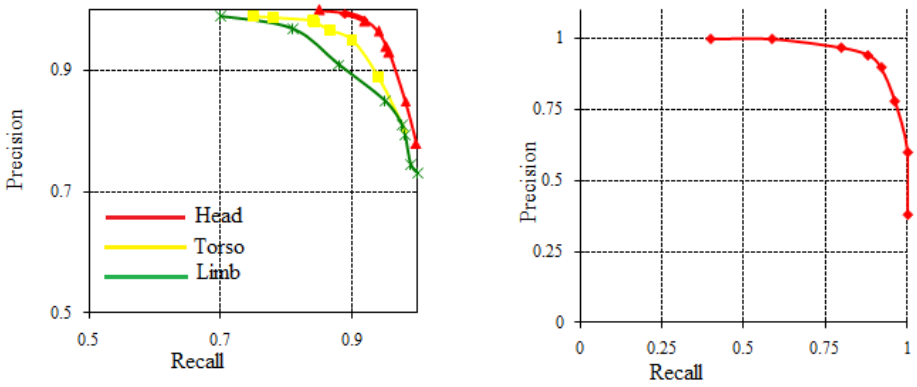
(c) Leftward left hand (d) Raised left hand

Figure 35 Body parts classification for different poses.

4.7 EXPERIMENTS

In this section we describe the experiments conducted to evaluate our body part classification and pose estimation methods. The algorithms were evaluated on the real-time image sequences involving a variety of upper body movements of a single person. The goal of this evaluation is two-fold. First, we evaluated the body parts classification performance of our system. Then, we examined our pose estimation method. In the experiments, the raw depth data was preprocessed by (i) scaling into grayscale depth image; (ii) downsizing the VGA image to QVGA resolution (320×240) and (iii) filtering out the background by setting a depth threshold. Figure 37(a) gives the precision-recall curves for the CPP classification from a real-time depth stream data containing 769 frames with 10928 human body CPPs.

Each point on the precision-recall curves (Figure 37(a)) corresponds to a specific threshold against the head likelihood confidence evaluated by the criteria defined in section 4.3. The head, torso and limb CPP classification tightly relies on the head detection results. When the head threshold likelihood is set high, the precision is high and the recall is low, vice versa. From the CPP classification results, the performance of the body parts segmentation and classification are sound. Especially the head CPP detection is performed very well from all frames. The error rates of torso CPPs and limb CPPs are relatively higher. There are mainly two reasons causing the classification failures: 1) head



(a) Body parts CPP classification (b) Hand tracking

Figure 37 Precision and recall of experimental results.

is not visible (Figure 38(a)); 2) regions of torso and head are defined in a way lack of flexibility (Figure 38(b)). A solution to these issues is to train a reference system and a backup body model which will be triggered automatically for classification when the confidence value is low.

The pose estimation for hand detection test was conducted on a 532-frame image sequence. Figure 37(b) shows the precision-recall (PR) curve of single hand tracking results. Similar to the classification, each point on the PR curve is a threshold against the likelihood value defined in Eq. 4.7. A lower threshold value will cause higher recall and lower precision, vice versa. Overall the hand tracking works well.

(a) Wrongly detected head (b) Missed head

Figure 38 Some failed cases of the body parts classification.

4.8 CONCLUSION

Human body parts classification and pose estimation are two fundamental tasks for developing successful human gesture/action recognition applications. This chapter presents a 3D perceptual shape feature-based body part classification approach facilitating the pose estimation process. By utilizing the prior knowledge of the human body structure, the bottom-up feature level salience maps are enhanced for efficient body part classification without involving heavy training process. The CPP-based classification results and pose prior knowledge speed up the process of body pose estimation, in that both the object level salience map, limb tree structure, and top-down search criteria work together to facilitate the estimation process. PO-based grouping laws play roles during

the salience map construction. The major advance of the proposed approach is that, following the visual attention and perceptual attention mechanisms, shape-based visual salience entities 3D CPPs/GETs and limb MST index provide both global and local views to facilitate the high-level interpretation tasks, which has good potential for modeling more complicated body poses. The object level salience map reduces the search space and computational costs for gesture and action interpretation.

CHAPTER 5 MODEL OF 4D SPATIOTEMPORAL SALIENCE

5.1 INTRODUCTION

One of our goals is to provide a type of generic perceptual representation for image sequences containing human gestures/actions, and evaluate how well these perceptual representations perform in recognition and classification tasks. Human gestures and actions are motion spatial-temporal patterns in images captured by a 3D camera, and their features are sequentially summarized from static properties of consecutive frames. It is challenging to let machines understand the human gestures/actions consisting of multiple dynamic structural components of multiple body parts in spatiotemporal space. We argue that limitations of the gesture/action representations cause the barriers for high-level gesture/action interpretation, and there are mainly two factors: low-level spatiotemporal features with less semantics, and holistic gesture representations without considering internal structures. Gesture representations directly from low-level features will contain more noise data and cause ambiguities for deriving 4D patterns. To deliver more semantics in the representation, some state-of-the-art approaches use intermediate gesture features to express visual salience, such as codebook, visual words or Bag-of-Words (BoW) that are made up by clustering low-level features according to the spatial and appearance similarities [64][73][74][75]. However, they are still far from being able to express the effective visual information for efficient high-level interpretation.

Besides the local and global appearances of gesture features, there exist internal structural properties that are less visually apparent and very difficult to describe. For example, the actual temporal relationship among multiple gesture features is not as simple as a sequential chain. It could be a very fuzzy structure, i.e. a mixture of chain, overlapped and stride modes with uneven steps. Due to its complexity and uncertainty, many previous approaches model gestures by either ignoring its intrinsic structures or using a simplified assumption so as to limit the performance of corresponding representations. In fact the intrinsic structural properties reflect the discriminative features for different gestures with different styles, and contribute to the gesture recognition and classification.

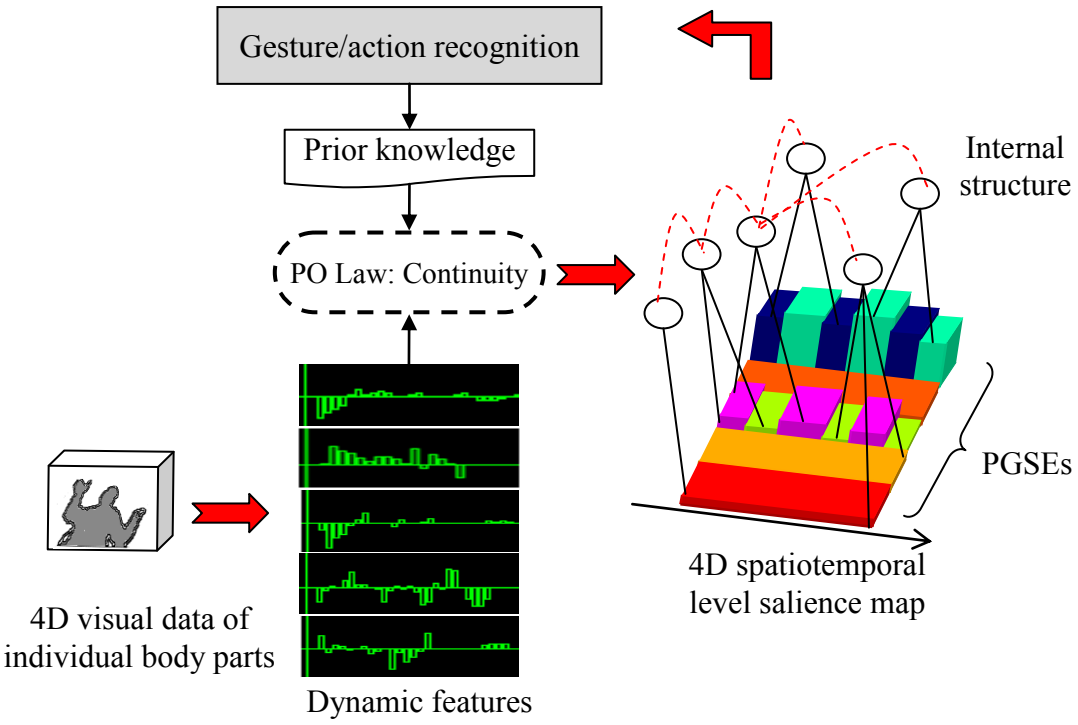


Figure 39 Perceptual gesture representation model.

Some works [89] [136] show that incorporating latent structures into the system can improve the performance.

Previously, we have introduced the solutions to human body part classification and pose estimation, which focus on extracting the static visual salience from a single frame. In this chapter, we introduce our approach that takes advantage of the bottom-up object recognition results to build the gesture representation at the object level, which is not only able to reflect perceptual visual attention salience, but also provide high-level qualitative reasoning. Based on the prior knowledge of different gesture types, corresponding body parts are tracked to generate five perceptual spatiotemporal dynamic change sequences including X, Y, Z, Shape and Orientation. According to the perceptual organization law of continuity, a set of generic perceptual gesture/action descriptors, Perceptual Gesture Salience Entities (PGSEs), are grouped as the gesture descriptor from the sequences to represent the extrinsic properties of gestures/actions at the 4D spatiotemporal level. In Figure 39, dynamic gesture features are extracted from 4D body pose sequences, and grouped into PGSEs (3D blocks). Each PGSE contains rich extrinsic

properties, such as duration, change type, change value, change rate, temporal moment, which are coded into the color and shape of a 3D block for better visualization. Within a set of 3D blocks with a certain pattern, the gesture/action intrinsic properties can be modeled as the temporal context relations among PGSEs. This novel representation acts as a visual salience map at the 4D spatiotemporal level, which encodes both attentional semantics and internal structures, and is able to facilitate the challenging human action/gesture interpretation tasks.

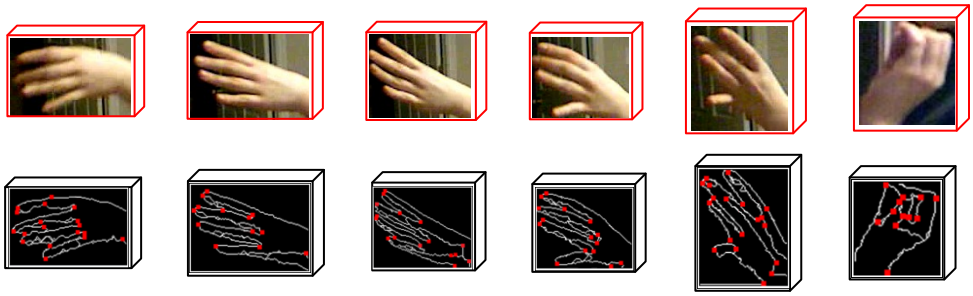


Figure 40 3D Boxes enclosing target objects (hands).

5.2 Perceptual Gesture Features

Instead of from pixel-level visual data directly, gesture features are derived from corresponding body parts at the object-level. In the previous chapter, we have introduced the approaches for human body parts classification. Among multiple body parts, the limb parts often play crucial roles in gestures/actions because of the following reasons: 1) limbs are the most active body parts; 2) gestures/actions are often characteristically determined by limb movements; 3) well classified limb parts lead to low-dimensional feature vectors with less computational cost. For instance, two hands are the most important body parts for many gestures/actions, such as driving, throwing, waving etc. On the other hand, motions of head, torso and feet/legs contribute less in these actions/gestures, and draw less attention with low salience. Each body part contains many perceptual shapes salience entities (3D GETs and CPPs). For the sake of simplicity and intuition, a 3D box is used as the representative to enclose all feature level salience entities of each individual interested body part. In Figure 40, a hand and its salience

entities are enclosed in a 3D box at each moment. As long as the object (hand) is detected correctly, all dynamic properties along the time serial can be expressed by the differences of the visual salience entities within 3D bounding boxes from neighboring frames. Overall there are three types of perceptual gesture features: motion trajectory, shape and orientation with five parameters, i.e. X, Y, Z, size, and angle.

1. Motion trajectory: The motion trajectory of a body part is measured by the dense trajectories of its salience entities (3D CPPs). Similar to Sun *et al.*'s method [63], the trajectories are obtained by matching the 3D CPPs between two consecutive frames. For a video sequence with k frames $\{f_1, \ldots f_k\}$, matches of all CPPs within a classified box were established between f_i and $f_{i+1}$for $1 \leq i \leq k$-1. A constraint is imposed to discard matches that are too far apart so as to reduce motion noise. The final displacement of a certain body part between consecutive frames is collectively determined by all 3D CPP's matches. A trajectory is formed by extending the matches over several frames. In Figure 41(a), it shows a frame f_i with a solid white boundary 3D box surrounding the person's left hand. In its previous frame f_{i-1} (not shown), everything is same except that his left hand is farther away from his left side body, and the location of the left hand in f_{i-1} is marked by a dash lined 3D box. According to our method, the motion trajectory of his left hand is collectively estimated by the CPP matches between f_i and f_{i-1}. Figure 41(b) shows a bunch of green lines each of which represents the motion direction of each pair of matched CPPs. The overall moving trajectory of the left hand is weighted by all pairs of CPPs. The extracted trajectory consists of changes on 3 axes, X, Y and Z, and their change values represent motion dynamics of the corresponding body part.

(a) depth image at frame f_i (b) CPPs in frame f_i

Figure 41 Collective motion trajectory estimation.

- Sequence of depth (Z) changes

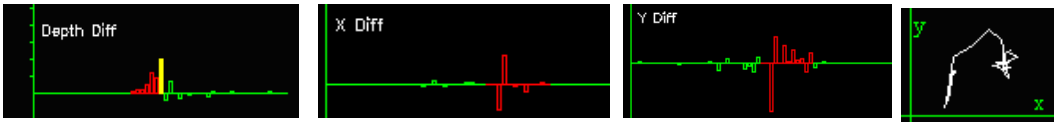
  The depth changes reveal the motion information regarding the speed and direction towards the camera. Each bar of the sequence in Figure 42(a) is the depth change value of an object within a short time period (100ms if the rate is 10fps). If the Z distances between two neighboring frames are unchanged, the value is ZERO. If the object is getting closer to the camera, several consecutive positive values appear on the sequence, and vice versa. If the frequency of the system is relatively stable, a bigger absolute value on the sequence means a faster speed on the Z axis, vice versa.

