Thesis
Doctor of Philosophy

A STUDY ON
HUMAN MOTION ACQUISITION AND RECOGNITION
EMPLOYING STRUCTURED MOTION DATABASE

By

S. M. Ashik Eftakhar

Faculty of Engineering,
Department of Mechanical & Control Engineering,
Kyushu Institute of Technology,
JAPAN

THESIS

A STUDY ON HUMAN MOTION ACQUISITION AND RECOGNITION EMPLOYING STRUCTURED MOTION DATABASE

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

BY

S. M. ASHIK EFTAKHAR

STUDENT NO.: 09584202

SUPERVISED BY

PROFESSOR SEIJI ISHIKAWA

DEPARTMENT OF MECHANICAL AND CONTROL ENGINEERING

KYUSHU INSTITUTE OF TECHNOLOGY

2012

Doctoral Thesis

A Study on Human Motion Acquisition and Recognition Employing Structured Motion Database

By

S. M. Ashik Eftakhar

(Student Number: 09584202)

Supervisor:

Professor Seiji Ishikawa

Member of Thesis Committee:

Professor Seiji Ishikawa

Professor Seiichi Serikawa

Professor Yoshihiko Tagawa

Professor Hyoungseop Kim

Kyushu Institute of Technology
Faculty of Engineering
Department of Mechanical and Control Engineering

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

……………………………………………………………….

Dr. Seiji Ishikawa

(Professor, Mechanical and Control Engineering)

Contents

List of Figures

List of Tables

Abstract

Human motion analysis is an emerging research field for the video-based applications capable of acquiring and recognizing human motions or actions. The automaticity of such a system with these capabilities has vital importance in real-life scenarios. With the increasing number of applications, the demand for a human motion acquisition system is gaining importance day-by-day. We develop such kind of acquisition system based on body-parts modeling strategy. The system is able to acquire the motion by positioning body joints and interpreting those joints by the inter-parts inclination. Besides the development of the acquisition system, there is increasing need for a reliable human motion recognition system in recent years. There are a number of researches on motion recognition is performed in last two decades. At the same time, an enormous amount of bulk motion datasets are becoming available. Therefore, it becomes an indispensable task to develop a motion database that can deal with large variability of motions efficiently. We have developed such a system based on the structured motion database concept. In order to gain a perspective on this issue, we have analyzed various aspects of the motion database with a view to establishing a standard recognition scheme. The conventional structured database is subjected to improvement by considering three aspects: directional organization, nearest neighbor searching problem resolution, and prior direction estimation. In order to investigate and analyze comprehensively the effect of those aspects on motion recognition, we have adopted two forms of motion representation, eigenspace-based motion compression, and B-Tree structured database. Moreover, we have also analyzed the two important constraints in motion recognition: missing information and clutter outdoor motions. Two separate systems based on these constraints are also developed that shows the suitable adoption of the constraints. However, several people occupy a scene in practical cases. We have proposed a detection-tracking-recognition integrated action recognition system to deal with multiple people case. The system shows decent performance in outdoor scenarios. The experimental results empirically illustrate the suitability and compatibility of various factors of the motion recognition.

Acknowledgements

All praise to the Almighty Allah.

I like to show my utmost gratitude and gratefulness to my Academic Supervisor **Professor Seiji Ishikawa** for his continuous guidance and valuable advices during my Doctoral Course period. Without his advices, suggestions and support, I would have not forwarded in research to this extent. Besides, I am very much thankful to **Dr. Joo Kooi Tan** for her time-to-time advices and support concerning my research and other things in *Ishikawa Lab* during study period in Kyushu Institute of Technology that really made everything smooth for me. I would like to express my sincere acknowledgements to **Professor Hyoungseop Kim**, and **Professor Seiichi Serikawa**, and **Professor Yoshihiko Tagawa** for their valuable suggestions as the co-supervisors of my thesis. I would like to show my gratitude to all other Professors in Kyushu Institute of Technology whose conducted course subjects I registered for their consistent favor on me.

I am very much grateful to my wife **Ms. Lantana Begum** and my only son **Muhammad Faaris Ahmed** for encouraging me in my research and other activities, and especially, my wife's contrition is invaluable to make my life comfortable, and this inspired me to concentrate on my research to accomplish my goal. I am equally grateful to *my parents* and *relatives* for supporting me physically and mentally.

I would like to thank my respectable teacher **Professor Dr. M. M. A. Hashem** for having faith in my ability, and inspiring me to perform quality research. I am really grateful to **Dr. Ahmed Galal** who has inspired and suggested me in different timely matters to lead my life a good way and face the challenges of life. I like to show my thankfulness to **Dr. Md. Atiqur Rahman Ahad and his family** who have helped and supported me a lot from the very first day I arrived Japan. I like to show equal thankfulness to **Dr. Md. Arifur Rahman Khan and his family** who are also very close to me and my family. I want to express my heartiest thanks to two Japanese personnel, **Mr. Kenji Kurokawa** and **Ms. Keiko Matsuoka**, for their great support to me and my family in many circumstances. I am also thankful to all of my current and past lab members, especially **Dr. Hirofumi Uemura, Mr. Satoru Masaoka, Mr. Makoto Miyoshi, and Mr. Ahmed Boudissa**, for their great support to accomplish my experiments and make my University life smooth. Some other names I like to mention are **Dr. Budi Sugandi, Dr. Hendra, Dr. Noor Danish, Dr. Noor Aamer**, and many other Muslims and Japanese people for their friendly relations that enriched my social life.

I dedicate this work to my beloved wife, parents, and my only son.

Chapter 1
Introduction

1

Introduction

1.1 Background

It is expected that in near future human beings and intelligent robots live together in our society. Robots will move into our home, offices, and other places. It is then required that such a robot should perform its actions based on the human-like degree of vision. Therefore, it would be necessary for a robot to have learning mechanisms that would enable the robot to adapt to and operate in a dynamic environment, because it would be impossible to pre-program a robot with all possible world states that it might encounter. It would be more desirable to teach the robot through examples. Then probably users can make a demonstration of a task and the robot can learn to do it. For example, we can think of a table setting scenario described in [1] where a robot has to learn to recognize the plates, glasses and other objects. Then it has to learn to grasp them in a robust manner, and transport them to the correct location on the table. The robot should understand, from example, the glasses can go on top of plates but plates cannot go on top of glasses, etc. It is also to be noted that the objects could be in a different location each time. Therefore it is not enough just to imitate the motion trajectory. Robots can be taught to perceive the surrounding environment and perform tasks in several ways [1-10]. According to [11], there are two diametrically opposite ways to teach robots: Tell the robot in detail what it has to do or give the robots some learning strategy and let the robot figure out what the appropriate action is. The former strategy was common in the beginning and the robots were preprogrammed to operate in a specific and highly controlled environment and to perform some pre-specified tasks for which controls were specified [12]. Such an approach is not suitable when the robot is required to learn a new task, or when it needs to adapt to a changing environment.

Moreover, it is difficult to program complex tasks in detail and specify exhaustively all new situations the robot might encounter [11]. An example of this is the Honda robot [13-14] that can walk, climb stairs and manipulate objects. It took nearly 10 years to program the robot with these capabilities. On the other hand, learning strategies are meant to prepare robots to deal with new situations. The learning techniques such as reinforcement learning [15, 16, 5] and genetic algorithms [17] used so far have the capacity to learn anything theoretically but in practice their learning power is limited [11, 18]. A combination of programming and learning strategies can be found in [19] where a robot is programmed with a set of basic behaviors and is expected to learn to use these behaviors. This approach does not scale well in modeling higher-level behaviors [20].

As we pointed out earlier, robots are to become common in our daily lives. In such a scenario, a robust vision-based module is to be developed within the robot so that the learning and teaching methods of the robot may improve its performance, and it can deal with unforeseen circumstances by visual analysis. In order to implant the self-acquisition capability within a robot, it must gather sufficient information from the environment. In this context, the perception of the robot is limited to the recognition, understanding, and interpretation of human behaviors, in terms of human motions and actions. In this respect, it is necessary for a robot to recognize human behaviors irrespective of which direction it observes a human. Since the real scenario is a three-dimensional (3-D) space, an intelligent robot should also correspond to the pragmatic situation. Thus the example-based learning strategy can be comprehensively employed with huge number of motions or actions of several persons from a number of varying viewpoints. In this way, the robot can be able to deal with the view-invariant or view-independent human behaviors. For this reason, researches are concentrating on the view-invariant motion capture [21-23]. Therefore, the view-invariance property is very much essential whenever comes the matter of motion analysis. This task can be accomplished by applying a range of technologies and methods to provide imaging-based automatic inspection, process control and guidance for robots. The basic tasks within an intelligent robot system thus can be separated into a number of phases; robots are to learn from demonstrated examples, form an internal representation of the observed action, use the representation to recognize actions, and finally, take decision based on the recognized output and act upon the decision, or reproduce the observed actions in the form of motion acquisition. This process is illustrated in **Fig. 1.1**.

1.2 Computer Vision-based System

Visual analysis of human movements or motions is an emerging research in the computer vision domain. The background of these sorts of researches lies in the basic understandings of computer vision itself. In this section, the ins and outs of computer vision domain are discussed. The next sections describe the detail of a computer vision system.

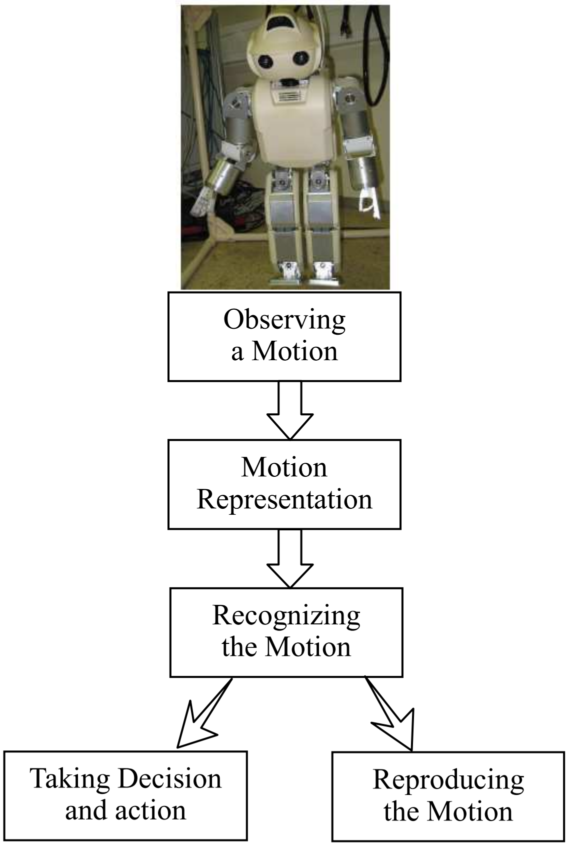


Fig. 1.1 Basic tasks within an intelligent robot system

1.2.1 Definition of computer vision

Computer vision, basically a science and technology, or combination of sciences and technologies, is concerned with the computational understanding and use of visual information which exists in the images. As a scientific discipline, computer vision deals with the theory for building artificial systems that obtain information from images. In part, computer vision is analogous to the transformation of visual sensation into visual perception in biological vision. For this reason the motivation, objectives, formulation, and methodology of computer vision frequently intersect with knowledge about their counterparts in biological vision. Thus it can also be described as a complement of biological vision. In biological vision, the visual perception of humans and various animals are studied, resulting in models of how these systems operate in terms of physiological processes. Computer vision, on the other hand, studies and describes artificial vision systems that are implemented in software and/or hardware. Interdisciplinary exchange between biological and computer vision has proven increasingly fruitful for both fields. However, the goal of computer vision is primarily to enable engineering systems to model and manipulate the environment by using visual sensing.

Computer vision begins with the acquisition of images. A camera produces a grid of samples of the light received from different directions in the scene. The position within the grid where a scene point is imaged is determined by the perspective transformation. The amount of light recorded by the sensor from a certain scene point depends upon the type of lighting, the reflection characteristics, and orientation of the surface being imaged, and the location and spectral sensitivity of the sensor. Using the imaging through digital cameras within digital computers, techniques are adopted to extract, characterize, and interpret information in visual images of a three-dimensional (3-D) world. However, images are sometimes interpreted in such a way to infer 3-D information from two-dimensional (2-D) images. The two-dimensional structure of an image or the three-dimensional structure of a scene must be represented so that the structural properties required for various tasks are easily accessible. Moreover, there are different sorts of imaging for interpretation and recognition of objects and scene contents. Some of these techniques and their analyses will be described in the latter chapters.

1.2.2 Computer vision domains

A significant part of *artificial intelligence* deals with autonomous planning or deliberation for systems which can perform mechanical actions such as moving a robot through some environment. This type of processing typically needs input data provided by a computer vision system, acting as a vision sensor and providing high-level information about the environment and the robot. Other parts which are sometimes described as belonging to artificial intelligence and used in relation to computer vision are pattern recognition and learning techniques. As a consequence, computer vision is sometimes seen as a part of the artificial intelligence field or the computer science field in general. *Physics* is another field that is strongly related to computer vision. A significant part of computer vision deals with methods which require a thorough understanding of the process in which electromagnetic radiation, typically in the visible or the infra-red range, is reflected by the surfaces of objects and is finally measured by the image sensor to produce the image data. The third field which plays an important role is *neurobiology*, specifically the study of the *biological vision* system. This has led to a coarse yet complicated description of how "real" vision systems operate in order to solve certain vision related tasks by studying extensively on visual stimuli in both human and animals. Yet another field related to computer vision is *signal processing*.

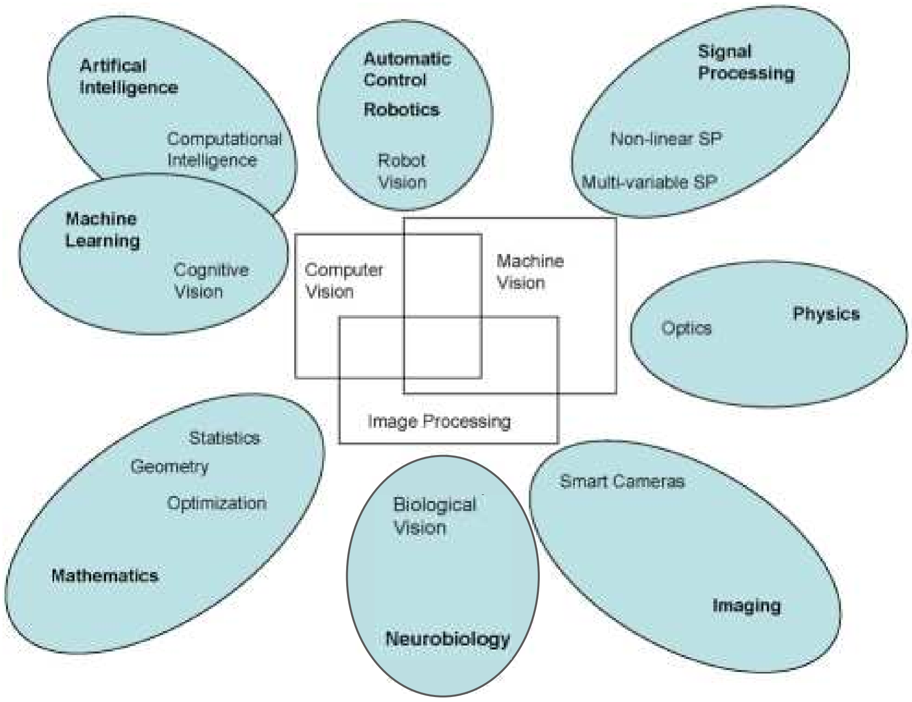


Fig. 1.2 Relation between computer vision and various other fields

Many methods for processing of one-variable signals, typically temporal signals, can be extended in a natural way to processing of non-linear two-variable signals or multi-variable signals in computer vision. Beside the above mentioned views on computer vision, many of the related research topics can also be studied from a purely mathematical point of view based on *statistics*, *optimization*, or *geometry* so as to implement different methods realizing in various combinations of software and hardware, or to analyze how these methods can be modified in order to gain processing speed without losing too much performance. The fields, most closely related to computer vision, are *image processing*, *image analysis*, *robot vision*, and *machine vision*. There is a significant overlap in terms of techniques and applications they cover. This implies that the basic techniques that are used and developed in these fields are more or less identical. However, image processing and image analysis tend to focus on 2-D images, how to transform one image to another, e.g., by pixel-wise operations such as contrast enhancement, local operations such as edge extraction or noise removal, or geometrical transformations such as rotating an image. This characterization implies that image processing/analysis neither require assumptions nor produce interpretations about the image content. Computer vision tends to focus on the 2-D and 3-D scene projected onto one or several images, e.g., how to reconstruct the structure or other information about the scene from one or several images. Machine vision tends to focus on applications, mainly in industry, e.g., vision based autonomous robots and the systems for vision based inspection or measurement. This implies that image sensor technologies and control theory are often integrated with the processing of image data to control a robot and that real-time processing is emphasized by means of efficient implementations in hardware and software. It also implies that the external conditions such as lighting can be and are often more controlled in machine vision than they are in general computer vision, which can enable the use of different algorithms. There is also a field called *imaging* which primarily focuses on the process of producing images, but sometimes also deals with processing and analysis of images. For example, medical imaging contains lots of work on the analysis of image data in medical applications. Finally, *pattern recognition* is a field which uses various methods to extract information mainly based on statistical approaches. A significant part of this field is devoted to applying these methods to various classification and recognition purposes. Sometimes for the accomplishment of a complete research, one might need image processing and/or image analysis at the first step, apply different methodologies to extract scene information, and finally interpret the extracted information for robots to take decisions by applying artificial intelligence. The above described fields constitute the domains of computer vision (See **Fig 1.2**).

1.2.3 Typical tasks of computer vision

Due to large variety of applications in the field of computer vision and in the fields relating computer vision, the fulfillment of those applications requires accomplishing diverse tasks. Some examples of typical computer vision tasks are presented below.

a) Recognition

The classical problem in computer vision, image processing and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g., polyhedrons, cube, etc.), human faces, human shapes, printed or hand-written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera. Different varieties of the recognition problem are described in the literature:

Recognition:

One or several pre-specified or learned objects or object classes can be recognized, usually with respect to their 2-D positions in the image or 3-D poses in the scene. The task of recognition may include either pose of any object or the activity the object engages.

Identification:

An individual instance of an object is recognized such as, identification of a specific person's face or fingerprint, or identification of a specific vehicle, etc.

Detection:

The image data is captured for the detection of specific condition. For example, detection of possible abnormal cells or tissues in medical images, or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interest from a captured image which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

However, several specialized tasks based on recognition also exist. Some of those are described below.

Content-based image retrieval:

It is a task of finding all the images in a larger set of images which have a specific content. The content can be specified in different ways, for example, in terms of similarity to a target image (query for all images similar to image X), or in terms of high-level search criteria given as text input (query for all images which contain many houses, taken during winter and have no cars in them).

Pose estimation:

Estimating the position or orientation of a specific object relative to a camera is a typical pose estimation problem. An application for this technique would be assisting a robot arm in grasping objects from a conveyor belt in an assembly line situation, or interpreting human poses in different situation for the robot to take suitable decisions.

Optical Character Recognition (or OCR):

It is very important in some cases to identify characters in the images of printed or handwritten texts, usually with a view to encoding the text in a suitable format for recognition and interpretation. Some of the most common applications of OCR are reading bank checks, letter mails, or credit-card slips. A bank-check reader may scan just the courtesy-amount field (where the amount of the check is written numerically) and a postal OCR system may scan just the address block on a mail piece. Other major applications are to handle a broader range of documents such as business letters, technical writings, and newspapers. These systems are able to capture an image of a document page and separate the page into text regions and non-text regions.

b) Motion estimation

Several tasks relate to motion estimation, in which an image sequence is processed to produce an estimate of the velocity either at each point in the image or in the 3-D scene. Examples of such tasks are:
Tracking: Following the movements of objects (e.g. vehicles or humans).
Understanding: Recognizing the actions or activities performed by any subject.
Ego-motion: Determining the 3-D rigid motion of the observer (e.g., a camera).

c) Scene reconstruction

Given one or (typically) more images of a scene, or a video, scene reconstruction aims at computing a 3-D model of the scene. In the simplest case, the model can be a set of 3-D points as feature points. More sophisticated methods produce a complete 3-D surface model.

1.2.4 Applications for computer vision

With the recent advancement in the computer vision field, there are numerous applications of computer vision which influence our life to a great extent. This sort of system has basically three major application areas: *surveillance*, *control*, and *analysis*. The *surveillance* area covers applications where one or more subjects are being tracked over time and possibly monitored for abnormal actions under specific situation. A classic example is the surveillance of a parking lot, where a system tracks humans to decide whether they may be about to commit a crime, e.g., stealing a car. The *control* area relates to applications where the captured motion is used to provide controlling functionalities. The *analysis* area is concerned with the detailed analysis of the captured motion data. This may be used in clinical studies of orthopedics patients, choreography of dance and ballet, help athletes understand and improve their performance in sports analysis, and so on. These systems would observe the skills of the pupils and make suggestions for improvement.

Besides the aforesaid areas of applications, a number of other domains of vision-based applications are available. One of the domains of application is virtual reality. In order to create an object in a virtual space, one needs to first recover the body pose in the physical space. Application areas lie in interactive virtual worlds, with the internet as a possible medium. The development of interactive spaces on the internet is still in its infancy; it is in the form of “chat rooms" where users navigate with icons in 2-D spaces while communicating by a text. A more enriched form of interaction with other participants or objects will be possible by adding gestures, head pose and facial expressions as cues. Other applications in this domain are games, virtual studios, motion capture for character animation (synthetic actors) and tele-conferencing. An important application area in the user interface domain involves social interfaces. Social interfaces deal with computer-generated characters, with “human-like" behaviors, who attempt to interact with users in a more personal way [27]. Alternative application areas in the user interface domain are sign-language translation, gesture driven control of graphical objects or appliances, and signaling in high-noise environments such as factories or airports. In the motion analysis domain, a possible application is content-based indexing of sports video footage; in a tennis context, one may want to query a large video archive with “give me all the cases where player *X* came to the net and volleyed". This would eliminate the need for a human to browse through a large data set. Other applications lie in personalized training systems for various sports.

Other most prominent application field is medical computer vision or medical image processing. This area is characterized by the extraction of information from image data for the purpose of making a medical diagnosis of a patient. Generally, image data is in the form of MRI (Magnetic Resonance Imaging), microscopy images, X-ray images, angiography images, ultrasonic images, and CT (Computer Tomography). An example of information which can be extracted from such image data is detection of tumors, arteriosclerosis, any malign changes, or detection of abnormal status of inner organs of human body. It can also be measurements of organ dimensions, blood flow, etc. This application area also supports medical research by providing new information, e.g., about the structure of the brain, or about the quality of medical treatments.

Another application area of computer vision is in industry. Here, information is extracted for the purpose of supporting a manufacturing process. One example is quality control where details or final products are being automatically inspected in order to find defects. Another example is measurement of position and orientation of details to be picked up by a robot arm.

One of the newer application areas is autonomous vehicles, which include submersibles, land-based vehicles (small robots with wheels, cars or trucks), aerial vehicles, and unmanned aerial vehicles (UAV). The level of autonomy ranges from fully autonomous (unmanned) vehicles to vehicles where computer vision based systems support a driver or a pilot in various situations. Fully autonomous vehicles typically use computer vision for navigation, i.e. for knowing where it is, or for producing a map of its environment (SLAM) and for detecting obstacles. It can also be used for detecting certain task specific events, e. g., a UAV looking for forest fires. Examples of supporting systems are obstacle warning systems in cars, and systems for autonomous landing of aircraft. Several car manufacturers have demonstrated systems for autonomous driving of cars, but this technology has still not reached a level where it can be put on the market. However, recently the computer vision system is employed in Intelligent Transport System (ITS) for various vehicle automations.

1.2.5 Organization of a computer vision system

The organization of a computer vision system is highly application dependent. Some systems are stand-alone applications which solve a specific measurement or detection problem, while other constitute a sub-system of a larger design which, for example, also contains sub-systems for control of mechanical actuators, planning, information databases, man-machine interfaces, etc. The specific implementation of a computer vision system also depends on if its functionality is pre-specified or if some part of it can be learned or modified during operation. There are, however, typical functions which are found in many computer vision systems.

- ❑ **Image acquisition:** A digital image is produced by one or several image sensor which, besides various types of light-sensitive cameras, includes range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is a 2-D image, a 3-D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or color images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.
- ❑ **Pre-processing:** Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are
  - ▪ Re-sampling in order to assure that the image coordinate system is correct.
  - ▪ Noise reduction in order to assure that sensor noise does not introduce false information.
  - ▪ Contrast enhancement to assure that relevant information can be detected.
  - ▪ Scale-space representation to enhance or reduce image structures at locally appropriate scales.

- ❑ **Feature extraction:** Image features at various levels of complexity are extracted from the image data for specific purposes. Typical examples of such features are
    - ▪ Primitive features such as lines, edges, and ridges;
    - ▪ Localized interest points such as corners, blobs or points;
    - ▪ Color-based features;
    - ▪ Texture-based features;
    - ▪ Shape-based features;
    - ▪ Motion features, etc.

- ❑ **Detection/Segmentation**: At some point in the processing, a decision is made about which image points or regions of the image are relevant for further processing. Examples are
    - ▪ Selection of a specific set of interest points;
    - ▪ Segmentation of one or multiple image regions which contains a specific object of interest.
- ❑ **High-level processing:** At this step, the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example,
    - ▪ Verification that the data satisfy model-based and application specific assumptions;
    - ▪ Estimation of application specific parameters, such as object poses or object size;
    - ▪ Classifying a detected object into different categories.

1.3 Human Motion Analysis

The systematic analysis of human motion dates back at least to Aristotle. However, it was only in the late 19th century that sequences of photographs could be recorded at sufficient speed for vision-based motion analysis. Pioneers in this field of chronophotography were Marey [24] and Muybridge [25]. Their recordings allowed for qualitative and quantitative analysis of human motion. The shift to automatic human motion analysis largely found its origin in the work by Johansson [26], who placed reflective markers on human joints. He showed that such the representation enabled human observers to recognize human action, gender and viewpoint. These compact representations of human motion also proved to be suitable for automatic recovery and recognition of human motion. However, since markers are usually absent in the image sequences, we focus on markerless, vision-based analysis of human movement. The analysis of images involving humans gains much interest in the recent years.

Generally, the task of human motion analysis refers to tracking, estimation, and recognition. Some of these analysis tasks involve face recognition, hand gesture recognition, human activity recognition, and whole-body tracking. The strong interest in this domain has been motivated by the desire for improved man-machine interaction for which there exists many promising applications. The number of analyses was accomplished in the field of registering human body motion using computer vision. Various categories were defined to characterize those sorts of researches: kinetic and kinematic, model-based and non-model-based, 2-D approaches and 3-D approaches, one person or multiple persons, number of tracked limbs, distributed and centralized processing, various motion type assumptions (rigid, non-rigid, elastic), etc.

Motion analysis of a human body usually involves the extraction of low-level feature, such as body part segmentation, joint detection and identification, and the recovery of 3-D structure from the 2-D projections in an image sequence. There are two typical approaches to the motion analysis of human body parts depending on if *a priori* shape models are used; non-model based and model-based. In each type of approach, the representation of human body parts evolves from stick figures to 2-D contours and to 3-D volumes as the complexity of the model increases [28]. The stick figure representation is based on the observation that human motion is essentially the movement of the supporting bones. The use of 2-D contours to represent the human body is directly associated with the projection of the human figure in images. Volumetric models, such as generalized cones, elliptical cylinders, and spheres, attempt to describe the details of a human body in 3-D and thus require more parameters for computation.

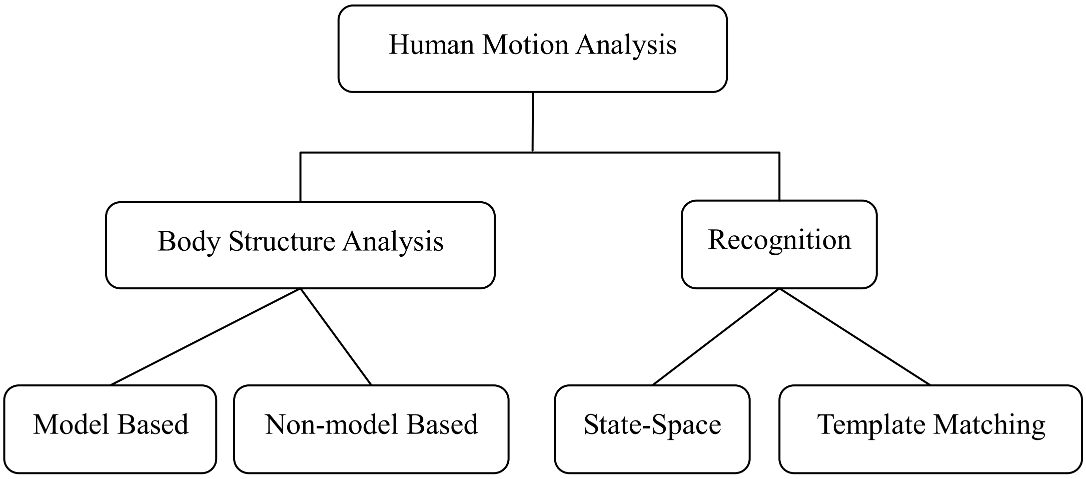


Fig. 1.3 Categorization of human motion analysis

However, for the task of recognition of human motions from an image sequence, researchers typically use one of the two types of approaches; approaches based on a state-space model or ones which use a template matching technique. In the first case, the features used for recognition have been points, lines, and 2-D blobs. Methods using template matching usually apply meshes of a subject image to identify a particular movement. A typical categorization scheme of human motion analysis is shown in **Fig 1.3**. We can deduce different taxonomies depending on the purpose of a system. We will focus on more general aspects such as the overall structure of a motion analysis system and the various types of information being processed. The functional structure of a comprehensive motion analysis system is shown in **Fig. 1.4**.

Before a system is ready to process data it needs to be *initialized*; e.g., an appropriate model of the subject must be established. Next the motion of the subject is *tracked*. This implies a way of segmenting the subject from the background and finding correspondences between segments in consecutive frames. The pose of the subject's body often needs to be estimated as this may be the output of the system, e.g., to control an avatar (the graphical representation of a human) in a virtual environment, or may be processed further by the recognition process. Some higher level knowledge, e.g., a human model, is typically used in *pose estimation*. The final process analyzes the pose or other parameters in order to recognize the actions performed by the subject. A system need not include all four processes, especially since many of the systems described in this survey are the researches in which only a method within one of the processes is investigated. Still, all systems can be described within the structure.

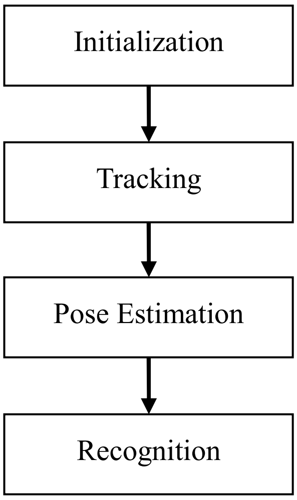


Fig. 1.4 A general structure for systems analyzing human motion

TABLE 1.1

Typical assumptions imposed by motion capture systems

Assumptions related to movements	Assumptions related to appearance
1. The subject remains inside the workspace	**Environment**
2. None or constant camera motion	1. Constant lighting
3. Only one person in the workspace at the time	2. Static background
4. The subject faces the camera at all time	3. Uniform background
5. Movements parallel to the camera-plane	4. Known camera parameters
6. No occlusion	5. Special hardware
7. Slow and continuous movements	
8. Only move one or a few limbs	**Subject**
9. The motion pattern of the subject is known	1. Known start pose
10. Subject moves on a flat ground plane	2. Known subject
	3. Markers placed on the subject
	4. Special colored clothes
	5. Tight-fitting clothes

Besides, human motion analysis systems also adopt various assumptions on the conditions for motion capture. The actual assumptions characterize various systems and provide a useful reference for evaluation. The typical assumptions may be divided into two classes: *movement assumptions* and *appearance assumptions*. The former concerns restrictions on the movements of the subject and/or the camera(s) involved. The latter concerns aspects of the environment and the subject. In **TABLE 1.1**, the relevant assumptions and their association with the two classes are listed. Some assumptions are very general and used in every system with a few exceptions: See, e.g., [29-31]. Some other assumptions simplify the motion or velocity calculation, tracking scheme, trajectory computation, distance calculation, subject or motion segmentation, camera calibration parameter estimation, and so on. Importantly, which assumptions a particular system uses depends on its goals. Generally the complexity of a system is reflected in the number of assumptions introduced; i.e., the fewer the assumptions, the higher the complexity.

We have already noticed that the visual analysis of human motion comprises many aspects. In this thesis, we limit our focus to human motion acquisition and human motion recognition. The former is a model or feature extraction task where the aim is to model the human motions by determining the locations or angles of key joints in the human body given an image of a human figure, or deriving a kinematic model from the body movements. The latter is the process of labeling image sequences with action labels, which is a classification task. Importantly, we do not consider the interpretation of the motion, which requires reasoning and is usually dependent on the specific application or application domain. For both the motion acquisition and the recognition task, we assume that a human figure in an image has been localized in a previous step. A part of human detection or human localization also falls inside our scope. However, we briefly discuss this topic in the following sections.

1.4 State of the Art

In this section we will discuss state of the art within motion analysis context. For the motion analysis carried out in an uncontrolled environment, the figure-ground segmentation relies mostly on motion data, since these are less dependent on various assumptions such as a known subject, known lighting, and different markers. For the same reason, object-based representation (i.e., point, box, silhouette, blob, etc.) is a natural way of representing images at a higher level. Due to uncertainty in the motion, pose estimation with no model or only an indirect use of a model can be used. The dynamic recognition approach is also widely used in such cases. An example of state of the art is the 3-D LSK (Local Steering Kernel) based system by Seo and Milanfar [32] where a novel feature representation is proposed that is derived from space-time 3-D LSKs, which capture the underlying structure of the data quite well, even in the presence of significant distortions and data uncertainty. In fact, 3-D LSKs measure the likeness of a voxel to its surroundings based on computation of a distance between points measured (along the shortest path) on a manifold defined by the embedding of the video data in 4-D. Second, we generalize a training-free nonparametric detection scheme to 3-D, which we developed earlier for 2-D object detection [33]. The state-of-the-art performance is reported on action category classification by using the resulting nearest neighbor classifier. In order to achieve better classification performance, we apply space-time saliency detection [36] to larger videos in order to automatically crop to a short action clip. However, with the extension of the method to a large-scale database requires significant improvement of computational complexity. An efficient database searching strategy can be adopted.

However, if the application is more in the form of direct animation, e.g., avatar control, different methods are used. This type of application is carried out in an indoor setting where a number of assumptions may be introduced, e.g., known subject, known background, and known start pose. Then the appearance-based figure-ground segmentation methods are applied. To obtain good accuracy, direct use of a human model is usually used. As an example of state of the art, we consider the work by Wren et al. [34]. First of all, they use the Pfinder algorithm [35] as the underlying tracking methods. It is a probabilistic method which segments the subject into a number of blobs and tracks those over time. This method has proven to be fast, robust, and able to directly estimate the positions of the head and hands, which are of great importance in control applications. They apply two Pfinder algorithms to obtain 3-D estimates of the hands and a head. Using a human model and kinematic constraints, they estimate the 3-D pose of the upper body. In the framework of a Kalman filter the model is predicted into the next frame to support the blob segmentation and tracking. The innovation of the Kalman filter is used to learn various motion patterns (behaviors) of the subject. These can then be incorporated into the filter to improve the state estimates and predictions, i.e., a better pose estimation result.

In the cases where the motion analysis is carried out in well-controlled environments, a number of assumptions are supposed to be introduced. A detailed model of a human is built for interpreting the motion extracted data which are obtained while analyzing the human motions. An example of direct use of a model is the work by Gavrila and Davis [37]. They use a model-based approach to track a subject in 3-D. A recognition cycle goes as follows. Based on the current and previous states, the allowed intervals for each body parameter (e.g., joint angles) are predicted. For each combination of the 22 body parameters, the human model is synthesized from the cameras' point of view. They compare edges between the synthesized model and the images and thereby (re)formulate the problem as a search problem—how to compare two edge images (a real image with a synthesized image). The search problem is solved using a robust variant of Chamfer matching. When they find the best fit (highest similarity measure), the model is updated using these parameters. They use four synchronized sequences from four different cameras and run the algorithm for each view. In order to obtain stable edges, they wear tight-fitting colored clothes. The high number of joints in their relatively detailed model, the four cameras, and relatively few assumptions make it a rather complex system which, to some extent, is able to estimate the pose of an entire subject. Although the aforesaid analysis-by-synthesis approach seems to be the right one, it is still rather slow and computationally demanding. Methods to prune the state space and faster optimization schemes are required.

1.5 Problem Specification

The focus of the thesis is on the design, development and evaluation of computer-assisted human-sensitive systems. The problem in hand is the recognition of human motions or actions based on video-to-video matching concept. Here, recognition is divided into two parts: motion localization or detection and motion classification. The motion detection deals with separating a motion or an action of interest from the background in a target video by some sort of spatial or spatio-temporal localization of moving region which is formed in the process of the motion performed by a person. Meanwhile, the goal of motion classification is to classify a given motion query into one of several pre-specified categories (for instance, ten categories from the *Avatar Dataset* [38]: *bend, carry, hop, jump, pickup, sit-down, stand-up, stomachache, walk, wave-hands*). This research opens a vast area of applications where it supposed to be fit. We have considered various constraints of different applications, for example, robustness, accuracy, speed, simplicity, database structurization, computational ambiguity, memory requirement, worthy of practical implementation, clutteredness of a scene environment, missing of sufficient features or important information, presence of non-human objects, multiple persons within a scene, etc. This sort of analysis has made our research context wide and extensive. This work is intended to tackle both motion detection and recognition problems simultaneously by searching for a motion of interest within other "target" videos with only a single "query" video. We focus on a sophisticated feature representation with an efficient and reliable similarity measure, which also allows us to avoid the difficult problem of explicit motion estimation. To accomplish the assigned tasks, the emphasis is always on the above measures of performance evaluation.

Though the target video may contain actions similar to the query, these might appear in completely different context. Examples of such differences can range from rather simple optical or geometric differences (such as, different clothes, lighting, motion speed, scale, and view changes) to more complex inherent structural differences. This contextual analysis is mainly within the scope of semantic study of motions. Thus it falls outside of our research context.

1.6 Objectives and Contribution of the Thesis

In this thesis, we propose a novel human motion recognition technique capable of efficient recognition. Our technique mainly deals with the task of recognition by constructing a structured motion database. The objective of our research is to recognize different types of motions captured from a number of viewpoints where those are performed in different surroundings, e.g., controlled indoor, synthesized, cluttered outdoor, etc. At the same time, our target is to make the system efficient by keeping the recognition time shorter, maintaining higher accuracy, dealing with the computational complexity, and making use of useful features of the motions. We also want to establish our recognition scheme as adaptable recognition scheme for computer vision systems. The integration of the various characteristics with the recognition system remains our research objective. Moreover, we have introduced a motion acquisition scheme that makes an intelligent robot to acquire and learn motions directly from the observations, and take further decisions and actions.

In this thesis, we make several contributions, which are summarized below. We consider the evaluation of our contributions as an important aspect. Therefore, we performed extensive experiments on various datasets.

- We give an extensive overview of the related literature in human motion recognition. Moreover, we have also described the concept of motion database and its structurization for effective data searching and retrieval. We describe directions within each field and the advantages and limitations of different approaches, while focusing on recent work. (Chapters 2 and 3)
- We discuss on the concept of human motion acquisition, and present various recent work on pose analysis. We propose an automatic human motion acquisition system for acquiring and understanding limb movements within the motion video. For this purpose, we derive a human model consisting of nine joints: neck, shoulders, elbows, hip, and knees. These joints are detected and the inter-joint angles are computed for acquisition. The motions are captured from frontal and side cameras. (Chapter 4)
- We present a structured motion database approach to human motion recognition. In such an approach, the directional organization of motion database is adopted, and motion recognition of an unseen motion is obtained by searching the directional databases corresponding to the directional feature spaces. However, it has the drawbacks of recognizing the observed motion which is similar to several motions within the motion database. This is called boundary problem that occurs because of the mis-selection of neighboring motions in the retrieval process. We propose a novel resolution approach to this problem to improve the system's performance. Nevertheless, the direction oriented approach searches all the directional feature spaces for similarity measurement. This leads to the ambiguous or unnecessary searching within the database. We have proposed a recognition scheme by pre-estimating the possible orientations of an unknown motion. Thus we are able to eliminate unnecessary searching load and make the system faster. (Chapter 5)
- In realistic situations, overwriting motions or partial occlusion of the human body in a motion is quite common. In order to make use of the former motion information that is lost due to the overwriting problem, we propose a directional motion template based recognition system. Moreover, in real-life scenarios, the surrounding environment of a person is much cluttered with non-uniform background, along with subtle movements of background objects (e.g., trees, shadows, sky, sun, waves, etc.). Because of the

non-uniform nature of outdoor environment, the background, if not subtracted and handled properly, may vastly affect the system's performance. We propose a recognition system that is able to cope with the clutteredness of the background by background modeling and flow estimation. (Chapter 6)

- A practical case with the motion recognition system is the presence of multiple persons in a scene. We use HOG features to detect human objects, and track and accumulate motion features with a view to performing template-based motion recognition. (Chapter 7)

1.7 Thesis Outline

Various human motion representation methods and the motion representation method employed in the current research are discussed in Chapter 2. Existing human motion recognition approaches and how we have adopted the motion recognition strategy using structured motion database are described in Chapter 3.

Various pose estimation methods are analyzed, and a human motion acquisition scheme is proposed in Chapter 4. We analyze various aspects of structured motion database for efficient human motion recognition and the experimental results are also presented in Chapter 5. In Chapter 6, we analyze various recognition constraints in terms of missing motion information and cluttered environment. In Chapter 7, we present a recognition system that can handle the situation where there exist multiple persons and non-human objects in a scene.

In Chapter 8, we summarize our main contributions and discuss the strengths and limitations of our approaches. We also present some future directions of research in this chapter.

Chapter 2
Human Motion Representation

2

Human Motion Representation

2.1 Background

In order to recognize various motions, a motion needs to be described compactly by some convenient representation. There are different forms of representations available, for example, point [39, 40], box [35, 41], silhouette [43-48], blob [49-51], shape [52-55], volume [56-62], and so on. We consider the task of representing human motion so that it can be easily manipulated for high-level processing. A large body of research has been carried out, mainly in recent years. However, in order to derive a suitable representation, motions are often needed to ensure that a system commences its operation with a correct interpretation of the current scene. This is called preprocessing, e.g., [63-64]. Some of the preprocessing may be performed offline prior to the start of operation, while other parts are preferably included as the first phase of operation. Preprocessing may be simplified by relying on some assumptions based on the context of application. Preprocessing mainly concerns segmentation of ROI, camera calibration, adaption to scene characteristics, and so on. The segmentation of the ROI can be obtained by thresholding or subtraction methods, background/foreground modeling, image-based morphological operations, or customized preprocessing techniques. However, for some computer vision systems, camera parameters often need to be known. These can be obtained through offline camera calibration, and for a stationary camera setup occasionally recalibration will suffice. If something in the setup regularly changes, a procedure for online calibration may be preferred as in [65]. However, all other systems are eventually based on offline calibration. Preprocessing to adapt the scene characteristics mainly relates to the appearance assumptions and the segmentation methods. In systems based on the assumptions, a typical offline preprocessing is carried out to find the thresholds and capture reference images which will be used during processing. In some systems, initialized parameters are used in an adaptive procedure to calculate (and update) scene characteristics on the fly [48]. However, rather going into detail of the preprocessing concepts, we concentrate on detail of various state of the art techniques of motion representation. Since motion representation is completely application oriented task, it is wise to adopt one scheme which is feasible with the current motion capture system. At the same time, sufficient motion information must be presented to uniquely distinguish each motion uniquely.

In this section, we first present the scope of this overview, and discuss related surveys within the motion or action representation context. In vision-based human motion recognition, motion representation can be regarded as a combination of feature extraction, and subsequent accumulation of these features for motion representation. We discuss two such kind of well-known motion representation schemes those are adopted in our work in Sections 2.2 and 2.3, respectively. Finally, we summarize the motion representation and its importance in brief in Section 2.4.