- Sequence of horizontal (X) changes

  The horizontal changes show the moving distance and direction along the X axis relative to the camera. Each bar of this sequence is the distance change value along the X axis of the target object within a short period. If there is no movement on the X axis, the value is ZERO. If the object is moving towards right-hand direction of a player, several consecutive positive values appear on the sequence (time serial), vice versa. If the acquisition frequency of the system is stable, this bar value is equivalent to the motion speed on the X axis (Figure 42(b)).

- Sequence of vertical (Y) changes

  Similar to the horizontal change sequence, the vertical changes reveal the moving distance and direction along the Y axis. Each bar of the sequence is the object distance change along the Y axis within a short period. If there is no movement on the Y axis, the value is ZERO. If the object is moving up, several consecutive positive values appear on the sequence, vice versa. If the acquisition frequency of the system is stable, this value is equivalent to the motion speed on the Y axis (Figure 42(c)).



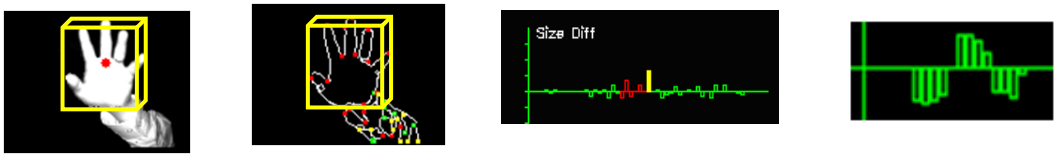
(a) (b) (c) (d)

Figure 42 Motion trajectory dynamics.

- Path on X-Y coordinates

  The motion path of the object in the X-Y plate shows its motion properties projected in 2D space (Figure 42(d)). Motion trajectories projected on other 2D plates (i.e. X-Z or Y-Z) or in 3D space can also be obtained by taking different combinations of 2 dimensions from XYZ.

2. Shape dynamics: To track shape dynamics, we currently use the object size as the metric, which is simple but still effective for estimating the shape properties. The object size is determined by the 3D box size of the corresponding body part. The box size changes reflect the dynamics of the object salience entity layout. Figure 43(c) is the size changing sequence. If a value on the size sequence is ZERO, it means no size change during a short interval. The positive or negative values indicate the object becomes bigger or smaller respectively. The larger the absolute value, the bigger the size change rate is, vice versa. Besides the object size, additional shape properties could be available by adding other statistical measures, e.g. GET type distribution etc..



(a) Object position (b) Edge features (c) Size changes (d) Orientation changes

Figure 43 Object position, edge features, size and orientation.

3. Orientation dynamics:

The orientation of the target object is an important gesture feature describing the status of the target body part. Orientation changes are discriminative for some gestures/actions, such as wave hand and flip over palm. For some gesture-based games and HCI applications, such as the dart throw and book page flip actions, the palm orientation can be used as a quantitative parameter to determine the release moment of a virtual dart, and the turning moment of the book page. To determine the object orientation, we need a reference system according to the environment setting, i.e. suppose a user is facing the camera while performing gestures/actions, and the target objects (e.g. hands) are in front

of the camera. Take the hand as an example, the palm can be treated as a plate in 3D space and can be determined by at least 3 spatial points. Since CPPs are the critical shape points along the object contour, an object plate could be generated by 3 CPPs. Each plate is a possible representative of the palm. The orientation of the palm can be measured by the angle between the camera and the palm plane, which equals the angle between their normal vectors. In Figure 44, 2 planes are in yellow and green colors respectively. *m* and *n* are their normal vectors, and the angle θ of both planes equals the angle θ' between 2 normal vectors. The orientation is measured as:

$$\cos\theta = \frac{n \bullet m}{|n||m|}. \tag{5.1}$$

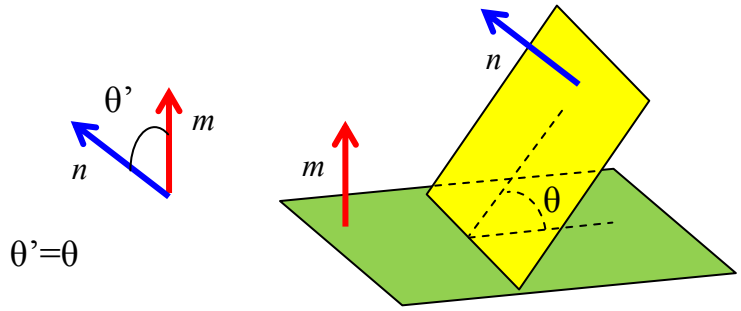


Figure 44 The angle between 2 planes.

The palm plane can be determined by any 3 CPPs within a target 3D box, and therefore there would be multiple candidate planes. To achieve reliable performance, the palm is collectively determined from multiple candidates with different weights that are factorized by the triangle size formed by 3 CPPs, i.e. a plane with larger triangle size is more likely the palm:

$$w_i \propto size_i \tag{5.2}$$

where w_i is the weight or parameter of a plane *i* formed by 3 CPPs, and is proportional to the triangle size. In practice, if there are many CPPs, too many palm candidates that always contain noise and outliers will be generated. We therefore apply RANSAC (RANdom SAmple Consensus) in order to get a robust orientation estimate. RANSAC is a resampling technique that divides data into inliers and outliers. The orientation angle

can be estimated from the minimal set of inliers with greatest support [144]. If the orientation angle of plane *i* is θ_i from Eq. 5.1, the final orientation of the palm is:

$$\overline{\theta} = \sum_{i=1}^{n} w_i \theta_i \tag{5.3}$$

where n is the number of inliers candidate planes selected by the RANSAC method. Figure 43(d) shows the orientation dynamics.

5.3 PERCEPTUAL GESTURE SALIENCE ENTITIES (PGSES)

There are a few high-level gesture descriptors in relevant literature, but they are too application-specific to provide generalities, and consequently, difficult to extend to other domains. Here we introduce a set of generic salience-based gesture descriptors. Those five perceptual dynamics are measured as the gesture/action properties for each body part, and well presented on the change sequences. These feature values are derived directly from local neighboring frames without further summarization. Inter-frame differential data contains outliers due to noise/errors from the bottom-up features. Not every value of these features reflects the 4D spatiotemporal salience since it lacks global views about the overall dynamic properties. Perceptually, there are three dynamic types of qualitative values on each sequence: increase, decrease and unchanged. To capture the dominant features of overall motion semantics, based on three qualitative values, each sequence can be grouped into several segments, so-called Perceptual Gesture Salience Entities (PGSEs), according to the perceptual organization law of continuity. PGSEs are the qualitative gesture descriptors that reflect the salience in 4D space, and can be applied in various human activity analysis applications.

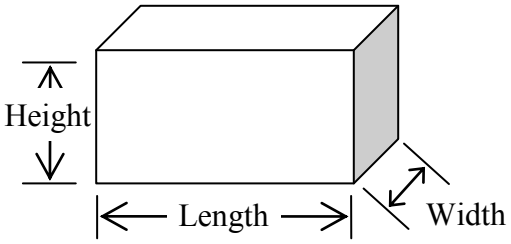


Figure 45 PGSE block representation.

5.3.1 PGSE Grouping

For each gesture or action, the motion of each corresponding body part are represented by five parallel dynamic feature sequences, X, Y, Z, size and orientation respectively. The PGSE grouping process is to find the consecutive dynamic changes on each sequence with the same or similar types according to the grouping Gestalt law of continuity. It is equivalent to finding partition points where changes of bar signs occur along time serials. If a discontinuity (change of bar sign) happens at a certain moment, the sequence is partitioned into smaller segments, PGSEs, accordingly. In Figure 46, each gray bar is derived from two neighboring frames, and represents their dynamic changes on a certain motion property. The signs of the first three bars are negatives (-), and it becomes positive (+) on the fourth bar. If this sequence is the changes in the Z direction, it means that the target object first moves a bit backwards, and then turns to forward towards the camera. After the fourth frame, the bar values remain positive (+) until the eighth bar. The value becomes negative (-) at the ninth frame. Thus, the discontinuities happened after the fourth and eighth bars so that the change sequence is segmented into three PGSE groups. For a better presentation, a 3D block list is used to represent segmented PGSEs. The width is fixed, and the length and height of each block represent the duration and the change volume respectively, and the ratio of the volume to the length is proportional to the normalized average speed (Figure 45).

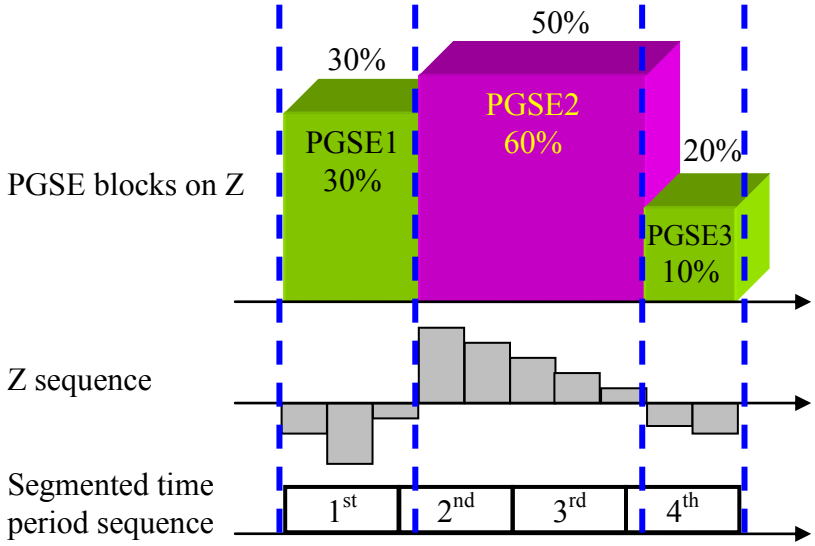


Figure 46 Time period of a video sequence containing a gesture/action.

5.3.2 Vector Descriptor for PGSE

For each grouped PGSE d, we define a feature vector as its descriptor

$$f(d) = \{Type, Start\ time, Duration, Volume, Speed, body\ part\}. \quad (5.4)$$

PGSE Type has 15 possible values. Each gesture/action has 5 different parallel PGSE dynamic sequences, including X, Y, Z, Size and Orientation, each of which has three qualitative change values: increase (+), decrease(-), and unchanged (0). Table 4 lists all PGSE types describing 4D gesture salience qualitatively.

Table 4 Perceptual Gesture Salience Entity type list.

No	Color	Property	Change	Description
1		X	+	Towards Right
2		X	-	Towards Left
3		X	0	No movement on X
4		Y	+	Towards Up
5		Y	-	Towards Down
6		Y	0	No movement on Y
7		Z	+	Forward
8		Z	-	Backward
9		Z	0	No movement on Z
10		Size	+	Increase
11		Size	-	Decrease
12		Size	0	Unchanged
13		Orientation	+	Front plane turning to the camera
14		Orientation	-	Front plane turning to the left or right side
15		Orientation	0	Unchanged

Start time is related to the sequential order of the PGSE in the video sequence. Each PGSE is a segment of the time period on a corresponding dynamic sequence. The *Start time* is the beginning position of the segment on the temporal serial. In Figure 46, if the time period of a gesture/action is equally divided into 4 zones, PGSE1's start time falls into the 1st zone, PGSE2's start time is within the 2nd one, and PGSE3 is on the 4th zone. *Start time* can be a normalized value invariant to various gesture/action durations.

Duration is the time period of a PGSE. Either it could be a ratio of the time period of a PGSE to the whole time period; or, it is a real time value which would be crucial for motion parameter measurement, e.g. speed. In Figure 46, 3 colored PGSE blocks have their relative duration values with 30%, 50% and 20% respectively. The length of a PGSE block represents this time duration.

Volume represents the dynamic change value with respect to the dynamic property within the video sequence. It is always Zero if the PGSE change type is 0; otherwise the volume of each PGSE is the sum of its underlying inter-frame changes. Similar to the *Duration* feature, *Volume* could be either a normalized or real value. The change volume would be useful for deriving demanded motion parameters, such as distance, speed, turning angle. In Figure 46, the 2D gray bars represent the underlying dynamic changes in inter-frame basis. Colored 3D PGSE blocks are the higher-level representatives that cover the underlying 2D gray bar sequence. Relative change volumes of 3 PGSE blocks are 30%, 60% and 10% respectively. The height of a PGSE block is proportional to the change volume.

Speed is the volume change rate of certain property for a gesture/action. The speed value will be Zero if the change type is Zero (0). For the other positive and negative changes, the speed value would be either an average speed or the peak speed within the duration. To get normalized average speed value, we can take the ratio of the average speed to the peak speed within each sequence (one property type).

Body part could be head, torso, and limbs.

The PGSE feature vector contains both qualitative and quantitative descriptions about gesture salience. Here we take a real human action for demonstration. A throw action is performed in front of a 3D camera with the right hand, and its dynamic features are presented in Figure 47(a) showing 5 different parallel feature channels. According to the PGSE grouping method, 6 PGSEs are obtained from 5 sequences, and all are marked with red boxes and labels with d_1 to d_6, PGSEs=$\{d_1, d_2, d_3, d_4, d_5, d_6\}$.

- d_1: It indicates the silence (no movement) on X axis. Since all values are small enough, the volume of this PGSE is close to zero. The small height of this PGSE block reflects the motion static on the X direction within a time period. This is the only segment on the X sequence. Therefore the start time is 1; duration is 100%; volume is zero; speed is 0. The vector descriptor of the PGSE is:

  $$f(d_1) = \{X0,1,1, 0, 0,Right\ hand\}.$$

  Its 3D block is a flat plate due to its zero volume.

- d_2: it represents positive change property about motion trajectory on Y direction. There are 2 PGSEs on this Y change sequence, d_2 and d_3. d_2 with positive changes takes 50% of whole time period; it starts at the beginning, 40% of changes with respect to the whole sequence. The normalized average speed value is 0.7. The vector is:

  $$f(d_2) = \{Y+, 1, 0.5, 0.4, 0.7, Right\ hand\}$$

- d_3: it is the negative PGSE on the Y direction. Its Start time is at 3rd zone, and the duration is up to 50% of the whole period. It contains 60% changes with respect to the all Y changes. The change rate is higher than that of d_2. Its vector is:

  $$f(d_3) = \{Y-, 3, 0.5, 0.6,0.8, Right\ hand\}.$$

  The blocks of d_2 and d_3 are shown in Figure 47(b).

- d_4: it is the PGSE block with large positive volume on the Z direction. It is the biggest one among 6 blocks occupying the whole sequence. The relative change value is 100% over the Z changes. The vector is:

  $$f(d_4) = \{Z+, 1, 1, 1, 0.8, Right\ hand\}.$$

- d_5: there is no big change on the hand size. d_5 is the only object size related PGSE. The volume and speed are zeros. The vector is:

$$f(d_5) = \{Size0,\ 1,\ 1,\ 0,0,\ Right\ hand\ \}.$$

Similar to d_1, its 3D block is a flat plate due to its zero volume.

- d_6: it reflects the orientation changes. Negative changes mean the object front plane tends to be turned away from the camera. d_6 is the only orientation PGSE for this action. its vector descriptor is:

$$f(d_6) = \{\ O\text{-},1,\ 1,\ 1,\ 0.7,\ Right\ hand\}.$$

Figure 47(c) shows the 3D block pattern of the throw action.

A gesture/action can be characterized by a combination of various PGSEs, which are a set of perceptual gesture descriptors in 4D spatiotemporal space. We emphasize that PGSE is not limited to the five features (X, Y, Z Shape and Orientation) and the upper limb parts (hands), and it can be generalized to any other dynamics involving multiple body parts with more complex human activities. It is worth noting that even the PGSEs group the dynamic properties to provide qualitative description about gesture salience, the detailed inter-frame data is still preserved for quantitative computation, e.g. the motion parameters for gesture control applications.

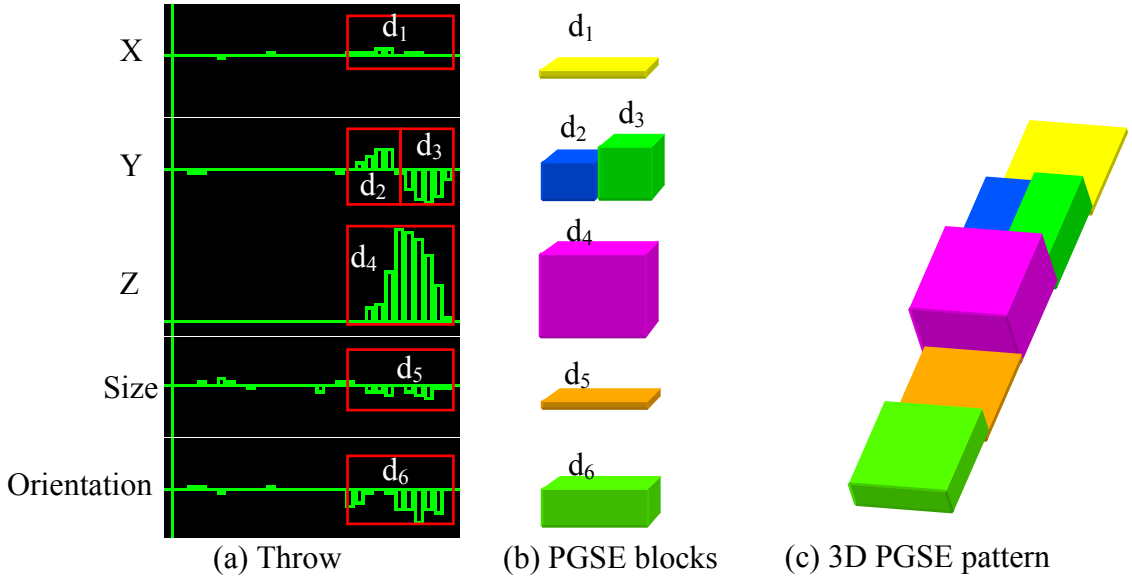
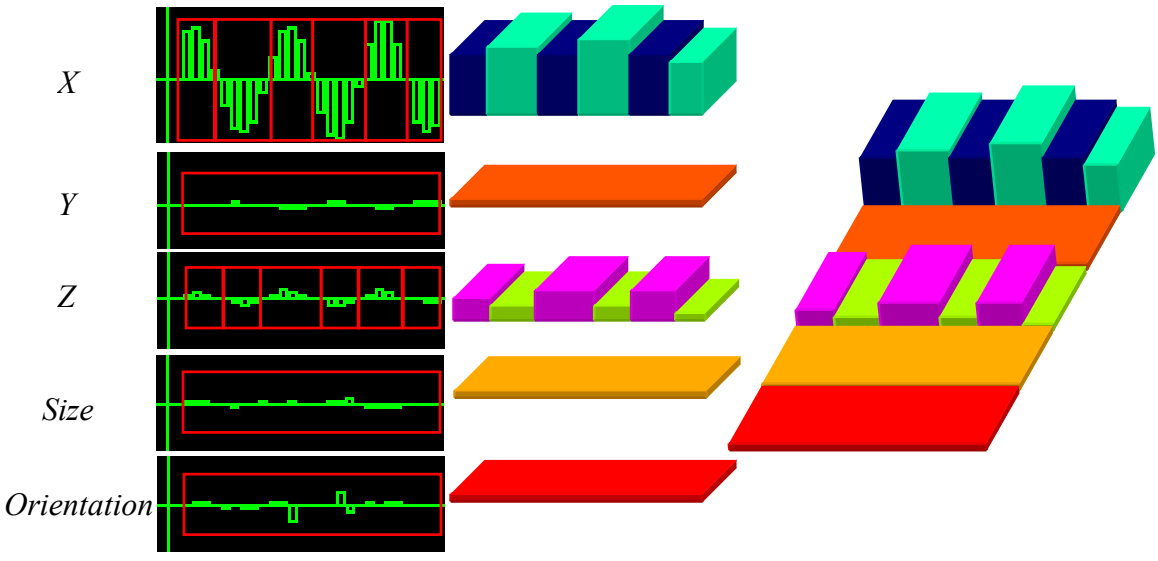


Figure 47 Five dynamic sequences for a throw action.
Each red box is a PGSE reflecting one dynamic property.

5.4 PGSE-BASED GESTURE/ACTION PATTERN

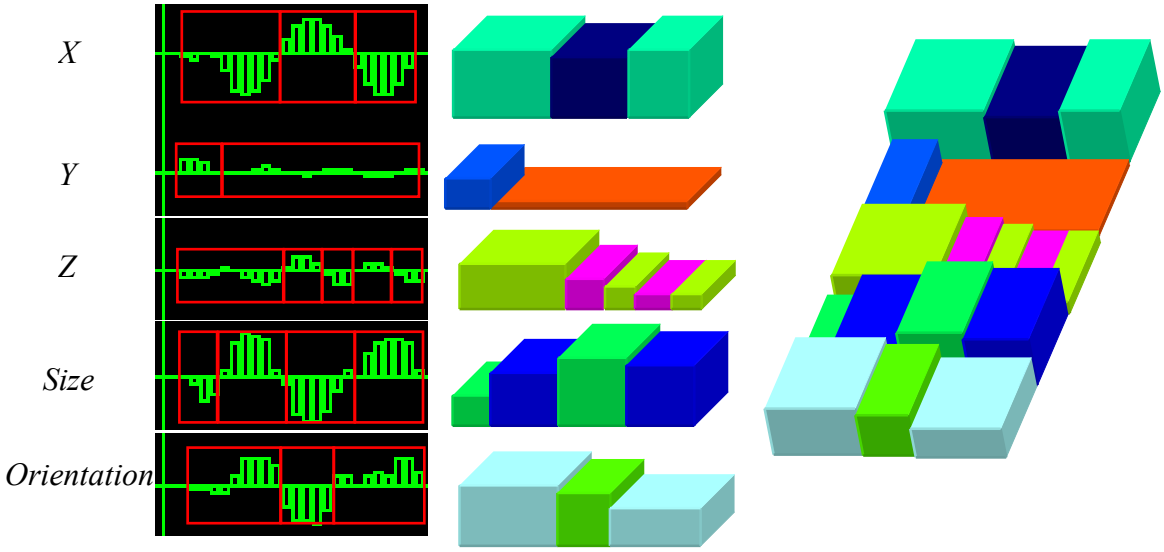
A gesture or action consists of a set of PGSEs, each of which represents one aspect of multiple channels of dynamic properties. PGSE's 3D block representative provides semantics for modeling. The combination of blocks with different shape and color patterns well reflect the motion visual salience, and acts as the salience map at the 4D spatiotemporal space for high-level interpretation. This salience organization encodes both visual salience and complex intrinsic relationships to provide selective visual evidence for recognition tasks. Meanwhile the underlying inter-frame data is still preserved for quantitative computation. We show two additional human actions, wave and flip palm, in Figure 48 and Figure 49. Gestures and actions involve multiple channels of dynamic properties, i.e. multiple types of PGSEs (XYZ, size, orientation etc.) from multiple body parts. Rather than independent, these multi-channel dynamics are tightly coupled. These temporal and spatial relationships of multiple channels are discriminative for patterns recognition, however, they may not be visually apparent, and are difficult to describe. From the 4D gesture salience map (PGSE blocks), the dynamic changes and their coherence of multiple channels (different body parts and properties) are well presented in a different view. For example, the Start time and Duration elements of



(a) Wave (b) Segmented PGSEs (c) 3D block pattern

Figure 48 Wave action with its 5 dynamic sequences and 3D PGSE block pattern.

each PGSE clearly state the internal temporal relations, such as overlap, interval, or simultaneous. By applying some computational models from other domains, the challenging recognition tasks can be accomplished effectively.



(a) Flip palm (b) Segmented PGSEs (c) 3D block pattern

Figure 49 Flip palm action with its 5 dynamic sequences and 3D PGSE block pattern.

CHAPTER 6 SALIENCE MAP-BASED GESTURE/ACTION REPRESENTATION

Based on the PGSE gesture descriptors, several ways, such as histogram, HMM, MRF, CRF, HCRF, can be used to describe the complex spatiotemporal patterns of human gestures/actions for high-level interpretation tasks.

6.1 SEVERAL MODELS

Here we conceptually introduce four PGSE-based methods: HMM, MRF, CRF and HCRF. More detailed descriptions about these methods are in Appendix A.

- **Hidden Markov Model (HMM)**

Hidden Markov Model is an effective approach to model spatial-temporal series related events. Many gesture recognition and natural language processing applications have adopted HMM as the model platform. HMM is based on a strong Markov property assumption that the conditional probability distribution of future states of the process depends only upon the present state. In our case, we can collapse the parallel multi-channel PGSE blocks into a sequential state model, each ordered block's state depends on its previous one only. HMM model is suitable for labeling sequential PGSEs, and makes sense for some human actions/gestures that only have PGSEs on one channel with a simple ordering structure, such as wave action containing several X PGSEs. A limitation of HMMs however, is that they cannot naturally handle the cases in which observed data elements overlap in arbitrary ways. The imposed Markov chain assumption on PGSEs from multiple parallel channels does not reflect the true temporal properties among PGSEs from complex gestures/actions.

- **Markov random Field (MRF)**

The temporal relationships among dynamic motion properties can be viewed as spatial

dependencies among PGSE blocks, which are usually not as simple as the sequential pattern. The spatial dependencies among PGSE blocks are the intrinsic properties of a human action. Markov Random Field (MRF) describes a set of observed data as random variables in an undirected graphic model which is able to incorporate the contextual constraints in a principled manner. MRF models have been used extensively for various segmentation and labeling applications in computer vision, such as image restoration, image segmentation, texture synthesis. Using the MRF model to describe a set of PGSEs is a probabilistic way to reveal internal relationships of a gesture or action, which may play a role in gesture/action understanding. MRF is a generative model to estimate the posterior properties. To give descriptions of the human gestures from a MRF framework, two distributions are required: the target label relationships and the target-observation likelihood. Unlike the HMM model, MRF tries to model the PGSEs from multiple property channels occurring in parallel and overlapping patterns. The local and pair-wise Markov properties of the MRF capture the relationship among the target labels of PGSEs as the model prior of the human activities, and the compatibility between a target label and a PGSE is modeled as the likelihood probability. An energy function needs to be defined to characterize the structure and compatibility properties. Since Markov properties of a non-regular arbitrary graph model are difficult to establish, usually in a MRF model, the graph nodes are factorized by a set of cliques. In our case, we can have cliques that only include the PGSEs from a same period (with the same start time zone). The intuition of this clique definition is that the motion dynamics within a time period occur almost simultaneously, and have a high degree of correlation.

- **Conditional Random Filed (CRF)**

Both HMM and MRF are the generative model. One limitation of the generative model is that to make the model computationally tractable they have to assume the independence of the observed data. HMM only takes the previous linked variable and one observation into the consideration. Traditional MRF model puts local pair-wise variable dependencies into potential functions to enrich the dependency description, but still only considers the observation data at one site without others. Therefore, the imposed restrictive observation

independence assumptions make these models limited in global feature modeling. In our case, the relationships among PGSEs over different time periods are not well modeled if using HMM. And the assumed prior distribution (e.g. Gibbs distribution) will be biased and cannot reflect the true PGSE distribution if using MRF. However, as a variant of a Markov random field, the conditional random field (CRF) models the variables conditioned upon a set of global observations, i.e. the label assignment decision depends not only on the current observation, but also the surrounding data within a certain size of neighborhood. CRF is a discriminative method which deals with the modeling of conditional distribution directly from the observation and target gestures/actions without providing the prior distribution. Similar to the MRF model, the CRF graph could consist of several cliques, each of which contains a set of edges with corresponding potential functions and parameters describing the relationship among PGSEs. The collection of the potential functions and parameters are formed into the CRF model for a PGSE-based human action/gesture pattern. Any human activities can be represented by this statistical graph model.