2.1.1 Scope of the overview

The area of human motion representation is closely related to other lines of research that extracts and segments human motion. Different motion representations are proposed. Ideally, the features that are extracted from the image sequences should be able to generalize over small variations in person appearance, background, viewpoints, and motion execution. At the same time, the representations must be sufficiently rich to allow for robust classification of the action. The temporal aspect plays an important role in the performance of actions. Some of the representations explicitly take into account the temporal dimension; others extract image features for each frame in the sequence individually. In this case, the temporal variations need to be dealt with in the classification step. Various taxonomies of motion representation are proposed and here we adopted the representation categorization illustrated in [66]. According to this, motion representations are divided into two categories: holistic representations and patch-based representations. The former encodes the visual observation as a whole. Holistic, or global, representations are powerful since they encode much of the information. However, in general they require more preprocessing, such as localization, background subtraction, or tracking. They are also more sensitive to viewpoint, noise and occlusions. When the domain allows for good control of these factors, holistic approaches usually perform well. On the other hand, patch-based, or local, representations describe the observation as a collection of independent patches. Such patches are often centered around spatio-temporal interest points. Since the number of interest points varies depending on the observation, a histogram of code-words is often used. This ensures that the feature vector has a fixed length, but the spatial and temporal information is discarded. Since global information based motion representations require less detailed information than that of patch-based representations, the former can be easily adopted in different motion analysis applications.

However, in many recent researches, a number of markers are attached to the subject to ease the motion capture process. Also, markers are used in many other sophisticated motion capture applications, where subtle changes of motions are needed to be tracked, especially for motion animation. Due to the inconvenience for the subject, computer vision systems allows for touch-free and pure capture mechanism. Therefore, we shall limit our scope on the markerless motion capture and representation. Moreover, we shall focus on those representations which make use of the image sequence, i.e., motion frame, comprising each motion, and, rather than analyzing each frame separately, it sequentially aggregates useful information from the successive motion frames. The approaches that analyze each frame as a distinct unit or consider local features individually within a motion fall outside the scope of this overview. In this overview, we will discuss mainly those approaches that can deal with a variety of motions with different spatial and temporal characteristics.

2.1.2 Surveys

There are several existing surveys within the area of vision-based human motion analysis and motion representation. Recent overviews in [67-69] focus on the vision-based motion analysis from image sequences. This can be regarded as a feature extraction problem. In many cases, the task is to localize persons within the image and extract distinguishable features from the person's continuous movements. We shall discuss the survey on holistic representations.

Holistic representations regard the observation as a whole. Often, this requires localizing the person, which is the task of determining the region of interest (ROI) in the image. The observation within the ROI is subsequently encoded into a convenient image representation. Common global representations are derived from silhouettes, edges or optical flow, and we discuss these in this section. Such representations are global, and are therefore sensitive to noise, partial occlusions and variations in viewpoint. Multiple images over time can be stacked, to form a 3-dimensional space-time volume, where time is the third dimension. Such volumes can be used for action recognition, and we present work in this area in this section.

a. Global representation schemes

When information about the background is given, the silhouette of a person in the image can be obtained by using background subtraction. In general, these silhouettes contain some noise due to imperfect extraction. Moreover, they are sensitive to different viewpoints, and implicitly encode the anthropometry of the subject. However, they encode a great deal of information, and are insensitive to changes in appearance. When the silhouette is obtained, there are many different ways to encode either the silhouette area, or the contour. One of the earliest uses of silhouettes is found in [70]. They extract silhouettes from a single view, calculate differences between subsequent frames and aggregate these overall frames of an action sequence. This results in a binary motion energy image (MEI), which indicates where motion occurs. Also, a gray-scale motion history image (MHI) is constructed, where pixel intensities are a recency function of the silhouette motion. Two templates are compared using Hu moments. In the work of [71], a $\Re$ transform is applied to extracted silhouettes. This results in a translation and scale invariant representation, which is reduced in dimensionality using principal component analysis (PCA). In [72], a $\Re$ transform surface is calculated, where the third dimension is time. Contours are used in [73], where the star skeleton describes the angles between a reference line, and the lines from the center to the gross extremities (head, feet, hands) of the contour. A codebook of star skeletons is used to compare sequences. The scheme in [74] uses either a silhouette or a contour descriptor. Given a sequence of frames, an average silhouette is formed by calculating the mean intensity over all centered frames. Similarly, the mean shape is formed from the centered contours of all frames. The work of [75] matches two silhouettes using Euclidean distance. In later work of [76], it is shown that silhouette templates can also be matched against edges using Chamfer distance, thus eliminating the need for background subtraction. A multi-view motion representation proposed in [83] transforms the postures comprising a motion into a single eXclusive-OR image for the task of storage and recognition. It generates the feature image by simple pixel-wise binary operations.

When multiple cameras are employed, silhouettes can be obtained from each image the cameras provide. Huang and Xu [77] use two orthogonally placed cameras at approximately similar height and distance to the subject. Silhouettes from both cameras are aligned at the medial axis, and an envelope shape is calculated. Cherla et al. [78] also use orthogonally placed cameras and combine the features obtained from the both camera images. Such representations are somewhat view-invariant, but focus on protrusions on the human body, which are not always present. In the work of [79], silhouettes from multiple cameras are combined into a 3-D voxel model. Such a representation is informative but accurate calibration of the cameras is needed. They use motion history volumes (see Figure 2.1(b)), which is an extension of the MHI [70] to the 3-D case. Matching is performed by first aligning the volumes using Fourier transforms on the cylindrical coordinate system around the medial axis. This makes the approach viewpoint-invariant.

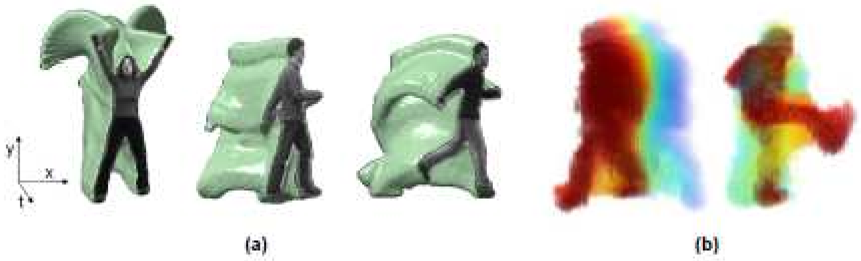


Fig. 2.1 Various motion representation schemes. (a) Space-time volume of stacked silhouettes [55, 84], (b) Motion history volumes [79]. (a) is viewed from a single camera, whereas (b) shows a recency function over reconstructed 3-D voxel models.

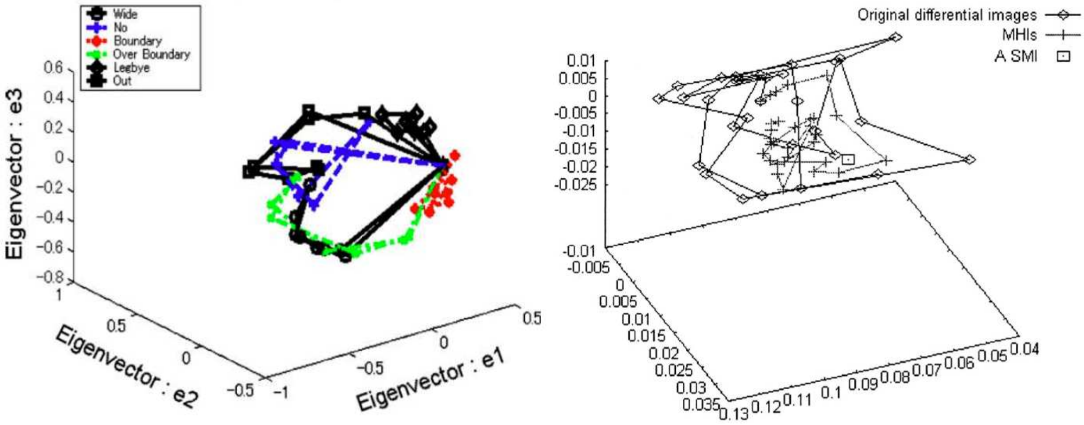
Instead of silhouettes, the observation within the ROI can also be described with optical flow. This is the pixel-wise oriented difference between subsequent frames and can be seen as a motion descriptor. Flow information does not depend on the person's appearance and is somewhat independent of a person's pose. However, dynamic backgrounds can introduce noise in the motion descriptor. Also, camera movement results in observed motion, which is usually compensated by tracking the person. In [80], optical flow is calculated in person-centered images that are obtained from a tracker. They use sports footage, where persons in the image are very small. Optical flow can result in noisy displacement vectors, therefore the result is blurred. To make sure that oppositely directed vectors do not even out, the horizontal and vertical components are divided into positively and negatively directed, yielding 4 distinct channels. The similarity between two flow descriptors is measured using cross-correlation distance. Ahad et al. [81] use these four flow channels to solve the issue of self-occlusion when using a MHI approach. Some other slight modifications of MHI are also presented in different works to customize it for specific tasks [87]. The work in [88, 89] proposes the Gait Energy Image (GEI) that targets specific normal human walking representation based on the concept of MEI. Similar to GEI, an Action Energy Image (AEI) is proposed that is computed by averaging silhouettes and incorporates information about both structure and motion [90]. Other similar methods include Gait History Image (GHI)[91], Gait Moment Image (GMI) [92], Dominant Energy Image (DMI) [93], Average Motion Energy (AME) and Mean Motion Shape (MMS) [94], Motion Energy Histogram (MEH) [95], Edge MHI (EMHI) [96], Multi-level MHI (MMHI) [97-99], Hierarchical Motion History Histogram (HMHH) [100], and many more. The work in [82] derives a number of kinematic features from the optical flow. These include divergence, rotation, symmetry and gradient tensor features. In a subsequent step, PCA is applied to determine dominant kinematic modes.

b. Space-time volumes

When frames over a given sequence are stacked together, a 3-D spatio-temporal volume (STV) can be formed. Usually, frames are aligned to account for translation of the person in the image. In several of the works, the STV is sampled locally. While this approach shares many similarities with patch-based approaches, an STV is a holistic descriptor. The construction of an STV therefore requires accurate localization and alignment and, in many cases, background subtraction or tracking. This makes them less suitable for domains where patch-based approaches typically perform well.

Blank et al. [55, 84] first stack silhouettes over a given sequence to form an STV (see Figure 2.1(a)). Then they use the solution of the Poisson equation to derive local space-time saliency and orientation features. Global features for a given temporal range are obtained by calculating weighted moments over the local features. The work in [85] uses a set of space-time volumes for each sequence, each of which covers only a part of the temporal dimension. The scheme in [86] extracts an STV by stacking frames, and it applies spatio-temporal snakes to carve the volume. By analyzing the periodicity in the XT-slices (obtained by slicing about Y-plane) at approximately knee height, different gait patterns, viewed from the side, are recognized. Instead of a global matching, several works sample the STV surface and extract local descriptors. Yilmaz and Shah [101] use local differential geometric properties on the STV surface. Such properties include maxima and minima in the space-time domain. An action sketch is the set of descriptors that are found on the surface. Given that the descriptors are local, the method is sensitive to noise on the surface. The idea is extended in [102] by first constructing 3-D exemplars from multiple views, for each frame in a training sequence. Then, for each view, an action sketch is calculated from the view-based STV and projected onto the constructed 3-D exemplars. The action sketch descriptors encode both shape and motion, and can be matched with observations obtained from arbitrary viewpoints. The work in [103] extends the shape context to 3-D and it is applied to STVs. The sampling of interest points is adapted to give more importance to moving regions. The work in [104] uses silhouettes, and samples the volume with small 3-D binary space-time patches. Oikonomopoulos et al. [105] extract salient points, and fit B-splines to these points to approximate an STV. The components of the partial derivatives of the volume are clustered into a codebook and used for training and recognition. The scheme in [106] constructs an STV of flow, and samples the horizontal and vertical components in space-time using a 3-D variant of the rectangle features proposed in [107]. Ogata et al. [108] combine the work of [106] with that of Efros et al. [80]. A combination of STVs of silhouettes and flow is used in [109] No background subtraction is needed, as 3-D super-pixels are obtained from segmenting the STV. Action classification is cast as

3-D object matching, where the distance to the segment boundary is used as a similarity measure. The work is extended in [110] to allow for the matching of parts, thus allowing recognition of actions under partial occlusion.



(a) (b)

Fig. 2.2 Eigenspace representations. (a) Graphical representation of an eigenspace containing 6 motions used in cricket match umpiring (Wide, No, Boundary, Over Boundary, Leg bye, Out) [114]. The motion representation is presented by a curve within the space. (b) An example of an eigenspace that is created from normalized differential images, and MHIs and a SMI [115]. Only three prominent dimensions are displayed.

A suitably compact and continuous representation of human appearance or pose is referred to as *eigenspace representation* [111] (See **Fig. 2.2**). This representation of motion consisting of a number of poses was adopted in several researches related to pose estimation [111], posture representation [112], 3-D object detection [113], human motion recognition [114], and so on. However, this sort of representation has been adopted after some preprocessing on motion frames. But it is also possible to use the eigenspace technique for motions to be compressed within a hyperspace [115, 83]. Yamato et al. [116] examines body silhouettes, and Akita [117] employs body contours/edges. Yamato utilizes low-level silhouettes of human actions in a Hidden Markov Model (HMM) framework, where binary silhouettes of background-subtracted images are vector quantized and used as input to the HMMs. In Akita's work [117], the use of edges and some simple 2-D body configuration knowledge (e.g., the arm is a protrusion out from the torso) are used to determine the body parts in a hierarchical manner (first, find legs, then head, arms, trunk) based on stability. Individual parts are found by chaining local contour information. These two approaches help alleviate some of the variability between people but introduce other problems, such as the disappearance of movement that happens to be within the silhouetted region and also the varying amount of contour/edge information (as in most natural scenes).

Also, the problem of examining the entire body, as opposed to only the desired regions, still exists, as it does in much of 3-D work.

So far, from previous discussion we have noticed that whether or not using 2-D or 3-D structural information, many approaches consider a motion to be comprised of a sequence of static poses of an object. Underlying all of these techniques is the requirement that there will be individual features or properties that can be extracted and tracked from each frame of the image sequence. Hence, motion understanding is consequently accomplished by recognizing a sequence of static configurations. However, we are interested in two-dimensional appearance-based representation of human motion where a motion is described by a sequence of 2-D instances/ poses of the object. And in order to use the motions for learning and recognition purposes, we extract features through some processing so that computer can learn first and become able to recognize afterwards.

In latter sections, we are going to discuss in details about our adopted motion representations: *Motion History Image* (MHI) and *Exclusive-OR* (XOR) representation.

2.2 Motion History Image

The concept of *Motion History Image* (MHI) was introduced by Bobick and Davis in 2001 [70] which is view-specific representation of movement, where movement is defined as motion over time. This has been a very famous and well-established motion representation strategy for many years. In generating MHI, the temporal information is embedded and specified by the pixel intensity. So, this is much effective representation of human motions. Therefore, this is represented as a frame-based temporal template for human motions. *Motion History Image*, as the name implies, keeps track of the motion history, i.e. representing *how* the motion is moving along a certain period of time. Let H_τ be a pixel intensity function of the temporal history of motion at a particular point. The function is represented in a simple way in Eq. (2.1).

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, H(x, y, t-1) - 1) & \text{Otherwise} \end{cases} \tag{2.1}$$

Here, D (x,y,t) is a difference binary image constructed by successive frame difference. The function $H_\tau(x, y, t)$ returns a scalar value, and according to the function, the more recently moving pixels are brighter than past moving pixels in the generated image. In the above equation, τ is taken as the temporal extent which is critical to define. But for the flexibility of the value of τ, it can be taken as the maximum gray level pixel value (255) or the maximum number of frames in a motion.

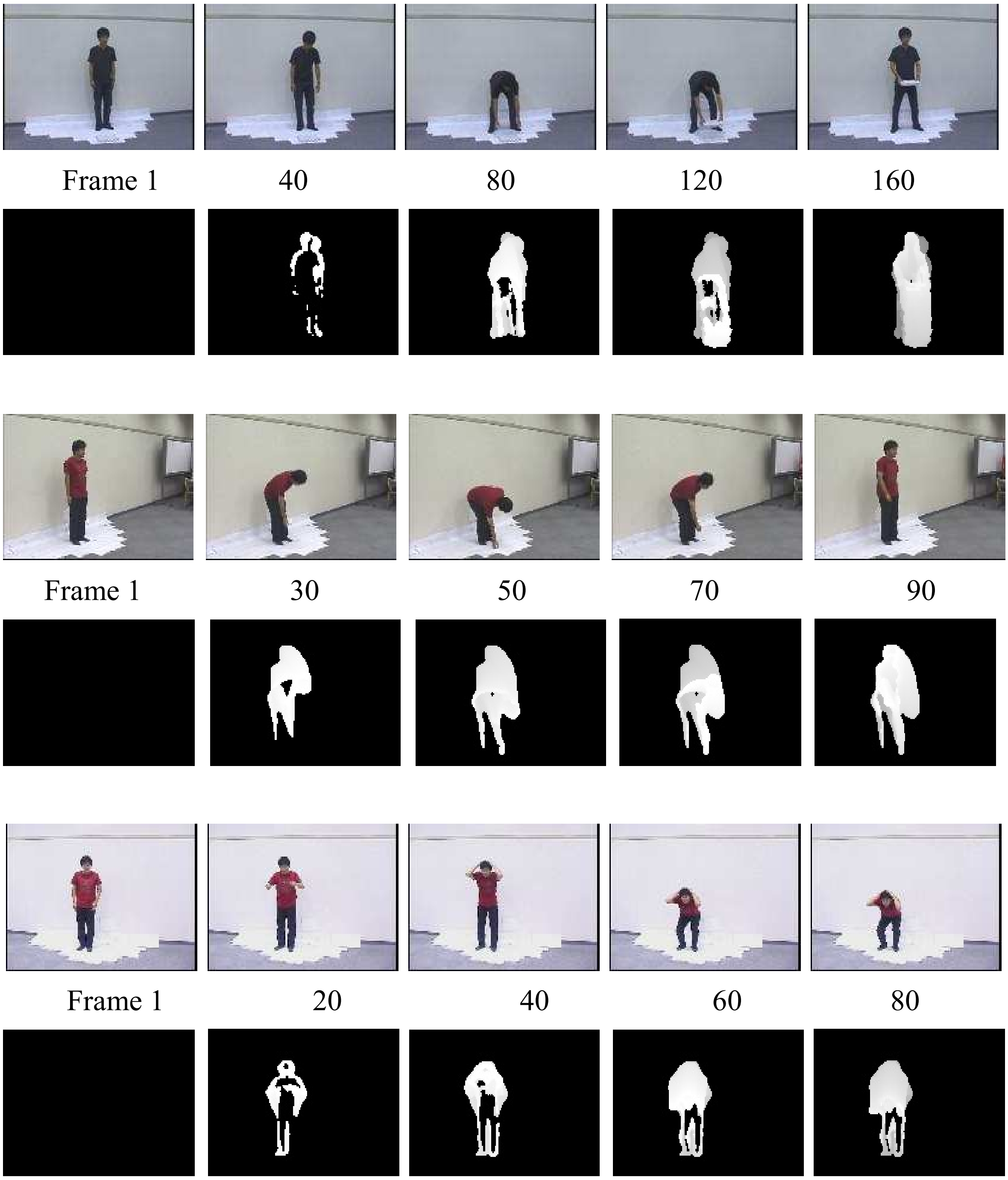


Fig. 2.3 Examples of different actions and corresponding MHIs; top row contains key frames and bottom row is motion history images starting form frame 1. Top to bottom: Carry (frontal view), Pickup (Right view), and Headache-and-Sit (Left view).

However, the MHIs are represented as vector-images (*how* motion is moving) that can be matched against stored representations of known movements. The motion history images implicitly represent the direction of movement. In our research, the MHIs were generated by first changing the original motion frames to gray-level and we segment the region of interest. After that, Eq. (2.1) is applied to all the frames included in each motion or action. Some examples of MHI used for our experiment are shown in **Fig. 2.3** where different actors performed different actions captured from different viewpoints.

2.3 Exclusive-OR Representation

The eXclusive-OR (XOR) representation is a simply logical manipulation between successive motion frames within each motion frame set. The concept of this representation was proposed by J. K. Tan [83] where the motion database was consisted of compressed form of the *XOR motion image* referred to as *JK motion database*. It is the cumulative logical exclusive-OR form of the frame set capable of representing the motion features by a single motion template. Eq. (2.2) is applied to every frame within the motion frame set for constructing this motion template.

$$\begin{aligned} U_c^{m,h}(2) &= f_c^{m,h}(1)\ XOR\ f_c^{m,h}(2) \\ U_c^{m,h}(r) &= U_c^{m,h}(r-1)\ XOR\ f_c^{m,h}(r) \\ U_c^{m,h} &\equiv U_c^{m,h}(R) \end{aligned} \tag{2.2}$$

where, $r = 3,4,\ldots,R$.

Here, the image of motion m of person h obtained from camera c is denoted by $U_c^{m,h}(r)$, whereas f, U denote *binarized motion frame* and *XOR frame*. Hence, for M motions of H persons each motion having R frames from C camera directions generates motion database of *MHC* XOR images. This is an appearance-based motion representation method representing one template image for each motion. This method is capable of avoiding the effect of background or stationary things in the scene. Only the moving portion in the scene is taken into account by using the logical equation. This is simple, effective, and fast generating motion representation method. But for overlapping and complex motions it is unable to handle and may lead to incomplete representation of motions. However, the degree of inability and its modification is of future research concern. Some examples of the generation of XOR representation of motions are illustrated in **Fig. 2.4**.

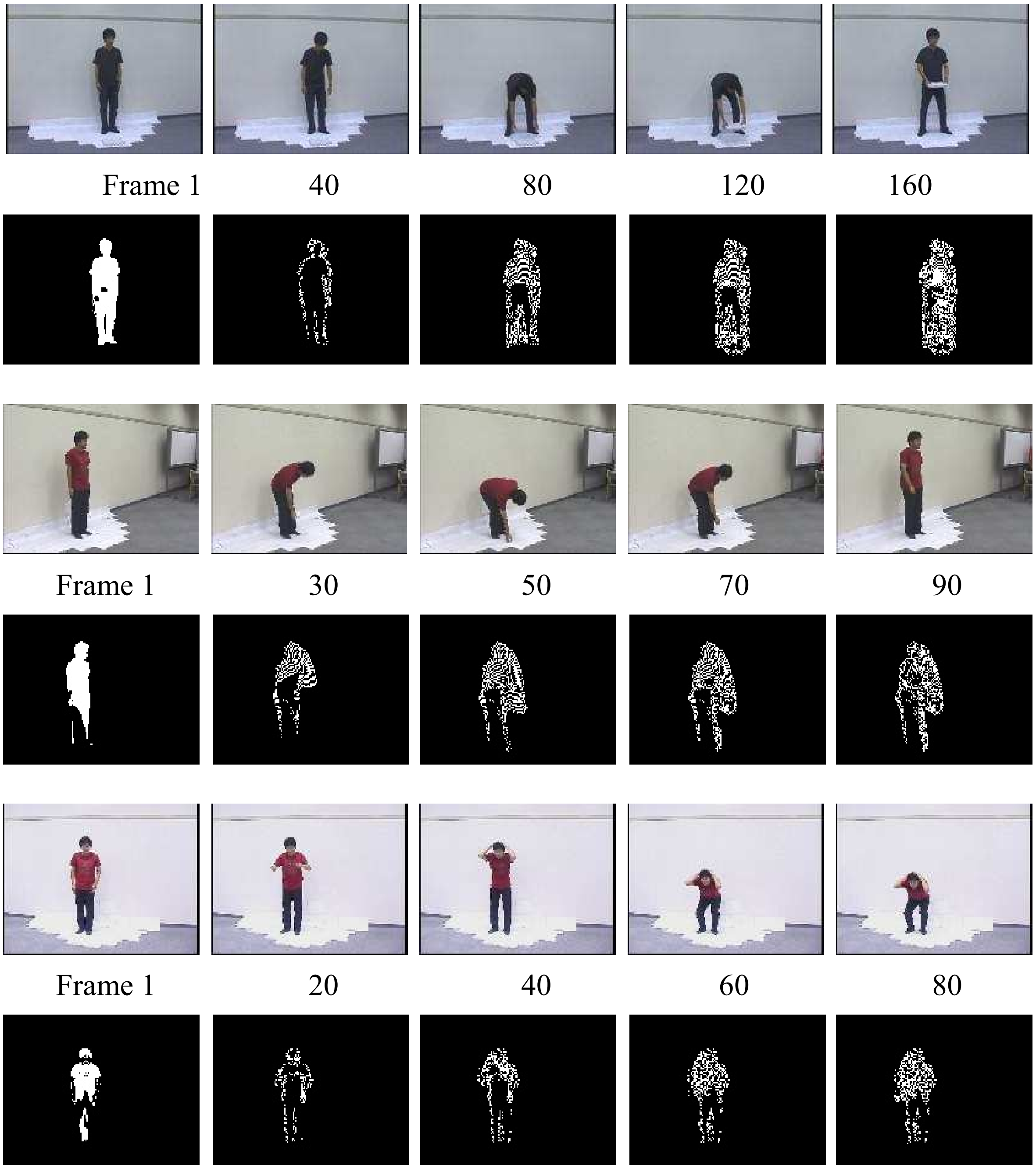


Fig. 2.4 Examples of different actions and corresponding XOR images; top row contains key frames and bottom row is cumulative motion images starting form frame 1. Top to bottom: Carry (frontal view), Pickup (Right view), and Headache-and-Sit (Left view).

2.4 Summary

In this chapter, we concentrate on the different forms of motion representations employed in recent years. We also describe, in detail, about two standard motion representations adopted in our work as *a feature image* from which we extract significant information for the task of human motion recognition. This form of motion images are used in the construction of a feature space which will be described in Chapter 3.

Chapter 3
Human Motion Recognition

3

Human Motion Recognition

3.1 Background

The recognition task of a human motion analysis system can be seen as the most important problem to solve. It is considered as the final or long term goal for many of the motion capture systems. It is a kind of classification problem whose purpose is to classify the captured motion as one of several types of the learned actions. The term 'action' and 'motion' will be used interchangeably throughout the thesis. The actions are normally simple, such as walking and running, but more advanced actions such as different ballet dance steps have also been studied. However, there is hierarchy of human movements to analyze and interpret it. We define the hierarchy as follows;

- *Movement*: A motion which is characterized by a definite space-time trajectory in some configuration space. For a given viewing condition, the appearance of movements is consistent. This is a basic motion that can be detected using low level processing of features.
- *Activity*: Activity describes motion consisting of a sequence of movements. Activities do not refer to external elements.
- *Action*: Actions can be considered as the highest level of abstraction. According to Bobick, actions are the boundary where perception meets cognition. This is because different instances of the same action can have different interpretation depending on the context or object being manipulated. So, the recognition of an action should be linked to the context of application.

Moreover, we can also categorize the motions based the formation of a motion. We categorize them either *simple* or *complex* motions. A *simple action* consists of only one type of movements in it; for example, bending down, sit-down, stand-up, stomachache, etc. On the other hand, a *complex action* contains more than one type of movement; for example, bend to pick something and then standup again with the picked object. In this thesis, we shall try to make use of these two kinds of motions.

Traditionally, two different paradigms exist for recognition: *static recognition* and *dynamic recognition.* Static recognition is concerned with spatial data, one frame at a time. The approaches usually compare prestored information with the current image. The information may be templates [119], transformed templates [120], normalized silhouettes [121], or postures [122]. Conversely, dynamic recognition employs the temporal characteristics for recognition. The methods based on this approach process either spatio-temporal data or temporal pose estimated data. Other forms of recognition approaches are available, as well.

We present related surveys on human motion recognition in Section 3.2. We outline the main characteristics and challenges of the field in Section 3.3, as these motivate the various approaches that are reported in literature. In Section 3.4, we mention the concept and construction of a feature space that is adopted in our work. We introduce the concept of motion database in Section 3.5. The section emphasizes significance of motion database, importance of a structured motion database and the employment of the structured motion database in the current work. Finally, we highlight the contributions of this chapter in brief in Section 3.6.

3.2 Surveys and Taxonomies

The recognition of an action can be performed at various levels of abstraction. Depending on the constraints and the requirements of specific applications, the recognition schemes are analyzed and the most suitable one is selected. Thus different recognition schemes and distance metrics are adopted in different works, and we shall intensively focus those schemes. However, we focus on actions and do not explicitly consider context such as the environment, interactions between persons or objects. These approaches fall outside the scope of this overview. Moreover, we consider only full-body movements. This excludes the work on gesture recognition and other limb extraction based approaches. Another focus of this survey is on motion classification. Practically, motion recognition implies motion classification, and this concept is focused in the survey.

Given the image representation of an unseen sequence, the recognition of human motions becomes the process of motion classification. In this process, a label is associated to the observed frame sequence. Alternatively, a probability distribution over the motion labels can be given. Motion classification is divided into template matching and state-space approaches (See e.g. [28, 69]). Recently, many different approaches have been proposed, and we feel that this traditional taxonomy does not capture these trends properly. Therefore, we use a different approach that is more focused on recent trends. Section 3.2.1 discusses approaches that directly match new sequences to training sequences or action prototypes. These methods do not explicitly model variations in the temporal domain. A subcategory is that of discriminative classifiers that does not match, but rather classify the motion representation directly. Grammars and graphical models are described in Section 3.2.2. These approaches have a state-space character and model temporal variation implicitly. A topic that is related, but is strictly not within the scope of our survey is the detection of query motions in video. These approaches are useful to temporally (and spatially) divide a video into segments, but they lack both the motion model and the labeling ability. We therefore discuss these works separately in Section 3.2.3.

3.2.1 Direct recognition

This section discusses approaches that classify the motion representation without paying special attention to variations in the temporal domain. In Section 3.2.1.1, we discuss the work that maps a new sequence to labeled sequences in the training set or to action class prototypes. The traditional class of spatio-temporal templates also falls into this category. A second class of approach is that of the discriminative classifiers. These learn a function that discriminates between two or more classes by operating directly on the image representation which we discuss in Section 3.2.1.2.

3.2.1.1 Nearest neighbor classification

k-Nearest neighbor (NN) classifiers are the simplest methods of classification. The idea is that motion representations of a given sequence are compared to those of labeled sequences in a training set. The most common label among the *k* most similar sequences is chosen as the classification. The ability to cope with variations in spatial and temporal performance, viewpoint and human appearance depends on the motion representation that is used, and the distance metric that is applied. NN classification can be either performed at the frame level, or for whole sequences. In the latter case, issues with different frame lengths need to be resolved. Due to their fixed descriptor length, global feature-based approaches lend themselves well for matching. For the local feature-based representations, a histogram of codewords can be used to obtain a fixed-length descriptor. For example, Blank et al. [55] apply 1-NN using Euclidean distance between global features; [104] uses Euclidean distance between histograms. In the work of [94], experiments are performed with various distance metrics. Bobick and Davis [70] describe their MHI templates using Hu moments. Given the different orders of these moments, Mahalanobis distance is used to compare a given sequence to an action class prototype. In [123], PCA is used to reduce the dimension. Another work [124] uses a learned discriminative distance metric in their NN classification.

Dynamic time warping: Dynamic time warping (DTW) is a distance measure between two sequences, possibly with different lengths. It simultaneously takes into account a pair-wise distance between corresponding frames and the cost of alignment of the sequences. For two sequences to have a low alignment cost, they need to be segmented similarly in time, and be performed at similar rates. Dynamic programming is used to calculate the optimal alignment. [125] uses DTW but observe that their normalized shape features lie on a spherical manifold. Therefore, they adapt a distance function between two shapes. In later work [126], they also address the alignment of sequences by considering the space of warping functions for a given activity. A related distance is the longest common subsequence (LCS), which is also applied between two sequences. It only takes into account similar elements of both sequences, and results in an increased distance when more inserts or deletions are necessary to warp one sequence onto the other. LCS was used by Yang et al. [127].

Manifold comparison: Different instances of a given action occupy only a part of the entire feature space. This subspace is a manifold, and it can often be embedded into a lower dimensional space. This embedding can be learned from training data, and allows for interpolation of the image representation. Elgammal and Lee [128] use this for human pose recovery, for which they construct manifolds for each action class, and learn mapping functions from image representation to manifold, and from manifold to pose space. For action recognition, pose information is not required. Instead, given a new sequence, the minimum distance of each frame to the manifold of a certain action can be determined. This approach has been taken by Masoud and Papanikolopoulos [129], who use PCA on motion recency images to determine the manifold. While the temporal order is neglected in such an approach, the burden of temporal alignment and variations in speed of performance can be overcome. Instead of using PCA, which is linear, some works learn a non-linear embedding. In [130], learned manifolds are adopted using either PCA or local linear embedding (LLE) on silhouette images. They experiment with different projection functions for LLE. Silhouettes and their distance transforms are also used by Wang and Suter [132] who use locality preserving projections (LPP) for the embedding. The use of Gaussian mixture models (GMM) to model the density of the low-dimensional embedding is investigated. Related work by the same authors [133] either uses the minimum mean frame-wise distance to the manifold, as in [129], or a frame-order preserving variant. Here, it is assumed that the time between two subsequent frames is equal for the entire sequence. More robust in this sense is the work in [134], where an adaptation of DTW is used. This requires adding a time dimension into the embedding, for which they use Isomap. Recent work in [135] focusses on parametric and non-parametric manifold density functions, and describes appropriate distance functions for Grassmann and Stiefel manifold embeddings. All these manifolds are learned in an unsupervised manner, which does not guarantee good discrimination between related classes. The work in [136] address this issue by learning an embedding that is discriminative both in a spatial and temporal sense. They propose local spatio-temporal discriminant embedding (LSTDE), which maps silhouettes of the same class close in the manifold, and model temporal relations in subspaces of the manifold.

Keyframes: It has been observed that many actions can be represented by a small number or even a single key frame or key pose. For example, [137] recognizes forehand and backhand tennis strokes by matching edge representations to stored and manually labeled key poses. Also based on edge distance is the work in [138], where action clusters are learned in an unsupervised fashion. They manually provide action class labels after the clustering. Weinland et al. [139] also learn a set of action key poses but use 3-D voxel representations. The previous methods used only a single frame for action classification. This is convenient when the frame contains a key pose, but in general they will generate many false matches. By considering a sequence of poses over time, ambiguities can be reduced. In [140], histograms of matches are used to manually selected key poses. The length of the histogram equals the number of key poses, and each bin contains the number of frames that best match the corresponding key pose. The histogram is normalized and 1-NN is used for classification.

3.2.1.2 Discriminative classifiers

Discriminative classifiers distinguish between classes without explicitly modeling each. The motion representation is simply regarded as a feature vector. Support vector machines (SVM) are popular classifiers that learn a hyperplane in a feature space that is described by a weighted combination of support vectors. SVMs have often been used in [141-144]. Here, the image representation must be of fixed length, for example a histogram of code-words over a sequence of frames. SVMs can be trained efficiently. Relevance vector machines (RVM) can be regarded as the probabilistic variant of the SVM. An additional advantage is that RVM usually results in a sparser set of support vectors. They have been used for action recognition in [145].

3.2.2 Graphical models

State-based models, or graphical models, are discussed in this section. They consist of states, connected by edges. These edges model probabilities between states, and between states and observations. For the task of action recognition, an observation corresponds to the image representation at a given frame. Usually, one model is trained per action class. In this case, states correspond to phases in the performance of the action. Hidden Markov models (HMM) are the most well-known generative graphical models. The adoption of HMM in motion recognition researches is immense. They use hidden states that correspond to different phases in the performance of an action. HMMs model state transition probabilities, and observation probabilities. To keep the modeling of the joint distribution over representation and labels tractable, two independent assumptions are introduced. First, state transitions are conditioned only on the current state, not on the state history. This is the Markov assumption. Second, observations are conditioned only on the current state, so subsequent observations are considered independent. We discuss the use of generative graphical models, in particular HMMs, for the task of action recognition.

HMMs have been used in a large number of works. Yamato et al. [116] cluster grid-based silhouette mesh features to form a compact codebook of observations. They train HMMs for the recognition of different tennis strokes. Training of an HMM can be done efficiently using the Baum-Welch algorithm. The Viterbi algorithm is used to determine the probability of observing a given sequence. When using a single HMM per action, action recognition becomes finding the action HMM that could generate the observed sequence with the highest probability per action. In [146], a set of HMMs is used, each of which models the action from a certain viewpoint. Weinland et al. [75] construct a codebook by discriminatively selecting a set of templates. In their HMM, they explicitly include the viewpoint, which allows them to condition the observation on the viewpoint. Related work in [147] uses an Action Net, which is constructed by considering key poses and viewpoints. Transitions between views and poses are encoded explicitly. Ahmad and Lee [148] take into account multiple viewpoints and use a multi-dimensional HMM to deal with the different observations. Instead of modeling viewpoints, the work in [149] uses a hybrid HMM, where one process denotes the closest shape-motion template, while the other encodes position, velocity and scale of the person in the image. [151] track persons in 2-D by learning the appearance of the body-parts. In [150], these 2-D tracks are subsequently lifted to 3-D using stored snippets of annotated pose and motion. An HMM is used to infer the action from these labeled code-word motions. However, [152] uses a slightly different approach by assigning one code-word observation to each state. This allows them to effectively train the dynamics, at the cost of reduced flexibility due to a simpler observation model.

Instead of modeling the whole human body as a single observation, an HMM can be made for every body-part individually. This makes training easier, as the combinatorial complexity is reduced to learning dynamical models for each limb individually. In addition, this has the advantage that composite movements that are not in the training set can be recognized. The work of [153] uses the 3-D body-part trajectories that are obtained using [150]. Instead of using labeled codeword motions, they construct HMMs for the legs and arms individually, where 3-D trajectories are the observations. This allows them to use much simpler action models. For each limb, states of different action models with similar emission probabilities are linked. This results in a HMM that allows for automatic segmentation of actions, for legs and arms separately. A similar approach has been taken in [154], where arms, legs and head are found with a set of view-dependent detectors. The work of [155] uses a different approach, but they also use 3-D joint locations as observations. First, they construct a large number of action HMMs, each of which uses a subset of the joints. This results in a large number of relatively weak classifiers. Subsequently, they use AdaBoost to select a set of these classifiers that form the final strong classifier.

In the work by Peursum et al. [156], a factored-state hierarchical HMM (FSHHMM) is used to jointly model image observations and body dynamics for each action class separately. By evaluating an image sequence using each of the action models, the action with the lowest log-likelihood is selected. Related work by Caillette et al. [157] uses a variable length Markov model (VLMM) to model observations and 3-D poses for a given action. The work is mainly aimed at improved 3-D pose tracking, but can also be used for recognition as in [156]. Natarajan and Nevatia [158] introduce a hierarchical variable transition HMM (HVT-HMM), which consists of three layers. The top layer models composite actions, the middle layer primitive actions and the bottom layer poses. Due to their variable window approach, actions can be recognized with low latency.

3.2.3 Video correlation

The approaches which fall within the context of this category do not explicitly model the image representation of subjects in the image, nor do they model action dynamics. Rather, they correlate an unseen sequence to video sequences in a database. Such work is mostly aimed at the detection of actions, rather than their recognition. However, since these works share many similarities to those previously discussed, we will describe them briefly in this section.

Zelnik-Manor and Irani [159] use histograms of appearance-normalized gradient patches, calculated at multiple temporal scales. Patches that exhibit low variance in the temporal dimension are ignored, which focusses the representation on the moving areas in the video. Consequently, for human action recognition, this restricts the approach to detection of movement against non-moving backgrounds. The work of [84] uses histograms of codewords, obtained from Gabor response instead of gradient patch histograms. Shechtman and Irani [59] consider the spatial dimension by correlating spacetime patches over different locations in space and time. Similarly to [159], they use space-time cuboids, but local motion information is used instead of gradients. To avoid calculating the optical flow, a rank-based constraint is used directly on the intensity information of the cuboids. Matikainen et al. [161] present an approximation of method that uses motion words and a look-up table to allow for faster correlation of the motion of different patches. In a recent work [160], a self-similarity descriptor is proposed, that correlates local patches. Such a descriptor is invariant to color, texture and can deal with small spatial variations. A query template is described by an ensemble of all descriptors, either at the frame level, or over a sequence of frames.

3.3 Challenges of the Domain

In human action recognition, the task is to analyze a video and to issue a corresponding action class label. The challenges of this domain encompass intra- and inter-class variations, differences in the recording, Spatial and temporal variations, difference in performance evaluation, and so on. In this section, we discuss these in detail.

a) Intra- and inter-class variations

For many actions, there are large variations in performance. For example, walking movements can differ in speed and stride length. Also, there are anthropometric differences between individuals. In fact, personal differences in gait have motivated its use as a biometric cue. Similar observations can be made for other actions, especially for non-cyclic actions or actions that are adapted to the environment (e.g. avoiding obstacles while walking, or pointing towards a certain location). For multiple classes, distinguishing becomes more challenging when the intra-class variation of each class is high. For example, slow running resembles jogging. A good human action recognition approach should be able to generalize over variations within one class, while at the same time to distinguish between actions of different classes.

b) Environment and recording settings

Even when actions are performed in the same manner, differences in the recording setup and environment result in differences in the captured movement. Since we focus on vision-based human action recognition, we address these differences explicitly. The environment, in which the action performance takes place, is an important source of variation in the recording. When this environment is cluttered or dynamic, it might prove harder to localize the person in the video. Moreover, the environment or recording setup might be such that parts of the person might be occluded in the recording. This introduces source of uncertainty and missing of information. Also, the fact that a single camera is only able to capture a projection introduces a source of variation. The same action, observed from different viewpoints, can lead to very different image observations. Often, a known camera viewpoint is assumed, but this restricts the use to static cameras. When multiple cameras are used, viewpoint problems and issues with occlusion can be alleviated, especially when observations from multiple views can be combined into a consistent representation. Dynamic (or irregular) backgrounds further increase the complexity of localizing the person in the image and robustly observing the motion. When using a moving camera, these issues become even harder. Different persons can appear differently due to differences in anthropometry, but also due to clothing, skin color and facial appearance. Lighting conditions can further influence the appearance. A robust approach should be able to generalize over these factors or employ an initialization phase.

c) Spatial and temporal variations

Since human motion is performed by a person, a common approach is to localize the person in the image or video first. There may be variations in the localization, and human action recognition algorithms should be able to cope with them. There can also be variation in the detection in the temporal domain. Often, actions are assumed to be segmented in time before the actual action classification takes place. Such an assumption moves away the burden of the segmentation from the recognition task, but requires a separate segmentation process in advance. This might not always be realistic. Also, there is substantial variation in the rate of performance of an action. We already discussed inter-personal variations, but the rate at which the action is recorded also has an important effect on the temporal extent of an action, especially when motion features are used. A robust human action recognition algorithm should be invariant to these different rates of execution.

d) Evaluation criteria

Within the domain, much of the evaluation efforts are focused on either publicly available or customized self-developed datasets. While adopting those datasets, there is the risk of tuning the algorithms to the datasets. In particular, the presence of several of the above mentioned variations strongly guides design decisions. Therefore, databases with sufficient variation for these challenges are necessary. Another related concept is the reliability of the data labeling. Most existing data uses actors that perform predefined actions. This readily provides the data labeling. However, performance of an action might be perceived differently by different people. There may be the case of significant disagreement between human labeling and the assumed ground-truth on a specific dataset. When no ground truth is available, an unsupervised approach needs to be pursued. While such an approach will discover classes of similar movement, there is no guarantee that these classes are semantically meaningful.

3.4 Construction of a Feature Space

A feature is a salient attribute for characterizing a motion in such a way that one motion can be distinguishable from the other. A set of features, collectively termed as feature image or motion image, characterizes a motion to determine the similarity of it among a number of training motions. However, a feature space refers to a mathematical space where the features, i.e., feature values, form a compact representation. In the context of motion recognition, the feature space plays an important role. This construction and manipulation of the feature space is performed at the learning and recognition phase. *How well the motion has been characterized* – this specifies *how perfectly* the recognition task will be accomplished. So, the selection of feature space construction scheme is very crucial in upgrading the performance of the system. This includes statistical modeling (e.g., HMM [51, 146-152, 162-163]), mathematical modeling (e.g., Hu moments [70]), time series-based modeling (e.g., motion time series [164]), feature extraction [165-166], eigenspace representation [111], or many other techniques.

Though the feature space construction scheme varies from system to system, the fundamental thing is to choose the scheme which can adapt to the system considering the system input. We need to keep in mind some advantages of the scheme – ease of employment, flexibility, reliability, simplicity, proof of excellence, etc. – before we select one. Among the aforesaid schemes, *eigenspace representation* is much acceptable and highly appreciated strategy that can be adopted in the current work, while coping with the successful transformation of motion data. In the next sections, this scheme will be described elaborately.

3.4.1 Eigenspace

Mathematical Definition

In linear algebra, the *eigenvectors* (from the German *eigen* meaning "inherent, characteristic") of a linear operator are non-zero vectors which, when operated on by the operator, result in a scalar multiple of them. The scalar is then called the *eigenvalue* associated with the eigenvector. In applied mathematics and physics, the eigenvectors of a matrix or a differential operator often have important physical significance. In classical mechanics the eigenvectors of the governing equations typically correspond to natural modes of vibration in a body, and the eigenvalues to their frequencies. In quantum mechanics, operators correspond to observable variables, eigenvectors are also called *eigenstates*, and the eigenvalues of an operator represent those values of the corresponding variable that have non-zero probability of occurring.

Formally, we define eigenvectors and eigenvalues as follows: If A: $V \rightarrow V$ is a linear operator on some *vector space* V, **v** is a non-zero vector in V and λ is a scalar (possibly zero) such that

$$\mathbf{Av} = \lambda\mathbf{v} \tag{3.1}$$

Then we say that **v** is an eigenvector of the operator **A**, and its associated eigenvalue is λ. Note that if **v** is an eigenvector with eigenvalue λ, then any non-zero multiple of **v** is also an eigenvector with eigenvalue λ. In fact, all the eigenvectors with associated eigenvalue λ, together with **0**, form a subspace of V, the eigenspace for the eigenvalue λ. In other words, If **A** is an $n \times n$ square matrix and λ is an eigenvalue of **A**, then the union of the zero vector **0** and the set of all eigenvectors corresponding to eigenvalues λ is a subspace of $\mathbb{R}^n$ known as the eigenspace of λ.

Practical Definition

According to object recognition concept, an efficient coding scheme is necessary for storing and retrieving the views representing the object or image. The most common coding scheme is based on representing each view using a relatively low-dimensional space which captures important characteristics of the entire set of views. This low-dimensional space is generally formed using an eigen-decomposition (or principal components analysis) to denote a subspace which provides a reasonable approximation to the set of stored views. Each stored model view is represented in terms of its projection into this subspace which is quite compact in comparison with the number of pixels in each view. An unknown object is recognized by projecting its image into the subspace and then finding the closest model views in the subspace using some similarity measure. We refer to this subspace as an *eigenspace*.

Subspace methods are attractive when there is a relatively large database of model views because the set of model views can be represented using a small number of coefficients each rather than the thousands of pixels in each image. This both saves storage and speeds of the process of finding the closest matching images in the database. Moreover, when the subspace is relatively low-dimensional (e.g., 25-30 dimensions), we can approximate the closest matches in the database easily [167]. Subspace methods can also be viewed as a form of generalization or learning. To the extent that a subspace captures the important characteristics of a given set of images while omitting the unimportant characteristics, it can be insensitive to unimportant variations in the images. The most effective applications of subspace methods have been limited to those tasks where the objects that are to be recognized appear fully visible i.e. not partially occluded against a uniform background and where the images are nearly correctly registered with each other in advance. For example, a particularly successful application is the recognition of faces from mugshots, where the head is generally about the same size and location in the image, and the background is a fixed color [168]. The main reason for these limitations is that when extraneous information from the background of an unknown image is projected into the subspace, it tends to cause incorrect recognition results. This is analogous to the problem that occurs with template matching techniques, using measures such as the sum of squared differences (SSD) or correlation, where background pixels included in a matching window can significantly alter the correlation value and cause incorrect matches. One standard way of addressing this problem in template matching is to use sub-regions of the views, such that the regions do not contain any background. A similar approach has also been taken in eigenspace matching [111, 169]. One drawback, however, is that sub-regions are generally less distinctive and thus can lead to more possible matches being found. This issue of distinctiveness has been addressed in [170] where they use a selection procedure for image regions based on a minimum description length principle.

Therefore, the eigenspace technique is a coding technique which is based on principle component analysis (PCA). It is a variation of Karhunen-Loeve transform; this method computes eigenvectors from an orthogonal basis for the representation of individual images in the image set. Though a large number of eigenvectors may be required for very accurate reconstruction of an image, only a few eigenvectors are generally sufficient to capture the significant appearance characteristic of an image. These eigenvectors constitute the dimensions of what we refer to as the eigenspace for the image set. If any two images from the set are projected onto the eigenspace, the distance between corresponding points in the eigenspace is a measure of similarity of the images [111].

3.4.2 Construction of an Eigenspace

In an eigenspace, each image or view is represented as a single point in the multi-dimensional space; we refer to this point as a compressed form of the image. So, for the construction of an eigenspace, the first step is the acquisition of an image set. After that, the eigenspace is computed with the image set with a set of multi-dimensional points corresponding to the images in the set. The construction process is described below.

3.4.2.1 Normalization of image set

For constructing an image set, all the images of the set should be same size (e.g., 32 X 32 pixels for our experiment). This is done by scale normalization (e.g., **Fig. 3.1**). In our work, the motion images are in the form of MHI and XOR images which are treated as our input image set. Suppose, the motion image is symbolized as a vector $\hat{\mathbf{x}}$ represented as pixel brightness values from the image in raster scan manner:

$$\hat{\mathbf{x}} = \left[\hat{x}_1, \hat{x}_2, \ldots.., \hat{x}_N\right]^{\mathrm{T}} \tag{3.2}$$

where, N is the total number of pixels in each motion image and 'T' is the transpose of a matrix or a vector.

However, we would like our system to be unaffected by the intensity or brightness. So, the brightness normalization is performed on each image, such that the total energy contained in the image is unity, i.e. $\| \boldsymbol{x} \|= 1$. This brightness normalization transforms each motion image $\hat{\mathbf{x}}$ to a normalized image $\mathbf{x}$:

$$\mathbf{x} = \left[x_1, x_2, \ldots.., x_N\right]^{\mathrm{T}} \tag{3.3}$$

where,

$$x_p = \frac{\hat{x}_n}{\|\hat{x}\|}, \quad \|\hat{x}\| = \sqrt{\sum_{n=1}^{N} \hat{x}_n{}^2} \tag{3.4}$$

This scale and brightness normalized motion image will be later used for computation of an eigenspace.

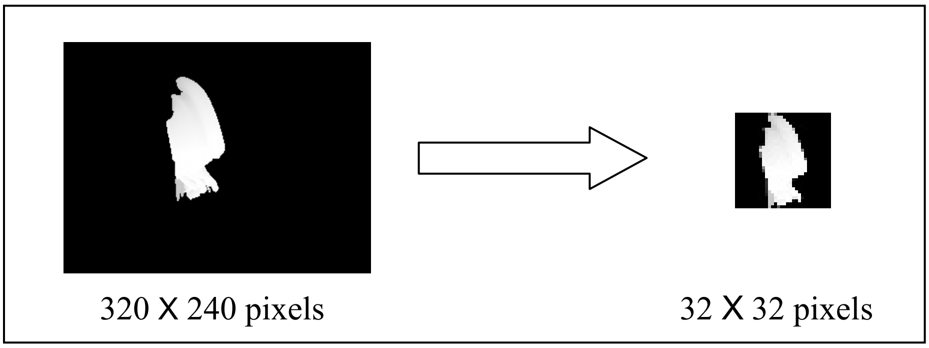


Fig. 3.1 An example of scale normalization

3.4.2.2 Computation of an Eigenspace

Similar motion images, corresponding to same actions, tend to be correlated to a large degree, since motion images for same actions performed by different actors are almost alike. Keeping this in mind, we compute the eigenspace consisting of different actions performed by different actors. In our experiment, there are two types of eigenspaces: *local eigenspaces* consisting of actions from one camera direction (frontal, left, or right) each, and a *global eigenspace* with all the actions from all camera directions. So, we have computed four eigenspaces in our experiment. But the construction scheme is same for all the eigenspaces. We shall describe here the eigenspace construction, in general.

To compute the eigenspace, we first subtract the average of all images in the motion image set (local or global image set). This ensures that the eigenvector with the largest eigenvalue represents the dimension in an eigenspace in which the variance of images is the maximum in the correlation sense. However, this is the most important dimension of the eigenspace. The average **c** of the motion image set is determined as:

$$\mathbf{c} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m \qquad (3.5)$$

where, *M* is the total number of motion images or total number of learning actions.

A new set of motion images are obtained by subtracting the average image from each of the normalized motion image in the set.

$$\mathbf{X} \triangleq (\mathbf{x}_1 - \mathbf{c}, \mathbf{x}_2 - \mathbf{c}, \ldots, \mathbf{x}_M - \mathbf{c}) \qquad (3.6)$$

The image matrix **X** is *N* X *M*, where *M* is the total number of motion images, and *N* is the total number of pixels in each image. To compute eigenvectors of the motion image set, a *covariance matrix* ***Q*** is defined as:

$$\mathbf{Q} \triangleq \mathbf{X}\mathbf{X}^{\mathrm{T}} \qquad (3.7)$$

The covariance matrix is *N* X *N*, obviously a large matrix, since a large number of pixels constitute the motion image. However, the eigenvectors $\mathbf{e}_i$ and the corresponding eigenvalues λ_i of **Q** are determined by solving the following eigen-equation:

$$\mathbf{Qe}_i = \lambda_i \mathbf{e}_i \tag{3.8}$$

By solving the above equation, *N* eigenvectors are obtained with eigenvalues $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_N$. The eigenvalues of the covariance matrix is real and nonnegative. All the *N* eigenvectors in the set constitute a complete eigenspace. However, it is noticeable that the variance is higher for higher eigenvalues and the variance becomes smaller as the eigenvalue decreases. Considering this fact, in place of taking large number of eigenvectors, corresponding to the eigenvalues, it is reasonable to reduce the number of dimensions of the eigenspace. Among *N* eigenvectors, *k* most prominent eigenvectors $(\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_k)$ are chosen to create an eigenspace *ES* consisting of the learning motions using the metric expressed in Eq. (3.9).

$$\kappa = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{3.9}$$

The value of κ is taken to be greater than or equal to 0.80 practically. The eigenvectors for which the variances are more, those are chosen as the prominent eigenvectors. There are other algorithms (e.g., *spatial temporal adaptive* [248]) for computing *k* eigenvectors directly from the covariance matrix. The result is a set of eigenvalues $\{\lambda_i \mid i = 1, 2, \ldots, k\}$ where $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_k$, and a corresponding set of eigenvectors $\{\boldsymbol{e}_i \mid i = 1, 2, \ldots, k\}$. These *k* eigenvectors constitute the eigenspace which is an approximation to the complete eigenspace with *N* dimensions.

After computing the eigenvectors for the construction of an eigenspace, the only task left is the projection of the motion images onto the constructed eigenspace. This is done by first subtracting the average image **c** from the image vector $\mathbf{x}_m$ and performing the dot product with each axis of the *k* dimensional eigenspace. Thus the projected point $\boldsymbol{g}_m$ onto the eigenspace is obtained by Eq. (3.10).

$$\mathbf{g}_m = (\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k)^T (\mathbf{x}_m - \mathbf{c}) \tag{3.10}$$

This represents a multi-dimensional point within the hyperspace which will be used later for storage and similarity measurement. An example of the eigenspace structure is illustrated in **Fig. 3.2**.

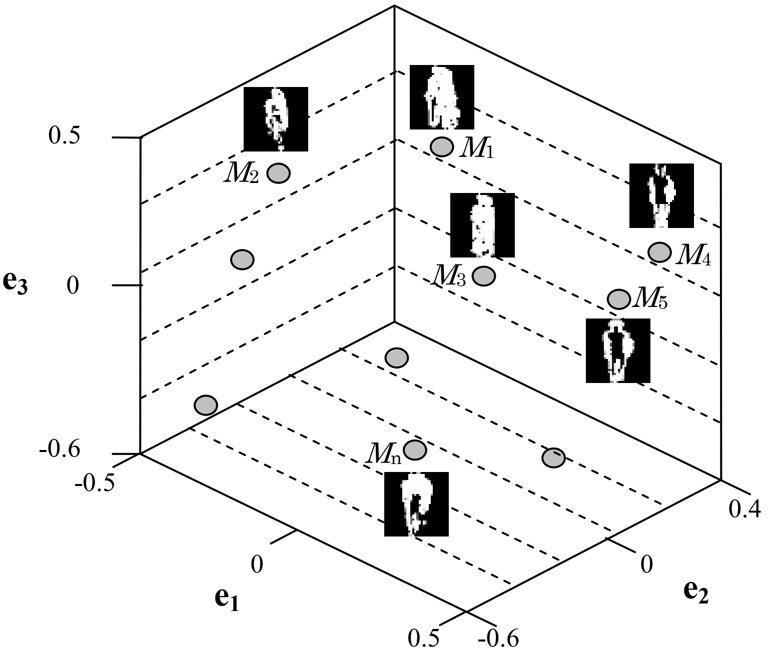


Fig. 3.2 An example of an eigenspace and projected motion images (M_1, M_2,…, M_n). MHI motion representations are used for the illustration. Each motion is represented as a single point within the space. Three major dimensions of the eigenspace are displayed.

3.5 Motion Database

Database is a collection of data stored in a computer system. The database, solely, relies upon the organization or the storage within the computer. Irrelevant to the conventional database concepts, the most common database organization is linear – arranges data in the order of its input. There are no predefined rules for the storage. Therefore, at the time of query, it becomes brute-force searching for the content within the database. In order to overcome this limitation of query time, many of the researches have been performed in developing a suitable motion database capable of quick retrieval.

3.5.1 Overview

Enormous growth of motion archives has significantly increased the demand for research efforts aimed at efficiently finding similar motions within a large motion database. One popular strategy of searching for motions within a motion database is that in which the motion query is expressed as a motion template or a compressed representation of motion. However, the motions are also sometimes represented as multi-dimensional spatial data which are to be stored within the database [83, 115]. Thus there are many variations for motion representations based on different applications. Therefore, considering the variability, there are many non-linear databases available for storing the motions, as a whole. Therefore, considering the variability, there are many non-linear databases available for storing the motions, as a whole. Some examples of this kind of databases are: AVL Trees [171], B-Trees [172], B+-Trees [187], R-Trees [173], R+ -Tree [174], PK-Tree [175], etc. Most of these structures deal with the storage of the multi-dimensional data within the database. B-Tree [172] structure is considered to be an effective database structure to store and retrieve motion data efficiently. In recent work, B-Tree structure has been successfully adopted as a high-speed retrievable Structured Motion Database (SMoDB) [176-179]. Another human motion database proposed uses a binary tree and node transition graphs [180]. The database consists of a binary tree representing the hierarchical clustering of states observed in the motion clips, as well as node transition graphs representing the possible transition among the nodes in the binary tree.

In order to store a motion, it is required to be transformed into a suitable form for the ease of storage and retrieval, which we call an *index*. The process of generating an index is termed as *indexing*. As the size of the motion database is increasing, the problem of indexing of motion databases has attracted great interest in the database community. There are many researches in the past years involving the task of indexing depending on the variability of the format of the motion data [181-185]. A hash table index database of human motions is adopted in [186]. However, motion database development is an emerging research area that requires much attention to make the overall motion analysis more robust, accurate and fast. In our current work, we have adopted the B-tree structure as the structured motion database approach. We shall discuss about B-tree and the reasons for adopting it in detail in Section 3.5.3.

3.5.2 Various non-linear database structures

As we mentioned earlier, a number of non-linear tree structure are available as database to structurize the motion database. We describe in brief some of these structures.

- **AVL Tree**

An AVL tree is a self-balancing binary search tree, and it was the first such data structure to be invented. In an AVL tree, the heights of the two child sub-trees of any node differ by at most one. Lookup, insertion, and deletion all take $O(\log n)$ time in both the average and worst cases, where n is the number of nodes in the tree prior to the operation. Insertions and deletions may require the tree to be rebalanced by one or more tree rotations. The balance factor of a node is the height of its left sub-tree minus the height of its right sub-tree (sometimes opposite) and a node with balance factor 1, 0, or −1 is considered balanced. A node with any other balance factor is considered unbalanced and requires rebalancing the tree. The balance factor is either stored directly at each node or computed from the heights of the sub-trees.

AVL trees are often compared with red-black trees because they support the same set of operations and because red-black trees also take $O(\log n)$ time for the basic operations. Because AVL trees are more rigidly balanced, they are faster than red-black trees for lookup intensive applications. However, red-black trees are faster for insertion and removal.

- **B-Tree**

A B-tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions in logarithmic time. The B-tree is a generalization of a *binary search tree* in that a node can have more than two children. Unlike self-balancing binary search trees, the B-tree is optimized for systems that read and write large blocks of data. It is commonly used in databases and filesystems. A B-tree is kept balanced by requiring that all leaf nodes are at the same depth. This depth will increase slowly as elements are added to the tree, but an increase in the overall depth is infrequent, and results in all leaf nodes being one more node further away from the root.

B-trees have substantial advantages over alternative implementations when node access times far exceed access times within nodes, because then the cost of accessing the node may be amortized over multiple operations within the node. This usually occurs when the nodes are in secondary storage such as disk drives. By maximizing the number of child nodes within each internal node, the height of the tree decreases and the number of expensive node accesses is reduced. In addition, rebalancing the tree occurs less often. The maximum number of child nodes depends on the information that must be stored for each child node and the size of a full disk block or an analogous size in secondary storage. While 2-3 B-trees are easier to explain, practical B-trees using secondary storage want a large number of child nodes to improve performance. The detail of B-Tree will be described in the next section.

- **B+-Tree**

B+-tree is a variant of B-tree while there is no single paper introducing the B+ tree concept. Instead, the notion of maintaining all data in leaf nodes is repeatedly brought up as an interesting variant. An early survey of B-trees covers the concept of B+ trees [187]. A B+ tree or *B plus tree* is a type of tree which represents sorted data in a way that allows for efficient insertion, retrieval and removal of records, each of which is identified by a key. It is a dynamic, multilevel index, with maximum and minimum bounds on the number of keys in each index segment (usually called a "block" or "node"). In a B+ tree, in contrast to a B-tree, all records are stored at the leaf level of the tree; only keys are stored in interior nodes. The primary value of a B+ tree is in storing data for efficient retrieval in a block-oriented storage context—in particular, filesystems. This is primarily because unlike binary search trees, B+ trees have very high fan-out (typically on the order of 100 or more), which reduces the number of I/O operations required to find an element in the tree.

The NTFS, ReiserFS, NSS, XFS, and JFS filesystems all use this type of tree for metadata indexing. Relational database management systems such as IBM DB2, Informix, Microsoft SQL Server, Oracle 8, Sybase ASE, PostgreSQL, Firebird, MySQL and SQLite support this type of tree for table indices. Key-value database management systems such as CouchDB and Tokyo Cabinet support this type of tree for data access. InfinityDB is a concurrent B-tree.

- **R-Tree**

An R-tree is a tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles or polygons. The R-tree was proposed by Guttman in 1984 [173] and has found significant use in both research and real-world applications.

The key idea of the data structure is to group nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree; the "R" in R-tree is for rectangle. Since all objects lie within this bounding rectangle, a query that does not intersect the bounding rectangle can also not intersect any of the contained objects. At the leaf level, each rectangle describes a single object, at higher levels the aggregation of an increasing number of objects. This can also be seen as an increasingly coarse approximation of the data set. Similar to the B-tree, the R-tree is also a balanced search tree (so all leaf nodes are at the same height), organizes the data in pages and is designed for storage on disk (as used in databases). Each page can contain a maximum number of entries, often denoted as M. It also guarantees a minimum fill (except for the root node), however best performance has been experienced with a minimum fill of 30% − 40% of the maximum number of entries (B-trees guarantee 50% page fill, and B*-trees [187] even 66%). The reason for this is the more complex balancing required for spatial data as opposed to linear data stored in B-trees. As with most trees, the searching algorithms (e.g., intersection, containment, nearest neighbor search) are rather simple. The key idea is to use the bounding boxes to decide whether or not to search inside a sub-tree. In this way, most of the nodes in the tree are never read during a search. Like B-trees, this makes R-trees suitable for large data sets and databases, where nodes can be paged to memory when needed, and the whole tree cannot be kept in main memory.

- **R+-Tree**

An R+ tree is a method for looking up data using a location, often (*x*, *y*) coordinates, and often for locations on the surface of the earth. Searching on one number is a solved problem; searching on two or more, and asking for locations that are nearby in both x and y directions, requires craftier algorithms. Fundamentally, an R+ tree is a tree data structure, a variant of the R tree, used for indexing spatial information.

R+ trees are a compromise between R-trees; and kd-trees; they avoid overlapping of internal nodes by inserting an object into multiple leaves if necessary. Coverage is the entire area to cover all related rectangles. Overlap is the entire area which is contained in two or more nodes. Minimal coverage reduces the amount of "dead space" (empty area) which is covered by the nodes of the R-tree. Minimal overlap reduces the set of search paths to the leaves (even more critical for the access time than minimal coverage). Efficient search requires minimal coverage and overlap. R+ trees differ from R trees in that:

- Nodes are not guaranteed to be at least half filled
- The entries of any internal node do not overlap
- An object ID may be stored in more than one leaf node

- **PK-Tree**

A PK-tree is a dynamic spatial indexing structure that can be conceived as a variation of PR quad-trees. However, it differs from all existing trees by employing a unique set of constraints to eliminate unnecessary nodes that can result from a skewed spatial distribution of objects. This ensures that the total number of nodes in a PK-tree is $O(n)$ and the average height of a PK-tree is $O(\log n)$ under some special conditions. In addition, PK-tree has a set of nice properties: non-overlapping of sibling nodes, uniqueness of a PK-tree for a given set of data points, and so on.

3.5.3 Structured motion database

Due to the increased number of registration within the database, the maintenance of the database organization is becoming a very crucial issue. Database system also faces a number of challenges for its implementation. We mention here some challenges that are faced at the time of adopting a suitable database structure and also how B-tree has fulfilled the required degree of efficiency.

3.5.3.1 Database challenges and its handling

□ Time to search a sorted file

Usually, sorting and searching algorithms have been characterized by the number of comparison operations that must be performed using order notation. A binary search of a sorted table with *n* records, for example, can be done in $O(\log_2 n)$ comparisons. If the table had 1,000,000 records, then a specific record could be located with about 20 comparisons: log 21,000,000 = 19.931.... Large databases have historically been kept on disk drives. The time to read a record on a disk drive can dominate the time needed to compare keys once the record is available. The time to read a record from a disk drive involves a seek time and a rotational delay. The seek time may be 0 to 20 or more milliseconds, and the rotational delay averages about half the rotation period. For a 7200 RPM drive, the rotation period is 8.33 milliseconds. For a drive such as the Seagate ST3500320NS, the track-to-track seek time is 0.8 milliseconds and the average reading seek time is 8.5 milliseconds. For simplicity, assume reading from disk takes about 10 milliseconds. Naively, then, the time to locate one record out of a million would take 20 disk reads times 10 milliseconds per disk read, which is 0.2 seconds.

The time won't be that bad because individual records are grouped together in a disk block. A disk block might be 16 kilobytes. If each record is 160 bytes, then 100 records could be stored in each block. The disk read time above was actually for an entire block. Once the disk head is in position, one or more disk blocks can be read with little delay. With 100 records per block, the last 6 or so comparisons don't need to do any disk reads—the comparisons are all within the last disk block read. To speed the search further, the first 13 to 14 comparisons (which each required a disk access) must be sped up.

□ An index speeds the search

A significant improvement can be made with an index. In the example above, initial disk reads narrowed the search range by a factor of two. That can be improved substantially by creating an auxiliary index that contains the first record in each disk block (sometimes called a sparse index). This auxiliary index would be 1% of the size of the original database, but it can be searched more quickly. Finding an entry in the auxiliary index would tell us which block to search in the main database; after searching the auxiliary index, we would have to search only that one block of the main database—at a cost of one more disk read. The index would hold 10,000 entries, so it would take at most 14 comparisons. Like the main database, the last 6 or so comparisons in the aux index would be on the same disk block. The index could be searched in about 8 disk reads, and the desired record could be accessed in 9 disk reads. The trick of creating an auxiliary index can be repeated to make an auxiliary index to the auxiliary index. That would make an aux-aux index that would need only 100 entries and would fit in one disk block.

Instead of reading 14 disk blocks to find the desired record, we only need to read 3 blocks. Reading and searching the first (and only) block of the aux-aux index identifies the relevant block in aux-index. Reading and searching that aux-index block identifies the relevant block in the main database. Instead of 150 milliseconds, we need only 30 milliseconds to get the record. The auxiliary indices have turned the search problem from a binary search requiring roughly $\log_2 n$ disk reads to one requiring only $\log_b n$ disk reads where b is the blocking factor (the number of entries per block: $b = 100$ entries per block; log b1,000,000 = 3 reads). In practice, if the main database is being frequently searched, the aux-aux index and much of the aux index may reside in a disk cache, so they would not incur a disk read.

❑ Insertions and deletions cause trouble

If the database does not change, then compiling the index is simple to do, and the index need never be changed. If there are changes, then managing the database and its index becomes more complicated. Deleting records from a database doesn't cause much trouble. The index can stay the same, and the record can just be marked as deleted. The database stays in sorted order. If there are a lot of deletions, then the searching and storage become less efficient.

Insertions are a disaster in a sorted sequential file, because room for the inserted record must be made. Inserting a record before the first record in the file requires shifting all of the records down one. Such an operation is just too expensive to be practical. A trick is to leave some space lying around to be used for insertions. Instead of densely storing all the records in a block, the block can have some free space to allow for subsequent insertions. Those records would be marked as if they were "deleted" records. Both insertions and deletions are fast as long as space is available on a block. If an insertion won't fit on the block, then some free space on some nearby block must be found and the auxiliary indices adjusted. The hope is that enough space is nearby so that a lot of blocks do not need to be reorganized. Alternatively, some out-of-sequence disk blocks may be used.

The B-tree uses all those ideas by the following way:

- It keeps records in sorted order for sequential traversing.

- It uses a hierarchical index to minimize the number of disk reads.
- It uses partially-full blocks to speed insertions and deletions.
- The index is elegantly adjusted with a recursive algorithm.
- B-tree minimizes waste by making sure that the interior nodes are at least ½ full. A B-tree can handle an arbitrary number of insertions and deletions.

Moreover, the B-Tree has some advantages over other structures; some of those are as follows.

- It works with one-dimensional data, numeric or real numbers.
- With simple indexing of data, multi-dimensional data can be stored structurally within the B-Tree.
- Storage utilization is considerably better in the average.
- Storage is requested and released as required.
- Simple but effective searching within the database for retrieval.

3.5.3.2 Overview of B-Tree

A B-Tree is a multi-way tree branching structure with adaptable capability of storage and retrieval. It is an efficient approach to external searching discovered by R. Bayer and E. McCreight [172]. The idea is to make the structure possible both to search and to update a large file with guaranteed efficiency.

The B-Tree consisting of *m* descendants (order of the tree) with height *h* is defined by $\tau(m,h)$. Then the B-Tree *T* is either empty (*h* = 0) or has the following properties:

(a) Every node has at most *m* children.

(b) Each path from the root to any leaf has the same length *h*.

(c) Root has at least 2 children, unless it is a leaf.

(d) Each node has $\lfloor m/2 \rfloor$ to *m* descendants, except for the root and leaves.

(e) A non-leaf node with *k* children contains *k*-1 keys.

Figure 3.3 shows a B-Tree of the class $\tau(4,3)$ with order 4 and height 3. Therefore, each node (except for the root and the leaves) has between $\lfloor 4/2 \rfloor$ and 4 children, so it contains 2, 3, or 4 keys. The root is allowed to contain 1 to 4 keys; in this case it is 1. All of the leaves are at level 3. It is noticeable that (i) the keys appear in increasing order from left to right, using a natural extension of the concept of symmetric order; and (ii) the number of leaves is exactly one greater than the number of keys.

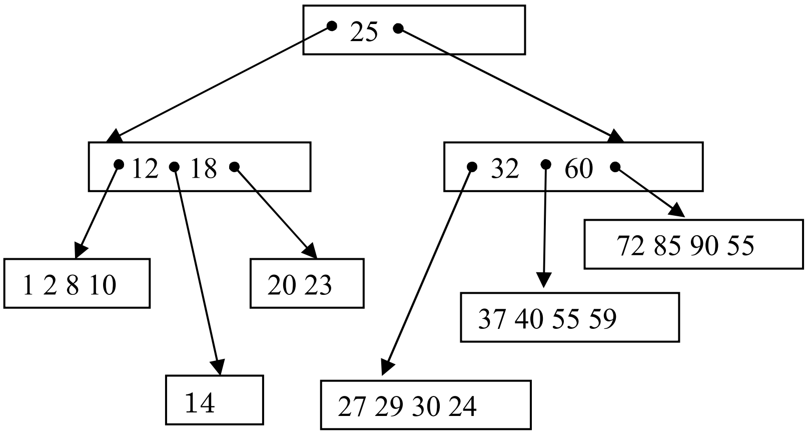


Fig. 3.3 A B-Tree structure in τ (4, 3)

Each node of B-Tree is called a *page*. Within each page of B-Tree, there are two kinds of data: a pointer and a key (shown in Fig. 3.3). A page containing *l* keys and *l*+1 pointers is represented in **Fig. 3.4**.

p_0	x_1	p_1	x_2	p_2	…………	x_l	p_l	Unused Space

Fig. 3.4 Organization of a page of a B-Tree

Here $x_1 < x_2 < \ldots < x_l$ and p_i points to the sub-tree for keys between x_i and x_{i+1}. Therefore, searching in B-Tree is quite straightforward: page (1) has been fetched into the internal memory, we search for the given argument among the keys $x_1, x_2, \ldots, x_l$. When *l* is large, it is better to use binary search, otherwise sequential search is the best. However, if the search is successful, the desired key is found; but if the search is unsuccessful because the argument lies between x_i and x_{i+1}, the page pointed by p_i is fetched and the process continues. The pointer p_0 is used if the argument is less than x_i, and p_l is used if the argument is greater than x_l.

3.6 Summary

In this chapter, we discuss a number of issues concerning human motion recognition. At first, we present detailed survey on human motion recognition and its various challenges. Then we explain the feature space construction scheme adopted in our work. Finally, we present the overview of motion database, mention the challenges for a suitable structured database, and give a brief description of B-tree structured database.

Chapter 4
Automatic Human Motion Acquisition

4

Automatic Human Motion Acquisition

4.1 Human Pose Acquisition

Human pose acquisition is the process of acquiring a configuration or a model of human body, and thus making use of this information to learn within an intelligent system or a robot. When poses are collected over time, it can be applied in human motion analysis. Traditionally, motion capture systems require that markers are attached to the body. These systems have some major drawbacks as they are obtrusive, expensive and impractical in applications in which the observed humans are not necessarily cooperative. As such, many applications, especially in surveillance and human-computer interaction (HCI), would benefit from a solution that is markerless. Vision-based motion capture systems attempt to provide such a solution using cameras as sensors. Over the last two or three decades, this topic has received much interest, and it is considered as an emerging research domain. In this chapter, we present the characteristics, challenges and related surveys on pose acquisition within the vision-based human motion analysis context.

4.2 Survey on Pose Acquisition

The study and research on human pose acquisition cover a significant portion with Human motion analysis concept. In theory, as many details as the human body can exhibit could be acquired, such as facial movement and movement of the fingers. We mainly focus on large body parts (torso, head, and limbs). We limit ourselves to estimating body part configurations over time, and we do not precisely evaluate the motion recognition. For some applications, the positioning of individual body parts is not important. Instead, the entire body is analyzed as a single object, and body configurations are extracted for pose analysis.

Body parts segmentation refers to the process of locating the body parts or body joints within an image. Most of the cases, the background of the image does not have any significant effect on the segmentation. Such kind of human body parts segmentation based modeling is illustrated in **Fig. 4.1**. Theoretically, the goal of body modeling is to construct the function that gives the likelihood of an input image, given a set of parameters. These parameters include body configuration parameters, body shape and appearance parameters and camera viewpoint. Some of these parameters are assumed to be known in advance, for example a fixed camera viewpoint or known body part lengths. Estimating a smaller number of parameters makes the search for the optimal model instantiation more tractable but also poses limitations on the visual input that can be analyzed. Due to the variations between people in shape and appearance, and a different camera viewpoint and environment, the same pose can have many different observations. Also, different poses can result in the same observation. Since the observation is a projection (or combination of projections when multiple cameras are deployed) of the real world, information is lost. When only a single camera is used, depth ambiguities can occur (See **Fig. 4.2**). Also, because the visual resolution of the observations is limited, small changes in pose can go unnoticed.

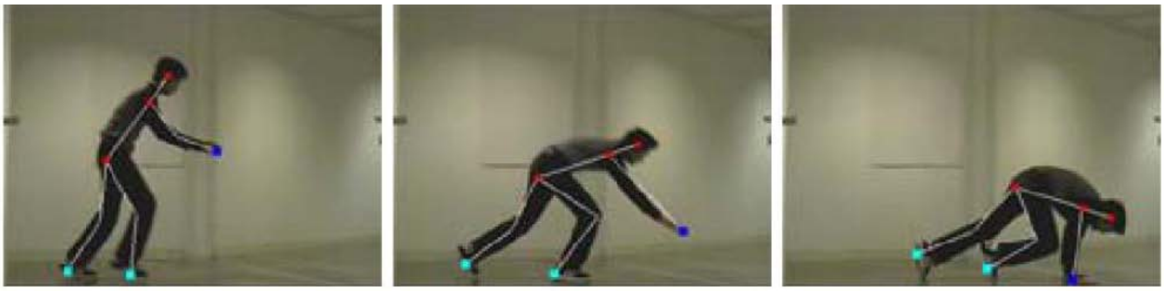


Fig. 4.1 The result of body modeling for the *falling* sequence as adopted in [188].

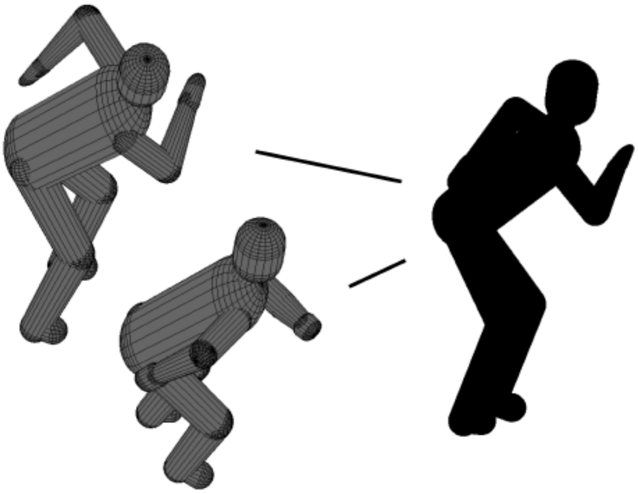


Fig. 4.2 The result of body modeling for the falling sequence as adopted in [189].

4.2.1 Body parts segmentation and modeling

Model-based approaches use a human body model, which may include the kinematic structure and the body dimensions. In addition, a function is used that describes how the human body appears in the image domain, given the model's parameters. Instead of using the original visual input, the image is often described in terms of edges, color regions or silhouettes. A matching function between visual input and the generated appearance of the human body model is then needed to evaluate how well the model instantiation explains the visual input.

Analyzing human motions by tracking or modeling body-parts is a much investigated research subject. A number of researches have been performed for such kind of motion analysis. In [190], a markerless capture of a human in full 3-D with a high-dimensional body model is addressed. It employs a simple 29-dimensional model of the human body parameterized by 24 joint rotations and five global variables (*x*, *y*, *z*, *orientation*, *scale*). A view-based human activity recognition method is proposed in [191]. In this work, an activity is represented by a set of pose and velocity vectors for the major body parts (hands, legs, and torso). A sequence-based voting approach is used to recognize activity invariant to the activity speed [192]. A Human Body Part Decomposition (HBPD) technique is used in the spatiotemporal analysis of the deforming apparent contour of a human moving according to a protocol of movements for the 3-D human body model acquisition from three mutually orthogonal views. In addition, the process of simultaneous 2-D part determination and shape estimation is modularized by employing Supervisory Control Theory of Discrete Event Systems. Finally, a novel algorithm is proposed in which the selectively integrates the apparent contours from three mutually orthogonal viewpoints to obtain a 3-D model of the subject's body parts. However, a hand gesture extraction method is proposed in [193]. The human position and the rising hand gestures are estimated by integrating silhouettes from multiple cameras by a background image subtraction and a frame subtraction. In [194], a human body kinematic acquisition is proposed that extends Shape-From-Silhouette (SFS) from the traditional SFS formulation to apply to dynamic articulated objects. This method recovers shape and motions in two steps: (1) correctly segmenting the silhouettes to each articulated part of the object, (2) estimating the motion of each individual part using the segmented silhouette. An efficient method for detecting and segmenting multiple and partially occluded objects is proposed in [195]. A part hierarchy is maintained, and the segmentation and detection tasks are formulated as binary classification problem. By maximizing the joint likelihood defined based on the part detection responses and the object edges part detection responses are grouped, merged and assigned to multiple object hypotheses.

A human body can be represented by its joints through stick-figure, since it reflects anatomic features of the human. A lot of researches have been performed on this concept. However, it may be hard to obtain the joints directly from an image and usually various assumptions are introduced to simplify the matter. In the work by Lee and Chen [196], the positions of the joints in the image and the 3-D length of each segment are beforehand. Given the 3-D position of the neck, a partial tree is build. At each node one of two solutions for the next joint is possible, since the joint's projection in the image and the 3-D length are known. A path through the tree is equal to one body pose. The tree is pruned using kinematic constraints and the assumption that the subject is walking is posed. In [197], they improve their system by introducing a smooth motion constraint. In the work by Attwood [198], a similar approach is taken, except that he uses three static postures (standing, kneeling, and sitting) instead of a walking assumption to prune the solution space. In [199], the 3-D positions of the joints are estimated using markers and stereo. These are compared with 3-D model joints using a graph-based scheme to find the correct pose of the subject. Some other works addressing the joint-based stick representation are mentioned in [200-207].

However, automatic human body extraction has much impact on human motion acquisition. The modeling of human body is performed from the images captured from single or multiple cameras. As human body is rigid one, the full-body model comprises the model of each articulated parts. The motion analysis represents how the articulated parts change their state in response to a particular motion. In [208], a monocular camera-based automatic human body modeling technique is proposed. It starts by finding shoulder and hip after locating the head based on curvature. Multiple frames' information is integrated to identify the extremities as hands and feet. Moreover, connectivity energy is employed to locate the elbow and knee joint points. Finally, a complete human model is effectively built for motion recognition and analysis purpose. Another articulated human body model acquisition method using multiple cameras is introduced in [209]. The human body is modeled as a set of tapered super-quadrics connected in articulated structure and the parameter of the model is automatically estimated using the video sequences obtained from multiple calibrated cameras. In [210], a knowledge-based framework is presented to capture the meta-representations for real-life video with human walkers. The system models the human body as an articulated object where each body part's motion, shape and texture is extracted, and treats human walking as cyclic activity with highly correlated temporal patterns. An articulated motion modeling strategy for activity analysis is proposed in [211]. It combines robust optical flow estimation, RANSAC, and region segmentation using color and Gaussian shape priors. The combination results in a system that can robustly estimate and segment multiple motions. A view-insensitive 2-D-Model generation method using a high perspective camera is proposed in [21]. It focuses on using the 3-D principal directions of man-made environments, and also the direction of motion to transform both 2-D-Model and input images to a common frontal view before fitting process. Later, inverse transformation is performed on the resulting human features obtaining a segmented silhouette and pose estimation in the original image. A vision system for labeling the outline of a moving human body, named as First Sight, is proposed in [213]. Two processes constitute the system. The first process extracts the outline of a moving human object, whereas the second process builds a human body model, interprets the outline and produces a labeled 2-D human body stick figure for each frame. The proposed body model has two parts: the basic model that uses cylinders in modeling the body, and the extended model which models the body outline with self-occlusion and with varying changes in appearance. A novel human capture scheme employing visual hull and Iterative Closest Point (ICP) is proposed in [214]. In this method, the 3-D representation is achieved through visual hull from multiple 2-D cameras and the 3-D articulated body is tracked in the 3-D representations using an articulated body from a repository of subject-specific articulated bodies that would match the subject which is the closest based on a volume and height evaluation. An improvement of this method proposed in [215] maintains anatomical consistency by enforcing rotational and translational joint range of motion constraints for each specific joint.

However, *kinematic models* describe the human body as a tree, consisting of segments that are linked by joints. Every joint contains a number of degrees of freedom (DOF), indicating in how many directions the joint can move. All DOF in the body model together form the pose representation. These models can be described in either 2-D or 3-D. 2-D models are suitable for motion parallel to the image plane. The works in [49] and [225] use a so-called Cardboard model in which the limbs are modeled as planar patches. Each segment has 7 parameters that allow it to rotate and scale according to the 3-D motion. In [226], an extra patch width parameter was added to account for scaling during in-plane motion (See **Fig. 4.3(a)**). In [227-228], a human body is described by a 2-D scaled prismatic model [231]. These models have fewer parameters and enforce 2-D constraints on figure motion that are consistent with an underlying 3-D kinematic model. But despite their success in capturing front-parallel human movement, the inability to encode joint angle limits and self-intersection constraints renders 2-D models unsuitable for tracking more complex movement. 3-D models allow a maximum of three (orthogonal) rotations per joint. For each of the rotations individually, kinematic constraints can be imposed.

Instead of segments that are linked with zero-displacement, [230] models the connection by constraints on the limb ends. In a similar fashion, Sigal et al. [232] model the relationships between body parts as conditional probability distributions. Bregler et al. [229] introduce a twist motion model and exponential maps which simplify the relation between image motion and model motion. The kinematic DOF can be recovered robustly by solving simple linear systems under scaled orthogonal projection.

Due to the restriction of 2-D models in terms of camera's angle, many researchers have been trying to depict the geometric structure of a human body in more detail using some 3-D models such as elliptical cylinders, cones, spheres, etc. [216-224]. The more complex 3-D volumetric models, the better results may be expected, but they require more parameters and lead to more expensive computation during the matching process (See **Fig. 4.3(b)**). An early work by Rohr [216] made use of 14 elliptical cylinders to model a human body in 3-D volumes. The origin of the coordinate system was fixed at the center of torso. Eigenvector line fitting was applied to outline the human image, and then the 2-D projections were fit to the 3-D human model using a similar distance measure. Aiming at generating 3-D description of people by modeling, Wachter and Nagel [217] recently attempted to establish the correspondence between a 3-D body model of connected elliptical cones and a real image sequence. Based on the iterative extended Kalman filtering, incorporating information of both edge and region to determine the degrees of freedom of joints and orientations to the camera, they obtained the qualitative description of human motion in monocular image sequences.

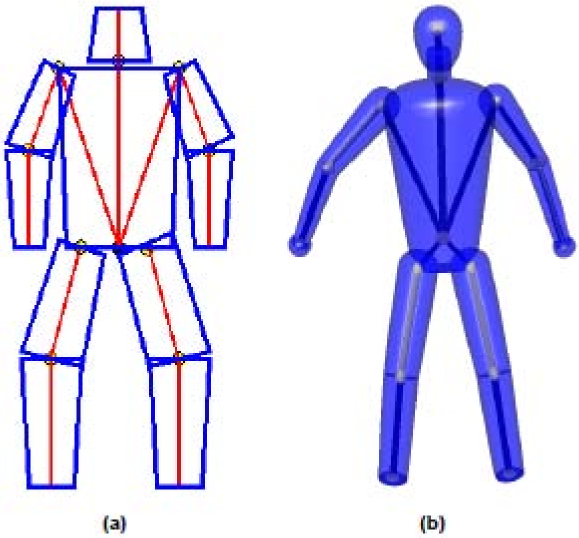


Fig. 4.3 Human shape models. (a) 2-D model [226], (b) 3-D volumetric model consisting of super-quadrics [224].