- **Hidden Conditional random field (HCRF)**

Internal temporal and spatial relationship among multiple dynamic properties is more complicated than what some simple models can handle. Rather than a sequential chain or a Markov network, intrinsic structures could be more complicated, i.e. a mixture of chain, overlapped and stride modes with uneven steps, and hard to identify. Therefore, many approaches model gestures/actions by either ignoring its intrinsic structures or using oversimplified assumptions so as to limit the performance of corresponding representations. Similarly, the PGSE-based gesture/action representation that is modeled by HMM or MRF will suffer the same difficulty. Even though CRF could model an arbitrary structure, it requires the knowledge about the connectivity structures of the random variables for a gesture/action class, which may not be feasible for various human activities since there exists wide variability of a same human motion and the PGSE patterns of one gesture/action class vary. To have more flexibility for structure modeling, we can assume that the internal temporal relationships are modeled by some hidden states that govern the pattern discrimination implicitly. Some works [89] [136] showed that

incorporating latent structures into the system can improve the recognition and classification performance. Hidden Conditional Random Field as an extended CRF is able to associate the hidden state layer to model the unknown internal substructure between the label and the observed data. In our case, the labels of HCRF are a fixed set of human actions/gestures characterized by the observed PGSE block patterns, and the relationship between labels and observations are modeled by a set of hidden states which reflect the temporal correlation among PGSEs. HCRF is able to reveal the sophisticated internal relationship among PGSEs.

6.2 PGSE-BASED HISTOGRAM REPRESENTATION

As we can see in Figure 47, Figure 48 and Figure 49, multiple channels of PGSE blocks of the single hand gestures are tightly coupled because the gestures and actions we currently focus on contain continuous atomic motions with less complexity, and their internal temporal ordering relationships among dynamic components are highly coherent, i.e. as simple as concurrency. Therefore, using computationally expensive probabilistic models like CRF, HCRF and MRF for them is over-kill. Instead, a histogram is a non-parameter representation of data distribution, and is suitable for these simple gestures and actions. In our case, there are total 15 types of PGSEs. Aggregate the volumes and time durations according to the PGSE types, a histogram representation for a human gesture/action is built. Figure 50 shows 2 histograms for throw and wave actions respectively. Each 3D bar represents one of 15 PGSE type, and the height and length of

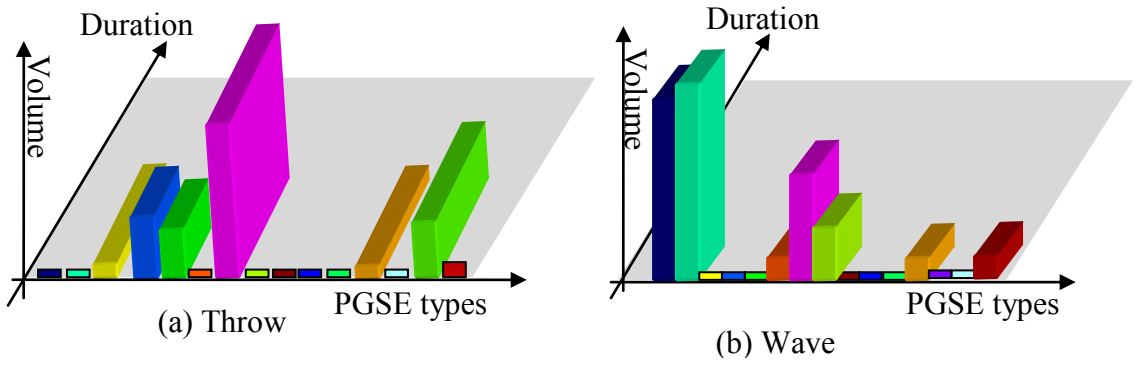


Figure 50 Histogram representation of PGSE blocks.

the 3D histogram bar represent the overall change volumes and durations of a PGSE within the motion period. However, histograms have the limitation of missing temporal relation among internal gesture components after aggregation. Temporal relations among PGSEs are gesture intrinsic properties which are not visually apparent, but are discriminative for classification. Therefore we add a metric, *sparseness*, as a weak temporal parameter for each aggregated PGSE. Each sparseness is proportional to the intervals among PGSEs with a same type before aggregation, and its value is normalized between [0, 1]. The bigger the value, the sparser it is. For example in Figure 51, 3 PGSE blocks with 2 types have the durations 9, 8, 8 respectively. The *sparseness* for the pink PGSE after aggregation will be 8/25 where 8 is the interval between 2 pink blocks, 25 is the overall duration; and the sparseness for the green one is 0.

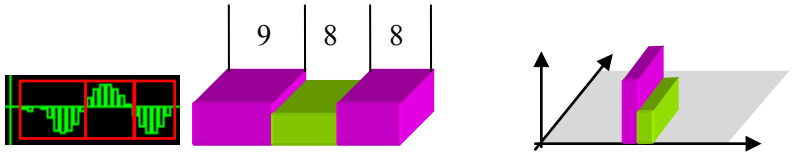
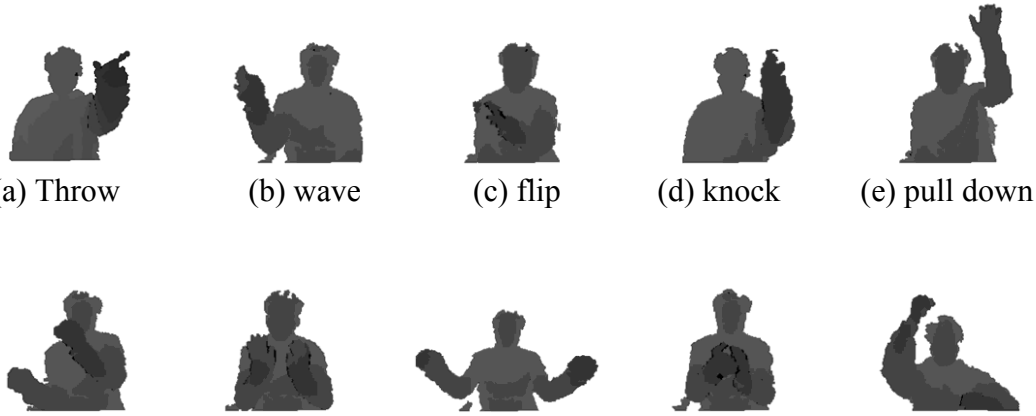


Figure 51 Sparseness for the PGSE histogram.

In sum, each aggregated PGSE can have 3 elements: volume, duration and sparseness, and the size of the histogram vector for each body part is 45 (3×15). Since gestures/actions are mainly conducted by the 2 upper limbs, a gesture can be represented by a histogram with 90 elements. This PGSE histogram with temporal element representation (PGSE_WT) is in the same spirit of the BoW method, but with smaller vocabulary size, qualitative descriptions and intrinsic properties. Without the temporal element sparseness, the PGSE histogram (PGSE_NT) with 2 elements (volume and duration) would be a 60-element vector for the 2 upper limbs (30 for each). To examine the effectiveness of gesture intrinsic properties in recognition tasks, we will compare the performance of PGSE_WT and PGSE_NT in experiments.

6.3 EXPERIMENTS AND EVALUATION

Since 3D camera-based video analysis is relatively new in the computer vision community, currently there is no 3D benchmark gesture dataset available. We instead created our own 3D gesture dataset for training and evaluation. We recorded 3D videos containing 10 types of human gestures/actions performed several times by 5 subjects individually. The environmental settings from video recording are as following: a single user in front of a fixed Kinect camera, interacts with a computer by performing gestures/actions including: throw, wave, flip palm, knock and pull-down for one hand, push, drive, expand, clap, climb rope with 2 hands (see Figure 52). We use depth images from a Kinect camera for our tests. Each depth image is a 640×480 array of raw depth values. 3D data accuracy is affected by the lighting condition, motion speed and distance. To produce a better quality dataset, some constraints and pre-processes were imposed: (i) setting the distance between a subject and the camera 1.5-2m in that Kinect can provide the most accurate depth data; (ii) scaling into a grayscale depth image; (iii) filtering out the background by a depth threshold; (iv) smoothing by a 3×3 median filter and (v) downsizing the image to 320×240. Currently the dataset contains 506 sequences with one gesture/action for each. Roughly 50 samples per gesture were collected.



(a) Throw (b) wave (c) flip (d) knock (e) pull down

(f) Drive (g) push (h) expend (i) clap (j) climb

Figure 52 Snapshots for 10 gestures/actions in our 3D video dataset.

There is no existing performance baseline for gesture/action recognition reported on a publically available 3D datasets. To comprehensively evaluate the performance of the

proposed method, we instead compare our approach with state-of-the-art local spatio-temproal feature descriptor methods by using a standard BoW SVM approach against our 3D gesture dataset. We take 3 baselines from the combinations of 2 feature detectors (Harris3D, Dense) and 2 local feature descriptors (HOG/HOF, HOG3D):

- ***Harris3D detector +HOG/HOF descriptor***: The STIPs from the Harris3D detector [68] are the XYT salient locations with local maxima. HOG/HOF descriptor [77] is similar in spirit to the SIFT descriptor. For each STIP, a local support region is built and further divided into grid cells containing 4-bin HOG and 5-bin HOF for each. It gained good performance on the KTH dataset.

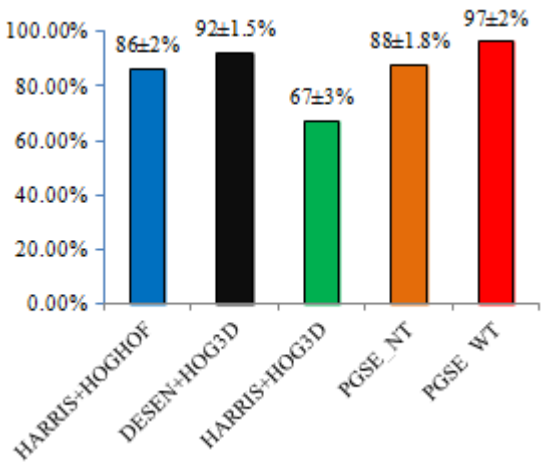
- ***Dense sampling + HOG3D descriptor***: Dense sampling [72] divides a video into sub-volumes (samples) with multiple spatial and temporal scales. HOG3D descriptor [121] is the histograms of 3D gradients. A local 3D patch around each sample is divided into cells with 10 gradient orientation bins for each. This method gained best performance against the UCF sports dataset.

- ***Harris3D detector + HOG3D descriptor***: Overall, STIPs from the Harris3D detector outperform the dense sampling results in terms of the feature effectiveness on the KTH datasets. HOG3D is ranked in the 2nd place in terms of descriptor performance [141]. To see the performance of this combination on our dataset is of interest to us.

For these detectors/descriptors, we used available tools from the authors' websites[1, 2] and applied the recommended parameters which are reportedly optimal for gesture videos. The outputs of 3 above methods are the list of long feature vectors. The BoW method quantizes features into visual words which were trained by the k-means clustering. Each descriptor has its own vocabulary with the size V=200 which empirically gives good results on our dataset. For each gesture video, its feature vectors are assigned to the closest visual words. The resulting histogram of visual word occurrences is the gesture representation.

[1] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html#stip
[2] http://lear.inrialpes.fr/people/klaeser/software_3d_video_descriptor

Support Vector Machines (SVMs) [142] are the large margin classifiers which have gained popularity for visual pattern recognition. Individual SVM classifiers were trained for 5 representations: i) 60-bin PGSE histogram without the temporal relation parameter (PGSE_NT), ii) 90-bin PGSE histogram with the temporal relation parameter (PGSE_WT), iii) Harris3D+HOG/HOF BoW histogram, iv) Dense sampling+ HOG3D BoW histogram and v) Harris3D+HOG3D BoW histogram. Each classifier was trained with a χ^2-RBF-kernel using Leave-One-Out (LOO) cross validation. Since it is the multi-class classification case, one-against-rest approach is applied to select the gesture class with the highest score as the recognized one. The performances of individual SVM classifier models were evaluated by measuring the average accuracy over all classes. Figure 53(a) shows the average classification rates for 5 representation methods. Each average rate value has its range covering 10 different gesture classes in the dataset. The PGSE_WT gives the best performance on our dataset. Dense+HOG3D outperforms PGSE_NT due to its ability to capture useful context information e.g. head, torso etc. Context may be helpful for human gesture/action recognition. Figure 53(b) shows the confusion matrix for the PGSE_WT representation. As we can see, there is a clear separation between single hand and 2-hand gestures/actions. The most confusion occurs between the wave and flip gestures because both of them have similar motions and the palm orientation was not able to reflect their differences.



	Throw	Wave	Flip	Knock	Pull down	Push	Drive	Expand	Clap	Climb
Throw	0.98	0	0	0.02	0	0	0	0	0	0
Wave	0	0.91	0.09	0	0	0	0	0	0	0
Flip	0	0.07	0.93	0	0	0	0	0	0	0
Knock	0.01	0	0	0.99	0	0	0	0	0	0
Pull down	0.05	0	0	0	0.95	0	0	0	0	0
Push	0	0	0	0	0	0.97	0	0.03	0	0
Drive	0	0	0	0	0	0	1.00	0	0	0
Expand	0	0	0	0	0	0.02	0	0.98	0	0
Clap	0	0	0	0	0	0	0	0	1.00	0
Climb	0	0	0	0	0	0.03	0	0	0	0.97

(a) Recognition rates for 5 representations (b) Confusion matrix for PGSE_WT

Figure 53 Comparison results on our 3D gesture/action dataset.

6.4 CONCLUSION

In this chapter, we presented a novel generic gesture representation method. The complex spatiotemporal extrinsic gesture properties can be expressed by a set of qualitative descriptors Perceptual Gesture Salience Entities (PGSEs). A histogram of PGSEs provides the statistics about gesture's spatial and appearance properties in the same spirit of the Bag-of-Word-based representation. Meanwhile the intrinsic context relations among PGSEs are modeled as the correlation parameters. Both extrinsic and intrinsic properties encoded within the PGSE-based gesture representation contribute to qualitative reasoning for gesture recognition and classification tasks with robustness and efficiencies. Our approach has been tested on a 3D gesture dataset; the experimental results show it outperforms other state-of-the-art methods. Even though current model is only tested on atomic gestures and actions, the method is valid for generic action modeling, and should perform well for complex human activities in general.

CHAPTER 7 PROOF-OF-CONCEPT APPLICATION

Using the PGSE-based gesture representation method, a Dart game has been developed as a proof-of-concept application for system demonstration. The throw gesture is recognized qualitatively; its dynamic properties are measured quantitatively and converted into game control parameters. The gesture game environment is set up as following. A player sits/stands in front of the screen and the camera. The imaginary target board is behind the screen. The camera is either at a high or a low position, and is centered at the horizontal direction of the imaginary target board (see Figure 54).

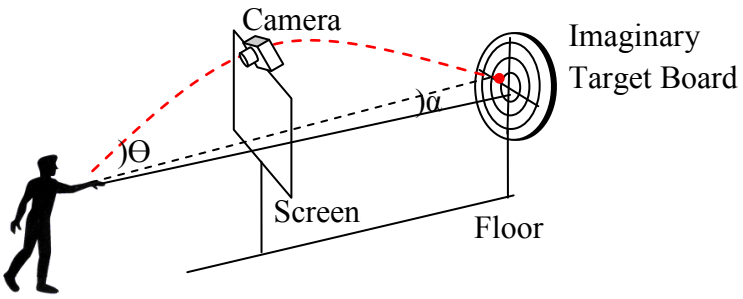


Figure 54 Dart game settings.

7.1 CAMERA CALIBRATION

The camera position is crucial in this gesture control game. According to the game environmental setting, a player faces the screen of the game terminal, and interacts with the game content on the screen. The possible camera position is usually above or under the screen so that performed motions are not directly towards the camera. However, all the spatial coordinate data (X, Y, Z) of the target objects from the 3D camera is derived from radial distances with the origin at the camera center, and they are biased and cannot reflect players' real motion parameters accurately. Therefore, the calculation relying on the (X,Y,Z) from the camera need to be adjusted according to the camera position and angle. In order to accommodate possible camera positions, a calibration function is needed for coordinate transformation. Figure 55 shows 2 possible camera positions, under the screen and above the screen.

If the desired camera position is under the screen, the camera lens needs to be pointed up with a certain angle β (0<=β<=90) to capture the player's motion, and centered at the horizontal direction of the screen. For the dart game, the appropriate origin of the 3D coordinate system is the imaginary target board center, rather than the camera center. The imaginary distance between the virtual target board and the camera in the depth direction is T_z. The imaginary distance from the camera to the center of the virtual target board in Y direction is T_y. The camera is centered at the target board in X direction, so T_x=0. Since the origin of the coordinate system is at the center of the imaginary dart board, the camera position is at (0,-T_y, T_z). The coordinate data of any object from the camera is (X_m, Y_m, Z_m) whose origin is the camera center. The new coordinate (X, Y, Z) with its origin at the center of the target board can be calculated by using a transformation matrix Eq.7.1.

$$\begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & \cos\beta & \sin\beta & Ty \\ 0 & -\sin\beta & \cos\beta & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{7.1}$$

Once T_x, T_y, T_z and β are known, the (X_m, Y_m, Z_m) measured by camera can be transformed into the true game coordinate data (X, Y, Z).

Similarly, if the camera is above the screen, and points down to the floor with an angle β (0<=β<=90), the (X,Y,Z) can be obtained by the transformation matrix Eq. 7.2:

$$\begin{bmatrix} 1 & 0 & 0 & Tx \\ 0 & \cos\beta & -\sin\beta & Ty \\ 0 & \sin\beta & \cos\beta & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} Xm \\ Ym \\ Zm \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \tag{7.2}$$

Based on the transformation formulas summarized above, the camera calibration function takes the following parameters to produce accurate gesture measurements:

- Position of the camera (top, bottom);
- Distance from the floor to the imaginary target board center;
- Perpendicular distance between the camera and the imaginary target board;

- Angle of the camera pointing up or down.

After calibration, the origin of the output spatial coordinate data is at the target board center, which assures the accuracy for the parameter estimation.

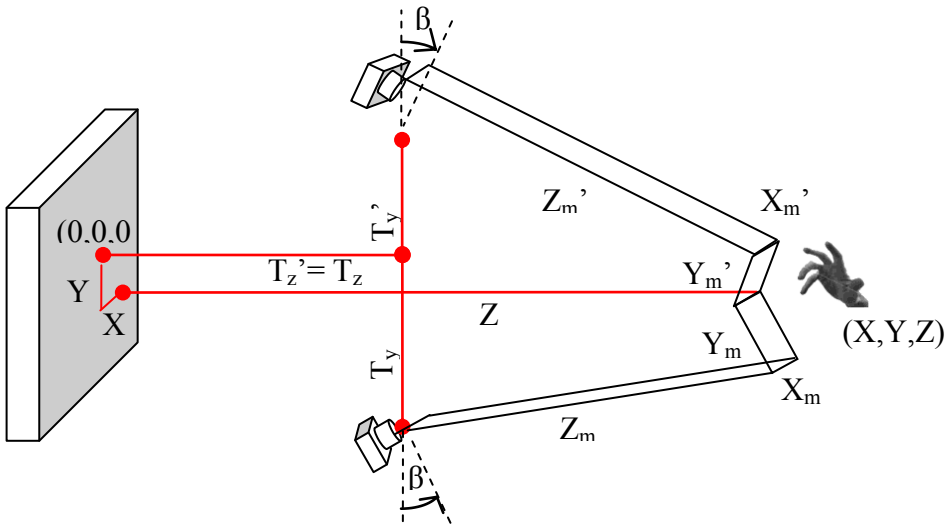


Figure 55 Camera at the high or low position.

7.2 Control Parameter Estimation

Having had a throw gesture recognized, we need to estimate parameters quantitatively for game controlling. The ultimate parameter of the Dart game is the final position of a thrown virtual dart. To get its accurate final destination, the imaginary trajectory of a flying dart needs to be estimated. The trajectory is calculated based on the physical law of gravity with the following parameters:

- Θ: the upward angle of the thrown dart;
- α: the forward angle of the thrown dart;
- V_0: the initial velocity of the thrown dart.

Eq. 7.3 and Eq. 7.4 are the formulas for the trajectory calculation:

$$x = tv_0 \cos\theta \tag{7.3}$$

$$h = tv_0 \sin\theta - \frac{1}{2}gt^2 \tag{7.4}$$

Here g is the acceleration of gravity, 9.8m/s^2; t is the time of flight; v_0 is the initial velocity; Θ is the launch angle. When a dart is flying in air, its motion under the influence of gravity is determined completely by the acceleration of gravity, its initial speed, and the launch angle. The air friction is directly proportional to the initial velocity. For a regular dart throw, the average speed of the dart is 60km/h, which is low, and the size of the dart is small, so the air friction is negligible and not considered into the trajectory calculation. Figure 56 shows trajectories with the same initial speed but different launching angles.

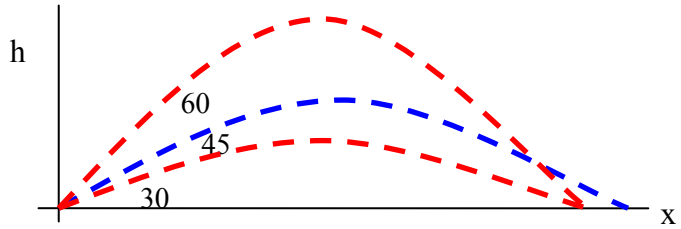


Figure 56 Flight trajectory of a dart without considering the air resistance.

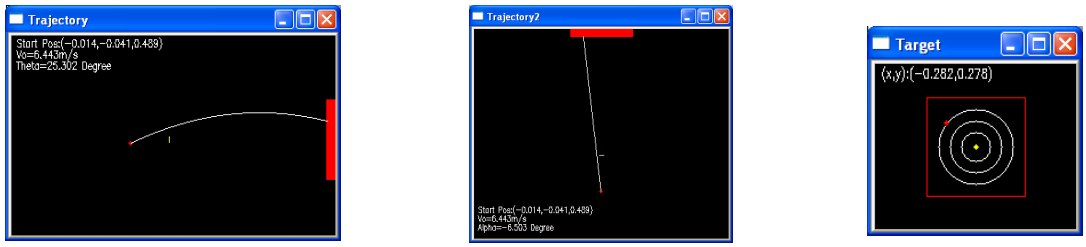
Those parameters are the instant measures derived at the moment when a virtual dart is thrown out, and can only be estimated accurately at the frame level when the dart release moment is identified. PGSE-based histogram is aggregated from the underlying frames which are still preserved for detailed parameter calculation. The assumption for the Dart game is that at the dart releasing moment, the hand speed on Z direction reaches the highest, the palm faces down, and the hand becomes open. We combine the three factors to determine the dart release moment:

$$\arg\max_{i=1}^{k}[\alpha \cdot Norm(Speed_i) + \beta \cdot Norm(Orientation_i) + \gamma \cdot Norm(Size_i)], \qquad (7.5)$$

where k is the total number of frames; the approaching speed, palm orientation and size changes are normalized into a range of (0,1). α, β and γ are the weights of three factors. The release moment is at the frame while the value of Eq. 7.5 reaches the maximum.

The game system captures the motion, calculates the trajectory and the destination of the virtual dart, and presents the result on the screen. Figure 57a shows a side view of the trajectory of a virtual dart for one throw. The red point is the dart position when released from the hand, the short yellow vertical line represents the camera position, and the red

thick bar on the right side represents the target board. The white curve is the virtual dart trajectory. Figure 57b shows a bird's-eye view (from ceiling to floor) of the same dart trajectory shown in Figure 57a where the red point is the initial dart position and the target board (red thick bar) is on the top. In Figure 57c, the red square is the boundary of the target board. The yellow point is the bullseye. If the dart hits the board, a red dot is shown within the red square.



(a) side view (b) bird's-eye view (c) front view

Figure 57 The side, bird's-eye and front views of a trajectory of the dart and the target board.

7.3 User Validation

To evaluate this proof-of-concept application, we conducted three-round tests for the dart game. There were 10 participants in the first round who were required to make throw gestures without using real darts. The system captured motions, calculated the trajectories and the destinations of the virtual darts, and showed results on the screen. According to the test plan, after each valid throw was performed, and had been recognized by the system, the participant was asked for the satisfaction rate from 1-10 on the result. This rate is subjective, but reflects the player's feeling about the system accuracy. Within the 671 recorded gestures, there were 334 throws and 337 others. Among the throws, the system correctly captured 287, and missed 47. The average satisfaction value of the captured valid throws is 6.57. Among the other gestures, the system mistakenly captured 92 as throws and ignored 245. The 75.7% precision, 85.9% recall and 79.3% accuracy are the fair results (Table 5). The major reason for failures is the insufficient training dataset does not cover all the gesture styles.

The objective of the second round is to further test the throw gesture recognition rate after the throw gesture model has been refined. 20 people participated in this round. They are a group of diverse people in terms of gender, age, body size, and 90% of them were new to this system. The average recognition rate is 90.1%. During the tests, they were not given any instruction when got failed. The players just kept throwing until they found right adjustment, which decreases the overall recognition rate. The majority of the failures were due to slow speed. In order to hit the board centre, people played cautiously, and used less force than usual to make throws. Look at the results of three sets, the recognition rates are increasing from 87% to 93% (Table 6). It follows the natural learning curve for people doing things that they have never experienced before.

Table 5 Throw gesture accuracy for the real-time application.

Gesture type	Total	Captured	Missed	Precision	Recall	Accuracy
Throw	334	287	47	75.7%	85.9%	79.3%
Others	337	92	245			

Table 6 The 2nd round tests on a real-time gesture application.

Set 1		Set 2		Set 3		Overall	
Captured	318	Captured	332	Captured	341	Captured	991
Missed	48	Missed	34	Missed	27	Missed	109
Recall	87%	Recall	91%	Recall	93%	Recall	90%

The original raw Dart Game has been integrated into a platform with a complete game theme. The goal of the third round test mainly examines the satisfaction of the user experience about whole system in terms of the accuracy of the motion intention. The overall recognition rate has reached over 96%, and the satisfaction rate is over 8.

CHAPTER 8 CONCLUSION AND FUTURE DIRECTIONS

In this dissertation, we proposed a novel 3D gesture/action recognition framework based on the hierarchical visual attention and perceptual organization models. This is one step closer toward a high-level human activity understanding solution. Much remains to be done, both in improving and extending our current framework and in developing a broader understanding that involves human interactions, group activities, and semantic knowledge from the scene context. In this chapter, we discuss the achievements that have been made, and future directions for performance improvement and further efforts for full tier human activity understanding.

8.1 CONCLUSION

The contributions of this dissertation are three-fold. First, visual attention and perceptual organization theories and hypotheses are modeled into the 3D human gesture/action recognition framework by applying the salience map principle. Within this framework, visual features are selectively processed, grouped and integrated into hierarchical salience maps at the feature, object and 4D spatiotemporal levels step-by-step.

Secondly, a set of gesture/action salient feature descriptors, Perceptual Gesture Salience Entities (PGSEs), are defined from the extrinsic motion properties, describing the human gesture/action qualitatively. By using a cuboid representative for each PGSE, any human gesture or action can be coded as a set of colorful cuboids with various shapes in a certain pattern. Thus the challenging gesture/action understanding task is converted into a much simpler cuboid pattern search/matching problem.