4.2.2 Pose analysis

A number of researches have been performed to estimate pose rather than the body-parts segmentation for human motion analysis. Basically, the concept of pose or posture is more or less related to silhouette. Each silhouette image within a motion sequence represents a pose, which is further required to be analyzed to extract application-specific information. In [189], a pose tracking system, known as Silhouette Lookup (SiLo) tracker, is proposed that uses articulated pose and it is based upon looking up observed silhouettes in a collection of known poses. This technique exploits temporal continuity to choose the best hypothesis among multiple candidate poses at each frame via a Markov chain formulation. Relieved of the burden of finding the perfect match, simple yet effective metrics make feasible rapid retrieval of candidate silhouettes. Smoothing and optimization based upon polynomial splines is performed to create a plausible human motion. A real-time silhouette extraction technique is proposed that estimates human postures for analyzing a sequence of human posture images [233]. It has several processes except the silhouette extraction by YIQ color values: obtaining orientation of the upper body, contour image, tips of feet and hands, top of head and estimating major joint positions. A novel method to estimate the body configuration and pose in a 3-D space is introduced in [234]. Initially, a set of boundary sample points from the image are obtained. Later, the 2-D image positions of 14 keypoints (wrists, elbows, shoulder, hips, knees, ankles, head, and waist) are estimated on the image by deformable matching to a set of stored exemplars that have hand-labeled keypoint locations. These estimated keypoints can then be used to construct an estimate of the 3-D body configuration in the test image. Gavrila and Davis [37] take a top–down approach with search-space decomposition for pose estimation. Poses are estimated in a hierarchical coarse-to-fine strategy, estimating the torso and head first and then working down the limbs. The initial pose prediction is based on constant joint angle acceleration. An analysis-by-synthesis approach is applied in a discrete fashion, resulting in a limited number of possible solutions per joint. Drummond and Cipolla [42] introduce constraints between linked body parts in the kinematic chain. This allows lower parts to effect parts higher in the chain. A pose is described by the rigid displacement for each body part. This yields an over-parameterized system which is solved in a weighted least-squares framework.

Recent work has focused on the recovery of human poses in cluttered scenes. The work in [235] adopts a three-stage approach, based on [236], to subsequently find human bodies, namely, their 2-D body part locations and a 3-D pose estimate. Sminchisescu et al. [237] learn top–down and bottom–up functions in alternate steps. The bottom–up process is tuned using samples from the top–down process, which is optimized to produce estimates that are close to those predicted by the bottom–up process. The processes are guaranteed to converge to equilibrium. Another silhouette based human motion reconstruction technique is proposed in [238]. This technique reconstructs unconstrained motions captured from multiple cameras using volume intersection. Motion data are acquired by fitting a model of the performer to the reconstructed volume.

4.3 Extension to Motion Modeling

Within the context of this thesis, the goal of pose acquisition is to apply it in motion modeling. Motion modeling refers to the parameterized modeling of the human body performing the motions within successive motion frames. Rather than using a single pose or posture, we are to extend it in temporal direction. Thus motion modeling encompasses both the pose modeling and the aggregation of the models for motion analysis. In some earlier work, such concept is effectively focused.

In the work of [216], a 3-D model based approach is proposed to interpret the movements of articulated bodies, e.g., pedestrians walking movements are recognized from the data obtained from medical motion studies. A 3-D cylindrical model is built and model parameters are estimated in consecutive images by applying Kalman filter (See **Fig. 4.4(b)**). A fast skeletonization technique, named *star skeleton*, is proposed in [73] that uses HMM-based methodology for action recognition. Here, pose-wise start-skeletons are generated over time. In the training phase, the model parameters of the HMM of each category are optimized so as to best describe the training symbol sequences. For human action recognition, the model which best matches the observed symbol sequence is selected as the recognized category (See **Fig. 4.4(a)**). A shape and stick figure based model of human body leading to motion analysis is introduced in [212].

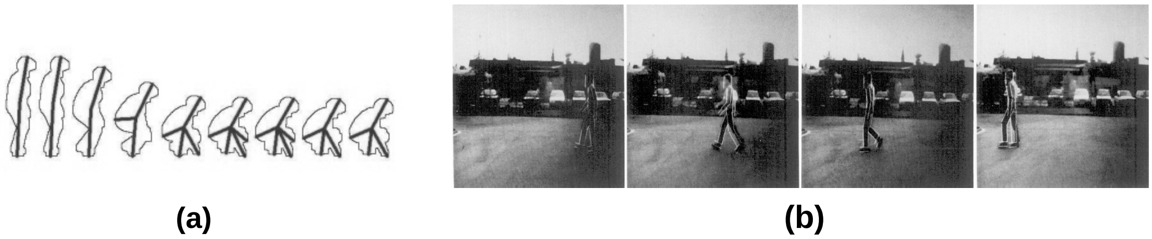


Fig. 4.4 Human motion modeling: (a) star skeleton in *pickup* sequence [73], (b) cylindrical model [216].

However, a human body tracking mechanism for human in motion is also adopted in [188]. An activity manifold learning with poses is proposed in [128]. A 3-D model based estimation of human motion is adopted in [239] that uses multiple cameras. In [240], a view-based method for the recognition of human action/activity is introduced where an activity is represented by a set of poses and velocity vectors for the major body parts (hands, legs, and torso), and stored in a set of multidimensional hash tables. The recognition of a sequence of body pose vectors is done by a method of indexing and sequencing with only a few pose vectors. Many other pose acquisition methods are available that focus on human motion analysis. Yet, this field of research within the domain of motion analysis is very immature. Further investigations are needed to make this domain rich with many research outcomes concerning the computational complexity, viewpoints, preciseness, accuracy, etc. to make the system worthy of practical implementation. With this view, we have proposed a pose-oriented human motion acquisition method by following motion modeling strategy. The goal of our work is to acquire, analyze and recognize different categories of motions by vision-based motion capture. This approach is discussed in Chapter 7.

4.4 Automatic Human Motion Modeling

As the issues of motion modeling introduced in Section 4.3, we propose a human motion modeling strategy based on the acquisition and understanding of limb movements. Analyzing the prior researches, we find those either complex or partially representative modeling of human body. However, we are interested in the modeling which is much simpler, and, at the same time, able to model the human body with maximum representative structure. So, we propose here a semi-3-D modeling of human limb movements, including head, for the task of motion acquisition by a robot. This involves the thorough analysis of a human by using a silhouette image, and a skeletonized model. We analyze frontal- and right side-view of a human body. The system framework is depicted in **Fig. 4.5**. Various tasks of the modeling system are described below.

4.4.1 Silhouette extraction

At first, each image frame comprising a motion is extracted. A *background subtraction* procedure is applied on the image. In our work, we use the static background which is subtracted from the image having the actor in it. After successful background subtraction from the background image, the outline of a human body is extracted in the form of *silhouette*.

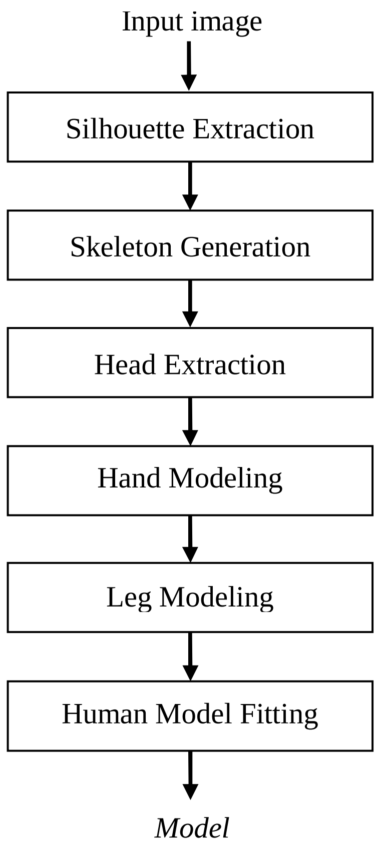


Fig. 4.5 System framework for motion modeling

4.4.2 Generation of a skeleton

A raw skeleton is constructed by *thinning*. Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It can be used for several applications, but is particularly useful for skeletonization. It is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images, and produces another binary image as output. In the following, we describe our adopted thinning algorithm in brief [41].

- The thinning of a set A by structuring element B, denoted $A \otimes B$, can be defined in terms of the hit-or-miss transform

$$\begin{aligned} A \otimes B &= A - A \circledast B \\ &= A \cap (A \circledast B)^c \end{aligned}$$

- The usual process is to thin A using a sequence of structuring elements $B^1, \ldots B^n$
- In other words, A is thinned by successive passes of structuring elements B^1, $B^2, \ldots$
- The entire process is repeated until no further change occurs

An example of the thinning algorithm is shown in **Fig. 4.6**.

However, the procedure from silhouette extraction to skeleton generation is shown in **Fig. 4.7**. The raw skeleton is modeled according to our proposed *human model* shown in **Fig. 4.8**.

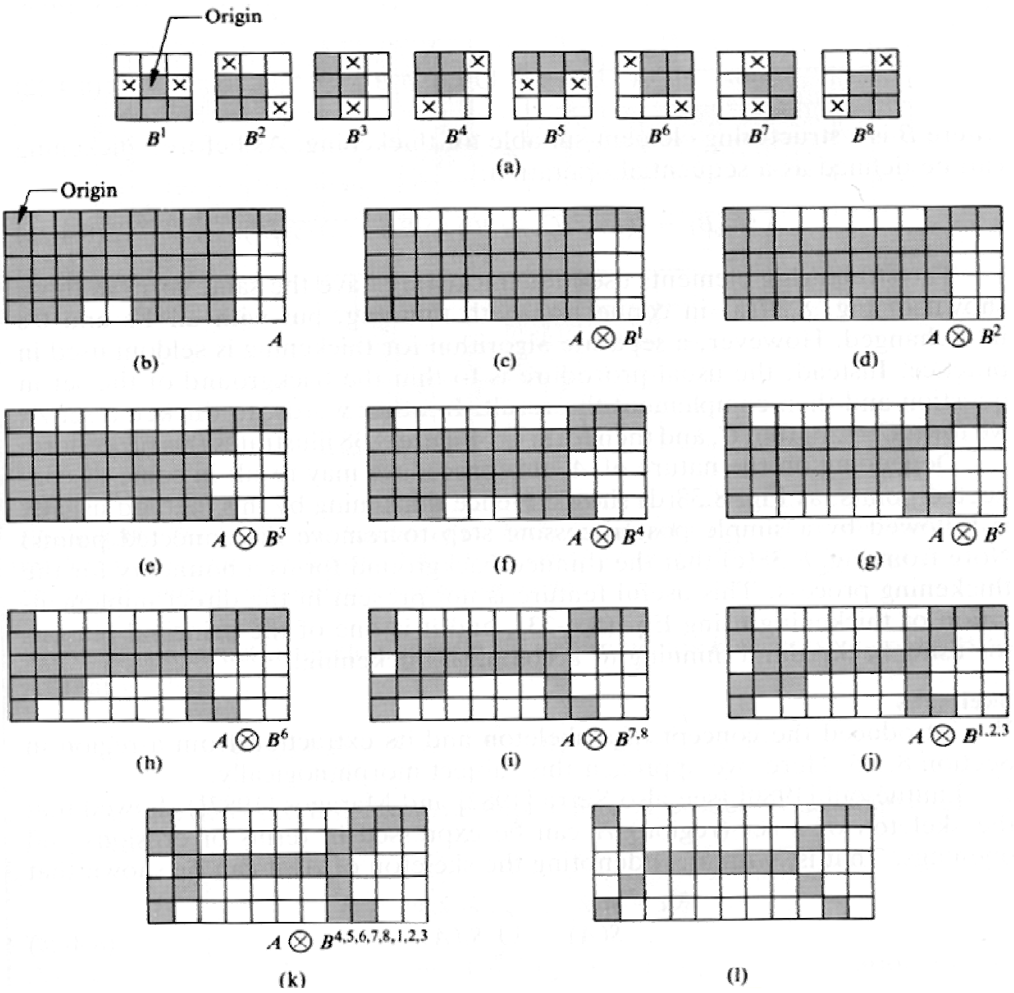


Fig. 4.6 An example of the implementation of the *thinning algorithm* on an image *A*.

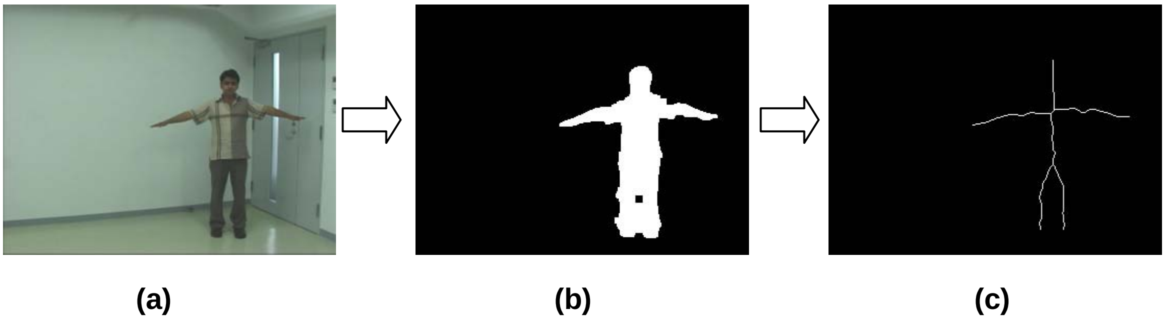


Fig. 4.7 Skeleton construction: (a) an original frame, (b) silhouette image, (c) constructed skeleton.

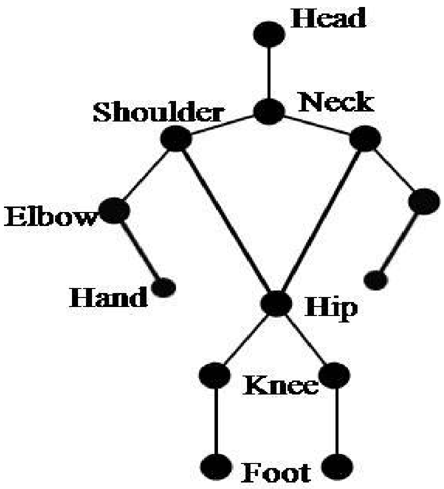


Fig. 4.8 Proposed *human model*.

4.4.3 Head and neck positioning

A template of *head* is constructed by estimating neck from top of the human body. The template is matched to locate the head position inside each image frame. The head is to be tracked in successive frames.

Moreover, the neck is also obtained by vertical histogramming as shown in **Fig. 4.9**. The neck is relocated on the human body skeleton.

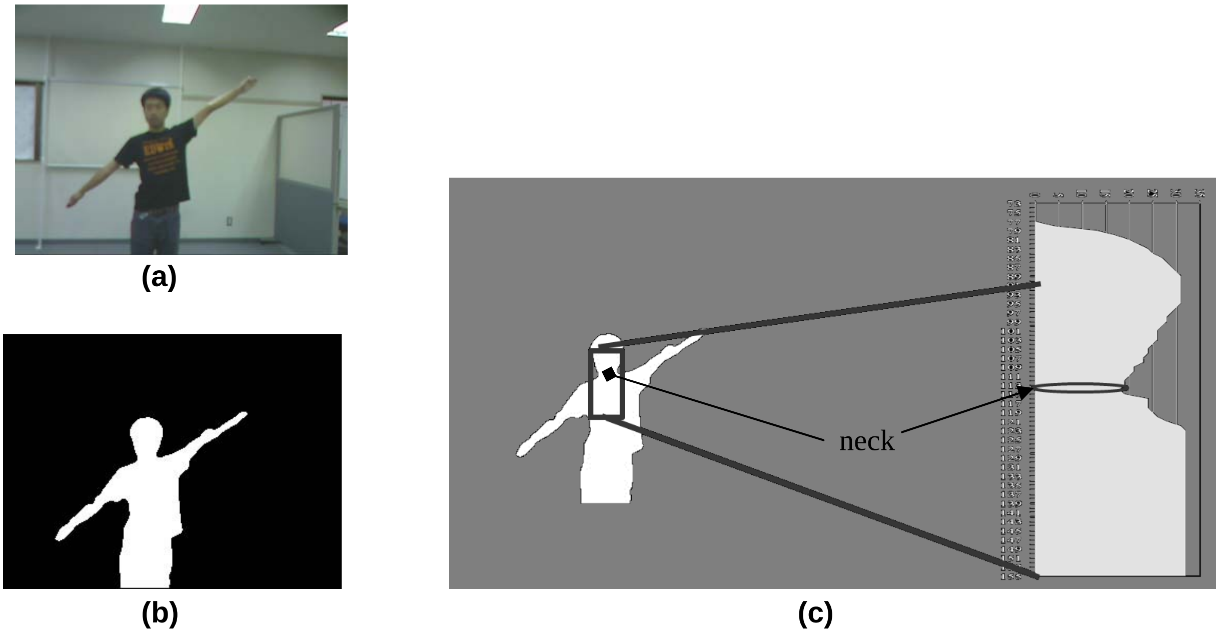


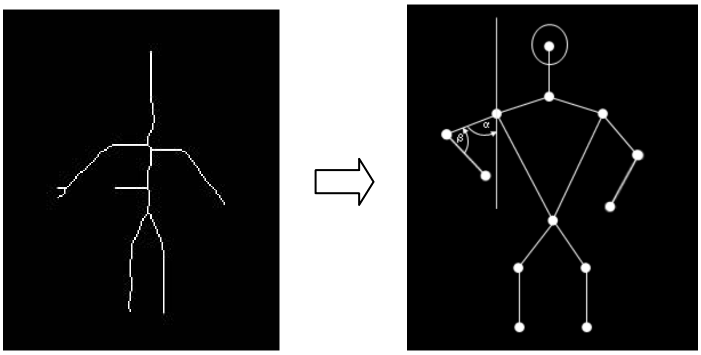
Fig. 4.9 Neck positioning. (a) original frame, (b) silhouette image, (c) neck positioning by vertical histogramming

4.4.4 Hand modeling

The shoulder points of both sides are necessary to model hands from frontal-view. Shoulder points are located on the skeleton by histogram tracing from neck to an assumed hip. The left and right shoulder points are assumed as the *starting points* of left and right hands, respectively. Efficient DFS (Depth-First-Search) algorithm traverses through the skeleton to obtain the maximum pixel length hand region.

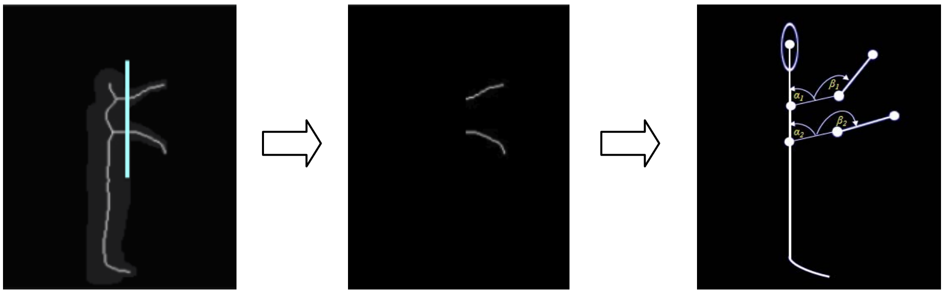
After getting the positioned hand pixels for both hands, they are modeled based on the *elbow* joint. Assuming the shoulder-elbow and elbow-hand regions as rigid, the elbow is positioned based on maximum inclination of region at a point from shoulder to hand. In the modeling process, shoulder-elbow and the elbow-hand joint angles are computed. Hand modeling for frontal view is shown in **Fig. 4.10**.

Moreover, the side-view of the body helps to model hand movements observed from the right side. This makes our hand model a 2-DOF model. Similar to the frontal analysis, the visible portions of hands are extracted and inter-relation of the joints is calculated in terms of joint angles. However, the right- and left-hand models are specified in the modeling. Hand modeling for the right side view is shown in **Fig. 4.11**.



(a) **(b)**

Fig. 4.10 Hand modeling for a frontal view. (a) a hand skeleton, (b) hand modeling



(a) **(b)** **(c)**

Fig. 4.11 Hand modeling for a right side view. (a) isolation process of the hands, (b) isolated hands, (c) hand modeling

A number of assumptions are imposed at the time of hand modeling using both hands' information.

- Label hands as either left or right.
- Mostly frontal view hand position is employed to solve the ambiguity between two hands
- We consider the hands are not at the maximum pick position.
- Intersecting hands are allowed in the computation.
    - Calculate angles for intersecting hands by using DFS-branching
- Limitations:
    - Overlapping or occluded hands
    - Non-extracted hands

4.4.5 Leg modeling

Leg modeling starts with the identification of a *waist* point from a frontal view. Similar to hands, DFS algorithm traverses through the skeleton to obtain the maximum pixel length leg region. The point with the maximum-length branching in the DFS-tree is estimated as a *waist*. The *knee* joint is located based on some predefined assumptions.

Moreover, the side-view of the human body provides information for the modeling of the legs observed from the right side. By positioning the waist on the side-view skeleton, the side-view based legs are modeled. The interrelation of the leg joints is described numerically in terms of angles between waist-knee and knee-foot. The positional information of the both legs is also specified. The leg modeling strategy is depicted in **Fig. 4.12**. The leg model is represented numerically by angles. Side-view modeling of legs is performed like front-view modeling. Small leg movements are unidentified thorough out the leg modeling. However, there are some assumptions imposed on leg modeling. Those are mentioned as follows.

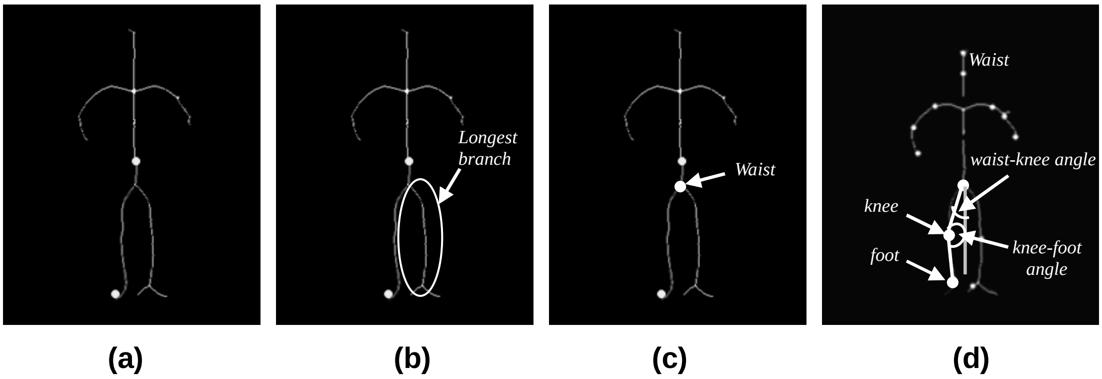


Fig. 4.12 Leg modeling. (a) Leg pixels are traversed, (b) the longest branch obtained by DFS, (c) *waist* is located, (d) *knee* and *foot* are positioned, and joint angles are shown.

- Both leg pixels are obtained
    - In *waist location* stage
    - Colorize leg pixels with two different colors
- Locate *knee* points based on:
    - *knee* height = $\frac{\text{Total length of leg pixels}}{2}$
- Locate *foot* points
    - Denoted by the *end* point in the DFS traversal

4.5 Experiments

We have modeled different sorts of human movements using our proposed modeling strategy in order to evaluate the precision of the modeling. The modeling rate is calculated by matching against the joint position and joint angles on the original human body and the derived model (See **Fig. 4.13**). We have experimented on 1129 motion frames consisting of four different body and head movements performed by two actors. We have obtained 93% modeling rate based on the strategy depicted in **Fig. 4.13**. **TABLE 4.1** summarizes the modeling rate and error rate for all the joint angles. However, the right-view movements of the legs are not taken into account for experimentation. **Figure 4.14** shows an example of modeling for frontal view and right side view.

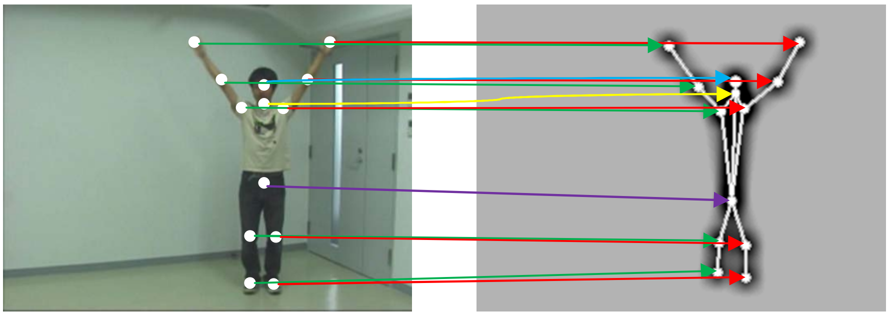
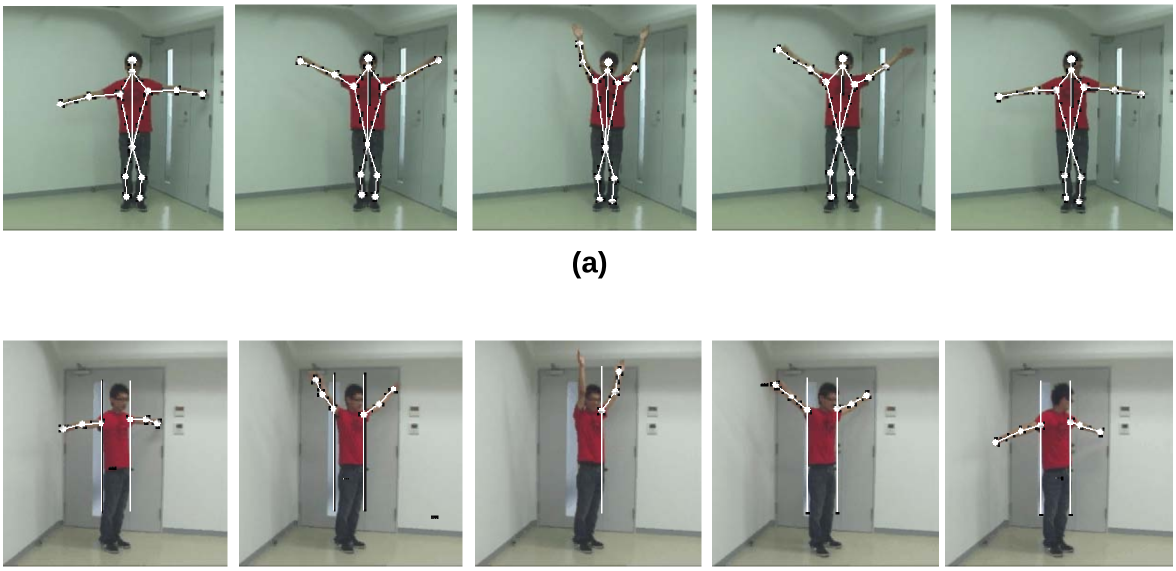


Fig. 4.13 Joint-by-joint matching scheme for evaluation

TABLE 4.1
Modeling rate of the proposed model

Joints	Modeling Rate (%)	Error Rate (%)
Head-Neck	93%	7%
Shoulder-Elbow(Front)	93%	7%
Elbow-Hand(Front)	93%	7%
Shoulder-Elbow(Side)	86%	14%
Elbow-Hand(Side)	86%	14%
Hip-Knee	99%	1%
Knee-Foot	99%	1%
Average	93%	7%



(a)

(b)

Fig. 4.14 The result of modeling of a hand waving sequence. (a) Frontal view, (b) right-side view.

4.6 Discussion

A semi-3-D modeling system is proposed that can be used for the acquisition of body and limb movements by a humanoid robot. The acquired motions are supposed to be recognized by the robot or an intelligent system in real-life scenario. We have developed the system for acquisition by modeling the movements of the body parts. We use frontal- and side-view capture of motions. As two cameras are used, the 2-DOF movements are corresponded. However, there are some limitations in the modeling in terms of modeling errors arising from generation of silhouette image, hands' overlapping and non-extracted hand regions. Modeling of torso region corresponding to the deformation occurred in various motions is our future consideration.

4.7 Summary

In this chapter, we discuss a number of related issues concerning human pose acquisition. We present a detailed survey on human pose acquisition. We also present some recent works that emphasize the concept of motion modeling using the poses. We present our proposed human motion modeling strategy, and show the performance of the strategy by experiments. This motion modeling concept has much potential for the acquisition of rather complex motions in future motion recognition applications.

Chapter 5

Human Motion Recognition Employing Structured Motion Database

5

Human Motion Recognition Employing Structured Motion Database

5.1 Introduction

In recent years, with the increasing interest in the field of computer vision and image processing, the development of an efficient human motion recognition system has become an indispensable part of the intelligent systems and Human-Computer Interaction (HCI) systems. Developing a reliable intelligent system that is capable of manifesting what a human is performing in a scene is a very much challenging task. This sort of system has wide variety of applications, especially in surveillance, virtual and augmented reality, animation, intelligent robots, diagnostics of orthopedic patients in clinics and hospitals, supporting aged people in rehabilitation centers, performance evaluation and training of the athletes in sports, and so on. Due to diverse applications for such systems, it requires robustness as well as accuracy. Moreover, the system is subjected to be in use in real-time; this urges for relatively fast response of the system. This attribute constrains on the system development that the system should have the capability of high-speed recognition. Therefore, having a number of aspects for the recognition system, the literature related to the problem of recovering and recognizing human motions in a scene is intensive [28, 128, 129, 131, 134, 241, 242]. However, we focus on the methods addressing the specific problem of recognizing human motions from image sequences without using markers, tracking devices, or special body suits. Based on whether or not a priori knowledge about the object's shape is required, the methods for human motion analysis can be classified broadly into two categories: model-based and appearance-based approaches [115, 243]. However, other forms of categories are also available [67]. Both the approaches have their own advantages and disadvantages. Appearance-based approaches are applicable to diverse situations, since they do not require a specific object model. Those methods are sensitive to noise in general, because they lack any mechanism to distinguish noise from signal in visual input. Appearance-based approaches build a body representation in a bottom-up fashion by first detecting appropriate features in an image, whereas in model-based approaches, the fitting process involves either an optimization scheme such as the least square method [244] or a stochastic sampling scheme such as the particle filtering method [245]. In practice, the degree of detail in the body representation (e.g., head, torso, limbs, etc.) is not a mandatory requirement for recognition purpose; rather motion-specific representation composed of adequate features to represent each motion uniquely is enough to accomplish the task of recognition. Therefore, advanced image processing techniques are being comprehensively investigated in search of effective representation of a motion. Standard techniques for the motion representation include the ones based on Motion History Image (MHI) and its variants [70, 129, 115, 81]. Motion history-based representations include not only the movement of a body itself but also the change of position of a person in a scene. However, object's silhouette information alone can be used as an input for a recognition system. Wang and Suter [131] used silhouettes as the input to their recognition system. Elgammal and Lee [128] also used silhouettes without motion history. Moreover, another motion recognition approach was also proposed which considered multi-view motion representation and recognition [83]. In this approach, the motion postures are iteratively transformed into a single eXclusive-OR (XOR) template image for the task of registration and recognition.

However, in order to deal with the high dimensional complex information extracted from human motion, it is necessary to find reduced representation of the motion while maintaining sufficient discriminating data for performing the recognition. To accomplish these goals, existing researches have used simple data reduction techniques such as Principal Component Analysis (PCA) [129], Eigenspace technique [111], Locality Preserving Projections (LPP) [131, 246], etc. Moreover, a statistical matching method using Hidden Markov Model (HMM) that allows for a principled probabilistic modeling of the temporal sequential information is also adopted in various works [128,162]. An alternative approach for matching the data sequences using Dynamic Time Warping (DTW) is also employed in recent works [134, 247]. The recognition methods mentioned in the aforesaid literature commonly use single or frontal cameras to capture motions in the case of view-based motion analysis. But an intelligent system may have the flexibility of orientation-independent recognition. Therefore, it is necessary to handle the orientation-specific data in an effective way.

5.2 Contribution of the Work

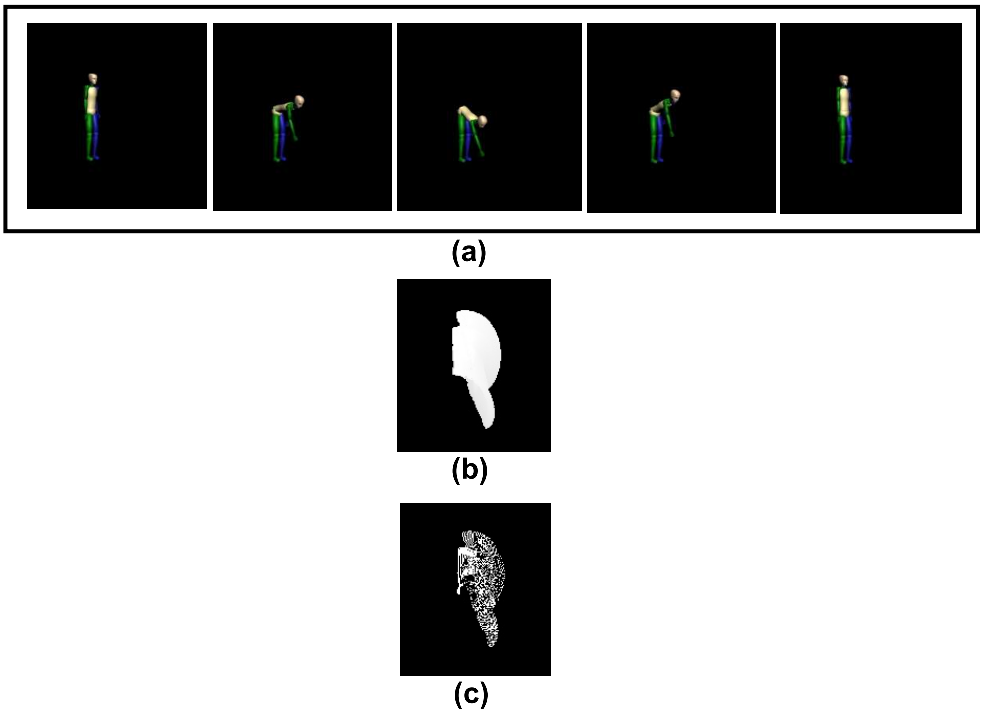
In this chapter, we propose a human motion recognition approach capable of distinguishing the orientation-specific motions effectively by means of a structured data organization. The novelty of the proposed approach lies in improving the precision and robustness of the recognition by making use of the directional organization of the motion database (referred to as “directional motion sub-database”) corresponding to the varying viewpoints and the nearest index searching strategy with the database. The directional candidates obtained from the directional motion sub-databases play an effective role to find similar motions. Unlike earlier researches, we propose an adoption of multi-viewpoint concept without integrating the orientation-wise information, and thus reducing the load of detailed analysis.

The aforementioned structured motion database system faces a significant problem, called *a boundary problem* or *a nearest neighbor searching problem*, which might degrade the performance of the overall recognition system. The boundary problem, as the name implies, is the misclassification of motion points residing near the boundary of the search space or the miss-selection of the candidate motion points within the space. The novelty of this work lies in resolving the boundary problem for the improvement of a human motion recognition system. The current work makes the following contributions: (1) We present a novel approach for overcoming the existing problem within the structured motion database. (2) The misrecognition is highly recovered by increasing the extent of searching. (3) We represent experimental results showing the significant improvement by adopting the proposed approach.

However, in the above system, an unknown motion is exhaustively searched over the stored data without any prior cue indicating the possible direction of motion capture to which it may belong. This often leads to the redundant searching by including the searching spaces where possibly it does not reside. Therefore, the time requirement as well as the recognition accuracy is subjected to be improved to make the system more efficient. Therefore, we propose a structured database based direction-oriented motion capture by pre-estimating the possible orientations of an unknown motion. Thus we are able to eliminate unnecessary search load and make the system faster so that it can be implemented in online applications.

5.3 Motion Representation

We use the concept of motion representation to generate a form to characterize a motion for the computer to understand and to use the motion for recognition. This is a very crucial task. However, as mentioned earlier, we have used two standard representations of motions: MHI, and Exclusive-OR image. We generate the motion images (or feature images) from preprocessed motion frames (See **Fig. 5.1**). Details of these representations are described in Section 2.2 and 2.3.



(a)

(b)

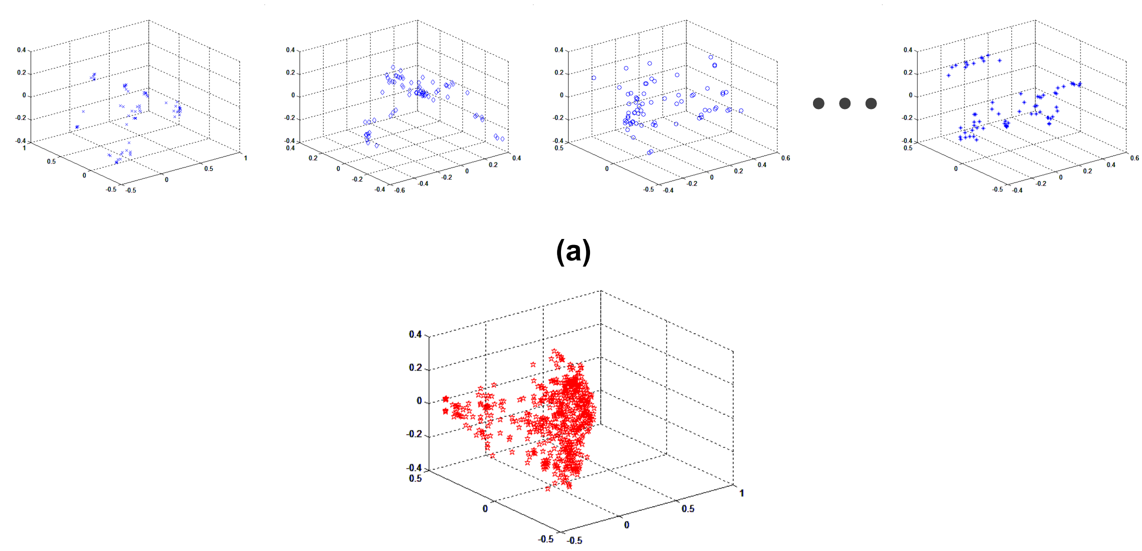
(c)

Fig. 5.1 *Pickup* motion and corresponding motion representations: (a) Some frames representing the motion, (b) MHI, (c) XOR image.

5.4 Construction of Directional Feature Spaces

A feature is a significant attribute for characterizing a motion and determining the similarity of it among a number of training motions [111]. The aggregation of the feature should have the quality of distinguishing unique motions. Based on this concept, a feature space is constructed in the form of an eigenspace, where the eigenvectors corresponding to the prominent eigenvalues construct an eigenspace of projected motion data. An eigenspace is a high-dimensional feature space that represents the proximity among the set of data. It is a modified form of Karhunen-Loeve Transform (KLT) that is used to derive the relationship among different random variables. In practice, a large set of learning motions is required to be projected onto the eigenspace by finding prominent eigenvectors. In Section 3.4, we have presented detailed description of the construction of a feature space.

We compute the eigenspace consisting of different actions performed by different actors. For each camera viewpoint, separate eigenspaces are created, which we call *directional eigenspaces* or the first hierarchy of eigenspaces, by projecting corresponding motions onto those using Eq. (3.10), which can be used to characterize motions as well as camera directions (See **Fig. 5.2(a)**). For constructing an image set, all the images of the set should be same size. In our work, feature images generated in the form of MHI or XOR image can be successfully characterized by 32X32 pixels. Correspondingly, equal number of sub-databases are also built and maintained. Each sub-database returns a single *candidate* motion for a motion query. Moreover, a *global eigenspace* or the second hierarchy of eigenspaces containing all the learning motions is also built, and maintained to decide the most similar one among several candidate motions (See Fig. 5.2(b)). The global eigenspace is constructed in a similar fashion as directional eigenspaces.



(a)

(b)

Fig. 5.2 Hierarchy of eigenspaces: (a) *Directional eigenspaces*, (b) *global eigenspace*. First three dimensions of these eigenspaces are shown for visualization.

5.5 Development of a Motion Database

A database, solely, relies upon the organization of data within the computer memory. The most common database organization is linear, *i.e.*, the data arranged in the order of its input. At the time of query, the database performs sequential blind searching among the data. In order to overcome the problem of sequentiality in the query, many researchers have been comprehensively involved in the development of a suitable database that is organized in a non-sequential manner and also capable of quick and successful retrieval. Moreover, due to the increased number of motion archives, maintenance of the database organization is also drawing much attention. As a result, the B-tree [172] database structure is adopted in our research as a structured non-sequential motion database.

5.5.1 Indexing of motion data

According to the concept of structured database as mentioned in Section 3.5.1, the motions which are projected onto the eigenspaces are required to be indexed (i.e., generating *an index*) into a numeric format for the flexibility of storage. The dimension of an eigenspace is taken as an important cue in space partitioning. The eigenspace is uniformly divided into several divisions. Each eigen-axis $\mathbf{e}_k$ (k = 1,2,..,K) is divided into S (S >1; *integer*) sections leading to S^k hypercubes with equal edge length of L along each eigen-axis. Each hypercube is referred to as a bin in spatial term and an index in numeric term (See **Fig. 5.3**). In this thesis, the edge length L of the hypercube is termed as *bin length*. Each motion point, represented as a bin or index, is assigned a digit from 0 to S-1 along each eigen-axis. Therefore, an index becomes a K-digit S-nary number (See **Fig. 5.4**). However, in the case of no division of the space, in fact, no database system exists; it is rather sequential storage of the motion data. Therefore, the structured form of the database is preferable to realize a profound database system.

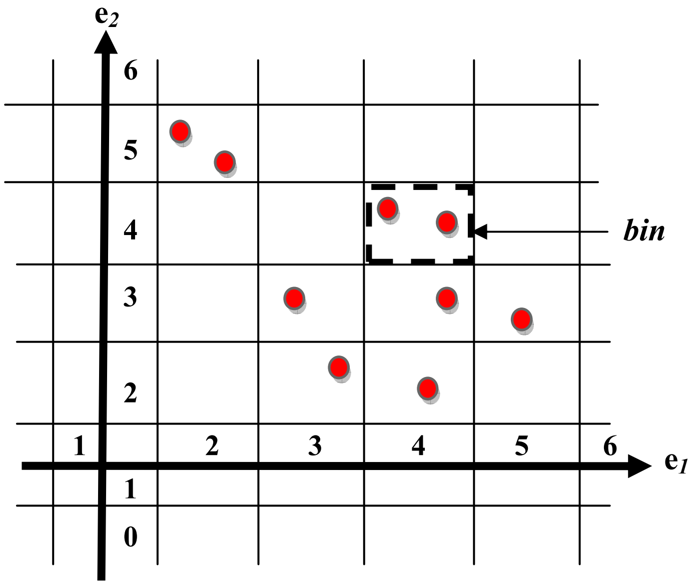


Fig. 5.3 Sectioning of each eigen-axis for indexing within an eigenspace

Digit-(K-1)	....	Digit-4	Digit-3	Digit-2	Digit-1	Digit-0
(0~S-1)	....	(0~S-1)	(0~S-1)	(0~S-1)	(0~S-1)	(0~S-1)

Fig. 5.4 A format of a K-digit S-nary index

5.5.2 Directional organization of structured database

In the case of the motions with several orientations, the motions can be grouped into several motions sets based on the orientation. The steps for constructing a directionally organized structured database are as follows:

a. Capture the training motions having c (c >1; *integer*) orientations by maintaining motion synchronization.
b. Create motion sets M_i (i=1,2,…,c) based on the orientation.
c. Construct eigenspaces ES_i corresponding to each motion set M_i using the scheme described in Section 3.4.2.
d. Construct the structured B-tree sub-database BSB_i corresponding to each eigenspace ES_i taking the division parameter S as described in Section 5.5.1.
e. Combine all the sub-databases to develop directionally organized database.

5.5.3 Robustness of the directional organization

The B-tree structured database maintains the ordered arrangement of data within the tree structure [83, 172, 249, 178]. Each index, as generated in Section 5.3.1, is also assigned a decimal value based on the radix of the index. Depending upon the assigned value, the indexes are stored in an orderly way within the database. The B-tree retrieval algorithm is then applied to retrieve the matched index, or the most appropriate position if matching fails. However, the decimal value does not efficiently represent the neighboring indexes of an index corresponding to its co-ordinate values within original feature space. Moreover, till now there is no standard algorithm for the indexing strategy to select the nearest neighboring point within the space when the exact match of an index is not encountered. Our adopted approach is to calculate the digit-wise *Sum of Squared Difference* (SSD) between the consecutively stored indexes within the B-tree. However, it is not the exact measure, rather an approximation, to select the nearest index within the space. The nearest index searching algorithm is described below. In the algorithm, *Keys*(..) denote the pointer to the successor (or children) within the tree structure.

<u>Algorithm 5.1</u>: *Nearest neighbor search*

```
1. For query index $y$, $y>x_i$ and $p_{i-1}$=NULL
   - Calculate $MinDist(MinDist(x_{i-1}, y), MinDist(x_i, y))$
2. For query index $y$, $x_i>y \geq x_i$ $p_i$=NULL
   - Calculate $MinDist(MinDist(x_i, y), MinDist(x_{i+1}, y))$
3. For query index $y$, $y<x_i$ and $p_{i-1} \neq$ NULL
   - Calculate $MinDist(MinDist(x_{i-1}, y), MinDist(x_i, y))$ as MD$_1$
```

```
        - Calculate $\underset{l=1,2,3,...}{MinDist}\,(x_{p_{i-1}^{(l)}}, y)$ as $MD_2$
        - Calculate $MinDist(MD_1, MD_2)$
4. For query index $y$, $x_i > y \geq x_i\ p_i \neq NULL$
        - Calculate $MinDist(MinDist(x_{i+1}, y), MinDist(x_i, y))$ as $MD_1$
        - Calculate $\underset{l=1,2,3,...}{MinDist}\,(x_{p_i^{(l)}}, y)$ as $MD_2$
        - Calculate $MinDist(MD_1, MD_2)$
```

The conventional database organization is represented by high dimensional eigenspace, whereas the directional organization is represented by comparatively low dimensional space due to the splitting up the whole dataset into directionally independent datasets. It is certain that with the increase in number of dimensions, the probability of miss-selection proportionally increases. Thus, the conventional organization exhibits lower possibility to select the exact nearest index than that of the directional organization based on the aforementioned searching algorithm. Therefore, our proposed directional organization theoretically and experimentally proves the robustness of the recognition system.

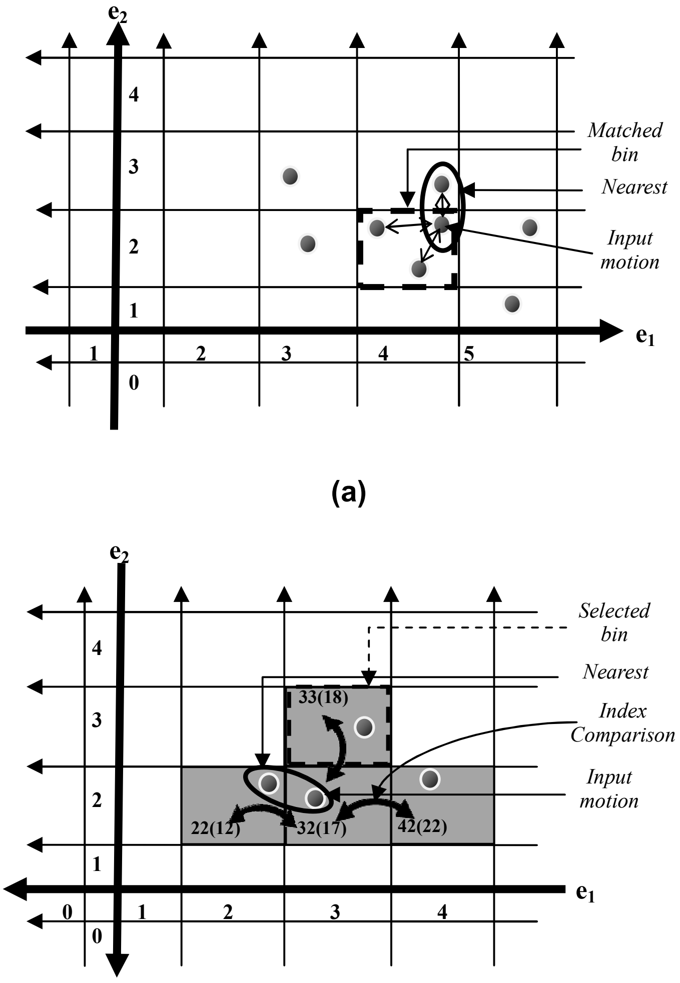
5.5.4 Boundary problem

5.5.4.1 Overview

A boundary problem, in this context, is referred to as possible miss-selection of a candidate bin, composed of motion points, within the feature space. In a general sense, we can mention it as the misclassification of motions. This problem commonly occurs in such kind of space searching due the similarity of a motion with several motions. According to the theory, there are two cases for the occurrence of this problem.

[1] Motion points lying on the edge of a query space imply the inaccurate selection of the bin, since other point in another bin may be near the point (See **Fig. 5.5(a)**).

[2] If the query index does not seem to reside within the database, it is necessary to find the least different index within it. But because of the multidimensionality of the feature space, no algorithm exists (except linear searching) that can do it accurately. Here, we adopted approximation algorithm that uses the decimal value corresponding to the S-nary number to match the nearest bin. The output of the algorithm may lead to misclassification of motions. This phenomenon is illustrated in **Fig. 5.5(b)**.



(a)

(b)

Fig. 5.5 Occurrence of the *boundary problem*: (a) Point inside the oval is the nearest; but other point will be selected, (b) Index comparison of input (32) with other three 22, 33 and 42. The point inside 22 is the nearest but 33 is selected.

5.5.4.2 Boundary problem resolution

In order to deal with the boundary problem, we maintain two sets of query space by shifting the space division to a certain scale. Thus we get two sets of query spaces, namely original and shifted query space set (See **Fig. 5.6**). In Fig. 5.6, we see how the boundary problem is resolved by maintaining two query space sets over the feature space. The proposed selected point searching shows visually better performance than the only using the conventional one. However, corresponding to two sets of query spaces, two parallel motion databases are developed which constitute the whole database system. In our work, we make use of two parallel B-Tree databases to cope with the significant problem.

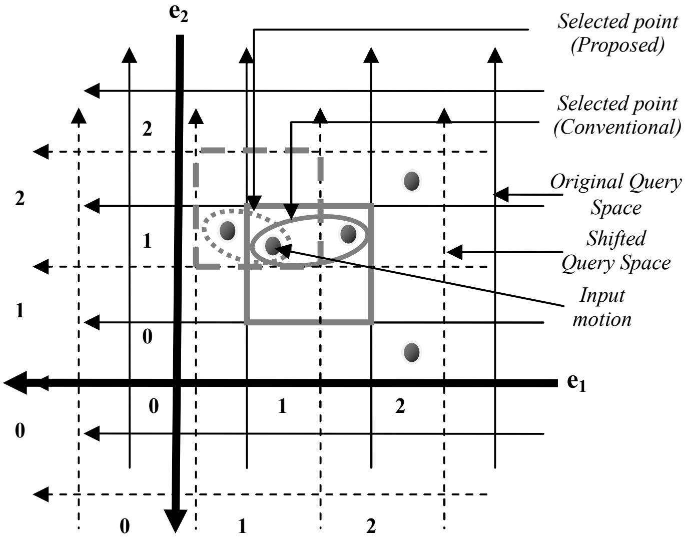


Fig. 5.6 Boundary problem resolution. Two query space sets are used for resolution.

5.5.5 Prior estimation of direction

In order to estimate the possible orientations of an unregistered motion, a global eigenspace containing all the training motions of all the camera views is analysed and information is manipulated. For prior estimation of directions, the projected training motion points within the global eigenspace are clustered based on the orientations from which those are viewed (See **Fig. 5.7**). For *D* number of orientations, *D* clusters are constructed correspondingly within the space. We enclosed those direction-wise motion points by hyperspheres within the space as *clusters*. Thus we obtain *D* hyperspheres in the space, either overlapping or non-overlapping.

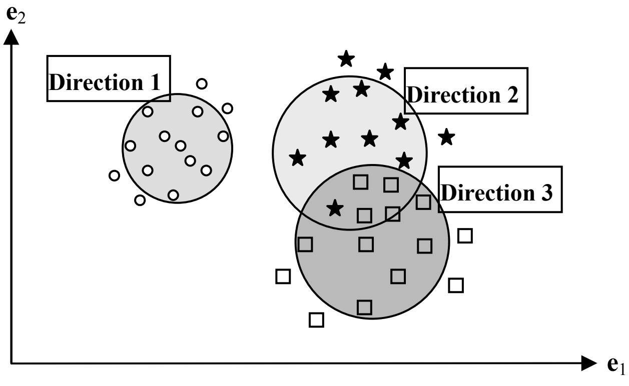


Fig. 5.7 Clustering of motion points on global eigenspace. A two-dimensional eigenspace is shown.

5.5.6 Database search scheme

The structured motion database searching strategy is quite simple, but effective. When an unknown motion comes, it is first represented as a sequence of image frames and is processed for generating motion representation. The MHI or XOR image is generated from the motion frames, its position within the global eigenspace is computed and the clusters it belongs to are also found out and selected as the possible estimation of directions. For example, we obtain d directions among the total of D directions. Then the prior selected d directional eigenspaces are searched by projecting onto the directional eigenspace. An index, representing motion identity within the directional sub-database, is generated from the point corresponding to the unknown motion after its projection onto each eigenspace. For the selected number of camera orientations, the equal number of similar motions is obtained by searching the corresponding B-Tree sub-database as mentioned in Section 5.5.3. The index searching process is repeated for the original and shifted query spaces. Based on the query of the motion, one of the two query space sets is selected adopting the distance measurement function as shown in Eq. (5.1). The closest motion point on a directional eigensapce is referred to as *a candidate motion*. Similarly, for selected number of camera directions, same number of candidate motions is obtained by searching the corresponding query space sets. If the query index resides within the database, it is found by simple index comparison. But for the case of the index not residing within the database, the nearest neighboring point approximation algorithm is employed to find the suitable query index (See Algorithm 5.1) corresponding to each camera direction.

Thus we get several candidate motions from camera-directional eigenspaces. Those candidate motions are projected onto the global eigenspace as $\mathbf{g}_{m_r} (r = 1,2,..,D)$, where D is the number of camera directions. The test motion is projected as $\mathbf{g}_m$ within the global eigenspace. The most similar motion is calculated from within the global eigenspace using the *Euclidian distance function* shown in Eq. (5.1).

$$d_m = \min_r \left\| \mathbf{g}_{m_r} - \mathbf{g}_m \right\| \tag{5.1}$$

5.6 Recognition Strategy

As we mentioned in Section 5.5.6, the database search scheme predominantly serve the purpose of recognition. Practically, after successful search within the structured motion database, we obtain the most similar motion to the input unknown motion. In this section, we summarize and illustrate the recognition strategy as one of the most important operations of the proposed motion recognition system. According to **Fig. 5.8**, the steps required for the task of recognition are stated below:

- The possible directions of the feature image corresponding to input motion are extracted from the clusters within the global eigenspace by direction estimation.
- It is projected onto the selected directional eigenspaces and an index is generated for each of the selected directional eigenspace.
- Each index is searched within each sub-database of B-tree set having original and shifted B-tree. Thus we get the closest point by searching the two. Among *D* number of directions, a subset, i.e., *d* sub-databases corresponding to *d* directional eigenspaces are searched, and we obtain *d* number of candidate motions.
- The candidate motions are projected onto the global eigenspace. The most similar motion is obtained from the global eigenspace by computing Eq. (5.1).

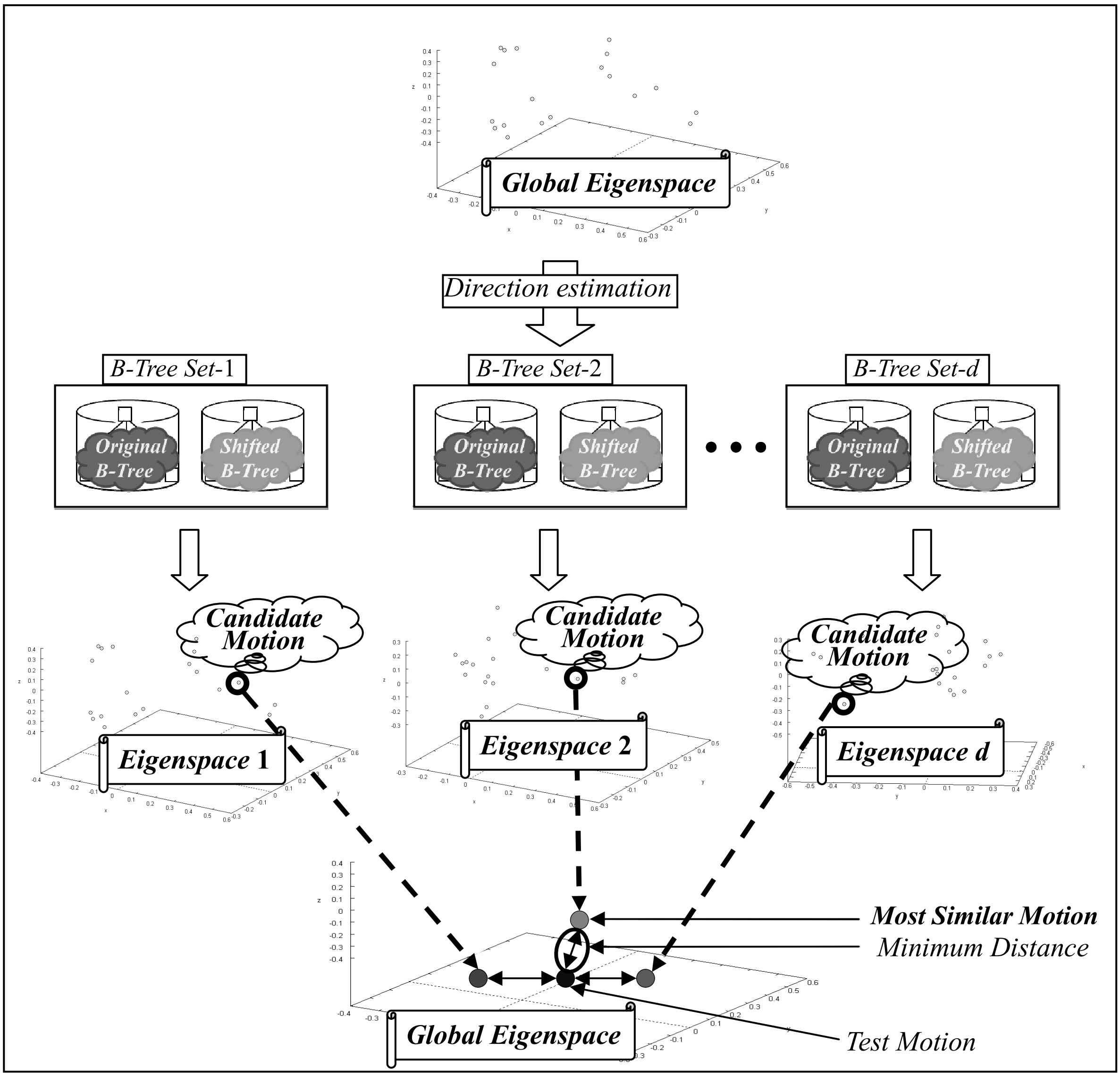


Fig. 5.8 Illustration of the recognition strategy

5.7 Experiments

5.7.1 Experimental setup

The experiments are performed on Avatar dataset with different synthesized human avatars (See **Fig. 5.9**). Each avatar actor performs ten different types of motions, namely *bend* (bending down), *carry* (carrying a box), *jump* (hopping in a place), *pjump* (jumping with two hands up and landing down), *pickup* (picking up something from the ground), *sitdown* (sitting down on a chair), *standup* (standing up from a chair), *stomachache* (touching stomach with pain and crouch), *walk* (walking motion), and *wave2* (waving two hands up in the air). The variations in motion are realized by a subject's height and shape, speed of motion, and field of view. The scene is assumed to be backgroundless. Eight uncalibrated cameras are placed surrounding the avatar at 45 degrees apart, having 0-, 45-, 90-, 135-, 180-, 225-, 270- and 315-degree camera orientations (See **Fig. 5.10**). Human surface is perpendicular to the viewing plane, i.e., parallel to the camera direction at 0-degree camera view and the viewing angles are considered in clockwise direction. **Figure 5.11** and **Fig. 5.12** illustrates different motions, and the corresponding MHIs and XOR images. The motion dataset consists of 800 motion data separated into eight orientations. Among those, 560 motion data are considered as training set, and the rest as testing set. Thus the directional sub-databases consist of 70 motion data corresponding to each orientation, whereas the global eigenspace consists of 560 motion data corresponding to all the orientations. Likewise, the test set consists of 30 motions each (10 motions performed by 3 actors) for every orientation. Each frame extracted from motion is 320X240 pixels that is represented as 32X32 pixels while generating the feature images. Eight directional eigenspaces are constructed corresponding to eight cameras. Thus a total of 16 B-Trees are constructed for eight directional eigenspaces each having two B-Trees for the problem resolution. The recognition system was implemented and tested on a Core2Duo 2.93 GHz-processor 4 GB-RAM computer. The system is tested for MHI and XOR images to illustrate the improvement in performance regardless of the representation adopted.

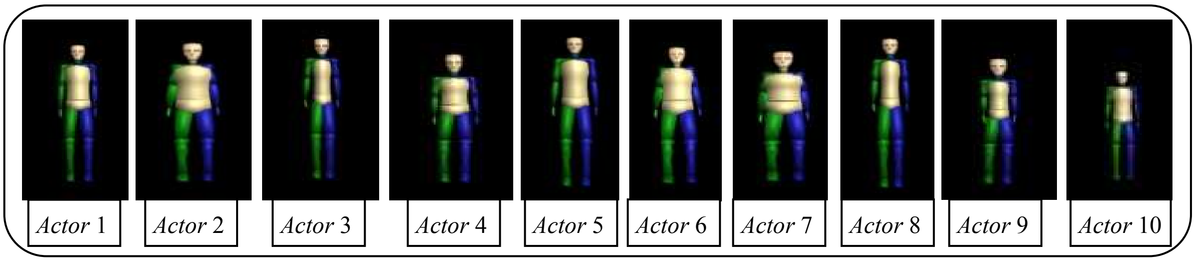
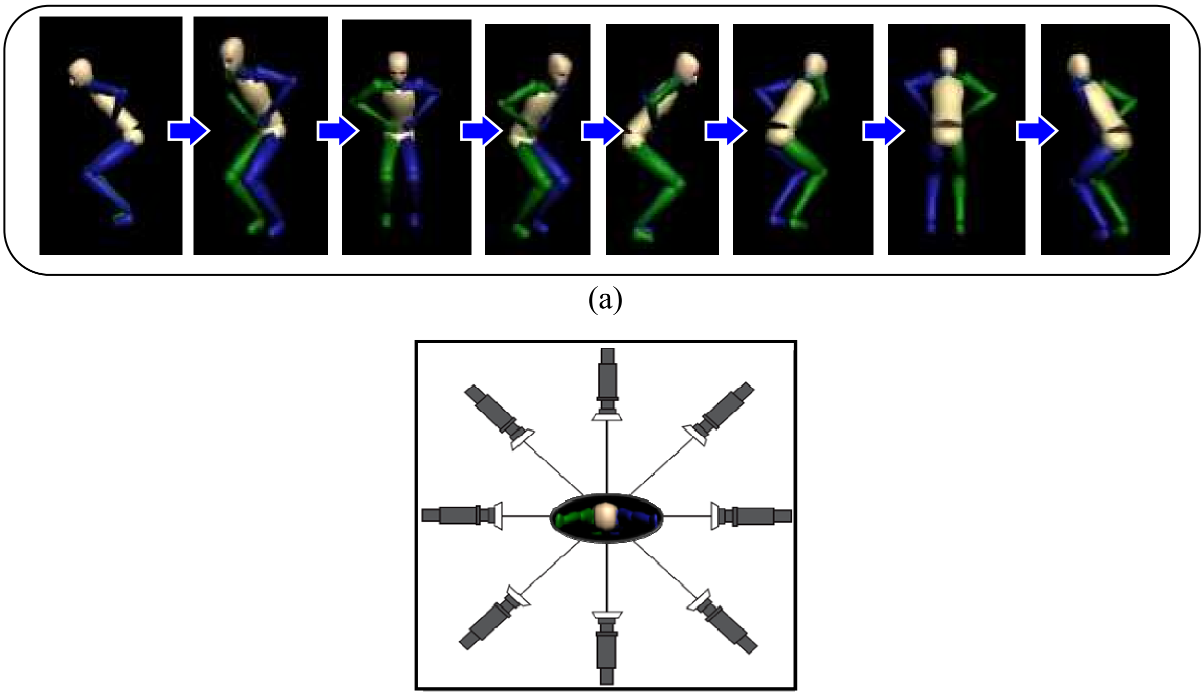


Fig. 5.9 Ten synthesized *avatar* actors



(a)

(b)

Fig. 5.10 (a) Eight views of an actor performing a motion, (b) surrounding camera concept.

5.7.2 Definition

5.7.2.1 Bin length

A bin is defined as the hypercube within each directional eigenspace by portioning each eigen-axis. By transforming the extent of each axis to *unit* with *S* divisions along the axis, the length of each edge of a bin is defined as:

$$\text{Bin Length}\,(L) = \frac{\text{Extent of an eigen-axis}}{\text{Total number of divisions along the axis}} = \frac{1}{S}$$

We shall refer to the bin length as *L* in the latter part of the thesis.

5.7.2.2 Recognition rate

Recognition rate is defined as the percentage of successfully recognized motions among total number of motions. It can be defined as:

$$\text{Recognition rate} = \frac{\text{Number of recognized motions}}{\text{Total number of motions}} \times 100$$

5.7.2.3 Searching rate

The time elapses for searching for a test motion within the database is known as searching time. We also compute searching rate which is defined as:

$$\text{Searching rate} = \frac{\text{Total number of data within the database}}{\text{Total searching time}}$$

The units for searching time and searching rate are *millisecond* and *data*/*millisecond*, respectively.

5.7.2.4 Recall

Recall is defined as the percentage of successfully recognized motions among the *ground truth* motions. It can be defined as:

$$\text{Recall } (R) = \frac{\text{Number of recognized motions}}{\text{Ground Truth}} \times 100$$

5.7.2.5 False Positive Rate (*FPR*)

False Positive Rate (FPR) is defined as the percentage of miss-recognized motions among the total number of motions which are recognized correctly and incorrectly. It can be defined as:

$$\text{FalsePositiveRate}(FPR) = \frac{\text{Numberof wronglyrecognizedmotions}}{\text{Numberof correctlyrecognizedmotions+ Numberof wronglyrecognizedmotions}} \times 100$$

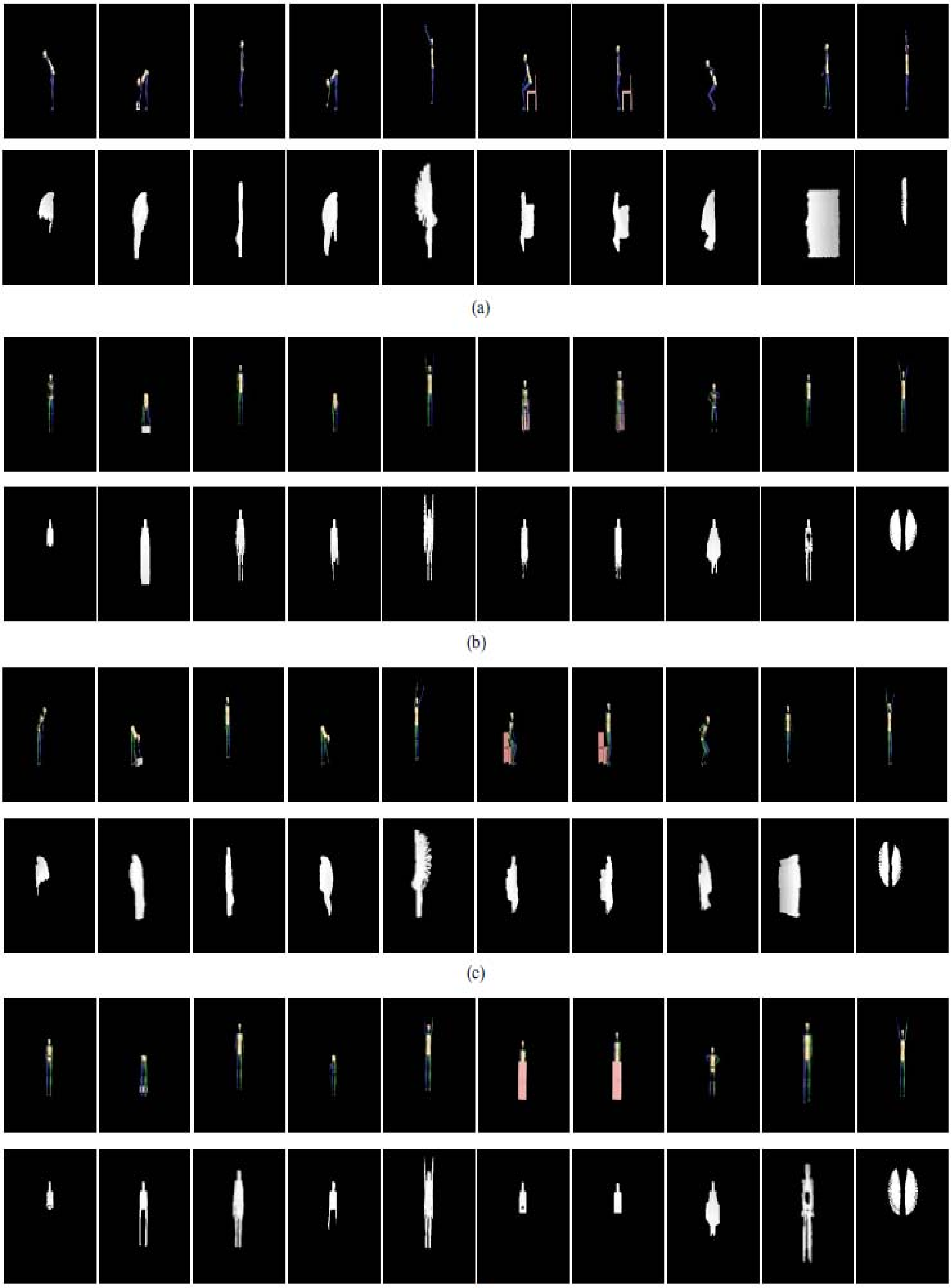
5.7.2.6 Precision

Precision is defined as the percentage of recognized motions among the total number of motions which are recognized correctly and incorrectly. It can be defined as:

$$\text{Precision } (P) = \frac{\text{Number of recognized motions}}{\text{Number of correctly recognized motions + Number of wrongly recognized motions}} \times 100$$

5.7.2.7 Scale of shifting

Scale of shifting refers to the extent within the eigenspace the space division is shifted to form a shifted query space (See Section 5.5.4). In general, we have taken it as *L*/2, i.e., half of the bin length. But we have also experimented for other scales of shifting.



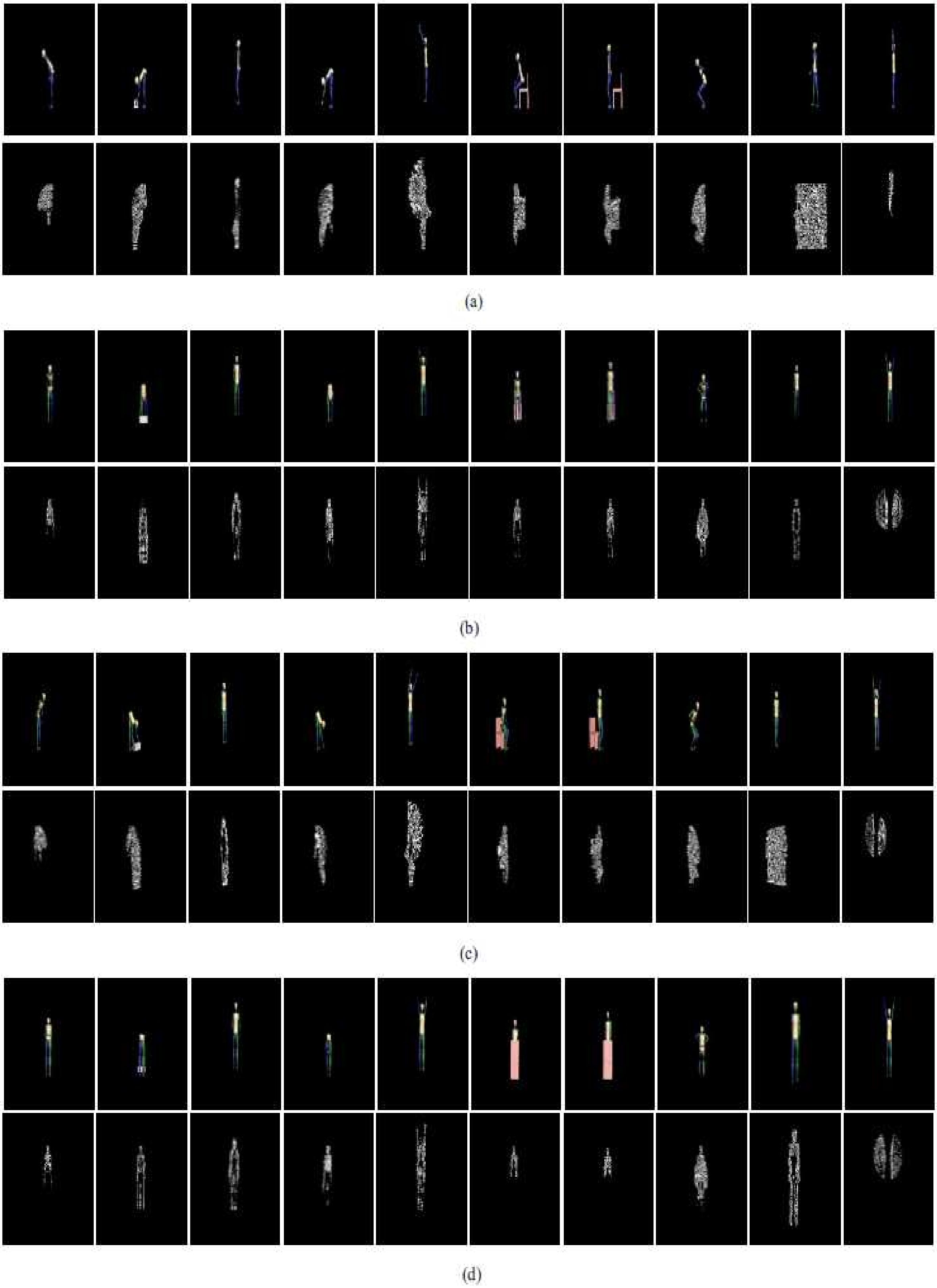
(a)

(b)

(c)

(d)

Fig. 5.11 A single motion frame and corresponding MHIs of each of the 10 motions from different camera angles (a) 0, (b) 90, (c) 135, (d) 270 degrees.



(a)

(b)

(c)

(d)

Fig. 5.11 A single motion frame and corresponding XOR images of each of the 10 motions from different camera angles (a) 0, (b) 90, (c) 135, (d) 270 degrees.

5.7.3 Experimental results and the analysis of the results

First of all, the structured motion database with the directional organization is used to evaluate the performance of the recognition system. The experimental results are obtained by varying the division parameter that we call bin length. The results are extensively compared with the methods proposed in [83, 249, 178] which we refer to as *existing methods/strategies.*

Using MHI, in the case of no division, the recognition rates for proposed and existing strategies are 97% and 96%, respectively. The average recognition rates for the bin lengths 1/2 to 1/10 employing *proposed* and *existing* strategies are 86% and 74%, respectively; while the maximum recognition rate is achieved in our proposed approach for bin length 1/5 as 89%. So, we notice significant difference in recognition rates between the proposed and other database organizations. Similarly, using XOR images, in the case of no division, the recognition rate is 95% for the both cases. The average recognition rates for proposed and existing strategies are 77% and 64%, while the maximum recognition rate is achieved in our proposed approach for bin length 1/8 as 85%. Therefore, our proposed approach with directional organization claims significant improvement over others for both the representations. The above results are tabulated in **TABLE 5.1**. From TABLE 5.1, it is noticeable that MHI outperforms XOR representation by considerable amount in terms of recognition rate.

Moreover, we have also investigated the time requirement for the proposed directional organization-based method and existing methods. In the case of MHI with no division, the searching rates are 31.4 data/ms and 8.5 data/ms for proposed and existing organization, respectively. The average searching rates using the structured motion database concept are 51.4 data/ms and 37.8 data/ms, respectively. Therefore, using MHI the proposed approach is very much faster than the earlier approach. Similarly, using XOR images, in the case of no division, the searching rates are 8 data/ms and 1.8 data/ms for proposed and existing organization, respectively. The average searching rates using structured database concept are 39 data/ms and 41 data/ms, respectively. In the case of XOR, time requirement does not vary due to the high dimensionality of the feature space, while the proposed directional organization-based approach is three to four times faster than the sequential search. **TABLE 5.2** tabulates the searching time and searching rates for both the aforesaid cases.

Secondly, we have solved the boundary problem within the motion database that implies higher performance than our approach with directional organization only. We have computed experimental results for the problem resolution scheme having adopted the directional organization. We have tabulated it in **TABLE 5.3**, and compared it with our aforesaid directional organization based results. We have computed the results for the scale of shifting of *L*/2. Moreover, a motion-wise performance graph is also shown in **Fig. 5.12**.

TABLE 5.1

Recognition rate (%) for existing and proposed directional organization

Bin Length	**MHI**		**XOR**	
	Existing	*Proposed*	*Existing*	*Proposed*
1	96	97	95	95
1/2	80	88	58	75
1/3	76	88	61	73
1/4	80	82	67	78
1/5	68	89	63	76
1/6	78	83	68	75
1/7	70	85	64	73
1/8	75	83	70	85
1/9	70	86	69	79
1/10	65	86	60	78
Average	74	86	64	77

TABLE 5.2

Time consideration for existing and proposed directional organization

Bin Length	**MHI**				**XOR**			
	Existing		*Proposed*		*Existing*		*Proposed*	
	Searching Time(ms)	*Searching Rate*(data/ms)	*Searching Time*(ms)	*Searching Rate*(data/ms)	*Searching Time*(ms)	*Searching Rate*(data/ms)	*Searching Time*(ms)	*Searching Rate*(data/ms)
1	66.17	8.5	18.1	30.94	308.55	1.8	70.16	8
1/2	13.72	40.8	9.98	56.11	12.81	43.7	14.625	38.3
1/3	14.6	38.4	11.11	50.41	14.66	38.2	13.825	40.5
1/4	13.89	40.3	11.48	48.78	13	43.1	13.9	40.3
1/5	15.15	37	11.52	48.61	11.69	47.9	13.95	40.1
1/6	15.96	35.1	11.59	48.32	16.4	34.1	15.28	36.6
1/7	14.68	38.1	11.83	47.34	14.15	39.6	15.18	36.9
1/8	15.51	36.1	13.63	41.09	15.02	37.3	13.72	40.8
1/9	16.5	33.9	13.76	40.7	12	46.7	14.175	39.5
1/10	13.97	40.1	12.87	43.51	14.5	38.6	14.82	37.8
Average	14.9	37.8	12	45.58	13.8	41	14.4	39

TABLE 5.3

Recognition rate (%) for directional organization only and with boundary problem resolution

Bin Length	MHI		XOR Image	
	Directional Organization only	*With Boundary Problem Resolution*	*Directional Organization only*	*With Boundary Problem Resolution*
1	97	97	95	95
1/2	88	94	75	90
1/3	88	94	73	82
1/4	82	90	78	83
1/5	89	91	76	84
1/6	83	91	75	85
1/7	85	92	73	82
1/8	83	92	85	88
1/9	86	90	79	84
1/10	86	90	78	85

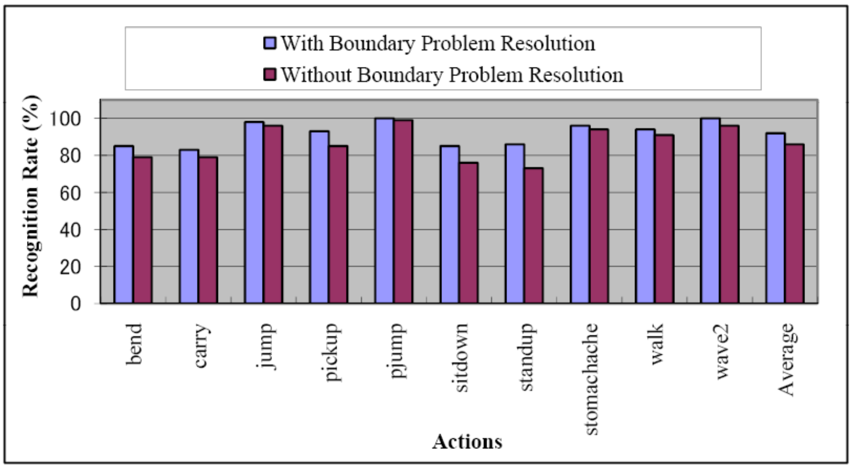


Fig. 5.12 Motion-wise comparative performance analysis according to boundary problem resolution

We obtain the average recognition rate of 92% for the system with the resolution to the boundary problem, whereas 86% for the system without the resolution using MHI (See **Table 5.3**). Similarly, we obtain the XOR-based average recognition rate of 85% for the system with the resolution to the boundary problem, whereas 77% for the system without the resolution (See Table 5.3). In Table 5.3, grey cells represent the recognition results having the boundary problem with directional organization only, whereas white cells represent the results after resolution. The results are computed with varying bin lengths. For both the representations, our introduced resolution scheme shows significant improvement. The motion-wise comparative performance analysis illustrated in Fig. 5.12 shows that for every motion the recognition rate is increased by a considerable amount. However, we have also computed the searching time corresponding to the boundary problem resolution scheme. Those are tabulated in **TABLE 5.4**.

Thirdly, as we mentioned in Section 5.5.5, we have introduced prior estimation of directions within the motion recognition system. We have also performed experiments employing this strategy for the directional organization *only*, and jointly with the directional organization and the boundary problem resolution scheme.

***Case*-1: *The directional organization*.** Using MHI, the recognition rates computed for bin length 1 for the directional organization and with the prior estimation strategies are 97% and 96%, respectively. The average recognition rates for the bin lengths 1/2 to 1/10 employing these two strategies are 86% and 85%, respectively; while the maximum recognition rate is achieved with prior direction estimation for bin length 1/2 as 88%. Similarly, using XOR images, in the case of no division, the recognition rates are 95% for both the cases. The average recognition rates for proposed and existing strategies are 77% for both of them, while the maximum recognition rate is achieved with prior direction estimation for bin length 1/8 as 86%. Therefore, we notice almost equal recognition accuracy for the proposed and existing strategy. The above results are tabulated in **TABLE 5.5**. From TABLE 5.5, it is noticeable that MHI outperforms XOR representation by considerable amount in terms of recognition rate. But the proposed approach merely affects the performance for the both cases.

However, we have comprehensively investigated the time requirement for prior estimation-based method and the existing methods. In the case of MHI with no division, the searching time is 15.1 milliseconds (ms) and 18.1 milliseconds (ms) for prior estimation-based method and existing method, respectively. The average searching time using the structured motion database concept is 10.2 ms and 12 ms, respectively. Therefore, using MHI the proposed approach that utilizes the prior direction estimation is much faster than that of the approach not using the direction estimation. If we consider the reduction of searching cost in terms of the number of eigenspaces, we found that 266 eigenspaces corresponding to 240 test motions remain unsearched due to direction estimation. So, in average, more than one eigenspace per motion is eliminated for redundancy at the time of searching for the candidate motions. Similarly, using XOR images, in the case of no division, the searching time is 68.3 ms and 70.16 ms for the new and existing method, respectively. The average searching time using structured database concept for the new and existing method is 13.6 ms and 14.4 ms, respectively. In the case of XOR, due to the scattered nature of the motion points within the space, the dimensionality of the feature space becomes comparatively high which leads to the less reduction of the searching spaces. Thus there is only slight reduction in time requirement for the proposed approach. **TABLE 5.6** also tabulates the searching time for the prior estimation-based method and existing method.

TABLE 5.4

Searching time (ms) for directional organization only and with problem resolution

Bin Length	**MHI**		**XOR Image**	
	Directional Organization only	*With Boundary Problem Resolution*	*Directional Organization only*	*With Boundary Problem Resolution*
1	18.1	18.1	70.16	70.16
1/2	9.98	26.96	14.625	49.56
1/3	11.11	22.9	13.825	28.34
1/4	11.48	24.69	13.9	28.88
1/5	11.52	23.53	13.95	29.57
1/6	11.59	24.77	15.28	31.12
1/7	11.83	25.36	15.18	30.35
1/8	13.63	28.22	13.72	30.55
1/9	13.76	26.85	14.175	32.18
1/10	12.87	25.63	14.82	31.93

TABLE 5.5

Experimental results with directional organization only and with prior direction estimation

Bin Length	**MHI**				**XOR**			
	Directional Organization only		*Prior direction estimation*		*Directional Organization only*		*Prior direction estimation*	
	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)
1	97	18.1	96	15.1	95	70.16	95	68.3
1/2	88	9.98	88	8.3	75	14.62	75	13.93
1/3	88	11.11	87	9.22	73	13.82	73	12.83
1/4	82	11.48	82	9.81	78	13.9	78	12.77
1/5	89	11.52	88	9.85	76	13.95	76	13.28
1/6	83	11.59	83	9.76	75	15.28	75	14.86
1/7	85	11.83	85	10	73	15.18	73	14.63
1/8	83	13.63	82	11.9	85	13.72	86	13.04
1/9	86	13.76	85	11.5	79	14.17	79	13.39
1/10	86	12.87	86	11.6	78	14.82	78	13.97

Case-2*: *Jointly with the directional organization and the boundary problem resolution scheme. Moreover, we have also computed the recognition rate and searching time for the prior direction estimation strategy by adopting the directional organization and boundary problem resolution scheme. We have computed the results for the scale of shifting of *L*/2. In this case, we obtained the maximum recognition rate of 93% and 90% for MHI and XOR image, respectively. However, the average MHI-based recognition rates for the bin lengths 1/2 to 1/10 employing the primitive directional organization-based methods and jointly with the directional organization and the boundary problem resolution scheme method are 91% and 92%, respectively; whereas the average XOR-based recognition rates for the bin lengths 1/2 to 1/10 employing the aforementioned strategies are 85% for both cases. After computing the searching time employing the resolution scheme with and without the prior direction estimation, we found the average searching time for MHI-based recognition with above strategies 20.9 ms and 25.4 ms, respectively. Similarly, the average searching time for XOR-based recognition obtained for the two cases is 31.7 ms and 32.5 ms, respectively. Therefore, we have also achieved a significant improvement in searching time with the boundary problem scheme with prior estimation of directions. The results are presented in **TABLE 5.6**.