Thirdly, besides the extrinsic gesture properties, the intrinsic properties that are not visually apparent and hard to be modeled, characterize the complexity of human gestures/actions. The contextual relations among PGSEs naturally encode the intrinsic gesture properties, and can be easily exploited by various probabilistic methods for modeling sophisticated human activities.

This framework significantly differs from others since the visual features and perception knowledge is modeled in a systematic, coherent and biologically plausible manner. The PGSE-based gesture/action representation is able to support qualitative reasoning for gesture recognition with robustness and efficiencies, in that it reduces the search space and retains the needed discriminative power. Meanwhile, low-level information is still preserved in the PGSE descriptors, and thus, gesture controlling parameters can be derived quantitatively. Our approach has been tested on 3D gesture datasets and a real-time gesture application, Dart game. The promising experimental results show our approach outperforms others and has great potential for different applications.

8.2 FUTURE DIRECTIONS

We have shown the potential of our framework for human gesture/action recognition. However, it is only the entry point of solving full tier human activity interpretation under arbitrary conditions. Here, we give several ideas for future work that extend our current framework to provide complete interpretations of human activities.

- *Human interaction recognition*

Human interactions are human activities that involve two or more persons, for example, two persons fighting, shaking hands etc. To do so, the human segmentation is required first, and thus body part segmentation and classification are applied on individual humans. A straightforward way to build human interaction representation is to make use of an existing PGSE-based method, i.e. by simply concatenating each person's PGSEs together. However, it will suffer the difficulties in the recognition stage because of the high dimensional feature space. Providing a more efficient and effective human interaction representation method is worthy of a closer look in future work. More generically, human interaction also includes human-object interactions, for example, a person picks up a phone and talks, or, a person hands over a ball to another. In order to model and recognize these type of activities, besides the human-human activity recognition, we need

to incorporate the object recognition model into the framework. Ideally, a generic human interaction representation template is able to model any object with its arbitrary motion patterns and human-human interactions.

- *Group activity recognition*

Group activities are the activities that involve one or more conceptual human groups. In order to recognize group activities, the analysis of both individual activities and overall structures are necessary. Some applications require more on overall motion of entire group members, such as moving direction of parading or marching, which are characterized by individual activities. Others focus more on individuals, e.g. finding uncommon behaviors in a crowed scene where the overall patterns act as the benchmarks that need to be identified first. The current framework can be extended by adding an additional group level salience map, which models both overall structures and individual abnormality of group activities as the salience entities. By taking the entire scene as one object, the group representation can be established by using the PGSE descriptors.

- *Scene context*

Scene contexts provide semantics for human activity interpretation. Humans cannot correctly perceive the world without context information, such as background, environment and relevant prior knowledge. Some works have proven that recognition algorithms equipped with context information are more robust. In the current framework, the background is filtered, and no other un-related objects are in the scene. In the future work, this constraint needs to be relaxed, and the scene context will be modeled to enrich the representations.

BIBLIOGRAPHY

[1] Isa N. Engleberg, Working in Groups: Communication Principles and Strategies, *My Communication Kit Series*, 2006.

[2] R. Blake and M. Shiffrar, Perception of Human Motion, *Annual Review of Psychology*, Vol. 58, January 2007.

[3] P. Cavanagh, AT. Labianca, IM. Thornton, Attention-based visual routines: sprites, *Cognition* 80:47–60, 2001

[4] IM Thornton, RA Rensink, M. Shiffrar, Active versus passive processing of biological motion, *Perception* 31:837–53, 2002.

[5] R. Kimchi, Perceptual organization and visual attention, *Progress in Brain Research* 176:15-33, 2009.

[6] V. Navalpakkam and L. Itti, Modeling the influence of task on attention, *Vision Research*, 45(2): 205-231, Jan 2005.

[7] L. Itti, C. Koch, E. Niebur, A model of saliency-based visual attention for rapid scene analysis, *IEEE Trans. Pattern Analysis and Machine Intelligence*. 20 (11): 1254–1259, 1998.

[8] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y.H. Lai, N. Davis, F. Nuflo, Modeling visual-attention via selective tuning, *Artificial Intelligence*, 78:507–545, 1995.

[9] G. Deco, B. Schurmann, A hierarchical neural system with attentional top-down enhancement of the spatial resolution for object recognition, *Vision Research*. 40(20): 2845–2859, 2000.

[10] J. Duncan, Integrated mechanisms of selective attention, *Current Opinion in Biology*, 7: 255-261, 1997.

[11] Y. Sun, R. Fisher, Object-based visual attention for computer vision, *Artificial Intelligence,* 20 (11): 77–123, 2003.

[12] A. Treisman and G. Gelade, A feature-integration theory of attention, *Cognitive Psychology*, 12(1): 97-136. 1980.

[13] A. Treisman and H. Schmidt, Illusory conjunctions in the perception of objects, Cognitive Psychology, 14:107-141. 1982.

[14] D. B. Walther and C. Koch, Attention in Hierarchical Models of Object Recognition, *Progress in Brain Research*, 165:57-78, 2007.

[15] M. Atsumi, A Probabilistic Model of Visual Attention and Perceptual Organization for Constructive Object Recognition, In Proc. of *the 5th International Symposium on Advances in Visual Computing,* Part 2:778-787, 2009.

[16] M. Ajallooeian, et al., Fast Hand Gesture Recognition based on Saliency Maps: An Application to Interactive Robotic Marionette Playing, *IEEE Robot and Human Interactive Communication*, Osaka, Japan. 2009.

[17] A. Borji, D. N. Sihite, L. Itti, Computational Modeling of Top-down Visual Attention in Interactive Environments, In: *Proc. British Machine Vision Conference* (BMVC 2011), 85:1-12, Sep 2011.

[18] A. Oikonomopoulos, I. Patras, and M. Pantic, Spatiotemporal salient points for visual recognition of human actions. *IEEE Trans. on Systems, Man, and Cybernetics*, *Part B: Cybernetics*, 36(3):710–719, 2006.

[19] C. L. Colby, and M. E. Goldberg, Space and attention in parietal cortex. *Annual Review of Neuroscience*, 22: 319–349, 1999.

[20] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg, The representation of visual salience in monkey parietal cortex, *Nature*, 391: 481–484, 1998.

[21] A. A. Kustov, and D. L. Robinson, Shared neural control of attentional shifts and eye movements, *Nature*, 384: 74–77, 1996.

[22] K. G. Thompson, and J. D. Schall, Antecedents and correlates of visual detection and awareness in macaque prefrontal cortex, *Vision Research*, 40(10–12): 1523-1538, 2000.

[23] D. Marr and T. Poggio, Cooperative computation of stereo disparity, *Science*, 194: 283–287, 1976.

[24] R. Szeliski. Computer Vision: Algorithms and Applications, *Springer*, *New York*, 2010.

[25] A. García, Z. Zalevsky, P. García-Martínez, C. Ferreira and Y. Beiderman, Three-dimensional mapping and range measurement by means of projected speckle patterns, *Applied Optics,* 47(16): 3032-3040, 2008.

[26] http://en.wikipedia.org/wiki/Wii [accessed 10 May 2012]

[27] http://en.wikipedia.org/wiki/PlayStation_Move [accessed 10 May 2012]

[28] A recent speech by the Microsoft's CEO, Steve Ballmer, on Xbox LIVE at CES 2011's Show Case Event. (http://www.youtube.com/watch?v=0dzbvRvMNA0&feature=youtu.be)

[29] Y. Zhai and M. Shah, Visual attention detection in video sequences using spatiotemporal cues, In *ACM Multimedia*, pages 815–824, 2006.

[30] R. Achanta and S. Susstrunk, Saliency detection using maximum symmetric surround, In *Proc. of Int'l Conf. on Image Processing* (ICIP), 2010.

[31] X. Hou and L. Zhang, Saliency detection: a spectral residual approach, In *Proc. of Conf*. on *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[32] C. Guo, Q. Ma, and L. Zhang, Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform, In *Proc. of Conf*. on *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[33] C. Koch, & S. Ullman, Shifts in selective visual attention: towards the underlying neural circuitry, *Human Neurobiology*, 4(4), 219–227, 1985.

[34] J. Harel, C. Koch, and P. Perona, Graph-based visual saliency, In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 545–552, 2006.

[35] D. Gao, V. Mahadevan, and N. Vasconcelos, The discriminant center-surround hypothesis for bottom-up saliency, In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2007.

[36] Y.F. Ma and H.J. Zhang, Contrast-based image attention analysis by using fuzzy growing, In *ACM Multimedia*, pages 374–381, 2003.

[37] S. Goferman, L. Zelnik-Manor, and A. Tal, Context-aware saliency detection, In *Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[38] B. Alexe, T. Deselaers, and V. Ferrari, What is an object? In Proc. *of Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[39] T. Liu et al., Learning to detect a salient object, *IEEE Trans.* on *Pattern Analysis and Machine Intelligence*, 33(2):353–367, 2011.

[40] C. Harris and M. Stephens, A combined corner and edge detector, In *Proc. of the Fourth Alvey Vision Conference*, pages 147–151, 1988.

[41] T. Tuytelaars and L. V. Gool, Matching widely separated views based on affine invariant regions, *International Journal of Computer Vision*, 59(1):61–85, 2004.

[42] J. Matas, O. Chum, M. Urban, and T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, *Image and Vision Computing*, 22(10):761–767, 2004.

[43] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, 60(2): 91–110, Nov 2004.

[44] H. Bay, et al., SURF: Speeded Up Robust Features, *Computer Vision and Image Understanding (CVIU),* 110(3): 346-359, 2008.

[45] B. Steder, G. Grisetti, M. Van Loock, and W. Burgard, Robust online model-based object detection from range images, In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, Oct. 2009.

[46] M. Ruhnke, B. Steder, G. Grisetti, and W. Burgard, Unsupervised learning of 3d object models from partial views, In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[47] C. Plagemann, V. Ganapathi, D. Koller, S. Thrun, Realtime Identification and Localization of Body Parts from Depth Images, In Proc. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 3108-3113, 2010.

[48] J. Yamato, J. Ohya, and K. Ishii, Recognizing human action in time-sequential images using hidden Markov model, In *IEEE Conf.on Computer Vision and Pattern Recognition (CVPR)*, 379-385, 1992.

[49] T. Starner, and A. Pentland, Real-time American Sign Language recognition from video using hidden Markov models, *International Symposium on Computer Vision*, 265, 1995.

[50] A. F. Bobick and A. D. Wilson, A state-based approach to the representation and recognition of gesture, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19(12): 1325-1337, 1997.

[51] D. Gavrila, and L. Davis, Towards 3-D model-based tracking and recognition of human movement, In *International Workshop on Face and Gesture Recognition,* 272-277, 1995.

[52] S. Park, and J. K. Aggarwal, A hierarchical Bayesian network for event recognition of human actions and interactions, *Multimedia Systems* 10(2): 164-179, 2004.

[53] Y. Yacoob and M. Black, Parameterized modeling and recognition of activities, In *IEEE International Conference on Computer Vision (ICCV)*, 120-127, 1998.

[54] A. Efros, A. Berg, G. Mori and J. Malik, Recognizing action at a distance, In *IEEE International Conference on Computer Vision (ICCV)*, 2:726-733, 2003.

[55] R. Lublinerman, N. Ozay, D. Zarpalas and O. Camps, Activity recognition from silhouettes using linear systems and model (in)validation techniques, In *International Conf. on Pattern Recognition (ICPR)*, 347-350, 2006.

[56] A. Bobick and J. Davis, The recognition of human movement using temporal templates, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(3): 257-267. 2001.

[57] E. Shechtman and M. Irani, Space-time behavior based correlation, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1: 405-412, 2005.

[58] Y. Ke, R. Sukthankar, and M. Hebert, Spatio-temporal shape and flow correlation for action recognition, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[59] M. D. Rodriguez, J. Ahmed and M. Shah, Action MACH: A spatio-temporal maximum average correlation height filter for action recognition, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[60] L. W. Campbell and A. F. Bobick, Recognition of human body motion using phase space constraints, In *IEEE International Conference on Computer Vision (ICCV)*, 624-630, 1995.

[61] C. Rao and M. Shah, View-invariance in action recognition, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:316-322, 2001.

[62] Y. Sheikh, M. Sheikh and M. Shah, Exploring the space of a human action, In *IEEE International Conference on Computer Vision (ICCV),* 1:144-149, 2005.

[63] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[64] H. Wang, A. Kläser, C. Schmid and C. Liu, Action Recognition by Dense Trajectories, *In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[65] O. Chomat and J. Crowley, Probabilistic recognition of activity using local appearance, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Vol. 2, 1999.

[66] L. Zelnik-Manor and M. Irani, Event-based analysis of video, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[67] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, Actions as space-time shapes, In *IEEE Int.Conference on Computer Vision (ICCV)*, 1395-1402, 2005.

[68] I. Laptev and T. Lindeberg, Space-time interest points, In *IEEE International Conference on Computer Vision (ICCV)*, 2003

[69] A. Yilmaz and M. Shah, Actions sketch: a novel action representation, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1: 984-989, 2005.

[70] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, Behavior recognition via sparse spatio-temporal features, In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS),* 2005.

[71] P. Scovanner, S. Ali and M. Shah, A 3-dimensional sift descriptor and its application to action recognition, In *ACM International Conference on Multimedia (ACM MM),* 357-360, 2007.

[72] K. Rapantzikos, Y. Avrithis and S. Kollias, Dense saliency-based spatiotemporal feature points for action recognition, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[73] J. Liu, J. Luo and M. Shah, Recognizing realistic actions from videos in the wild, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2009.

[74] J. Liu, B. Kuipers, and S. Savarese, Recognizing Human Actions by Attributes, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2011.