TABLE 5.6

Experimental results with boundary problem resolution, and with prior direction estimation and boundary problem resolution scheme

Bin Length	**MHI**				**XOR**			
	Boundary problem resolution		*Prior direction estimation+ boundary problem resolution*		*Boundary problem resolution*		*Prior direction estimation+ boundary problem resolution*	
	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)	Recognition rate(%)	Searching Time (ms)
1	97	18.1	96	15.1	95	70.16	95	68.3
1/2	94	26.96	93	18.5	90	49.56	90	47.6
1/3	94	22.9	93	17.4	82	28.34	83	27.8
1/4	90	24.69	90	20.4	83	28.88	83	28.4
1/5	91	23.53	89	19.4	84	29.57	84	28.8
1/6	91	24.77	91	20.6	85	31.12	85	30.6
1/7	92	25.36	91	20.7	82	30.35	82	29.8
1/8	92	28.22	91	23.5	88	30.55	88	29.9
1/9	90	26.85	90	24.2	84	32.18	84	31.4
1/10	90	25.63	89	22.6	85	31.93	85	31.2

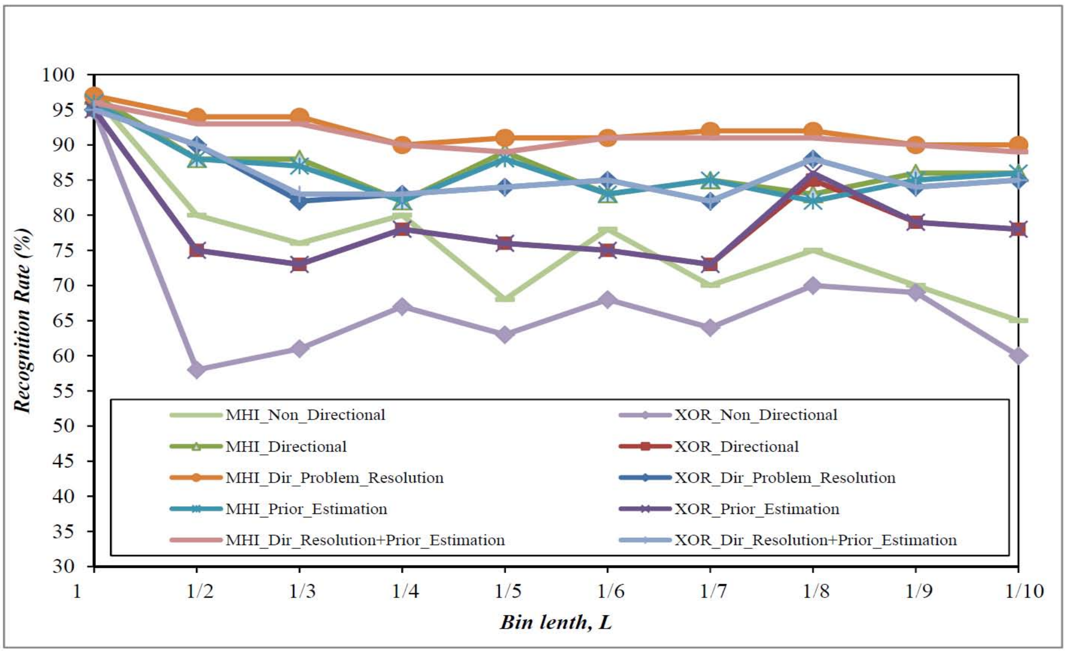


Fig. 5.13 Illustration of recognition rate for various schemes with XOR and MHI

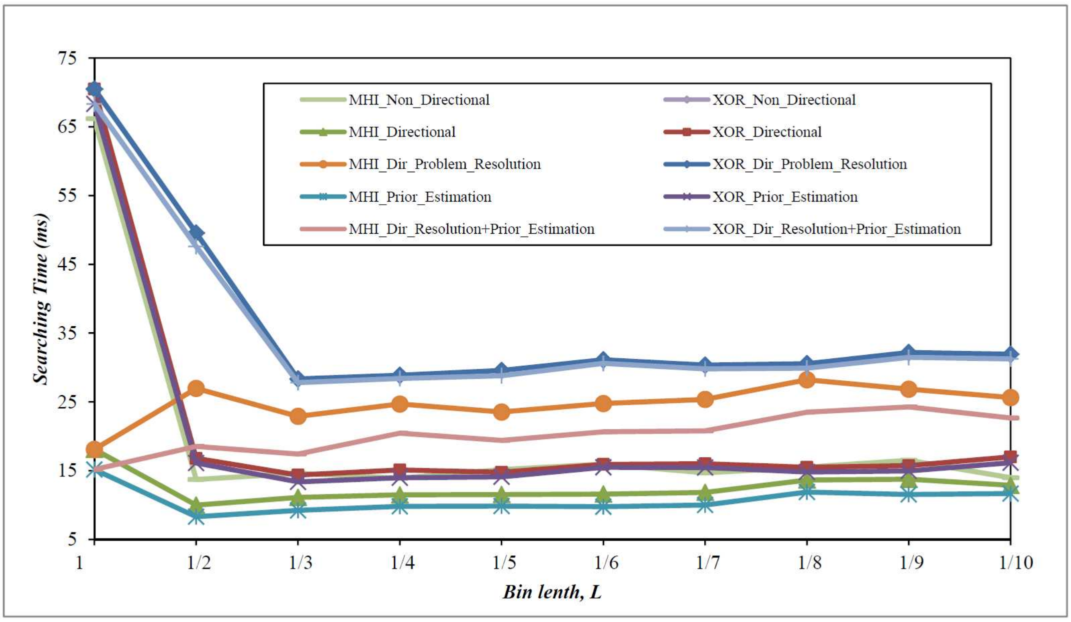


Fig. 5.14 Illustration of time requirement for various schemes with XOR and MHI

In the above analysis of experimental results, we have emphasized three aspects of structured motion database; *directional organization*, *resolution of the nearest neighbor searching problem*, and *prior estimation of directions*. We have demonstrated the performance of these aspects with comprehensive experimentation and analysis. We illustrate the overall performance comparison in terms of recognition rate and searching time in **Fig. 5.13** and **Fig. 5.14**, respectively. We notice the significant increase in performance from non-directional to problem resolution scheme for both MHI and XOR image representations. The eigenspace with bin length 1 is referred to as non-structured form of motion database that employs exhaustive search within the database for finding candidate motions. However, having employed the structured motion database with MHI templates, the average recognition rate for non-directional, directional, problem resolution (scale of shifting is *L*/2), prior direction estimation and problem resolution with prior estimation schemes are 74%, 86%, 92%, 85% and 91%, respectively. Similarly, with XOR image, the average recognition rate for non-directional, directional, problem resolution (scale of shifting is *L*/2), prior direction estimation and problem resolution with prior estimation schemes are 64%, 77%, 85%, 77% and 85%, respectively. We notice significant improvement in recognition rates from non-directional to problem resolution. From Fig. 5.14, we notice the average searching time requirements for the above five schemes in order are 14.9 ms, 12 ms, 25.4 ms, 10.2 ms and 20.9 ms with MHI, and are 13.8 ms, 14.4 ms, 32.5 ms, 13.6 ms and 31.7 ms with XOR, respectively.

We have also tabulated the performance evaluation of various schemes, namely, non-structured, basic structured (or non-directional), directional, prior direction estimation, problem resolution, and problem resolution with prior estimation, with the maximum recognition rates achieved and the corresponding time requirement (See **TABLE 5.7**). We find that 95% and 93% recognition rate is obtained for the problem resolution scheme at the scale of shifting 2*L* (*L*=1/2) with MHI and XOR image, respectively. However, for the scheme having prior direction estimation and problem resolution together, though the recognition rate (94%) is slightly less than the scheme with problem resolution only, it shows shorter searching time which is much acceptable in this case. Therefore, the scheme with problem resolution with prior estimation of directions presents the best performance for our experimentation utilizing the three aspects of the structured database altogether.

However, we have also calculated the motion-wise *recall*, *FPR* and *precision* at scale of shifting 2*L* (*L*=1/2); we found that standup and pickup motions show higher FPR and lower precision with MHI, and stomachache and pickup show higher FPR and lower precision with XOR image. These measures are tabulated in **TABLE 5.8**.

Moreover, we have also analyzed the effect of shifting parameter (i.e., scale of shifting) on the recognition rate by varying the parameter at *L*/4*, L*/2*, L* and 2*L*. We found that when the scale of shifting is 2*L*, the recognition rate is found to be the maximum for both the MHI and XOR image. We found 95% and 93% recognition rate with the shifted B-Tree concept without the prior estimation. Thus we can assume the shifting parameter experimentally for a specific dataset. This effect is illustrated by bar-graph in **Fig. 5.15**.

TABLE 5.7

Performance evaluation for various recognition schemes

Scheme	MHI		XOR	
	Recognition Rate (%)	Searching Time (ms)	Recognition Rate (%)	Searching Time (ms)
Non-structured	96	66.53	95	336.84
Basic structured	80	14.9	70	13.8
Directional	89	12	85	14.4
Prior estimation of directions	88	10.2	85	13.64
Directional with problem resolution	95	33.24	93	59.64
Directional with prior direction estimation and problem resolution	**94**	**22.85**	**93**	**57.28**

TABLE 5.8

Motion-wise performance evaluation

Motion	MHI			XOR		
	Recall (%)	Precision (%)	False Positive Rate (%)	Recall (%)	Precision (%)	False Positive Rate (%)
bend	87.5	100	0	87.5	95.5	4.5
carry	83.33	95.2	4.8	83.33	87	13
jump	91.67	88	12	100	96	4
pickup	95.83	85.2	14.8	79.17	86.4	13.6
pjump	100	100	0	100	100	0
sitdown	87.5	100	0	83.33	87	13
standup	91.67	84.6	15.4	100	100	0
stomachache	100	100	0	100	82.8	17.2
walk	100	100	0	100	100	0
wave2	100	88.9	11.1	100	100	0

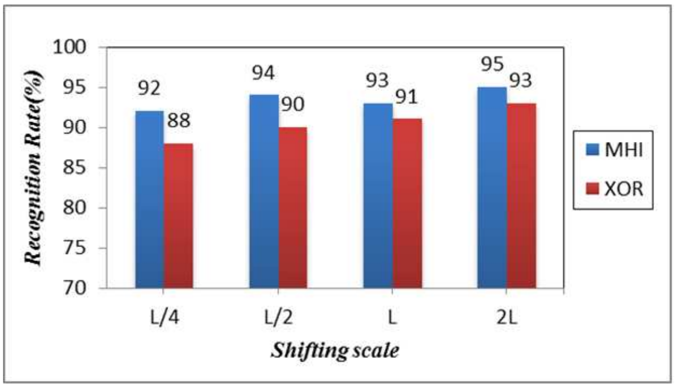


Fig. 5.15 Effect of *scale of shifting* on recognition rate

5.8 Discussion

We have proposed a novel recognition technique for identifying and interpreting human motions and actions from surrounding viewpoints using large avatar motion set. This method employs three essential aspects of the structured motion database; namely, *directional organization*, *resolution of the nearest neighbor searching problem*, and *prior estimation of directions*. In recent times, a bulk of motion/action datasets is available for action or behavior understanding and analysis. Due to the large amount of data, the structurization becomes an indispensable task. The structurization demands the database to be efficient with respect to recognition rate, time and space. We found that the aforesaid aspects jointly construct a database for the motion recognition that is suitable according to the above three criteria, i.e., recognition rate, time and space. Our proposed method utilizes the spatio-temporal information of motions and the concept of hierarchical eigenspaces for each camera direction to characterize the directional motion, as well as, to get some idea about the possible direction of the activities.

We propose a scheme where both the top-down and bottom-up strategies are followed one-after-another. We have estimated the directions for an unregistered motion by top-down manner, and we obtained the possible orientations of the motion that guides the searching algorithm. On the other hand, the motion is searched within the directional eigenspaces and candidate motions are obtained. These candidate motions are further projected onto the global eigenspace to confirm the category of the motion. This is accomplished by bottom-up manner. Previously, motion recognition is accomplished in bottom-up manner only [38, 83, 176, 177, 178, 249], whereas the newly introduced hybrid manner of problem solution proposed in this paper shortens the search time by reducing potentially unnecessary search cost within the feature spaces. The goodness of the system lies in reducing the search complexity that makes it improved and non-redundant. The scheme of prior direction estimation with nearest neighbor search problem resolution has significantly reduced the search time with high recognition rate: This proves its effectiveness in performance. We obtain 94% recognition rate at 22.85 ms with MHI, and 93% recognition rate at 57.28 ms with XOR image. Though both MHI and XOR image-based system shows similar recognition rate, the search time varies due to the high dimensionality of the feature space with XOR image. However, we notice a slight decrease of less than one percent in the recognition rate with this scheme; this is due to the inclusion of 90 percent of the total number of direction-wise motion points for constructing each direction-wise cluster. Moreover, with the increase in registration within the database, the search time tends to rise; but selective search may lead to time-efficient human motion recognition.

However, there is hardly any research that employs surrounding viewpoints for motion capture; rather frontal or sideways' motions are captured. Noticeably, we have obtained higher recognition rates with much higher data search rate for the surrounding camera arrangement. By analyzing the proposed system, we can figure out some findings which are stated below.

i. For conventional strategy with all the motions within a single eigenspace without using the hierarchical strategy will surely lead to longer search time due to the increase of dimensions to realize the motion space. For this case, the search is *x* times longer than the proposed approach, for *x* is a numeric factor.

ii. In our experimentation, though we vary the motion speed, the recognition rate is unaffected for the captured motions. Thus the proposed system has achieved high recognition performance with high-speed recognition.

iii. For XOR, compared to MHI, linear search time increases proportional to the number of data, whereas using our proposed technique the time increase is almost unaffected by the number of dimensions for same number of registered data. Surely, if the number of data registration increases, the dimension will consequently increase. But with this increment, the proposed approach is capable of keeping the search time short.

iv. The search time requirement for the proposed approach having the boundary problem resolution scheme is found to be somehow increased compared to that of directional organization-based approach due to the search of the B-Tree twice, rather than once. Though we can compromise a little amount of time for the precision, the adoption of the advent technique, e.g., parallel programming, multi-threaded programming etc., might be a possible alternative.

v. The division parameter *S* might be chosen based on the types of motions being recognized to make the system an unsupervised one. From Fig. 5.16, we get some idea about the proper bin length for the system. The bin length is to be selected in such a way to optimize the system's performance in terms of recognition rate and search time. For our experimental dataset, we can choose the length of 1/2 for both MHI- and XOR image-based system (as shown in *blue circle* in **Fig. 5.16**) according to recognition rate by compromising with time requirement.

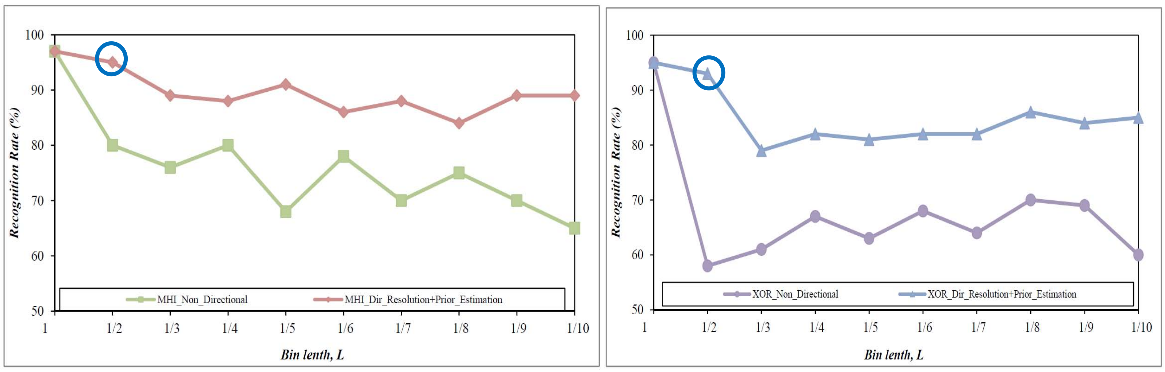


Fig. 5.16 Possible selection of *bin length*; (a) MHI, (b) XOR images.

In the performed experiments, the motions were formed by varying speed, and varying shape of the performing subjects. In spite of the variability in motions, our proposed motion database shows satisfactory performance on their recognition. In Section 5.5, the motion database development scheme and other factors are analyzed, and the reasons behind the superiority of the method over conventional or non-directional organization are also represented by introducing several novel aspects for the structured motion database. Although we have achieved satisfactory performance for our proposed system, there are, of course, some limitations in terms of occlusion of the body parts or overlapping movements, moving cameras, changing background, etc. The occurrence of the motions, i.e., *how* the motion is moving, is another cause of poor recognition. The experimental motions, however, are not so much complex, rather simple and have almost no overwriting problems. For the overwriting cases, the Directional MHI representation [81] has much potential to uniquely represent each motion. This form of motion template may also be incorporated in our proposed system (See Section 6.2). Moreover, it would be worthwhile to develop the system with real-life large motion database with a huge number of indoor and outdoor motions. Separate mechanism for background subtraction and foreground segmentation might be employed for real-life motion recognition cases (See Section 6.3).

5.9 Summary

In this chapter, we have presented a novel approach for human motion recognition by developing an efficient motion database that is organized based on camera orientation. We have discussed the detailed structure and operations of the motion database. We have also performed experiments with the proposed technique, and the performance evaluation is comprehensively analyzed. Thus the effectiveness of the proposed system is described through experimental data and graphical illustrations. Finally, we discuss some issues regarding the developed human motion recognition system.

Chapter 6
Analysis on the Constraints in Human Motion Recognition

6

Analysis on the Constraints in Human Motion Recognition

6.1 Introduction

Human motion recognition system imposes a number of constraints on performing of the motions, environmental condition, recording setup, camera orientations, and many others. As mentioned in Section 5.9, the overlapping of the body parts and the overwriting or repetitive characteristics of motions may lead to significant drop of system's performance. Similarly, from one camera viewpoint the moving regions may be visible, whereas from other camera viewpoints those may not be visible; the same action, observed from different viewpoints, can lead to very different image observations. Also, the fact that a single camera is only able to capture a projection introduces a source of variation. All this leads to the missing of sufficient information for representing a motion. Often, a known camera viewpoint is assumed, but this restricts the use of static cameras. In Chapter 3, we mentioned that the successful recognition of a motion depends on how well the motions are characterized by means of features. Likewise, missing of the information may occur due to the variability of the environmental conditions. Suppose that actions are performed in the same manner for different environmental conditions; the differences in the environmental condition result in differences in the captured movement. The environment or the surroundings, in which the action performance takes place, is an extremely important source of variation in motion capture. When this environment is cluttered or dynamic, it might prove harder to localize the person in the video. The environmental setup might be such that parts of the person might be occluded in the capture. This introduces source of uncertainty and missing of information. Dynamic (or irregular) backgrounds further increase the complexity of localizing the person in the image and robustly observing the motion. When using a moving camera, these issues become even harder. Different persons can appear differently due to differences in anthropometry, but also due to clothing, skin color and facial appearance. Lighting conditions can further influence the appearance. A robust approach should be able to generalize over these factors. Since we focus on vision-based human action recognition, we address the aforementioned constraints explicitly.

In this chapter, we deal with two important constraints in human motion recognition: *inclusion of directional information* to handle missing information, and *cluttered outdoor environment* for the recording environment to handle the irregular backgrounds in motion recognition. The recognition system employing the directional information is discussed in Section 6.2. Besides, a recognition system implemented for a cluttered outdoor scenario is discussed in Section 6.3.

6.2 Inclusion of Directional Information

6.2.1 Overview

Development of a robust human motion recognition system concerns with the fact that it adapts to the complexities imposed on recognition. Since the motion overwriting problem leads to the loss of sufficient information while representing a motion, the direction of movement can be treated as a significant clue for solving the aforesaid problem. Therefore, we exploit the directional vectors in motion analysis, and the robustness is enhanced for the current system over earlier methods. In our approach, we use the structured motion database concept as discussed in Section 5.5. Multi-directional distinct motions are represented, and compressed with the motion flow detection and compression technique, and prominent features are extracted. The extracted features are stored within a motion database that can cope with different forms of motion information. Our proposed system framework is illustrated in **Fig. 6.1**. However, we shall present the motion segmentation and motion flow computation techniques in the next sections.

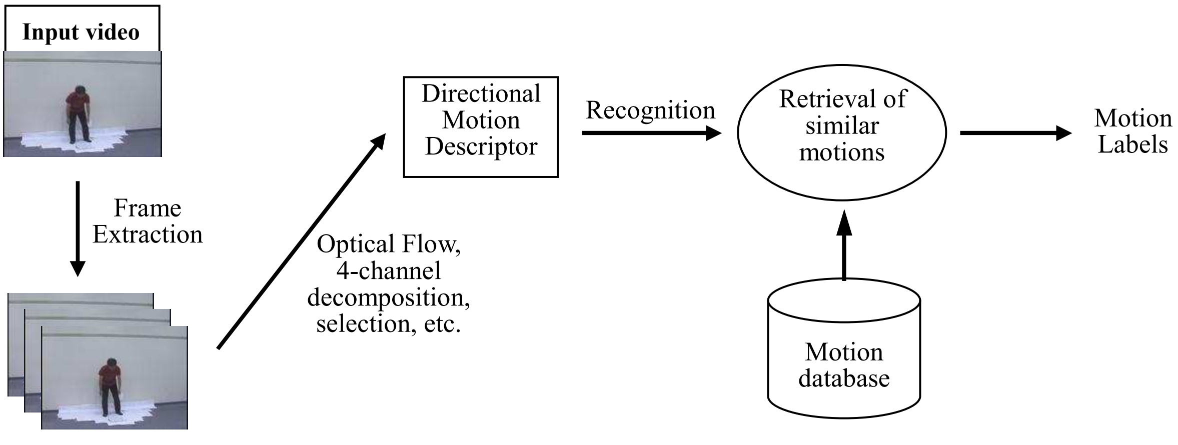

Input video
Frame Extraction
Optical Flow, 4-channel decomposition, selection, etc.
Directional Motion Descriptor
Recognition
Retrieval of similar motions
Motion database
Motion Labels

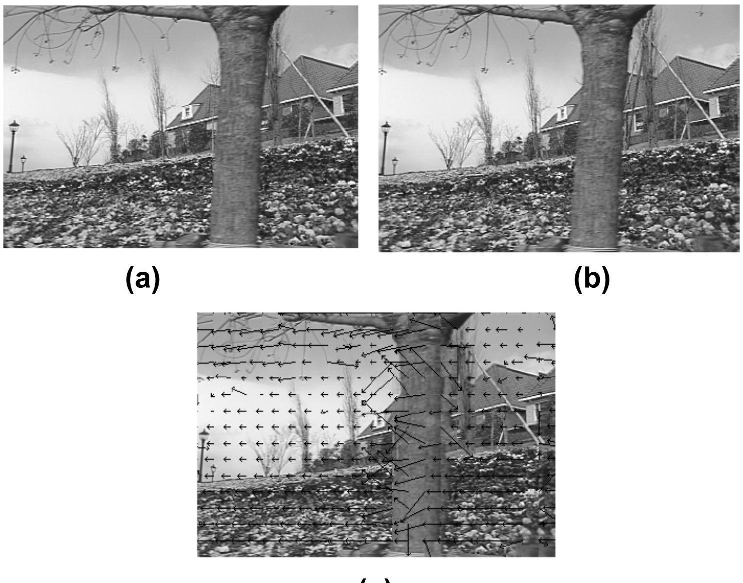
Fig. 6.1 System frame with the inclusion of directional information

❑ Motion Segmentation

Our proposed system computes a spatio-temporal volume for each motion. For this computation, the motion should be somehow segmented to determine the flow of the motion; otherwise unwanted noise will reduce the significant information embedded within the motion to be represented uniquely. Such kind of motion segmentation can be obtained by first transforming each motion frame by *Gaussian blurring*, and extract the moving region between successive frames. After successful segmentation of motion sequences, the direction and magnitude of the motion, i.e., flow of motion, is computed using optical flow computation.

❑ Optical Flow

An optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the subtle change of motion in temporal direction. Our concerned flow of motion is estimated and/or computed by determining the optical flow between successive motion frames (See **Fig. 6.2**). As the similarity measurement between different motions depends on both spatial and temporal information, optical flow can be used as a discriminative feature for determining the correlation. However, a number of optical flow methods are available to compute the optical flow [250]. The optical flow methods try to calculate the motion between two successive image frames which are taken at times t and $t + \delta t$ at every pixel position. In this work, we have adopted Lucas-Kanade (LK) method for computing the optical flow. Here, we present a mathematical explanation about the Lucas-Kanade method.



(a) (b)

(c)

Fig. 6.2 Illustration of *optical flow*. (a) previous frame, (b) current frame, (c) current frame with optical flow. The tree is moving with the wind in the figure.

Lucas-Kanade Method

In computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade [251]. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion. By combining information from several nearby pixels, the Lucas-Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

The Lucas-Kanade method assumes that the displacement of the image contents between two nearby instants (frames) is small and approximately constant within a neighborhood of the point *p* under consideration. Thus the optical flow equation can be assumed to hold for all the pixels within a window centered at *p*. Namely, the local image flow (velocity) vector (V_x,V_y) must satisfy

$$\begin{aligned} I_x(q_1)\,V_x + I_y(q_1)\,V_y &= -I_t(q_1) \\ I_x(q_2)\,V_x + I_y(q_2)\,V_y &= -I_t(q_2) \\ &\vdots \\ I_x(q_n)\,V_x + I_y(q_n)\,V_y &= -I_t(q_n) \end{aligned} \tag{6.1}$$

where $q_1, q_2, \ldots., q_n$ are the pixels inside the window, and $I_x(q_i)$, $I_y(q_i)$, $I_t(q_i)$ are the partial derivatives of the image *I* with respect to position *x*, *y* and time *t*, evaluated at the point q_i and at the current time.

These equations can be written in matrix form $Av = b$, where

$$\mathbf{A} = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ I_x(q_2) & I_y(q_2) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad \boldsymbol{\upsilon} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix} \tag{6.2}$$

This system has more equations than unknowns and thus it is usually over-determined. The Lucas-Kanade method obtains a compromise solution by the least squares principle. Namely, it solves the 2×2 system

$$\begin{gathered} \mathbf{A}^T\mathbf{A}v = \mathbf{A}^T\mathbf{b} \text{ or} \\ \boldsymbol{v} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{gathered} \tag{6.3}$$

where, $\mathbf{A}^T$ is the transpose of matrix **A**. That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (6.4)$$

with the sums running from i=1 to n. The matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is often called the structure tensor of the image at point p.

The plain least squares solution above gives the same importance to all n pixels q_i in the window. In practice, it is usually better to give more weight to the pixels that are closer to the central pixel p. For that, one uses the weighted version of the least squares equation,

$$\mathbf{A}^{\mathrm{T}}\mathbf{WAv} = \mathbf{A}^{\mathrm{T}}\mathbf{Wb} \text{ or}$$
$$\boldsymbol{v} = (\mathbf{A}^{\mathrm{T}}\mathbf{WA})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{Wb} \quad (6.5)$$

where, **W** is an $n \times n$ diagonal matrix containing the weights $\mathbf{W}_{ii} = w_i$ to be assigned to the equation of pixel q_i. That is, it computes

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i)I_y(q_i) \\ \sum_i w_i I_x(q_i)I_y(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i w_i I_x(q_i)I_t(q_i) \\ -\sum_i w_i I_y(q_i)I_t(q_i) \end{bmatrix} \quad (6.6)$$

The weight w_i is usually set to a *Gaussian function* of the distance between q_i and p.

❑ Half-wave rectification of the optical flow

The optical flow computed from the successive frame sequences are further represented through the *horizontal* and *vertical* velocities. However, the horizontal and vertical components are half-wave rectified to signify the four directional movements, namely *right*, *left*, *up*, *down* (See **Fig. 6.3**).

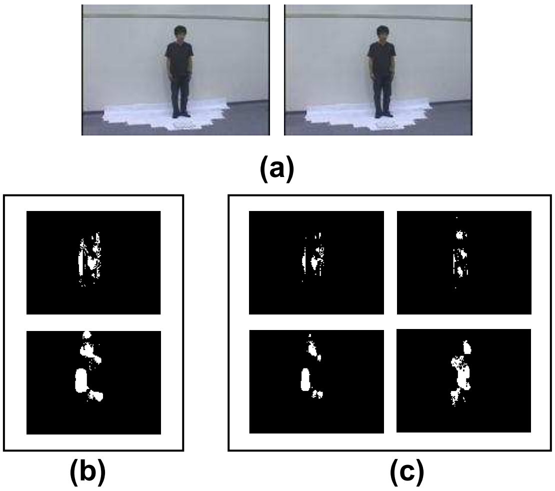


Fig. 6.3 Computation of an optical flow image. (a) Original consecutive frames, (b) Intensity image for the horizontal(x) and vertical(y) components of the optical flow generated from original consecutive frames, (c) Intensity image of four separate channels of x and y. top row: x^+, x^- , bottom row: y^+, y^-.

Thus the motions possess more significant information to be distinguished from other motion. Though the computation of optical flow might not be very accurate in coarse and noisy environment, it can be tuned to perform better. The optical flow, thus computed, can be used as *directional motion descriptor* for accomplishing the task of recognition. The optical flow computed in the aforementioned way is used as motion features which are further processed to generate motion representation for the recognition purpose. In Section 6.2, we present a MHI-based motion representation that uses *four* directional information to generate four directional MHIs for each motion.

6.2.2 Directional motion history image

The introduction of the directional concept in MHI was realized from the original Bobick and Davis paper [70]. According to [70], absence of motion flow was analyzed as the limitation or objection of using traditional MHI. It is necessary to establish some correlation between the construction of MHI and the direction of motion (e.g., optical flow). However, in other researches this directional concept was adopted for motion detection and recognition [87]. The optical flow vector between consecutive frames are computed which denotes the direction of movement of a person in motion. The optical flow vector is split along the horizontal and vertical directions by four components, namely x^+ (*right*), x^- (left), y^+ (*up*), and y^- (*down*). These optical flow images are referred to as vector images. Directional vector images are separated and are used to construct the MHI separately using Eq. (6.7).

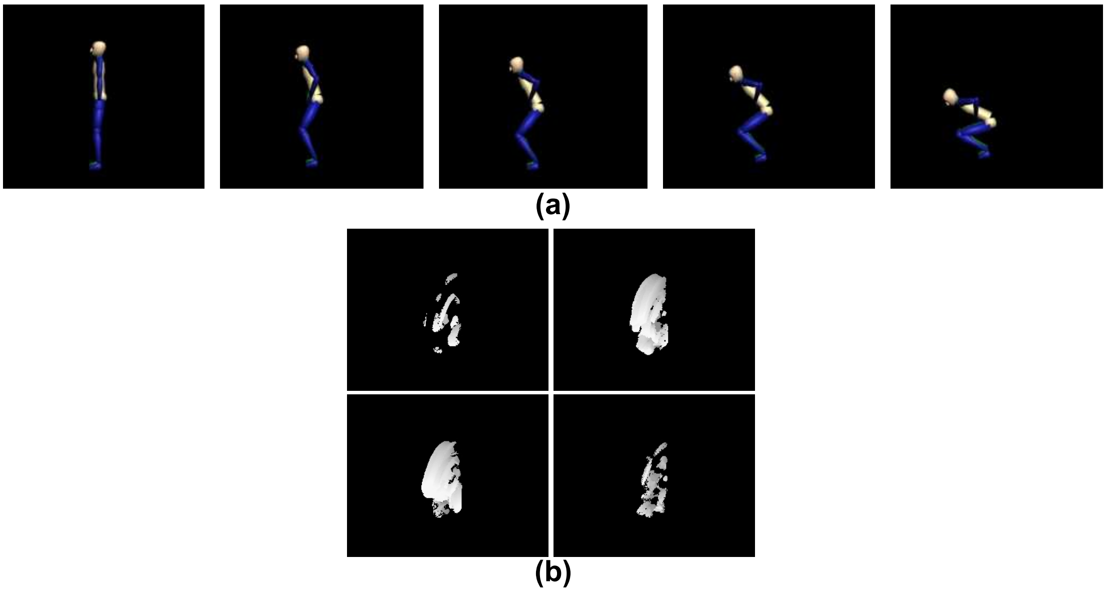


Fig. 6.4 Directional Motion representation a) *stomachcache* motion, (b) directional MHIs. Top row: MHI$_{x+}$,MHI$_{x-}$, bottom row: MHI$_{y+}$,MHI$_{y-}$.

$$
\begin{aligned}
H_\tau^{x+}(x,y,t) &= \begin{cases} \tau & \text{if } D(x,y,t) > th_{x+} \\ \max\left(0, H_\tau^{x+}(x,y,t-\delta)\right) & \text{otherwise} \end{cases} \\
H_\tau^{x-}(x,y,t) &= \begin{cases} \tau & \text{if } D(x,y,t) > th_{x-} \\ \max\left(0, H_\tau^{x-}(x,y,t-\delta)\right) & \text{otherwise} \end{cases} \\
H_\tau^{y+}(x,y,t) &= \begin{cases} \tau & \text{if } D(x,y,t) > th_{y+} \\ \max\left(0, H_\tau^{y+}(x,y,t-\delta)\right) & \text{otherwise} \end{cases} \\
H_\tau^{y-}(x,y,t) &= \begin{cases} \tau & \text{if } D(x,y,t) > th_{y-} \\ \max\left(0, H_\tau^{y-}(x,y,t-\delta)\right) & \text{otherwise} \end{cases}
\end{aligned}
\qquad (6.7)
$$

Here, D (*x,y,t*) is a binary difference image, $H_\tau^*(*= x^+, x^-, y^+, y^-)$ is a scalar valued function, δ is the decay parameter and th$_*$ is a threshold.

Using the directional representation, four directional motion history images are generated for each motion representing the right-, left-, up-, and down- directional movements, respectively (See **Fig. 6.4**). These four directional images represent a motion more vigorously compared to non-directional representation. However, as the accurate determination of optical flow is almost impossible, some of the generated directional images contain significant amount of noise within those. Neglecting the noise-factor, these motion images will act as the *motion descriptor* for the recognition system.

6.2.3 Selection of a representative feature image

As the directional feature image, generated from optical flow, may include noise, the employment of all the feature images for learning and recognition will surely lead to the erroneous result. So, the selection of the representative feature image is a very crucial issue for such kind of development of a robust system. A *representative directional MHI* is a feature image selected from a set of 4-directional MHIs to characterize a particular motion. The task of selection is likely to be accomplished by taking the *Motion Energy Image* (binarized MHI) from the MHI and by computing the pixel volume [70]. But for some cases, due to the existence of unwanted noise within the optical flow, the noise advances to the generation of directional MHI. So, the usage of inter-frame information, if extracted effectively, plays an important role in selecting the best representative directional MHIs. Based on this concept, we use two considerations for the selection of a representative feature image; *direction of global motion*, and *higher pixel volume*. If the selected feature image from the directional of global motion has much less number of pixels, the higher pixel volume comes into account.

The direction of global motion refers to overall direction of motion as the vector sum of the valid gradient directions. We compute the direction of global motion in the following way:

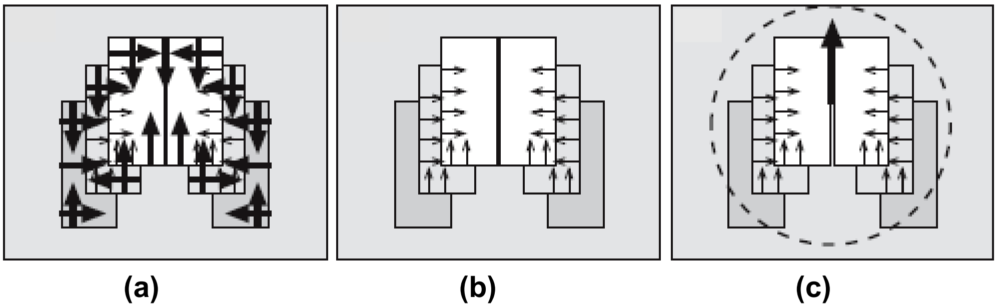


Fig. 6.5 Computation of the direction of global motion. (a) gradient magnitudes and directions, (B) large gradients are eliminated; (C) overall direction of motion is found

(a) We compute MHI for the motion.

(b) Motion gradient is also computed from MHI (See **Fig. 6.5(a)(b)**).

(c) Then the direction of global motion is computed by summing up the pre-computed motion vectors. One could compute the global motion from the center of mass of each of the MHI silhouettes, but the current method is much faster (See Fig. 6.5(c)).

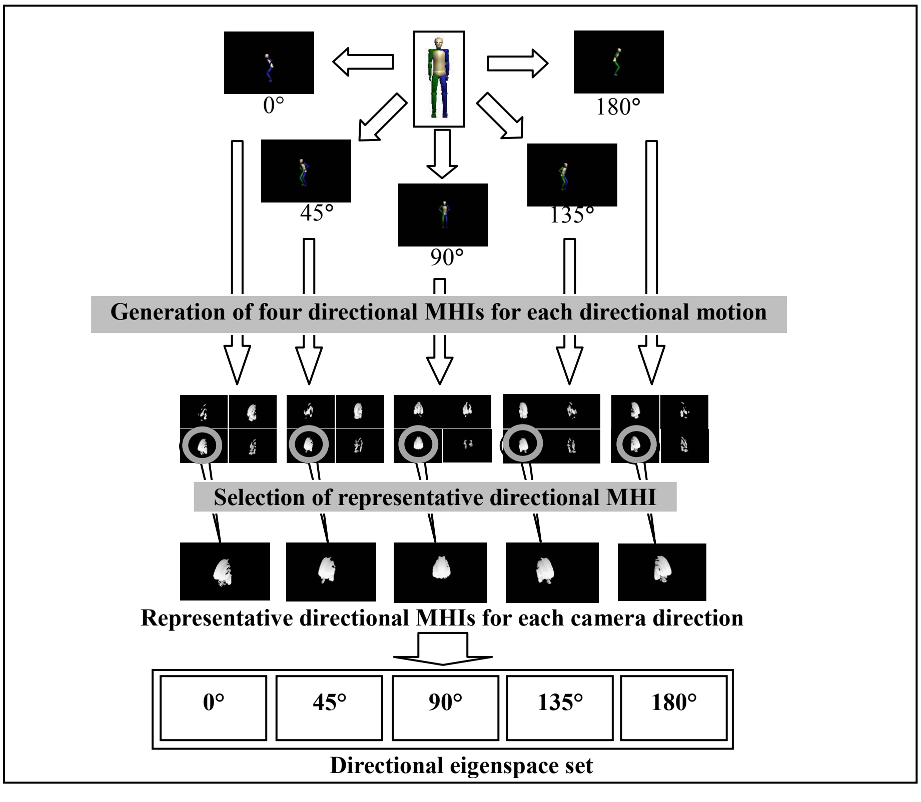

0°
180°
45°
90°
135°
Generation of four directional MHIs for each directional motion
Selection of representative directional MHI
Representative directional MHIs for each camera direction
0°
45°
90°
135°
180°
Directional eigenspace set

Fig. 6.6 Representation of a motion (*stomachache*) extracted from five directional video cameras.

The representative images thus selected are used to construct directional *eigenspaces* for further task of learning and recognition of the proposed system (See Section 5.4). The representation of a motion by representative feature image is depicted in **Fig. 6.6**.

6.2.4 Storage and retrieval within the motion database

In order to develop a structured motion database, we follow the database development strategy discussed in Section 5.5. We have adopted only the directional organization of the motion database. We summarize here the important steps for the database development.

- Construction of the directional eigenspaces
- Construction of global eigenspace
- Indexing of the training motions
- Store the indexes in the B-tree structured motion database

Similarly, for recognition, we use the similar concept as described in Section 5.6, except that the prior direction estimation and boundary problem resolution schemes are discarded. The recognition strategy is summarized as follows.

- An unknown motion is first represented as a sequence of 2-D image frames and directional motion image is constructed from successive optical flow images.
- One representative feature image is selected for each motion.
- Those feature images are projected onto each view-oriented eigenspace.
- An index, representing motion identity within the directional database, is generated after projecting onto each eigenspace.
- For each view, the number of similar motions, termed as candidate motions, is obtained by searching the corresponding B-Tree.
- Finally, these candidate motions are projected onto the global eigenspace as $\mathbf{g}_{m_r}\,(r=1,2,..,D)$, where D is the number of camera directions or views. The unknown motion is projected as $\boldsymbol{g}_m$ within the global eigenspace. The most similar motion is calculated within the global eigenspace using Eq. (5.1).

6.2.5 Recognition results

The experimentation of the system is performed by avatar actors of different size, shape, and the field of view (distance from camera). Ten humanoid avatars performs ten distinct motions: *bend* (bending down), *carry* (carrying a box), *jump* (hopping in a place), *pjump* (jumping up and landing down), *pickup* (picking up something from the ground), *sitdown* (sitting down on a chair), *standup* (standing up from a chair), *stomachache* (touching stomach with pain and sit), *walk* (walking motion), and *wave2* (waving two hands in the air). Five uncalibrated cameras are placed facing the avatar between 0 to 180 degree viewing angles, at 45 degrees apart. Motions are captured at 30*fps* (frames per second) with varying frame numbers performing realistic movements by the avatars. **Figure 6.7** illustrates different motions and their corresponding representative directional MHIs (Section 6.2.3).

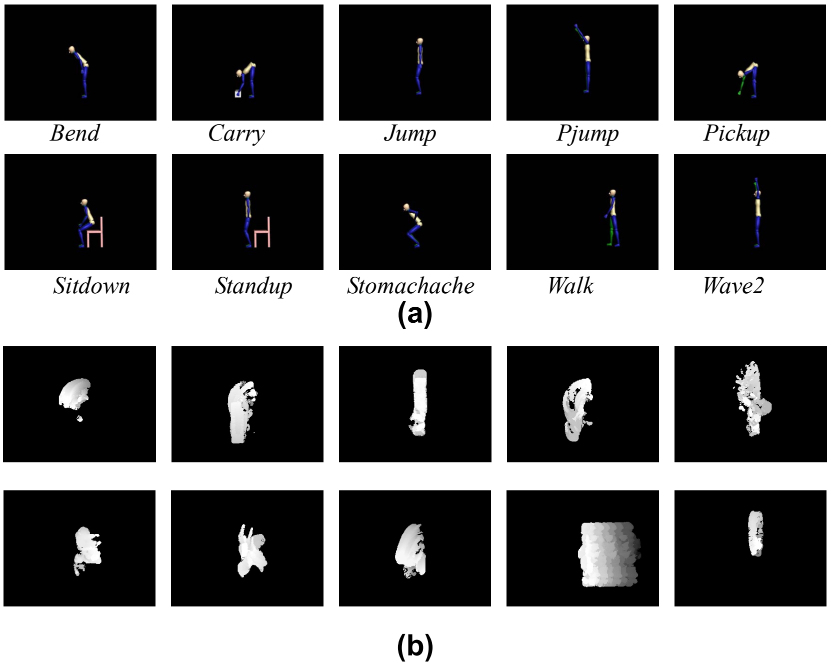


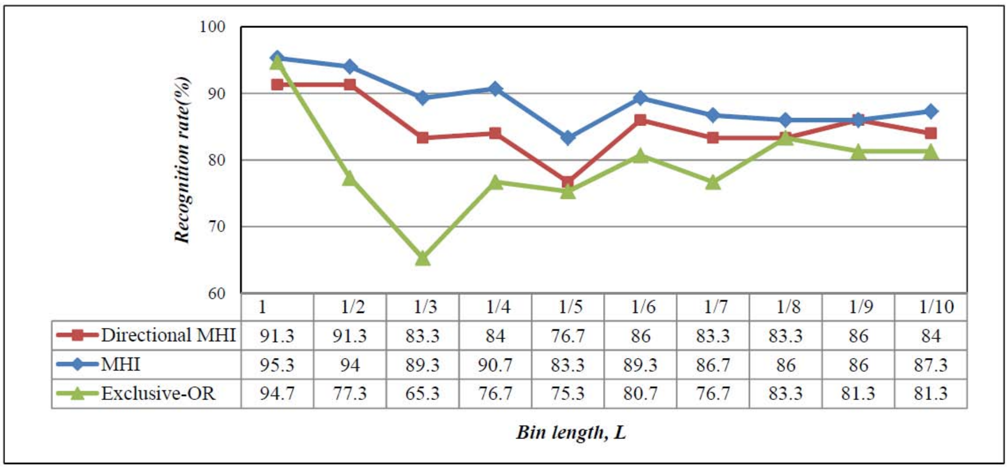
Fig. 6.7 Illustration of representative directional MHIs. (a) 10 types of motions, (b) corresponding directional MHIs.

TABLE 6.1
Performance Evaluation with directional MHI

Bin Length	*Recognition Rate (%)*	*Time consideration*	
		Searching time (*ms*)	Searching rate (*data/ms*)
1	91.3	78.9	4.4
1/2	91.3	50.2	7
1/3	83.3	33	10.6
1/4	84	42.2	8.3
1/5	76.7	40.3	8.7
1/6	86	54.1	6.5
1/7	83.3	55.5	6.3
1/8	83.3	60.2	5.8
1/9	86	60.4	5.8
1/10	84	63.6	5.5
Average	**85%**		**6.89 data/ms**

A total of 500 captured motion data are divided into the training set and the testing set; 350 motion data are used to construct the training set, and the remaining 150 are used for recognition. Varying bin lengths are used for the evaluation of the proposed technique. With large number of motions, if the length of the bins is long, it has higher possibility that many motion points will have an identical index and need much time to searching within the bin. Conversely, if the bin length is short, less motion points will be sought. So, we adopt varying bin length and calculate the recognition rate and searching time for the performance evaluation.

The recognition results are tabulated using the directional feature images in **TABLE 6.1**. We obtained about 92% recognition rate for the system having the bin length of 1. A satisfactory rate of 91.3% recognition rate is achieved at bin length 1/2. However, with shorter bin lengths, the system also performs well. Moreover, we have also analyzed the *database searching time* for the recognition (See TABLE 6.1). We have obtained the average searching rate by calculating average searching time per motion and it was found to be 6.89 data/ms. For bin length 1, the system shows much lower searching rate, whereas subdivision strategy (indexing) increases the searching rate up to a considerable amount. Suppose, for bin length 1, the searching rate is 7 data/ms, whereas for bin length 1/2 it is about 6.5 data/ms. Moreover, we have also tested the recognition using the MHI and XOR representations in place of directional MHI. It shows better performance using XOR images; but a slight decrease in performance using MHI. We obtained the average recognition rate of 89% and 79% using MHI and XOR images, respectively. However, due to the procedure for selecting the appropriate feature image the search time for DMHI is found to be longer than that of MHI, whereas it has shorter search time than that of XOR images. The results are graphically illustrated in **Fig. 6.8** and **Fig. 6.9**.



	1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	1/9	1/10
Directional MHI	91.3	91.3	83.3	84	76.7	86	83.3	83.3	86	84
MHI	95.3	94	89.3	90.7	83.3	89.3	86.7	86	86	87.3
Exclusive-OR	94.7	77.3	65.3	76.7	75.3	80.7	76.7	83.3	81.3	81.3

Fig. 6.8 Comparison of recognition rates using directional MHI, MHI and XOR images

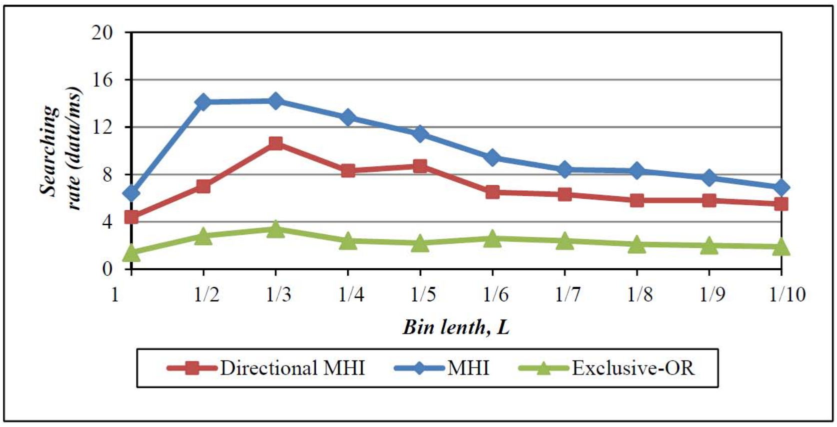


Fig. 6.9 Comparison of searching rates using directional MHI, MHI and XOR images

6.2.6 Discussion

We present a robust recognition technique for identifying and interpreting human motions and actions. This approach uses spatio-temporal directional feature-based representation of motions by exploiting optical flow over time. We adopt the directional MHI to include more precise information than the conventional MHI with no information of flow of motion. We obtained the average recognition rate of 85%, and searching rate of 6.89 data/ms. We notice that its performance is quite deficient than MHI-based system. The reason behind it is that most of our experimental motions are not so much complex for using the concept of directional MHI properly. In the work of [81], more complex motions are captured, and DMHI is proved a prominent one over MHI. However, for simple motions, MHI shows much better representation. Except for this fact, the proposed system with the inclusion of directional information achieves high recognition performance with high-speed recognition. As the directional MHI signifies each motion strongly by making use of the flow of motion, it certainly increases the robustness of the system. So, if the optical flow is computed near accurately, motion recognition with directional MHI leads to robust performance. Moreover, the selection of appropriate representative feature image is also vital to the system. The analyses of these points are left as our future work.

6.3 Cluttered Outdoor Environment

6.3.1 Overview

According to the literature on human motion recognition discussed in Section 3.2, almost all the recognition schemes employ indoor/experimental environment for learning and recognition. Usually, it is an existing demand to recognize human activities in both indoor and outdoor scenes. The system that is capable of working in indoor environment mostly fails to perform well outdoors, since every vision-based system is usually environment-dependent. As with the indoor activities, the background is almost uniform and the illumination condition is almost constant; so it does not affect the system performance in a large degree. On the contrary, the outdoor scenario is much cluttered with non-uniform background, along with subtle movements of background objects (e.g., trees, shadows, sky, etc). Because of the non-uniform nature of outdoor environment, the background, if it is not subtracted and handled properly, it may vastly affect the system's performance. So, the motion segmentation based on background modeling has crucial importance. Referring to the recognition approach presented in Chapter 5, the system is required to be adapted to the system for outdoor environment with real-life activities. Moreover, the viewpoint of different activities is also an important factor at the time of recognition. Most of the recognition systems deal with the activities either facing the camera or parallel to the camera plane [67]. But those systems perform poorly when the activities are to be viewed from back or from other camera angles. Therefore, it is supposed to be concerned to label various actions or activities within the motion database, and to apply those viewpoint-oriented activities to accomplish comprehensive recognition.

In this work, a novel motion recognition system is proposed, having the ability to cope with the clumsiness of the surrounding environment and to recognize activities from different viewpoints. Activities are represented as successive frames extracted from a video. The direction and magnitude of movement between consecutive frames are computed and directional motion templates are generated. We have also made use of the structured database as motion database for efficient storage and retrieval (see Section 5.5). Next sections describe different phases in the development of such a system. A system framework is shown in **Fig. 6.10**.

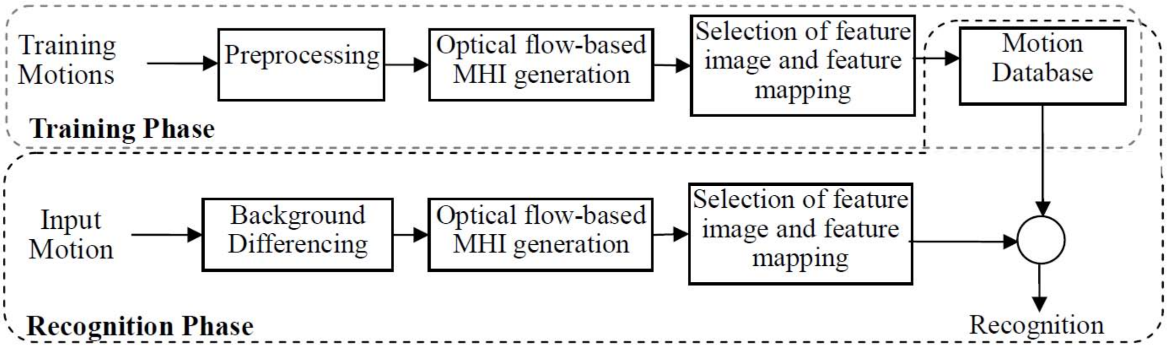

Training Motions
Preprocessing
Optical flow-based MHI generation
Selection of feature image and feature mapping
Motion Database
Training Phase
Input Motion
Background Differencing
Optical flow-based MHI generation
Selection of feature image and feature mapping
Recognition
Recognition Phase

Fig. 6.10 System framework for the cluttered outdoor recognition

6.3.2 Motion segmentation

Motion segmentation refers to the task of segmenting moving region from a video. A video is supposed to be composed of sequence of frames or images. Therefore, in this context, motion segmentation encompasses the preprocessing tasks on the frames to extract the moving region or to suppress the region not being the region of interest in successive frames.

In this work, this task has vital importance, since the system is designed to adapt with the outdoor environment where the static background is cluttered with non-uniformity (movement of unwanted objects) and subtle changes in illumination or lighting conditions. So, motion segmentation should be performed accurately so that those moving regions can be later used for computing the flow of motion. Dynamic adaptive *Gaussian Mixture Model* (GMM) is a very effective technique for background modeling which classifies the pixels of a video frame either background or foreground based on probability distribution [252]. We present a brief description on GMM here.

Gaussian mixture model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

A Gaussian mixture model is a weighted sum of *M* component Gaussian densities as given by Eq. (6.8).

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{M} w_i \, g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6.8}$$

where, x is a D-dimensional continuous-valued data vector (i.e. measurement or features), w_i, $i = 1, \ldots, M$, are the mixture weights, and $g(x|\mu_i, \Sigma_i)$, $i = 1, \ldots, M$, are the component Gaussian densities. Each component density is a D-variate Gaussian function of the form,

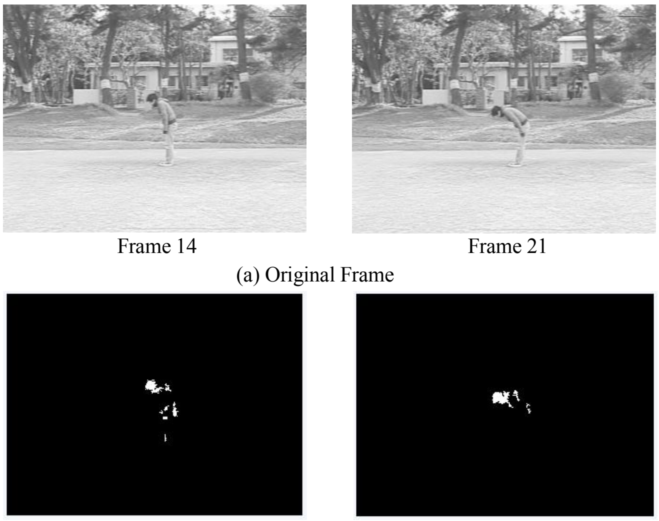
$$g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)' \, \Sigma_i^{-1} \, (\mathbf{x}-\boldsymbol{\mu}_i)\right\} \tag{6.9}$$

with mean vector μ_i and covariance matrix Σ_i. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. The complete Gaussian mixture model is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by the notation,

$$\lambda = \{w_i,\, \boldsymbol{\mu}_i,\, \Sigma_i\} \qquad i = 1, \ldots, M \tag{6.10}$$

There are several variants on the GMM shown in Eq. (6.10). The covariance matrices, Σ_i, can be full rank or constrained to be diagonal. Additionally, parameters can be shared, or tied, among the Gaussian components, such as having a common covariance matrix for all components, The choice of model configuration (number of components, full or diagonal covariance matrices, and parameter tying) is often determined by the amount of data available for estimating the GMM parameters and how the GMM is used in a particular biometric application. It is also important to note that, because the component Gaussians are acting together to model the overall feature density, full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modeling the correlations between feature vector elements. The effect of using a set of *M* full covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians.

In our work, GMM is employed within successive frames to model the background, and thus extract the moving regions (i.e., foreground) between consecutive frames (See **Fig. 6.11**). These extracted regions are used at the time of flow computation. The conventional frame difference technique cannot deal with the non-uniformity of the background, whereas the dynamic form of GMM effectively selects moving points within the cluttered background. From the extracted moving regions obtained using GMM are later used for estimating those points that are tracked for the computation of optical flow to signify motion flow.



Frame 14 Frame 21

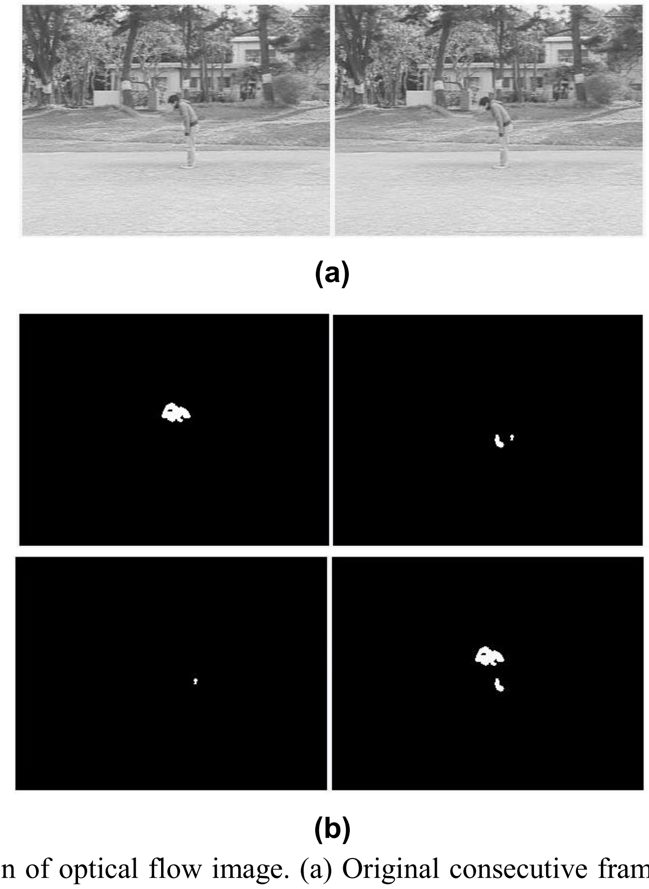
(a) Original Frame

(b) Extracted moving regions at frame 14 and frame 21

Fig. 6.11 Extraction of moving regions using dynamic GMM. (a) Single frame of the bend action, (b) extracted moving regions

6.3.3 Optical flow based MHI generation

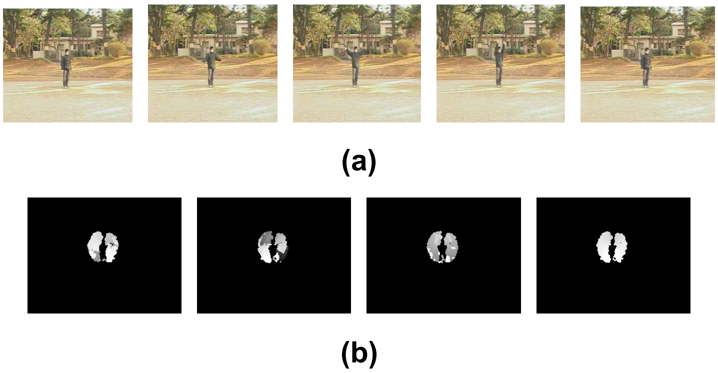
The flow of motion wraps up much information about the moving regions. This information is gathered by analyzing two consecutive frames. Optical flow is the most widely used flow computation technique. An optical flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the subtle change of motion in temporal direction. Our concerned flow of motion is estimated and/or computed by determining the optical flow between successive motion frames. As the similarity measurement between different motions depends on both spatial and temporal information, optical flow can be used as a discriminative feature for determining the correlation (See **Fig. 6.12**). Similar to the optical flow based directional feature image generation described in Section 6.2, we generate the same direction of flow represented motion representation, *directional MHI*. An example of the direction MHI is shown in **Fig. 6.13**. Moreover, the selection mechanism for the representative feature images are same as mentioned in Section 6.2.2. The reader is referred to Section 6.2.1 and 6.2.2 for the detailed methodology for the generation of the directional feature images.



(a)

(b)

Fig. 6.12 Computation of optical flow image. (a) Original consecutive frames, (b) intensity image for four separate channels of x and y. top row: x^-, x^+ , bottom row: y^-, y^+.



(a)

(b)

Fig. 6.13 Generation of four-directional MHIs. (a) Waving-two-hands action, (b) four directional MHIs, x^-, x^+, y^-, y^+, respectively.

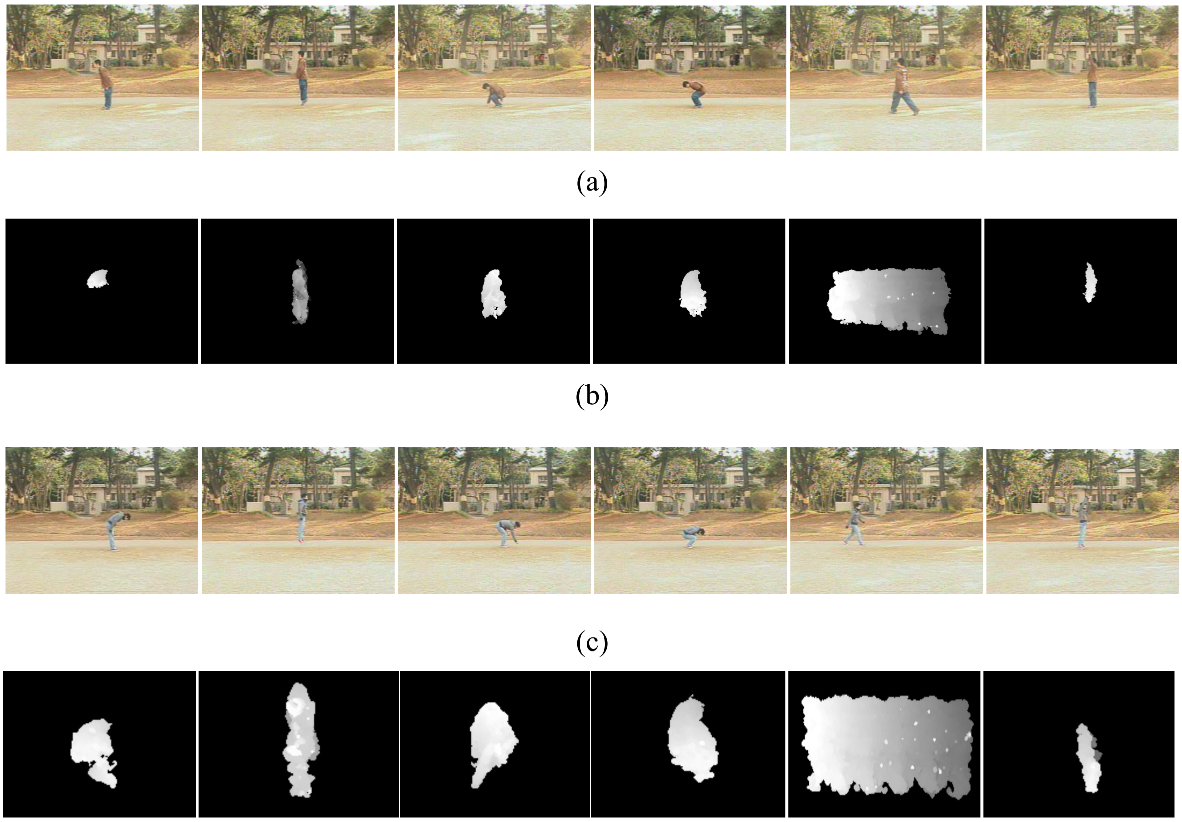
Likewise, the representative images for the learning motions are used to construct an *eigenspace* (See Section 3.4). The motion points are then indexed, and stored in the B-Tree structured database for the task of recognition. Due to the fact that the number of experimental motions is not large enough, we have not adopted the directional organization concept of the motion database.

6.3.4 Recognition strategy

The recognition strategy is also simple, but effective. When an unknown motion comes, it is first represented as a sequence of image frames and is processed for generating motion representation. Then 4-directional MHIs are obtained from the motion frames. Among the four MHIs, the MHI capable of signifying most of the features is selected. Then the feature image is projected onto global eigenspace. An *index*, representing motion identity within the directional database, is generated from the unknown motion after projection onto the eigenspace. The most similar motion is obtained by searching the corresponding B-Tree by calculating number base-based difference among the indexes. If the query index resides within the database, it is found by simple *index comparison*. But for the case of the index not residing within the database, the task becomes finding the most similar index which corresponds to the most similar motion within the database. The candidate motions, either residing within the same bin or within the nearest bin, are projected onto the eigenspace. The unknown motion is also projected within the eigenspace. The most similar motion is calculated within the global eigenspace using Eq. (5.1).

6.3.5 Experimental results

The evaluation of the system is performed by capturing human activities in outdoor with non-uniform, dense, unstructured background. Five human actors performed six distinct activities with movement overlapping: *bend* (bending down), *jump* (hopping in a place), *pickup* (picking up something from the ground), *stomachache* (touching stomach with pain and sit), *walk* (walking motion), and *wave2* (waving two hands upwards and moving down). Four uncalibrated cameras are placed in different angles relative to the human body. Moreover, they are captured at 30*fps* (frame per second) with varying speed to perform each motion. Optical flow is generated for each motion. As the system performance vitally depends on the generation of the optical flow and its separation into four channels, the optical flow computation technique adopted in [253] is modified to adapt to the current circumstance. The renowned Lucas-Kanade optical flow technique is used in the current work. After computing optical flow, the MHIs are constructed. Some examples of the directional MHIs generated and selected to represent the activities are shown in **Fig. 6.14**.



(a)

(b)

(c)

(d)

Fig. 6.14 Representation of MHIs for each motion from different viewpoints. (a) Original motion, one frame each, (b) represented directional MHIs of (a), (c) original motion, one frame each, (d) represented directional MHIs of (c).

TABLE 6.2
Recognition rate (%) for the experimental motions

Bin Length	Recognition Rate (%)
1	80
1/2	70
1/3	66
1/4	67
1/5	70
1/6	64
1/7	73
1/8	64
1/9	63
1/10	71

Varying bin lengths are used for the evaluation of our proposed recognition technique by simulation. With large number of motions, if the bin length is high, there is higher possibility that more motion points will have an identical index and need much time to search similar motions, and vice versa. Total of 100 captured motion data are divided into the training set and the testing set by taking into account *leave-one-out* cross-validation strategy, where one data sample is used for testing whereas the remaining data constitute a training set. Thus the motion database are constructed and tested against the testing data for recognition. There is much variability in the simulation as the motion is captured from variable viewpoints and overlapping moving regions. We have tabulated the recognition results in **TABLE 6.2**. We obtained a maximum of 80% recognition rate with the system. Due to the elimination of some significant pixels in generating the MHIs due to cluttered background subtraction, similar dress color with background and selection of field of view, the misrecognition occurs.

6.3.6 Discussion

The experimental results provide performance evaluation for the recognition system using directional MHI representations. We obtain satisfactory recognition rate with the system. The system deals with the outdoor motion data which is core issue in the novelty of the system. However, we see that the recognition results, though satisfactory, sometimes fails to recognize motions due to the failure of characterizing the motions, and thus less features are tracked in optical flow. This is due to the factors such as non-uniformity of illumination, distance of the actor from a camera, synchronization of the learning motions, dress color effect, movement of background objects, shadows, temporary presence of other objects, etc.

So, the system is subjected to be upgraded to obtain higher recognition rate by adapting with the aforesaid environment conditions.

6.4 Summary

In this chapter, we have presented two methods to deal with two constraints in human motion recognition. At first, we proposed the directional flow included motion representation to deal with the problem of directional motion overwriting problem. We have adopted our structured motion database to implement this concept, and also performed experimentation to evaluate the effectiveness of the technique. After that, we proposed a recognition system for the cluttered outdoor environment, and used GMM as the background modeling technique to extract the significant spatial information from the motions. We also performed experiments with the outdoor motions and obtain satisfactory performance within the cluttered outdoor scenario. We have also analyzed the advantages and limitations of the both techniques in separate sections.

Chapter 7
Multiple Persons' Action Recognition

7

Multiple Persons' Action Recognition

7.1 Overview

For the task of motion analysis, the detection and recognition of generic objects present an important issue. In a scene, there exist several moving objects in practical cases (See **Fig. 7.1**). So, to figure out the motion of either a human or other moving objects, it becomes necessary to perform detection procedure within the scene and then segment the scene from non-interest moving objects. It is possible to detect generic objects, for example, human, car, bicycle, tree, building, mountain, forest, and so on. There are a lot of challenges for object detection in a scene involving wide variety of articulated poses, variable appearance/clothing, complex backgrounds, unconstrained illumination, occlusions, etc. A number of object detectors are available that include Rectified Haar Wavelets [254], Histograms of Oriented Gradients (HOG) [255], Edge images [256], Haar-like Wavelets and Space-time Difference[257], 1st and 2nd Order Gaussian Filter [258], and many others. Dalal and Triggs [255] proposed HOG for human detection that has drawn much attention of many researchers for its effectiveness in detection. This method is able to detect a human region accurately and fast by extracting the gradient information from an image and exploiting the information for determining human existence. In [259], the cascade-of-rejectors based HOG method is proposed that is found to be better than the basic HOG in terms of higher processing rate, variable sized-blocks, and selection of best feature for detection. Another form of HOG is introduced in [260] to deal with the motion of subjects, the camera and the background and to variations in pose, appearance, clothing, illumination and background clutter. This method combines motion-based descriptor (differentials of optical flow) and HOG descriptor. Recently, an efficient HOG human detection is proposed that reuses the features in blocks to construct HOG feature for intersecting detection windows, and also utilizes sub-cell based interpolation to compute HOG features efficiently [261]. In [262], HOG-based human detector is adopted with partial occlusion handling mechanism.

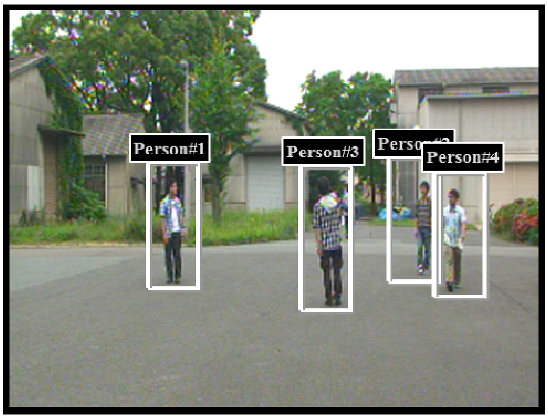


Fig. 7.1 An example of several persons occupying a scene

Several other approaches have been adopted to accomplish object detection/classification task. Bag-of-Words (BoW) is used in computer vision for object categorization [263]. This technique is also known as "Bag of Features model". The BoW model allows a dictionary-based modeling, and each document looks like a “bag” (thus the order is not considered), which contains some words from the dictionary. Computer vision researchers use a similar idea for image representation (Here an image may refer to a particular object, such as an image of a car). For example, an image can be treated as a document, and features extracted from the image are considered as the “words”. It usually includes following three steps: feature detection, feature description and codebook generation. A definition of the BoW model can be the “histogram representation based on independent features”. The BoW method has been used for human action classification in [203]. In this approach, a hierarchical model is proposed that can be characterized as a constellation of bags-of-features and that is able to combine both spatial and spatial-temporal features. Moreover, BoW model is also used for scene understanding and proved to be quite successful [212].

Among these methods, HOG has drawn the attention of many researchers for its effectiveness in detection. This method is able to detect a human region accurately and fast by extracting the gradient information from an image and exploiting the information for determining human existence. Thus it has been employed successfully to various environments in the recent years. In the earlier researches, the HOG features were mostly used for detection of human in successive frames, whereas we have employed these features in a detection-tracking fashion. A number of approaches have been adopted for action recognition since last decade. Robustness as well as accuracy is the key factor for the recognition.

In this chapter, we shall present a HOG-based multiple persons’ action recognition technique by integrating the human detection, tracking and action recognition techniques.

Moreover, we have also developed a Structured Motion Database (SMoDB) which is developed for action recognition that is capable of high-speed and high-precision searching within the registered actions (See Chapter 5). The system framework is illustrated in **Fig. 7.2**. According to the framework, the overall system is divided in order of its phases: *fast human detection*, *feature tracking, and recognition*. We shall discuss these phases in subsequent sections.

7.2 Fast Human Detection

Each individual is detected within a video sequence by fast human detector. In our work, Histograms of Oriented Gradients (HOG) method is adopted for the detection. Here below we describe the procedure of fast human detection in brief.

7.2.1 Object detection

The object detection architecture is based upon a method for classifying individual image regions. This is divided into two phases. The *learning phase* creates a binary classifier that provides object/non-object decisions for fixed sized image regions ("windows"); while the *detection phase* uses the classifier to perform a dense scan reporting preliminary object decisions at each location of the test image. These preliminary decisions are then fused to obtain the final object detections. Both the learning phase and the detection phase contain three stages which are depicted in **Fig. 7.3**. Overall this defines a fixed and relatively simple architecture for object detection. The final detector performance depends on the accuracy and reliability of the binary classifier and on how multiple detections are fused during the detection phase.

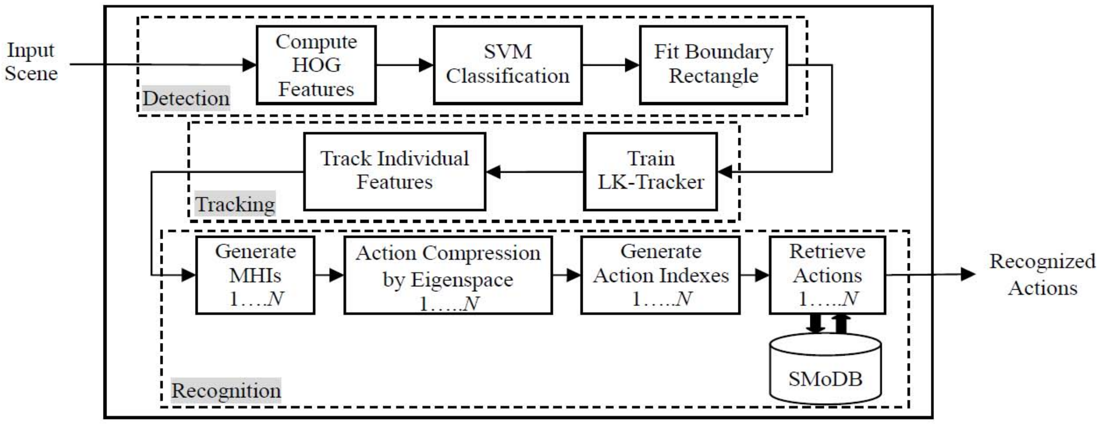


Fig. 7.2 System framework having detection, tracking, and recognition module

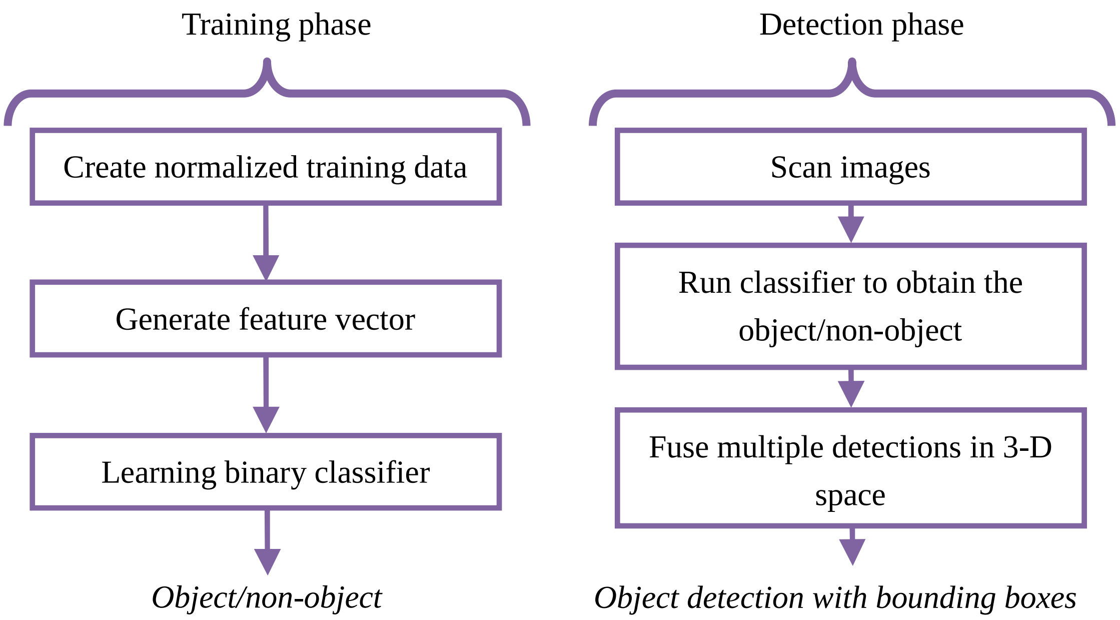

Training phase
Create normalized training data
Generate feature vector
Learning binary classifier
Object/non-object
Detection phase
Scan images
Run classifier to obtain the object/non-object
Fuse multiple detections in 3-D space
Object detection with bounding boxes

Fig. 7.3 Object detection architecture. (a) The learning phase extracts robust visual features from fixed size training windows, and trains a binary object/non-object classifier over them. (b) The detection phase uses the learned binary classifier to scan the test image at all locations for object/non-object decisions. These preliminary decisions are later fused to produce the final object detections.

The first stage of learning is the creation of the training data. The positive training examples are fixed resolution image windows containing the centered object, and the negative examples are similar windows that are usually randomly subsampled and cropped from set of images not containing any instances of the object. The binary classifier is learned using these examples. Ideally, each positive window contains only one instance of the object, at a size that is approximately fixed w.r.t. the window size. In some cases, the windows contain only a limited number of points of view of the object. However, the simple window architecture has various advantages. It allows a conventional classifier to be used for detection and relieves the classifier of the responsibility to be invariant to changes in position and scale (although invariance to other types of transformations, changes in pose and viewpoint, and illumination still has to be assured). It also means that the classifier works in relative coordinates (feature position relative to the center of the current window) which allows relatively rigid template-like feature sets to be used. On the other hand, it means that the classifier is run on a large number of windows, which can be computationally expensive and which makes the overall results very sensitive to the false positive rate of the classifier.

The image feature extraction process maps image windows to a fixed size feature space that robustly encodes visual form. These feature vectors are fed into a pattern recognition style classifier. Any classifier can be used for the purpose, but SVM or AdaBoost is common. In this thesis, we have selected a simple, reliable classification framework as a baseline classifier for most of the experiments. We use linear SVM as our baseline binary classifier as it proved to be the most accurate, reliable and scalable. Three properties of linear SVM make it valuable for comparative testing work: It converges in a reliable and repeatable manner during training; it handles large data sets gracefully; and it has good robustness towards different choices of feature sets and parameters. As the linear SVM works directly in the input feature space, it ensures that the feature set is as linearly separable as possible, so improvements in performance imply an improved encoding.

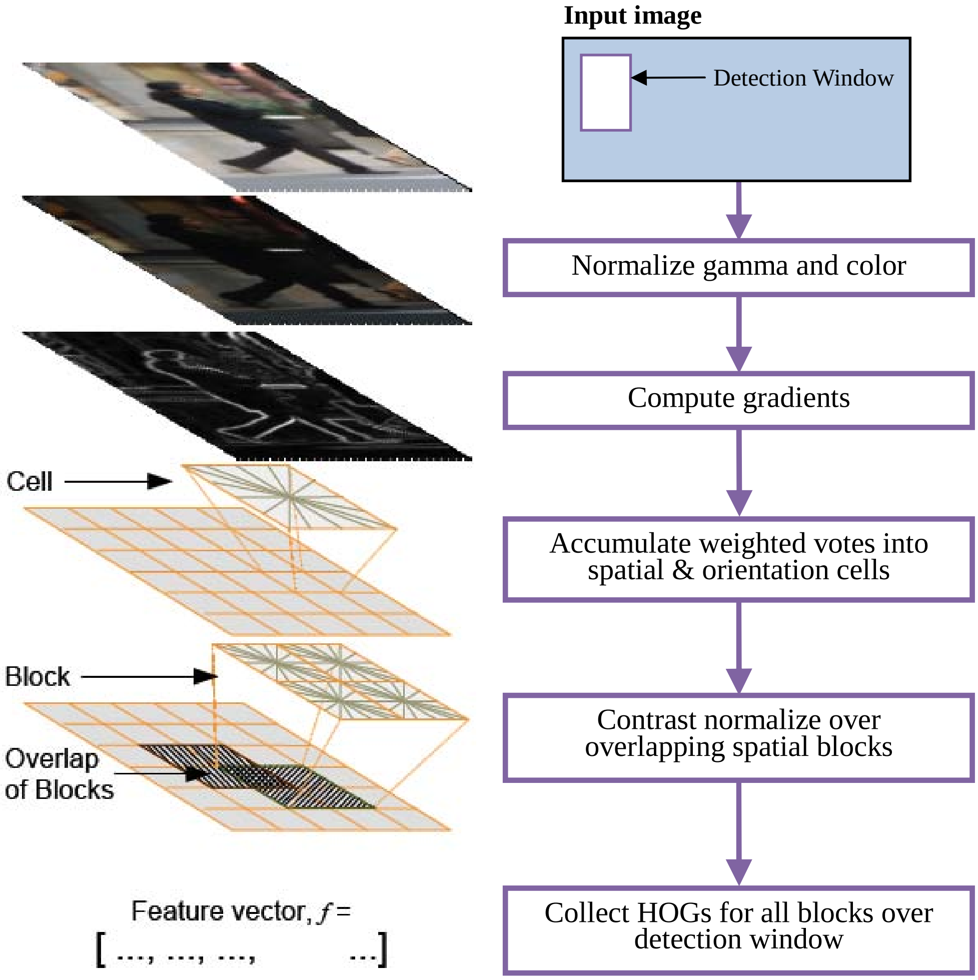

Input image
Detection Window
Normalize gamma and color
Compute gradients
Accumulate weighted votes into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOGs for all blocks over detection window
Cell
Block
Overlap of Blocks
Feature vector, f =
[..., ..., ..., ...]

Fig. 7.4 An overview of HOG feature extraction. The detector window is tiled with a grid of overlapping blocks. Each block contains a grid of spatial cells. For each cell, the weighted vote of image gradients in orientation histograms is performed. These are locally normalized and collected in one big feature vector.

7.2.2 Histogram of Oriented Gradient (HOG) descriptors

HOG was originally proposed by Dalal and Triggs in 2005 [255]. In HOG, the local object appearance and shape is characterized by the distribution of local intensity gradients or edge directions without precise knowledge of the corresponding gradients or edges. It is significantly robust to shape or illumination change. These gradient features are collected over a search window of an image. The search window is divided into small spatial regions, termed as cells, for each cell represents the local histogram of oriented gradients over the pixels contained in the cell. The histograms consist of the gradient orientations into a number of bins. Each orientation bin for HOG computation is evenly spaced over 0˚-180˚ that constitutes a 9-bin histogram. **Figure 7.4** presents the complete processing chain of the feature extraction algorithm. In practice, the implementation differs slightly from that presented in Fig. 7.4. Certain stages are optimized for efficiency.

The HOG representation has several advantages. The use of orientation histograms over image gradients allows HOGs to capture local contour information, i.e. the edge or gradient structure which is very characteristic of local shape. In conjunction with the spatial quantization into cells, it allows those to capture the most relevant information with controllable precision and invariance (e.g. by changing the number of bins in orientation histograms and the cell size). Translations and rotations make little difference so long as they are much smaller than the local spatial or orientation bin size. For example, in the human detector we find that rather coarse spatial sampling, fine orientation sampling and strong local photometric normalization turns out to be the best strategy, presumably because this permits limbs and body segments to change appearance and move from side to side provided that they maintain a roughly upright orientation. Gamma normalization and local contrast normalization contribute to another key component, illumination invariance. The use of overlapping blocks provides alternative normalization so that the classifier can choose the most relevant one. These steps ensure that as little information as possible is lost during the encoding process. Overall encoding focuses on capturing relevant fine grained features and adding the required degree of invariance at each step.

7.2.3 Integral image/histogram

The integral image is an intermediate representation for images which allows a rapid computing of rectangular features. The integral image at location *x*, *y* contains the sum of the pixels above and to the left of the pixel (*x*,*y*), including (*x*,*y*).

$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \tag{7.1}$$

where *ii*(*x*,*y*) is the integral image and *i*(*x*,*y*) is the original image.

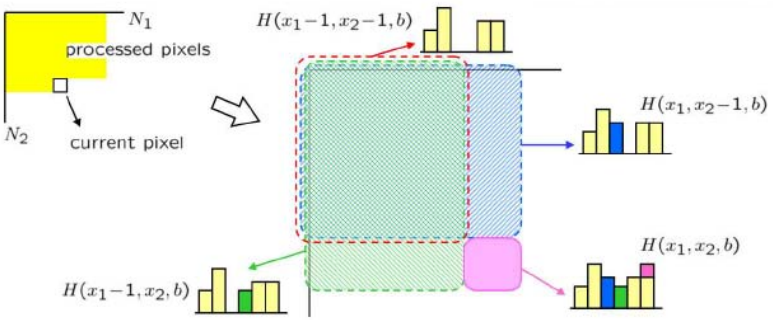


Fig. 7.5 Concept of integral histogram. Yellow indicates already traversed points. At each step, the current integral histogram is obtained from the integral histogram values of the three neighbors, and the bin that corresponds to current point's value is increased by one.

Porikli [94] suggested a similar method to efficiently compute histograms over arbitrary rectangular regions, called '*Integral Histogram*' (See **Fig.7.5**). The scan of the image requires updating the integral histogram for such data points that their left, upper, and upper-left neighbors are already scanned in case of an image data. The integral histogram at a point is obtained by three arithmetic operations for each bin of using the integral histogram values of the three neighbors as shown in Fig. 7.5. The integral histogram values of the previous point are copied to the current point before the propagation. Either the updated bin is copied to all of the remaining points, or all the previous bins are copied to the current bins.

7.2.4 Integral HOG

Inspired by the concept of integral histogram, Zhu et al. [259] developed a fast way to calculate the HOG feature, which is called *integral HOG*. First of all, each pixel's orientation (including its magnitude) is discretized into 9 histogram bins. Next, an integral histogram is computed and stored for each bin of the HOG. And finally they are used to compute the HOG efficiently for any rectangular image region. This method is fast to compute, while there are some differences with the original HOG method:

- First, they could not use a Gaussian mask and tri-linear interpolation in constructing the HOG for each block because this would not fit with the integral histogram approach.
- Second, they used L1 normalization instead of L2 normalization because it allowed a faster computation when using the integral images.

7.2.5 Human detection with integral HOG descriptor

A larger spatial region, termed as *block*, is formed by accumulating the cell-based oriented histograms. Contrast-normalization is accomplished to be invariant to illumination, shadows, etc. These normalized descriptor blocks are referred to as HOG (See **Fig. 7.6**). In our work, we have adopted another fast HOG-based method employing the integral histogram to detect the humans in a scene. Original HOG is quite time-consuming, since for every search window it computes the oriented gradients even though for some blocks within the neighboring windows it is already computed. On the contrary, the integral histogram representation allows rapid computing of the rectangular feature by pre-computing the cumulative histograms for all the bins corresponding to each point on an image. Finally, those are used to compute the HOG features for the blocks within the search window. Thus ambiguous block-based feature computation is eliminated. This makes the HOG feature generation accurate and fast.

After the detection of a probable human region, the classifier classifies the region whether there is a human or not. If a human exists, it is detected by a rectangular region enclosing the human. Support Vector Machine (SVM) is used to construct the classifier where a strong classifier is constructed from a number of weak classifiers. Accordingly, a scene with many persons is segmented into several regions based on the *number* of the occupying persons. The corresponding regions are detected in the scene. Thus the detector output provides several bounding boxes corresponding to the detected humans in a scene. However, in order to use the detector output for recognition, the distinguishable features corresponding to each human region are to be tracked in subsequent frames to accumulate the information of the motion flow.

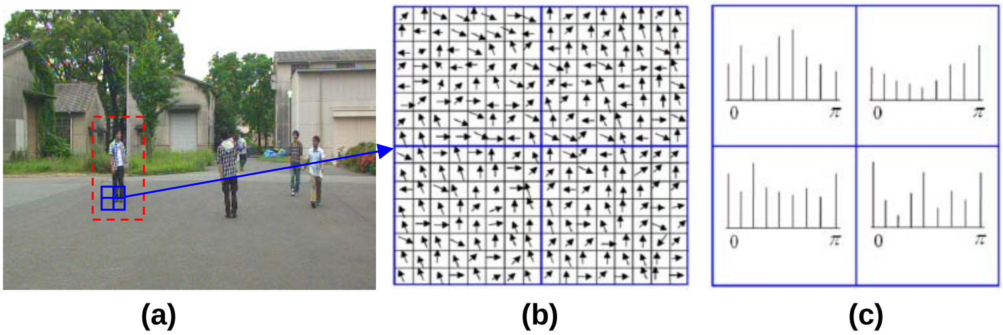


Fig.7.6 Histograms of oriented gradients: (a) A 64X128 search window (a rectangle enclosing the leftmost person), (b) a 16X16 block of 4 cells, (c) the histograms of oriented gradients corresponding to the 4 cells.

7.3 Feature tracking

Feature tracking is a process of finding the important features of an *object of interest* from one frame in a subsequent frame of a video stream. In order to keep track of the motion flow, the features corresponding to each person is tracked in successive frames. We have employed the well-established Lucas-Kanade Tracker for person-wise feature tracking. Lucas-Kanade optical flow tracker itself does not work very well because it works on a large window and a large window is too often unable to track smaller region of features. To solve this problem, we can track first over larger spatial scales using an image pyramid and then refine the initial motion velocity assumptions by computing down the levels of the image pyramid until we arrive at the raw image pixels. Hence, the technique is first to solve for optical flow at the top layer and then to use the resulting motion estimates as the starting point for the next layer down. We continue going down the pyramid in this manner until we reach the lowest level. Thus we minimize the violations of our motion assumptions and so can track faster and longer motions. This technique is known as *pyramid* Lucas-Kanade method (See **Fig. 7.7**).