[75] J. C. Niebles, C. Chen and L. Fei-Fei, Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification, In *European Conference on Computer Vision (ECCV),* 2010.

[76] S. Savarese, A. DelPozo, J. Niebles and L Fei-Fei, Spatial-temporal correlations for unsupervised action classification, *In IEEE Workshop on Motion and Video Computing (WMVC)*, 2008.

[77] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, Learning realistic human actions from movies, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2008.

[78] P. Hong, M. Turk, and T. Huang, Gesture modeling and recognition using finite state machines, in *Proc. 4th IEEE Int. Conf. Autom. Face Gesture Recogn., Grenoble,* France, pp. 410–415 Mar. 2000.

[79] M.S. Yang and N. Ahuja, Recognizing Hand Gesture Using Motion Trajectories, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1: 466–472. 1998.

[80] U. Gaur, Y. Zhu, B. Song, A "String of Feature Graphs" Model for Recognition of Complex Activities in Natural Videos, In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[81] W. Brendel, S. Todorovic, Learning Spatiotemporal Graphs of Human Activities, In *IEEE International Conference on Computer Vision (ICCV),* 2011.

[82] J. Yamato, J. Ohya and K. Ishii, Recognizing human action in time-sequential images using hidden Markov model, In Proc. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 1992.

[83] T. Starner,, J. Weaver and A. Pentland, Real-time American sign language recognition using desk and wearable computer based video, *IEEE Trans. Pattern Analysis Machine. Intelligence*, 20(12): 1371–1375. 1998.

[84] C. Vogler and D. Metaxas, A framework for recognizing the simultaneous aspects of American sign language, *Computer Vision and Image Understanding*, 81(3): 358–384, 2001.

[85] P. Turaga, R. Chellappa, V. Subrahmanian and O. Udrea, Machine Recognition of Human Activities: A Survey, *Circuits and Systems for Video Tech*. 18(11): 1473-1488, 2008.

[86] M. Elmezain, A. Al-Hamadi and B. Michaelis, Hand Trajectory-based Gesture Spotting and Recognition Using HMM, In *Proc. Int. Conf. on Image Processing*, 3577-3580, 2009.

[87] C. Sminchisescu, A. Kanaujia, Z. Li and D. Metaxas, Conditional models for contextual human motion recognition, In *IEEE International Conference on Computer Vision (ICCV)*, 2: 1808-1815. 2005.

[88] H. Yang, S. Sclaroff and S. Lee, Sign Language Spotting with a Threshold Model Based on Conditional Random Fields, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 31(7): 1264-1277, 2009.

[89] S. Wang, A. Quattoni, L. Morency, Demirdjian, D. and Darrell, T., Hidden Conditional Random Fields for Gesture Recognition, In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[90] L.P. Morency, A. Quattoni and T. Darrell, Latent-Dynamic Discriminative Models for Continuous Gesture Recognition, In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR),* 2007.

[91] M. Marszalek, I. Laptev and C. Schmid, Actions in context, In. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[92] D. Han, L. Bo and C. Sminchisescu, Selection and context for action recognition, In *IEEE Int.Conference on Computer Vision (ICCV),*, 2009.

[93] T. Lan, Y. Wang, G. Mori and S. Robinovitch, Retrieving Actions in Group Contexts, International Workshop on Sign Gesture Activity, In *European Conference on Computer Vision (ECCV)*, 2010.

[94] J. Shotton, J. Winn, C. Rother and A. Criminisi, TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation, In *European Conference on Computer Vision (ECCV)*, 2006.

[95] J. Shotton, A. Fitzgibbon, M. Cook, et al., Real-Time Human Pose Recognition in Parts from Single Depth Images, In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[96] A. Blake and K. Toyama, Probablisitic tracking in a metric space, In *IEEE Int.Conference on Computer Vision (ICCV)*, 2001.

[97] A Agarwal and B. Triggs, 3D Human Pose from Silhouettes by Relevance Vector Regression, In *Proc. of Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2:882-888, 2004.

[98] R. Urtasun, D.J. Fleet, A. Hertzmann and P. Fua, Priors for people tracking from small training sets, In Proc. of *IEEE International Conference on Computer Vision (ICCV)*, 1:403-410, 2005.

[99] K. Koffka, Principles of Gestalt psychology, *New York: Harcourt Brace Jovanovich*, 1935.

[100] M. Wertheimer, Gestalt theory. In: W.D. Ellis (Ed.), A source book of Gestalt psychology (pp. 1–16), *London: Routhedge and Kegan Paul.* (Originally published in German,1923,London.) 1923/1955.

[101] E. Rubin, Visuell Wahrgenommene Figuren, *Kobenhaven: Glydenalske boghandel*, 1921.

[102] S. Palmer and I. Rock, Rethinking perceptual organization: The role of uniform connectedness, *Psychonomic Bulletin and Review,* 1(1), 29–55, 1994.

[103] M.A. Peterson and B.S. Gibson, Object recognition contributions to figure-ground organization: Operations on outlines and subjective contours, *Perception & Psychophysics,* 56(5): 551–564, 1994.

[104] S.P. Vecera, E.K. Vogel and G.F. Woodman, Lower region: A new cue for figure-ground assignment, *Journal of Experimental Psychology: General,* 131(2):194–205, 2002.

[105] V. Klymenko and N. Weisstein, Spatial frequency differences can determine figure-ground organization, *Journal of Experimental Psychology: Human Perception and Performance,* 12: 324–330, 1986.

[106] J. Hulleman and G.W. Humphreys, A new cue to figure-ground coding: Top-bottom polarity, *Vision Research,* 44(24): 2779–2791, 2004.

[107] S.E. Palmer and T. Ghose, Extremal edges: A powerful cue to depth perception and figure-ground organization, *Psychological Science,* 19(1): 77–84, 2008.

[108] L. Itti, Visual salience, *Scholarpedia,* 2(9):3327, 2007.

[109] M.I. Posner, Orienting of attention, *Quarterly Journal of Experimental Psychology,* 32: 3–25, 1980.

[110] A.M. Treisman and G. Gelade, A feature-integration theory of attention, *Cognitive Psychology,* 12: 97–136, 1980.

[111] C.W. Eriksen and J.D. James St., Visual attention within and around the field of focal attention: a zoom lens model, *Perception & Psychophysics,* 40: 225–240, 1986.

[112] A.L. Rothenstein and J.K. Tsotsos, Attention links sensing to recognition, *Image and Vision Computing,* 2007.

[113] J.K. Tsotsos, Y. Liu, J. Martinez-Trujillo, M. Pomplun, E. Simine, K. Zhou, Attending to motion, *Computer Vision and Image Understanding,* 100:3–40, 2005.

[114] J.M. Wolfe, Guided Search 2.0: a revised model of visual search, *Psychonomic bulletin & review,* 1: 202–238, 1994.

[115] R. Egly, J. Driver and R.D. Rafal, Shifting visual attention between objects and locations: evidence from normal and parietal lesion subjects, *Journal of Experimental Psychology: General:* 123: 161–177, 1994.

[116] R.P.N. Rao, Visual attention during recognition, In: *Advances in Neural Information Processing Systems,* 10: 80–86, 1998.

[117] Lee, S.I. and Lee, S.Y. Top-down attention control at feature space for robust pattern recognition, In *Biologically Motivated Computer Vision,* Seoul, Korea, 2000.

[118] E.T. Rolls and G. Deco, Attention in natural scenes: neurophysiological and computational bases, *Neural Network.,* 19: 1383–1394, 2006.

[119] T. Kadir and M. Brady, Scale saliency: A novel approach to salient feature and scale selection, in *Proc. Int. Conf. Visual Information Engineering*, Surrey, U.K., Nov., 25–28, 2000.

[120] L. Itti and P. Baldi, Bayesian surprise attracts human attention, *Vision Research,* 49(10):1295–1306, 2009.

[121] A. Klaser, M. Marszalek, and C. Schmid, A spatio-temporal descriptor based on 3D gradients, In *British Machine Vision Conference,* 2008.

[122] H. Tamura, S. Mori, T. Yamawaki, Texture features corresponding to visual perception, *IEEE Transactions on Systems, Man and Cybernetics,* 8: 460-473, 1978.

[123] M. Amadasun, R. King, Textural features corresponding to textural properties, *IEEE Trans. on Systems, Man and Cybernetics,*19: 1264-1274, 1989.

[124] D. Zhang and G. Lu, Study and evaluation of different Fourier methods for image retrieval, *In Image and Visual Computing,* 23: 33-49, 2005.

[125] Q. Gao and A. Wong, Curve detection based on perceptual organization, *Pattern Recognition*, 26(1): 1039–1046, 1993.

[126] G. Hu and Q. Gao, A non-parametric statistics based method for generic curve partition and classification, In *Proc. of Int. Conf. Image Processing* , 3041-3044, 2010.

[127] X. Zheng, S. A. Sherrill-Mix, Q. Gao, Perceptual shape-based natural image representation and retrieval, In Proc: *The 1st IEEE Int. Conf. on Semantic Computing (ICSC2007)*, 622-629, 2007.

[128] G. Hu and Q. Gao, Interactive image feature visualization for supporting CBIR study, In Proc. *Int. Conf. on Image Analysis and Recognition*, LNCS 5627: 239-247, 2009.

[129] H. Chen and Q. Gao, Integrating color and gradient into real-time curve tracking and feature extraction for video surveillance, Book Title: Video Surveillance, Chapter 12, Publisher: *InTechOpen*, ISBN: 978-953-307-436-8: 217-230, 2011.

[130] Y. Li. Generic edge feature extraction based on perceptual curve partitioning, *Master thesis*, *Dalhousie University*, 2004.

[131] http://en.wikipedia.org/wiki/Non-parametric_statistics [retrieved 8 May 2012]

[132] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Journal of Computer Vision,* 60(2):91-110, 2004.

[133] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 27(10):1615-1630, 2005.

[134] M.A. Fischler and R.C. Bolles, Perceptual organization and curve partitioning, *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 8(1): 100-105, 1986.

[135] http://en.wikipedia.org/wiki/Prim's_algorithm [retrieved 8 May 2012].

[136] Y. Wang and G. Mori, Max-Margin Hidden Conditional Random Fields for Human Action Recognition, In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[137] G.R. Arce, Nonlinear Signal Processing: A Statistical Approach, *Wiley: New Jersey,* USA, 2005.

[138] S. Park, J.K. Aggarwal, Simultaneous tracking of multiple body parts of interacting persons, *Computer Vision and Image Understanding,* 102 (1):1–21. 2006.

[139] P. Viola and M. Jones, Robust real-time object detection, *International Journal of Computer Vision,* (57)2: 137–154, 2001.

[140] R.Tyrrell, R. Rockafellar, J-B Wets, Variational Analysis, *Springer-Verlag*, 2005.

[141] H. Wang, M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, In *British Machine Vision Conf.* 2009.

[142] V. N. Vapnik, The Nature of Statistical Learning Theory, *Springer*, 1995.

[143] Shu-Fai Wong, R. Cipolla, Extracting Spatiotemporal Interest Points using Global Information, In *IEEE Int. Conf. on Computer Vision (ICCV)*,1-8, 2007.

[144] M. A. Fischler and R. C. Bolles, Random Sample Consensus, A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Communications. of the ACM* 24(6): 381–395, 1981.

[145] M. K. Leung and Y. H. Yang, First Sight: A human body outline labeling system, *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 17(4): 359-377, 1995.

APPENDIX A Several Possible PGSE-based Probabilistic Models

Hidden Markov Model (HMM)

Hidden Markov Model is an effective approach to model spatial-temporal series related events. Many gesture recognition and natural language processing applications have adopted HMM as the model platform. HMM is based on a strong Markov property assumption that the conditional probability distribution of future states of the process depends only upon the present state only. In our case, we can collapse the parallel multi-channel PGSE blocks into a sequential state model, each ordered block's state depends on its previous one only. The model is formed as:

$$p(y,x) = \prod_{t=1}^{T} p(y_t \mid y_{t-1}) p(x_t \mid y_t), \tag{A.1}$$

where y_t and x_t are the state and observed PGSE features at t position of the sequence respectively. Given the training PGSEs of a gesture/action, a gesture/action is represented by two sets of learned parameters λ and μ that are for the transition distribution $p(y_{t-1}, y_t)$ and state-observation distribution p(xt, yt) respectively. A throw action is shown in Figure 58. Essentially, HMM model is suitable for labeling sequential PGSEs. It makes sense for some human actions/gestures that have PGSEs from one property, such as wave action containing only X PGSEs. A limitation of HMMs however, is that they cannot naturally handle the cases in which pattern instances overlap in arbitrary ways. The imposed Markov chain assumption on PGSEs from multiple parallel channels does not reflect the true temporal properties among PGSEs from complex gestures/actions.

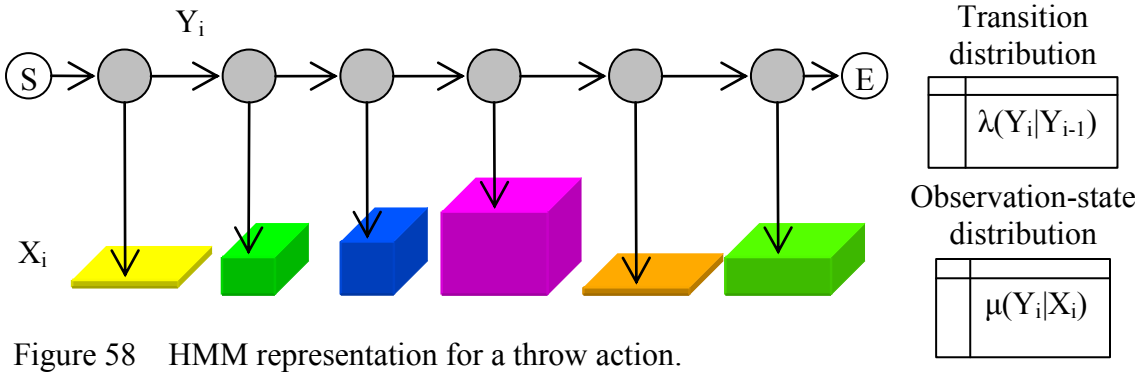


Figure 58 HMM representation for a throw action.

Markov random Field (MRF)

The temporal relationships among dynamic motion properties are converted into the spatial dependencies among PGSE blocks, which are usually not as simple as the sequential pattern. The spatial dependencies among PGSE blocks reflect the intrinsic properties of a human action. Unlike the HMM representation, Markov Random Field (MRF) describes a set of observed data as random variables in an undirected graphic model which is able to incorporate the contextual constraints in a principled manner. MRF models have been used extensively for various segmentation and labeling applications in computer vision, such as image restoration, image segmentation, texture synthesis. Since PGSEs are perceptual salient entities for different dynamic features, and the relationship among PGSEs may play a role in gesture/action understanding, using the MRF model to describe a set of PGSEs is a probabilistic way to reveal intrinsic properties of human activities. MRF is a generative model to estimate the posterior properties. To give descriptions of the human gestures from a MRF framework, two distributions are required: the target label relationships and the target-observation likelihood. Unlike the sequential pattern, MRF tries to model the PGSEs from different property channels occurring in parallel and overlapping patterns. The local and pair-wise Markov properties of the MRF capture the relationship among the target labels of PGSEs as the model prior of the human activities, and the compatibility between a target label and a PGSE is modeled as the likelihood probability. An energy function needs to be defined to characterize the structure and the compatibility properties. The PGSE gesture/action representation has rather than a rectangular lattice structure, non-regular planar structure instead. Since Markov properties of a non-regular arbitrary graph model are difficult to establish, usually in a MRF model, the graph nodes (Y) are factorized by a set of cliques. In our case, we can have cliques that only include the PGSEs from a same time period (in the same start time zone). The intuition of this clique definition is that the motion dynamics within a time period occur almost simultaneously, and have a high degree of correlation. According to the Hammersley-Cliord Theorem, the prior P(Y) follows the Gibbs distribution:

$$p(Y) = \frac{\exp(-U(Y))}{Z} = \frac{1}{Z}\exp(\sum_{c}^{C} w_c^T \varphi_c(y_c)), \quad \text{(A.2)}$$

where Z is the partition function as a normalizing constant, *U(Y)* is called an energy function which consists of a set of potential functions of all cliques, y_c is a set of nodes within a clique c, and $\varphi_c(y_c)$ is a potential function of a clique, w_c^T is a set of weight parameters of $\varphi_c(y_c)$. The value of the energy function *U(Y)* is determined by the sum of similarity degrees of pair-wise neighboring nodes associated with PGSEs.

Take an example in Figure 48, the PGSEs are from a wave action, and the whole time duration of the action is divided into 4 zones (yellow lines in Figure 59). The MRF graph has four cliques circled in the red dash lines. Each node is a variable corresponding to a PGSE block. The variables within each clique are from the same start time zone. The prior distribution of the MRF is written as the sum of potential functions $\varphi_c(x_c)$ of a log-linear function over the graph cliques. The likelihood between PGSEs and a random variable could be assumed in a Gaussian distribution with corresponding means μ and covariance matrices ∑. The MRF parameters $\theta=\{w_c^T, \sum, \mu\}$ are learned from training data. The human gesture/action can be represented as a PGSE-based MRF model P(*Y*, *PGSE*) factorized by the prior distribution P(*Y*) and the likelihood P(*PGSE*|*Y*) with parameters θ.

Conditional Random Filed (CRF)

Both HMM and MRF are the generative model. One limitation of the generative model is that to make the model computationally tractable they have to assume the independence of the observed data. HMM only takes the previous linked variable and one observation into the consideration. Traditional MRF model puts local pair-wise variable dependencies into potential functions to enrich the dependency description, but still only considers the observation data at one site without others. Therefore, the imposed restrictive observation independence assumptions make these models limited in global feature modeling. In our case, the relationships among PGSEs over different time periods are not well modeled if using HMM. And the assumed prior distribution (e.g. Gibbs distribution) will be biased and cannot well reflect true PGSE distribution if using MRF. However, the conditional

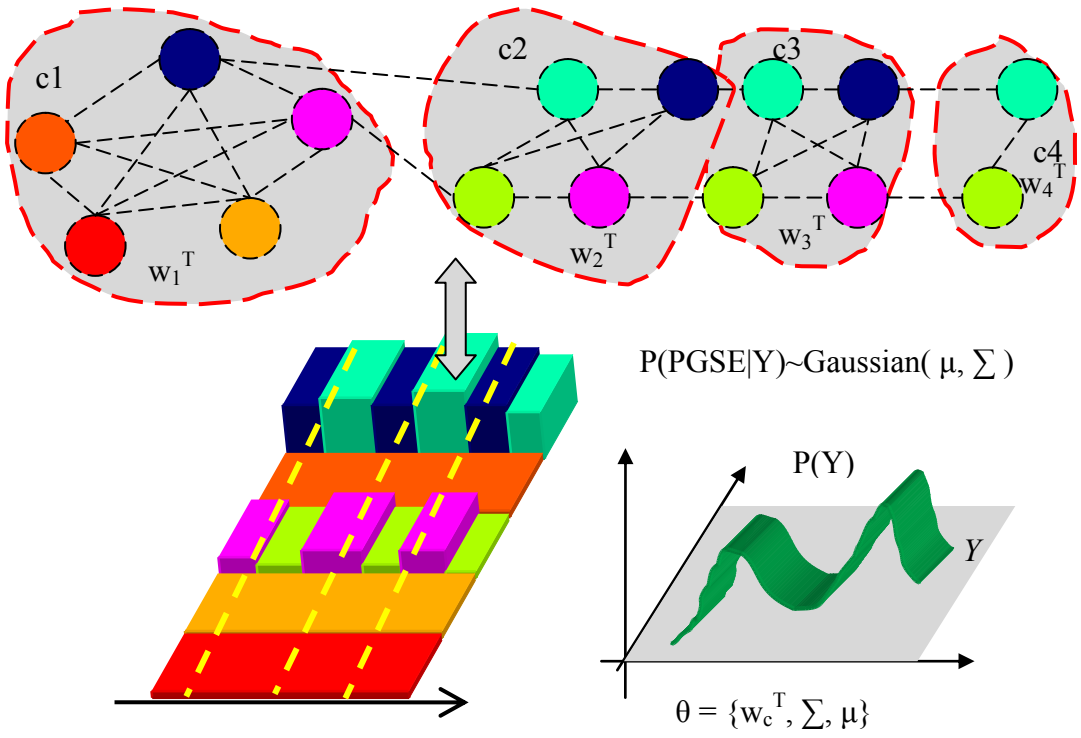


Figure 59 Markov random filed representation for PGSE blocks of a wave action.

random field (CRF) models the variables conditioned upon a set of global observations, i.e. the label assignment decision depends not only on the current observation, but also the surrounding data within a certain size of neighborhood.

CRF is a discriminative method which deals with the modeling of conditional distribution directly from the observation and target gestures/actions without providing the prior distribution. An exponential distribution (e.g. Gibbs distribution) is used to model the statistically correlated features of all variables given the observation data to obtain globally optimal results. It relaxes the independence assumption between variables and observations, and allows non-local dependencies among labels and observations. Furthermore, CRFs can model arbitrary features of observed data and can therefore accommodate complex feature structures.

Let G be a graph over the labels Y, X be the observed PGSE blocks. Similar to the MRF model, the graph could consist of several cliques $C=\{C_p\}$. Each clique C_p contains a set

of edges E with corresponding potential functions $\varphi=\{\varphi_c\}$ and parameters $\theta=\{\theta_p\}$. Each potential function $\varphi_c(x_c, y_c; \theta_p)$ describes the relationship among a variable (label) y_c and an observation data x_c, and weighted by θ_p. Since the distribution is assumed in a Gibbs distribution, the CRF can be written as,

$$p(y \mid x) = \frac{1}{Z}\exp\Big(\sum_{C_P \in C}\sum_{\varphi_c \in C_P}\varphi_c(x_c, y_c; \theta_p)\Big) . \tag{A.3}$$

The collection of the potential functions and weight parameters, $\varphi_c (x_c, y_c; \theta_p)$, are formed up the CRF model for a PGSE block-based human action/gesture pattern. Here, the graph nodes (variables) Y and their connectivity structure is pre-defined, X are the observed PGSEs. θ are the trained weights of potential functions. Any human activities can be represented by this statistical graph model.

Figure 60 (a) shows a CRF undirected graph for a human throw action. The graph nodes are from a set of pre-defined finite label variables (Y) describing human action/gesture properties. The CRF model assigns a label y_j to each observed PGSE block (x_i). A human action/gesture is represented by a set of variables labels tagged on the PGSE blocks. The dashed lines between graph nodes represent the structures of the internal of variables. Each solid line indicate the relation between a pair of PGSE and a variable $<x_j,y_i>$. Each of these edges is described by a corresponding potential function and its weights. In this example, there is only one clique; each variable considers all observations (PGSEs) within a clique.

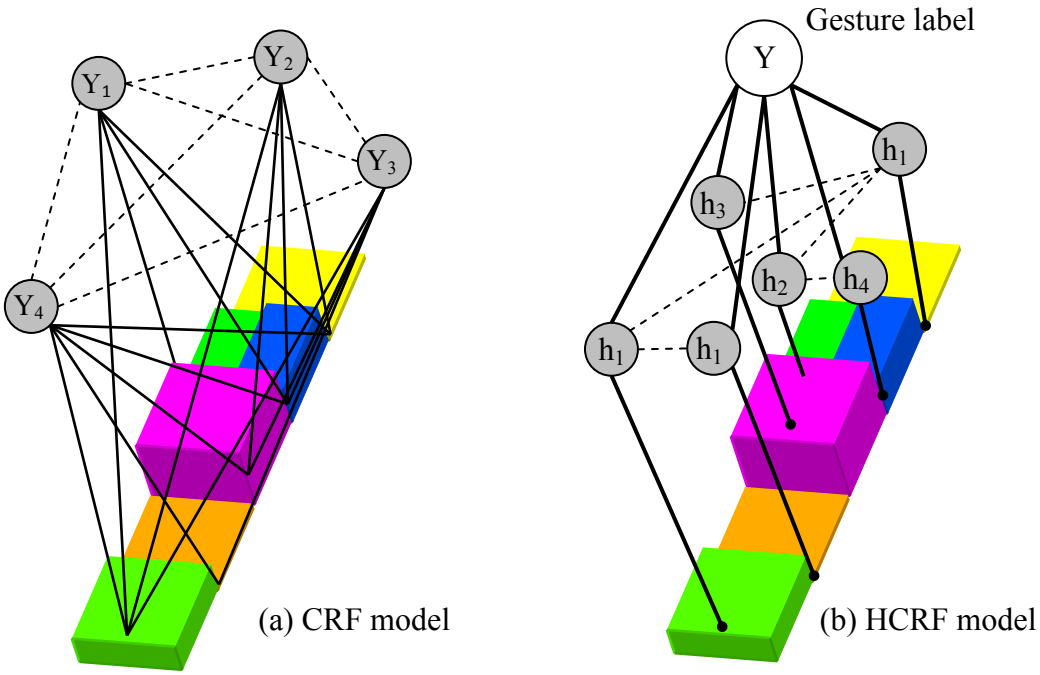


Figure 60 CRF and HCRF representations for PGSE blocks of a throw action.

Hidden Conditional random field (HCRF)

Internal temporal and spatial relationship among multiple dynamic properties is more complicated than what some simple models assume. Rather than a sequential chain or a Markov network, intrinsic structures could be very fuzzy, i.e. a mixture of chain, overlapped and stride modes with uneven steps, and hard to identify. Therefore, many approaches model gestures/actions by either ignoring their intrinsic structures or using oversimplified assumptions so as to limit the performance of corresponding representations. Similarly, the PGSE-based gesture/action representation that is modeled by HMM or MRF will suffer the same difficulty. The fuzzy intrinsic structures of multi-channels are ambiguous and hard to model. Even though CRF could model an arbitrary structure, it requires the knowledge about the connectivity structures of the random variables for a gesture/action class, which may not be feasible for various human activities. In the CRF model example, the neighborhood structures (cliques) are simply constructed by using the start time information which is only one element of PGSEs and may not be enough for building the true structures. Consequently, the oversimplified

graph structures would have less discriminative power. To have more flexibility for structure modeling, we assume that the internal temporal relationships are modeled by some hidden states that govern the pattern discrimination implicitly. Some works [89] [136] showed that incorporating latent structures into the system can improve the recognition and classification performance. In the models we mentioned before, HMM is only suitable for sequential order dataset, and an imposed strong independent assumption make it hard to represent complex and correlate relationships. MRF puts more emphasis on pair-wise relations, and its generative method cannot directly provide a way to estimate the conditional probability of a gesture/action for entire PGSE features. CRF is not able to capture hidden states which are more suitable to represent the latent relationships among multiple channel PGSEs.

Hidden Conditional Random Field as an extended CRF is able to associate the hidden state layer to model the unknown internal substructure between the label and the observed data. In our case, the labels of HCRF are a fixed set of human actions/gestures characterized by observed PGSE block patterns, and the relationship between labels and observed PGSE blocks are modeled by a set of hidden states which reflect the temporal correlation among PGSEs. HCRF is able to reveal the sophisticated internal relationship among PGSEs. Figure 60(b) shows the HCRF graphic model for a throw gesture. PGSEs are extracted as the observation data from the extrinsic properties of a human gesture/action. The gray circles are the hidden states which model the internal relationships between PGSEs. The upper level white circle is the gesture/action class.

HCRF model can be used either as a gesture/action class detector, where a single class is discriminatively trained against all other gestures/actions, or as a multi-way gesture/action classifier, where discriminative models for multiple gestures are simultaneously trained.