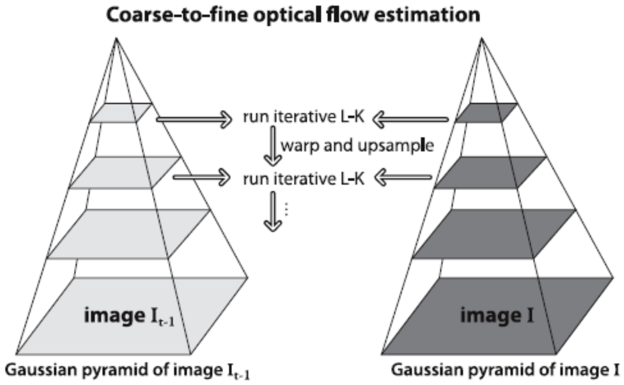
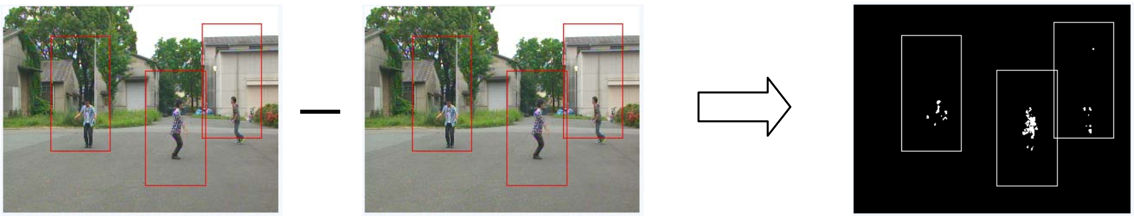

Coarse-to-fine optical flow estimation
run iterative L-K
warp and upsample
run iterative L-K
image I_{t-1}
image I
Gaussian pyramid of image I_{t-1}
Gaussian pyramid of image I

Fig. 7.7 *Pyramid* Lucas-Kanade optical flow.



Frame at *t*

Frame at *t*+1

Computed optical flow

Fig. 7.8 Optical flow computation from L-K tracker output.

In our system, the tracker is first provided with the prominent features which are to be tracked at every time-stamp. Then the tracker tracks those feature in the next frame. These tracked features are properly clustered on the frame to maintain individual feature sets. The intra-frame features are tracked and the inter-frame tracking synchronization is maintained for each individual. However, the HOG classifier output initially guides the feature tracker with the region of interest. Afterwards, the moving region corresponding to a particular action performed by a particular person is resized based on the tracker output. In this way, we compute the *optical flow* of a motion (See **Fig. 7.8**). The optical flow vectors are used in the recognition phase to construct a template for the action to be recognized.

7.4 Action Recognition

In the action recognition phase, the segmented moving regions for each individual are passed through a template generator to construct an accumulated feature image. We have adopted MHI template (See Section 2.2) as an action representation. The subsequent operations for the action recognition are described briefly as follows:

- An MHI is generated from frame sequence corresponding to an action.
- A number of directional eigenspaces are computed with the generated MHIs of the actions corresponding to each camera viewpoint (See Section 5.4).
- An index is generated for each motion points within the eigenspace, and those indexes are stored in a Structured Motion Database (SMoDB) for the task of recognition (See Section 5.5). The database development is done at learning phase.
- For the purpose of action recognition, an input motion is searched within the motion database and the most similar motion is obtained by the scheme described in Section 5.6, except that the prior direction estimation scheme is discarded.

7.5 Performance Analysis

The experiments were performed for the performance evaluation of the proposed system. For training the SVM human classifier using HOG features, 1506 positive training samples and 1226 negative training samples (negative training images are further sampled into 10 samples corresponding to each negative image) extracted from INRIA Dataset and other outdoor images were used (See **Fig. 7.9**). For action learning, five types of actions were captured: *Pickup, Jump, Jogging, Headache, Fall-Down* (See **Fig. 7.10**). Six actors performed total of 90 actions which were captured from three directions, namely, *front*, *left* and *right*. The learning actions have varying frame numbers ranging from 29 to 130 frames per action representing either fast or slow movements. The block size was 2X2 cells with cell size 8X8 pixels. The search window size was 64X128 pixels having 105 overlapping fixed-sized blocks. As each cell represents 1-D 9-bin histogram features, correspondingly each block consists of 36-D feature vector, and a total of 3780 dimensional feature vector was obtained. To test the detector performance, we have used 1126 test samples. From this experiment, 98% detection rate was obtained with the trained SVM classifier.

Fig. 7.9 Some examples of positive training samples.

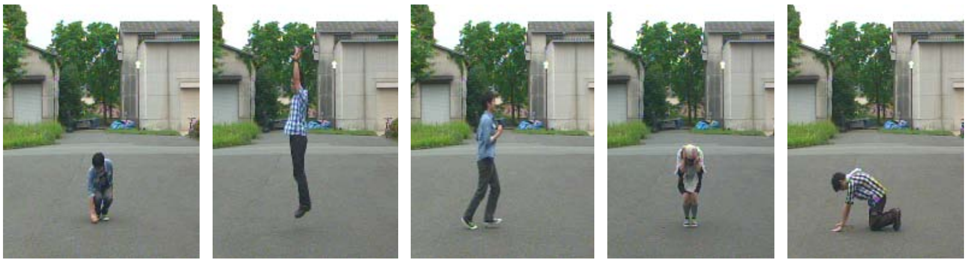
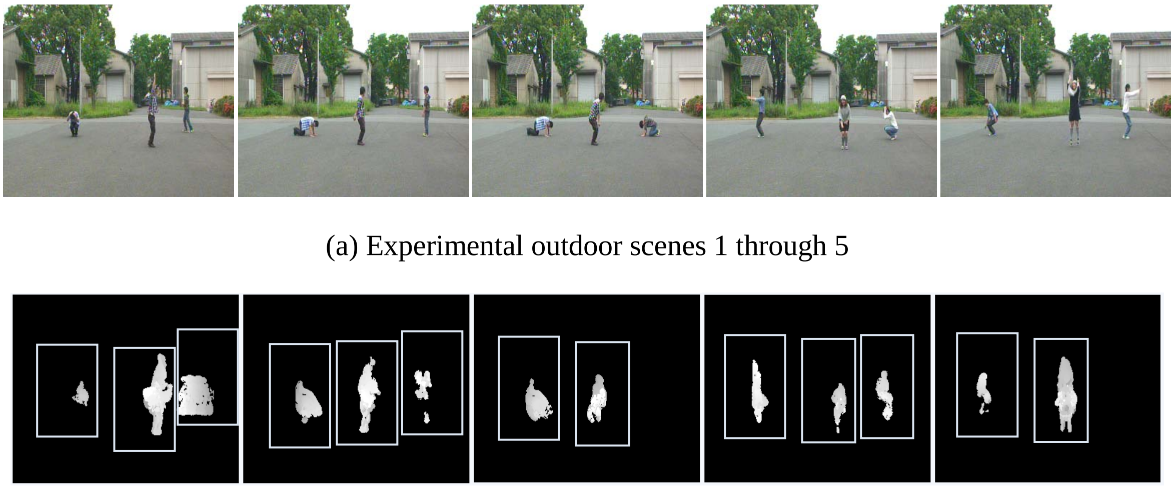


Fig. 7.10 Some learning actions captured from different directions.



(a) Experimental outdoor scenes 1 through 5

(b) Corresponding generated MHI motion templates for each individual

Fig.7.11 Experimental scenes and corresponding motion templates within the rectangular region for each person

TABLE 7.1 Experimental Results with five scenes.

	Detection	Tracking	Recognition
Scene 1	100%	55%	100%
Scene 2	100%	54%	67%
Scene 3	67%	52%	100%
Scene 4	100%	41%	100%
Scene 5	67%	42%	100%

To test our overall system, we have considered five outdoor scenes where multiple persons were performing different actions. **Figure 7.11** shows the detection results which demonstrate the robustness of the system in terms of detection, tracking, and recognition. The outputs of the HOG detector and the recognizer are analyzed separately. We have tabulated the experimental results in terms of detection rate, rate of successfully tracked points, and recognition rate for the five scenes in **TABLE 7.1**. We found the average detection rate for our experimental scenes is 87%. The average of successfully tracked features is about 49%. Finally, we have obtained the average recognition rate of 93% corresponding to the detected humans. In scene 3 and scene 5, one person each was found undetected and the actions were also not taken into account for recognition. However, tuning of the fusion parameter for human detection may be able to detect the human.

However, we have calculated the vital time requirement factors. Most of the time elapses within this system is due to *human detection*, *feature tracking*, and *MHI generation for all the existing persons*. We found that human detection takes most of the time (about 75%-80% of the total time), whereas the feature tracking and MHI generation steps does not take much time (20%-25% of the total time). **Figure 7.12** graphically illustrates the time requirement for the above factors.

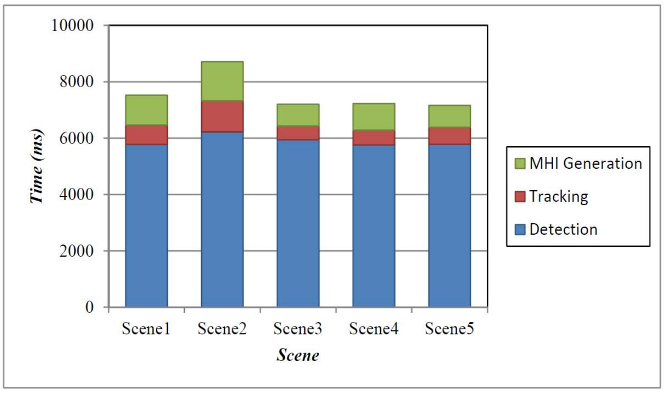


Fig. 7.12 Time distribution for major three steps

7.6 Discussion

We propose a novel multiple persons' action recognition system. This system is robust in detection, tracking and recognition in the sense that it can deal with the cases where several persons perform similar or dissimilar actions in a scene with unknown background, whereas the earlier researches have not adopted such an integrated strategy to handle this sort of situations. Moreover, in some cases, background subtraction is employed against known background: But our system is more generalized than those methods. Our proposed system is also capable of fast detection and recognition. Since an outdoor environment is usually non-symmetric and cluttered by nature, it is quite difficult to establish a system that can cope with each and every situation for human detection and its action analysis. In our system, we have adopted the following assumptions:

a. At each time stamp, the left and rightmost regions are searched for possible human entries.
b. For overlapping of multiple humans, the system is subjected to be terminated for the overlapping humans. A threshold is set for detecting the occurrence of occlusion.
c. If a reasonable number of feature points are not tracked in subsequent frames, the person is assumed to be out of the visible area.
d. All the persons in a scene perform actions simultaneously.

Although there are some limitations and assumptions in the form of occlusion and false detection, it is proved to be an efficient approach according to the performance of the system in terms of precision.

7.7 Summary

In this chapter, we have presented a human action recognition system capable of dealing with several persons' actions. We have discussed the framework for the recognition system. The overall system is divided into three phases: human detection, person-wise feature tracking, and action recognition. We have discussed each phases, and also performed experiments with several persons in outdoor environment. Finally, we have focused on some issues regarding the advantages and constraints of the system.

Chapter 8
Discussion and Conclusions

8

Discussion and Conclusions

8.1 Discussion

Our contribution has a number of merits over other existing methods which are intensively surveyed in different chapters of this thesis. Also, there are some limitations, and points of improvement for our proposed methods. As our contribution emphasizes on human motion acquisition and human motion recognition, we shall discuss on these topics separately in Section 8.1.1 and 8.1.2. In Section 8.2, we summarize our overall contribution, and finally, Section 8.3 presents some future research directions.

8.1.1 Human motion acquisition

Our proposed human motion acquisition system focuses on the modeling of human body in successive frames. It is based on the acquisition and understanding of limb movements. The proposed model employs a simple body parts or joints extraction scheme to model the motion. However, some matters are still unsolved which include the modeling of torso or trunk, accurate positioning of the joints, shape deformation of body and body parts, and so on. We have also assumed the human having no body parts occlusion and all the joints are mostly visible throughout the movements. However, the above factors are quite hard to deal with when there is a case of pose analysis in non-3-D environment. Our proposed scheme, though it employs a semi-3-D form of the model, is quite difficult to present the subtle body parts movements in such a form. The reason for not choosing the 3-D form of motions lies in the fact that it is computationally expensive and, having surveyed various related works, we notice that the accuracy of the model is really much perfect as the integration of different view's information is not an easy task. In this respect, our proposed method with two camera views performs satisfactorily for simple body movements. However, it may be worthwhile to use another frontal view camera to solve the ambiguities of limbs, and overlapping of body parts. Moreover, the ultimate target of such a kind of acquisition system is to extend to the self-development of motion database in real-time by self-learning of robots. At the same time, when an unknown new motion comes, it is acquired by a robot and the motion database is updated with the new one correspondingly. Based on this, we shall specify some ideas on the development of such a system in Section 8.3.

8.1.2 Human motion recognition

We present a novel recognition technique for identifying and interpreting human motions or actions utilizing a structured motion database. This approach used spatio-temporal representation of motions. In such an approach, the directional organization of motion database is adopted: Motion recognition of an unseen motion is obtained by searching the selected directional databases: And the nearest neighbor searching problem is resolved. A motion is represented by Motion History Image and Exclusive-OR image, and these are used to construct eigenspace. We obtained the recognition rate 94% and the recognition time is about 20 milliseconds. Thus the proposed system has achieved high recognition performance with high-speed recognition.

Although we have achieved satisfactory performance from our proposed recognition system, there are, of course, some limitations in the current system. A more sophisticated motion detection technique might increase the robustness of the recognition system. Moreover, with respect to the structured motion database, the selection of bin length is an important issue to solve. This can be solved by adopting specific datasets and the corresponding *best* bin length obtained empirically. Suppose that *tennis action dataset, cricket action dataset, kitchen activity dataset*, etc., imply the *best* bin length separately. Similar with the bin length, the scale of shifting should also be chosen in such a way to produce the best results. However, the system incorporates all the frames to generate the motion images. Therefore, a strategy is required to deal with this limitation.

We have also presented two significant constraints in human motion recognition in the form missing information and recognition in cluttered outdoor scenario. With the first constraints, though the directional information increases the robustness of the system, there should be some selection for the appropriate feature image. It is also recommended to use all the four directional feature images so that significant information is not lost. Rather than this, we consider to adopt a mechanism to switch the representation scheme between MHI and directional MHI, since MHI seems performing better in the case of simple motions. For the second constraint, the motions are captured in cluttered outdoor scenario, and the motion recognition scheme is applied on these captured motions. Though it shows reasonable performance for the outdoor data, practically its performance is subjected to be improved to be able to use it in real-life applications.

We propose another human motion recognition system by considering the practical cases where several persons occupy a scene. The system works well with the experimental outdoor motions. Analyzing the time requirement for various phases of the system, we find it a bit long in terms of the speed concerned. We may use the AdaBoost or other boosting algorithm to select prominent HOG feature (e.g., [259]), rather than selecting all the features. This will certainly reduce the detection time, and will make a high-speed recognition system.

However, with some limitations, our proposed system emphasizing certain factors works well and can be upgraded to adapt to the environment. The significant merit of the contribution is that it works with the extracted motion frames and the normalized human posture images. In other words, we are using only the relative posture change within successive motion frames to recognize a human motion. Therefore, if it is possible to extract a human region firmly, the system can recognize the motion using even low resolution cameras, in both indoor and outdoor. Another noticeable thing inside the implementation of the system is the employment of separate eigenspaces for each camera direction which corresponds to a system with multi-camera concept. Some fundamental advantages of the proposed technique are as follows;

(i) The human motions observed from multiple directions can be dealt with numerically using the eigenspace concept.

(ii) The motion database is also in compact form, since only one motion image is utilized for each motion with several frames.

(iii) The proposed technique is advantageous with respect to computational load, recognition rate, and steadiness of performance.

There is broad area of applications for such a human motion recognition system. The most desirable one is the control of an intelligent robot capable of human motion or action recognition with instant decision making in any security system, or in clinics or rehabilitation centers, or in surveillance system for tracking suspicious matter, etc. With the use of networks, it will become more effective, reliable and robust.

8.2 Conclusions

We summarize the contribution of this thesis in a brief way. Within the motion acquisition context, we proposed a motion modeling strategy for acquiring and understanding limb movements within the motion video. It extracts the joint location on the human body, and makes a human body model correspondingly.

We present a structured motion database approach to human motion recognition by the structurization and organization of large amount of human motion data. We adopted a novel directional organization, a boundary problem resolution scheme, a direction estimation method to accomplish a high-precision and high-speed recognition. Moreover, in order to make use of missing motion information, we propose a directional motion template based recognition system. Furthermore, for the real-life scenarios which are cluttered with non-uniform background, along with subtle movements of background objects, we propose a recognition system that is able to cope with the cluttered nature of the background by background modeling and flow estimation. We propose a detection-tracking-recognition based human action recognition system using HOG features, Lucas-Kanade tracker, MHI, and structured motion database. As a conclusion, we claim that we have developed suitable motion recognition systems which have reasonable capability to recognize motions.

8.3 Future Work

The proposed multi-factor based recognition systems require further investigation to make the recognition more improved and the recognition system more enhanced. It would be more efficient to develop the system with real-life indoor, outdoor, simple and complex motion datasets for the practical implementation of the system. We can include different sports motions within the system. The motion representation plays an important role in building templates for recognition. Other motion representations [87] could also be adopted to test the system's performance. In the motion modeling scheme, the system should be enhanced to the motion recognition. The possible directions for the enhancements are mentioned below.

- Three cameras can be used: frontal, side (left/right), and top.
- Detect the body parts: (a) head, (b) torso, (c) limbs: (i) hands (upper arm, lower arm), (ii) legs (thigh, leg).
- Detect corresponding joints.
- Track the body parts in the successive frames.
- Storage of feature points at each frame (for each pose) for the body parts, or store trajectory information of the body parts.
- Store the features in the database for recognition purpose.

However, the system is subjected to be comprehensively investigated in order to be practically implemented in crowded scenarios. The detector should be trained with huge number of positive and negative samples. Moreover, the detection strategy should be improved, if necessary, for accurate and fast detection. We can adopt the occlusion handling mechanism proposed in [212] to build our system to effective even in the situation where significant occlusion occurs.

A miss-recognition condition will occur if one person partially occludes another, making separation difficult. So, multiple cameras are recommended in such a situation. Besides, the occlusion of the body parts or repetition of same movement more than once, may also lead to worse performance. Possibly, multi-view method can cope with the problem, as well. For monitoring situation, one can use an overhead camera to select which ground based cameras have a clear view of a subject and to specify (assuming loose calibration) where the subject would appear in each image.

Above all, our proposed approach is a one-step forward to the development of a complete human action recognition system and has much potential to be applied to real-world scenarios.

References

REFERENCES

[1] S. Ekvall, and D. Kragic, “Robot learning from demonstration: A task-level planning approach”, *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223-234, 2008.

[2] R. Dillmann, “Teaching and learning of robot tasks via observation of human performance”, *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109-116, 2004.

[3] M. Kaiser, and R. Dillmann, “Building elementary robot skills from human demonstration”, In: *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2700-2705, 1996.

[4] M. N. Nicolescu, and M. J. Mataric, “Natural methods for robot task learning: instructive demonstrations, generalization and practice”, In: *Proceedings of the International joint conference on Autonomous Agents and multi-agent Systems*, pp. 241-248, 2003.

[5] P. Lima, and G. Saridis, “Hierarchical reinforcement learning and decision making for intelligent machines”, In: *Proceedings of the IEEE International Conference on Robotics and Automation*, 1994.

[6] A. Chella, H. Dindo, and I. Infantino, “Learning high-level manipulative tasks through imitation”, In: *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pp. 251-256, 2006.

[7] A. Billard, and M. J. Mataric, “Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture”, *Robotics and Autonomous Systems*, vol. 37, no. 2-3, pp. 145-160, 2001.

[8] J. Aleotti, and S. Caselli, “Robust trajectory learning and approximation for robot programming by demonstration”, *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 409-413, 2006,

[9] Y. Kuniyoshi, M. Inaba, and H. Inoue, “Learning by watching: extracting reusable task knowledge from visual observation of human performance”, *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799-822, 1994.

[10] S. Schaal, “Is imitation learning the route to humanoid robots?”, *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233-242, 1999.

[11] P. Bakker, and Y. Kuniyoshi, “Robot see, robot do: an overview of robot imitation”, In: *Proceedings of the AISB workshop on Learning in Robots and Animals*, pp. 3-11, 1996.

[12] S. Calinon and A. Billard, “Learning of gestures by imitation in a humanoid robot”, *Imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions*, C. L. Nehaniv and K. Dautenhahn, Eds. Cambridge University Press, 2007.

[13] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, “The development of honda humanoid robot”, In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1321-1326, 1998.

[14] K. Hirai, “Current and future perspective of honda humamoid robot”, In: *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 2, 7-11 1997, pp. 500-508.

[15] J. Peters, and S. Schaal, “Reinforcement learning for parameterized motor primitives”, In: *Proceedings of the International Joint Conference on Neural Networks*, pp. 73-80, 2006.

[16] A. McGovern and A. G. Barto, “Automatic discovery of subgoals in reinforcement learning using diverse density”, In: *Proceedings of the International Conference on Machine Learning*, pp. 361-368, 2001.

[17] Y. Davidor, *Genetic Algorithms and Robotics*. World Scientific Publishing Co., Inc., 1991.

[18] R. A. Brooks, and M. J. Mataric, “Real robots, real learning problems”, *Robot Learning*, J. H. Connel and S. Mahadevan, Eds. Kluwer Academic Press, pp. 193-213, 1993.

[19] P. Maes, “Behavior-based articial intelligence”, In: *Proceedings of the international conference on From animals to animats 2 : simulation of adaptive behavior*, pp. 2-10, 1993.

[20] J. K. Tsotsos, “Behaviorist intelligence and the scaling problem”, *Artificial Intelligence*, vol. 75, no. 2, pp. 135-160, 1995.

[21] G. Rogez, J.J. Guerrero, J. Martínez, and C. Orrite-Uruñuela, "Viewpoint independent human motion analysis in man-made environments", In: *Proceedings of the British Machine Vision Conference (BMVC)*, pp.659-668, 2006.

[22] M.-C. Roh, H.-K. Shin, S.-W. Lee, “View-independent human action recognition with Volume Motion Template on single stereo camera”, *Pattern Recognition Letters*, vol. 31, issue 7, pp. 639-647, 2010.

[23] X. Ji and H. Liu, “Advances in view-invariant human motion analysis: a review”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, issue 1, pp. 13-24, 2010.

[24] É.-J. Marey, *La Machine Animale, Locomotion Terrestre et Aérienne*, Germer Bailliére, Paris, 1873.

[25] E. J. Muybridge. *Animals in Motion*. University of Pennsylvania Press, 1887.

[26] G. Johansson. *Visual perception of biological motion and a model for its analysis. Perception and Psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[27] M. Turk, “Visual interaction with lifelike characters”, In: *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 368-373, 1996.

[28] J. K. Aggarwal, and Q. Cai, “Human motion analysis: a review”, *Computer Vision and Image Understanding*, vol. 73, no. 3, pp. 428-440, 1999.

[29] Q. Cai, and J. K. Aggarwal, “Tracking human motion using multiple cameras”, In: *Proceedings of International Conference on Pattern Recognition*, 1996.

[30] L. Davis, S. Fejes, D. Harwood, Y. Yacoob, I. Hariatoglu, and M. J. Black, “Visual surveillance of human activity”, In: *Proceedings of Asian Conference on Computer Vision*, Mumbai, India, 1998.

[31] A. Nakazawa, H. Kato, and S. Inokuchi, “Human tracking using distributed video systems”, In: *Proceedings of International Conference on Pattern Recognition*, 1998.

[32] H. J. Seo, and P. Milanfar, “Action recognition from one example”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 867-882, 2011.

[33] H. J. Seo, and P. Milanfar, “Training-free, generic object detection using locally adaptive regression kernels”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1688-1704, 2010.

[34] C. R. Wren, B. P. Clarkson, and A. P. Pentland, “Understanding purposeful human motion”, In: *Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2000.

[35] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, “Pfinder: real-time tracking of the human body”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[36] H. J. Seo and P. Milanfar, “Static and space-time visual saliency detection by self-resemblance”, *Journal of Vision*, vol. 9, no. 12, no. 15, pp. 1-27, 2009,.

[37] D. M. Gavrila, L. S. Davis, “Tracking of humans in action: A 3D model-based approach”, In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 73–80, 1996.

[38] S. M. A. Eftakhar, J. K. Tan, H. Kim, and S. Ishikawa, “An Effective Directional Motion Database Organization for Human Motion Recognition”, *International Journal of Innovative Computing, Information and Control (IJICIC)*, in press.

[39] M. C. Silaghi, R. Plänkers, R. Boulic, P. Fua, and D. Thalmann, “Local and global skeleton fitting techniques for optical motion capture”, In: *Proceedings of International Workshop on Modelling and Motion Capture Techniques for Virtual Environments*, 1998.

[40] O. Munkelt, C. Ridder, D. Hansel, and W. Hafner, “A model driven 3D image interpretation system applied to person detection in video images”, In: *Proceedings of International Conference on Pattern Recognition*, 1998.

[41] R. Gonzalez, and R. Woods, *Digital Image Processing*, Addison-Wesley, pp. 518-548, 1992.

[42] T. Drummond, R. Cipolla, “Real-time tracking of highly articulated structures in the presence of noisy measurements”, In: *Proceedings of International Conference On Computer Vision*, Vol. 2, pp. 315–320, 2001.

[43] Y. Li, S. Ma, and H. Lu, “Human posture recognition using multi-scale morphological method and Kalman motion estimation”, In: *Proceedings of International Conference on Pattern Recognition*, 1998.

[44] A. M. Baumberg and D. C. Hogg, “An efficient method for contour tracking using active shape models”, In: *Proceedings of Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 2–14, 1994.

[45] A. Bottino, A. Laurentini, and P. Zuccone, “Toward non-intrusive motion capture”, In: *Proceedings of Asian Conference on Computer Vision*, 1998.

[46] N. Jojic, J. Gu, H. C. Shen, and T. Huang, “3-D reconstruction of multipart self-occluding objects”, In: *Proceedings of Asian Conference on Computer Vision*, 1998.

[47] I. Haritaoglu, D. Harwood, and L. S. Davis, “Ghost: A human body part labeling system using silhouettes”, In: *Proceedings of the International Conference on Pattern Recognition*, 1998.

[48] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4: Who? When? Where? What? - A real time system for detecting and tracking people”, In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1998.

[49] S. X. Ju, M. J. Black, and Y. Yacoob, “Cardboard people: A parameterized model of articulated image motion”, In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1996.

[50] B. Heisele, and C. Wohler, “Motion-based recognition of pedestrians”, In: *Proceedings of the International Conference on Pattern Recognition*, 1998.

[51] C. Bregler, “Learning and recognizing human dynamics in video sequences”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1997.

[52] C. Carlsson, and J. Sullivan, “Action recognition by shape matching to key frame”, In: *Proceedings of the Workshop Models versus Examplars in Computer Vision*, 2001.

[53] A. Yilmaz, and M. Shah, “Actions Sketch: A Novel Action Representation”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[54] K. Cheung, S. Baker, and T. Kanade, “Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2003.

[55] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as Space-Time Shapes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247-2253, 2007.

[56] T. Mahmood, A. Vasilescu, and S. Sethi, “Recognition of action events from multiple video points”, In: *Proceedings IEEE Workshop Detection and Recognition of Events in*

Video, 2001.

[57] J. Liu, S. Ali, and M. Shah, “Recognizing human actions using multiple features”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.

[58] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition”, In: *Proceedings of ACM Multimedia Conference*, 2007.

[59] E. Shechtman and M. Irani, “Space-time behavior-based correlation—or—how to tell if two underlying motion fields are similar without computing them?”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 2045-2056, 2007.

[60] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.

[61] T. Kim, and R. Cipolla, “Canonical correlation analysis of video volume tensors for action categorization and detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1415-1428, Aug. 2009.

[62] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang, “Hierarchical space-time model enabling efficient search for human actions”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 808-820, June 2009.

[63] D. Meyer, J. Denzler, and H. Niemann, “Model based extraction of articulated objects in image sequences”, In: *Proceedings of the International Conference on Image Processing*, 1997.

[64] M. Rossi, and A. Bozzoli, “Tracking and counting moving people”, *Technical Report 9404-03, IRST, Trento, Italy*, 1994.

[65] A. Azarbayejani, C. R. Wren, and A. P. Pentland, “Real-time 3-D tracking of the human body”, In: *Proceedings of the International Conference dedicated to Image Communication* (IMAGE’COM 96), 1996.

[66] R. Poppe, “Discriminative vision-based recovery and recognition of human motion”, *PhD Thesis, University of Twente. CTIT Ph.D.-thesis series No. 09-136, Netherlands*, 2009.

[67] T. B. Moeslund, and E. Granum, “A survey of computer vision-based human motion capture”, *Computer Vision and Image Understanding*, vol. 81, pp. 231–268, 2001.

[68] L. Wang, W. Hu, and T. Tan, “Recent developments in human motion analysis”, *Pattern Recognition*, vol. 36, pp. 585-601, 2003.

[69] R. Poppe, “Vision-based human motion analysis: An overview”, *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4-18, 2007.

[70] A. F. Bobick, and J. W. Davis, “The recognition of human movement using temporal templates”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no.

3, pp. 257-267, 2001.

[71] Y. Wang, K. Huang, and T. Tan, “Human activity recognition based on ℜ Transform”, In: *Proceedings of the Workshop on Visual Surveillance (VS’07)*, pp. 1-8, 2007.

[72] R. Souvenir, and J. Babbs, “Learning the viewpoint manifold for action recognition”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’08)*, pp. 1-7, 2008.

[73] H.-S. Chen, H.-T. Chen, Y.-W. Chen, and S.-Y. Lee, “Human action recognition using star skeleton”, In: *Proceedings of the International Workshop on Video Surveillance and Sensor Networks (VSSN’06)*, pp. 171–178, 2006.

[74] L. Wang and D. Suter, “Informative shape representations for human action recognition”, In: *Proceedings of the International Conference on Pattern Recognition (ICPR’06)*, vol. 2, pp. 1266–1269, 2006.

[75] D. Weinland, E. Boyer, and R. Ronfard, “Action recognition from arbitrary views using 3D exemplars”, In: *Proceedings of the International Conference On Computer Vision (ICCV’07)*, pp. 1–8, 2007.

[76] D. Weinland, and E. Boyer, “Action recognition using exemplar-based embedding”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’08)*, pages 1-7, 2008.

[77] F. Huang, and G. Xu. “Viewpoint insensitive action recognition using envelop shape”, In: *Proceedings of the Asian Conference on Computer Vision (ACCV’07)*, part 2, no. 4844, *Lecture Notes in Computer Science*, pp. 477–486, 2007.

[78] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian, “Towards fast, view-invariant human action recognition”, In: *Proceedings of the* Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB’08), pages 1–8, 2008.

[79] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes”, *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[80] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance”, In: *Proceedings of the International Conference on Computer Vision (ICCV’03)*, vol. 2, pp. 726–733, 2003.

[81] M. A. R. Ahad, T. Ogata, J. K. Tan, H.S. Kim, and S. Ishikawa., “A complex motion recognition technique employing directional motion templates”, *International Journal on Innovative Computing, Information and Control (IJICIC)*, vol. 4, no. 8, pp. 1943-1954, 2008.

[82] S. Ali and M. Shah, “Human action recognition in videos using kinematic features and multiple instance learning”, *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, vol. 32, no. 2, pp. 288 – 303, 2010.

[83] J. K. Tan and S. Ishikawa, “High accuracy and real time recognition of human activities”, In: *Proceeding of Annual Conference of IEEE Industrial Electronics Society (IECON)*, pp. 2377-2382, 2007.

[84] H. Ning, Y. Hu, and T. S. Huang, “Searching human behaviors using spatial-temporal words”, In: *Proceedings of the International Conference on Image Processing (ICIP’07)*, vol. 6, pp. 337–340, 2007.

[85] C. Achard, X. Qu, A. Mokhber, and M. Milgram, “A novel approach for recognition of human actions with semi-global features”, *Machine Vision and Applications*, vol. 19, no. 1, pp. 27–34, 2008.

[86] S. A. Niyogi, and E. H. Adelson, “Analyzing and recognizing walking figures in XYT”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’94)*, pp. 469–474, 1994.

[87] M. A. R. Ahad, T. Ogata, J. K. Tan, H.S. Kim, and S. Ishikawa., “Motion history image: its variants and applications”, *Machine Vision and Applications*, pp. 1-27, 2010.

[88] J. Han, and B. Bhanu, “Individual recognition using gait energy image”, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, 2006.

[89] J. Han, B. Bhanu, “Gait energy image representation: comparative performance evaluation on USF Human ID database”, In: *Proceedings of the Joint International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 133–140, 2003.

[90] V. Chandrashekhar, and K. S. Venkatesh, “Action energy images for reliable human action recognition”, In: *Proceedings of the Asian Symposium on Information Display (ASID)*, pp. 484–487, 2006.

[91] J. Liu, and N. Zhang, “Gait history image: a novel temporal template for gait recognition”, In: *Proceeding of IEEE International Conference on Multimedia and Expo*, pp. 663–666, 2007.

[92] Q. Ma, S. Wang, D. Nie, and J. Qiu, “Recognizing humans based on gait moment image”, In: *Proceeding of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 606–610, 2007.

[93] C. Chen, J. Liang, H. Zhao, H. Hu, and J. Tian, “Frame difference energy image for gait recognition with incomplete silhouettes”, *Pattern Recognition. Letters*, vol. 30, no. 11, pp. 977–984, 2003.

[94] F. Porikli, “Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 829-836, 2005.

[95] C.-C. Yu, H.-Y. Cheng, C.-H. Cheng, and K.-C. Fan, "Efficient human action and gait analysis using multiresolution motion energy histogram", *EURASIP Journal on Advances in Signal Processing*, pp.1-13, 2010.

[96] D. Chen, and J. Yang, "Exploiting high dimensional video features using layered Gaussian mixture models", In: *Proceeding of the IEEE International Conference on Pattern Recognition*, p. 4, 2006.

[97] M. Pantic, I. Patras, and M. F. Valstar, "Learning spatio-temporal models of facial expressions", In: *Proceeding of the International Conference on Measuring Behaviour*, pp. 7–10, 2005.

[98] M. Valstar, M. Pantic, and I. Patras, "Motion history for facial action detection in video", In: *Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 1, pp. 635–640, 2004.

[99] M. Valstar, M. Pantic, and I. Patras, "Facial action recognition using temporal templates", In: *Proceeding of the IEEE Workshop on Robot and Human Interactive Communication*, pp. 253–258, 2004.

[100] H. Meng, N. Pears, and C. Bailey, "A human action recognition system for embedded computer vision application", In: *Proceeding of the Workshop on Embedded Computer Vision (with CVPR)*, pp. 1–6, 2007.

[101] A. Yilmaz and M. Shah, "A differential geometric approach to representing the human actions", *Computer Vision and Image Understanding*, vol. 119, no. 3, pp. 335-351, 2008.

[102] P. Yan, S. M. Khan, and M. Shah, "Learning 4D action feature models for arbitrary view action recognition", In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pp. 1-7, 2008.

[103] M. Grundmann, F. Meier, and I. Essa, "3D shape context and distance transform for action recognition", In: *Proceedings of the International Conference on Pattern Recognition (ICPR'08)*, pp. 1-4, 2008.

[104] D. Batra, T. Chen, and R. Sukthankar, "Space-time shapelets for action recognition", In: *Proceedings of the Workshop on Motion and Video Computing (WMVC'08)*, pp. 1-6, 2008.

[105] A. Oikonomopoulos, M. Pantic, and I. Patras, "B-spline polynomial descriptors for human activity recognition", In: *Proceedings of the Workshop on Computer Vision and Pattern Recognition for Human Communicative Behavior Analysis (CVPR4HB'08)*, pp. 1-8, 2008.

[106] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features", In: *Proceedings of the International Conference on Computer Vision (ICCV'05)*, vol. 1, pp. 166-173, 2005.

[107] P. A. Viola, and M. J. Jones, "Rapid object detection using a boosted cascade of simple features", In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*

(CVPR'01), vol. 1, pp. 511–518, 2001.

[108] T. Ogata, W. Christmas, J. Kittler, and S. Ishikawa, "Improving human activity detection by combining multi-dimensional motion descriptors with boosting", In: *Proceedings of the International Conference on Pattern Recognition (ICPR'06)*, vol. 1, pp. 295–298, 2006.

[109] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-temporal shape and flow correlation for action recognition", In: *Proceedings of the Workshop on Visual Surveillance (VS'07)*, pp. 1–8, 2007.

[110] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos", In: *Proceedings of the International Conference On Computer Vision (ICCV'07)*, pp. 1–8, 2007.

[111] H. Murase, and S. K. Nayar, "Visual learning and recognition of 3-D objects from appearance", *International Journal of Computer Vision*, vol.14, no.1, pp. 5-24, 1995.

[112] M. M. Rahman, and S. Ishikawa, "Representing human postures/motions using an eigenspace technique", In: *Proceedings of the International Conference on Artificial Intelligence in Engineering & Technology*, pp. 232-236, 2002.

[113] H. Murase, and S. K. Nayar, "Detection of 3D objects in cluttered scenes using hierarchical eigenspace", *Pattern Recognition Letters*, vol. 18, no. 4, pp. 375-384, 1997.

[114] M. M. Rahman, and S. Ishikawa, "Human Motion Recognition Using an Eigenspace", *Pattern Recognition Letters*, vol. 26, no. 6, pp. 687–697, 2005.

[115] T. Ogata, J. K. Tan, and S. Ishikawa, "High-speed human motion recognition based on a motion history image and an eigenspace", *IEICE Transactions on Information and Systems*, Vol. 89, Issue D, no. 1, pp. 281-289, 2006.

[116] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time sequential images using hidden markov models", In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.

[117] K. Akita, "Image Sequence Analysis of Real World Human Motion", *Pattern Recognition Letters*, vol. 17, no. 1, pp. 73-83, 1984.

[118] A. Bobick, "Movement, activity, and action: the role of knowledge in the perception of motion", *Philosophical Transactions on Royal Society London*, vol. 352, pp. 1257-1265, 1997.

[119] C. Sul, K. Lee, and K.Wohn, "Virtual stage: a location-based karaoke system", *Journal of Multimedia*, vol. 5, no. 2, pp. 42–52, 1998.

[120] M. Oren, C. Papageorigiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates", In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1997.

[121] I. Haritaoglu, D. Harwood, and L. S. Davis, *"Ghost: A human body part labeling system*

using silhouettes", In: *Proceedings of the International Conference on Pattern Recognition*, 1998.

[122] L. Campbell, and A. Bobick, "Using phase space constraints to represent human body motion", In: *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[123] R. E. Rosales, "Recognition of human action using moment-based features", *Technical Report BU-1998-020, Boston University, Computer Science, Boston, MA*, 1998.

[124] D. Tran, A. Sorokin, and D. A. Forsyth, "Human activity recognition with metric learning", In: *Proceedings of the European Conference on Computer Vision (ECCV'08)*, part 1, no. 5302, *Lecture Notes in Computer Science*, pp. 548-561, 2008.

[125] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1896–1909, 2005.

[126] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, "The function space of an activity", In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1, pp. 959–968, 2006.

[127] C. Yang, Y. Guo, H. S. Sawhney, and R. Kumar, "Learning actions using robust string kernels", In: *Proceedings of the International Conference on Human Motion: Understanding, Modeling, Capture and Animation (HUMO'07)*, no. 4814, *Lecture Notes in Computer Science*, pp. 313–327, 2007.

[128] A. Elgammal, and C. S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning", In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 681–688, 2004.

[129] O. Masoud, and N. Papanikolopoulos, "A method for human action recognition", *Image and Vision Computing*, vol. 21, no. 8, pp. 729–743, 2003.

[130] T.-J. Chin, L. Wang, K. Schindler, and D. Suter, "Extrapolating learned manifolds for human activity recognition", In: *Proceedings of the International Conference on Image Processing (ICIP'07)*, vol. 1, pp. 381–384, 2007.

[131] L. Wang, and D. Suter, "Analyzing human movements from silhouettes using manifold learning", In: *Proceedings of IEEE International Conference on Video and Signal Based Surveillance*, p. 7, 2006.

[132] L. Wang, and D. Suter, "Learning and matching of dynamic shape manifolds for human action recognition", *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1646-1661, 2007.

[133] L. Wang, and D. Suter, "Visual learning and recognition of sequential data manifolds with applications to human movement analysis", *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 153–172, 2008.

[134] J. Blackburn, and E. Ribeiro, “Human motion recognition using isomap and dynamic time warping”, *Human Motion- Understanding, Modeling, Capture and Animation: Lecture Notes in Computer Science*, vol. 4814, pp. 285-298, 2007.

[135] P. K. Turaga, A. Veeraraghavan, and R. Chellappa, “Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’08)*, pp. 1-8, 2008.

[136] K. Jia, and D.-Y. Yeung, “Human action recognition using local spatio-temporal discriminant embedding”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’08)*, pp. 1-8, 2008.

[137] J. Sullivan, and S. Carlsson, “Recognizing and tracking human action”, In: *Proceedings of the European Conference on Computer Vision (ECCV’02)*, vol. 1, no. 2350, *Lecture Notes in Computer Science*, pp. 629–644, 2002.

[138] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori, “Unsupervised discovery of action classes”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, pp. 1654-1661, 2006.

[139] D. Weinland, R. Ronfard, and E. Boyer, “Automatic discovery of action taxonomies from multiple views” In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2, pp. 1639–1645, 2006.

[140] Y. Dedeoğlu, B. U. Töreyin, U. Güdükbay, and A. E. Çetin, “Silhouette-based method for object classification and human action recognition in video”, In: *Proceedings of the Workshop on Computer Vision in Human-Computer Interaction (ECCVHCI’06)*, no. 3979, *Lecture Notes in Computer Science*, pp. 64–77, 2007.

[141] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, “A biologically inspired system for action recognition”, In: *Proceedings of the International Conference On Computer Vision (ICCV’07)*, pp. 1-8, 2007.

[142] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg, “Local velocity adapted motion events for spatio-temporal recognition”, *Computer Vision and Image Understanding*, vol. 108, no. 3, pp. 207–229, 2007.

[143] C. Schüldt, I. Laptev, and B. Caputo, “Recognizing human actions: A local SVM approach”, In: *Proceedings of the International Conference on Pattern Recognition (ICPR’04)*, vol. 3, pp. 32–36, 2004.

[144] D. Cao, O. T. Masoud, D. Boley, and N. Papanikolopoulos, “Human motion recognition using support vector machines”, *Computer Vision and Image Understanding*, vol. 113, pp. 1064–1075, 2009.

[145] A. Oikonomopoulos, I. Patras, and M. Pantic, “Spatiotemporal salient points for visual recognition of human actions”, *IEEE Transactions On Systems, Man, and Cybernetics (SMC) - Part B: Cybernetics*, vol. 36, no. 3, pp. 710–719, 2006.

[146] F. Niu, and M. Abdel-Mottaleb, “View-invariant human activity recognition based on shape and motion features”, In: *Proceedings of the International Symposium on Multimedia Software Engineering (ISMSE’04)*, pp. 546–556, 2004.

[147] F. Lv, and R. Nevatia, “Single view human action recognition using key pose matching and Viterbi path searching”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’07)*, pp. 1-8, 2007.

[148] M. Ahmad, and S.-W. Lee, “Human action recognition using shape and CLG-motion flow from multi-view image sequences”, *Pattern Recognition*, vol. 41, no. 7, pp. 2237-2252, 2008.

[149] *W.-L. Lu, and J. J. Little, “Simultaneous tracking and action recognition using the PCA-HOG descriptor”, In: Proceedings of the Canadian Conference on Computer and Robot Vision (CRV’06)*, p. 6, 2006.

[150] D. Ramanan, and D. A. Forsyth, “Automatic annotation of everyday movements”, *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, pp. 1-8, 2003.

[151] D. Ramanan, D. A. Forsyth, and A. Zisserman, “Tracking people by learning their appearance”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65–81, 2007.

[152] X. Feng, and P. Perona, “Human action recognition by sequence of movelet codewords”, In: *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT’02)*, pp. 717–721, 2002.

[153] N. İkizler, and D. A. Forsyth, “Searching for complex human activities with no visual examples”, *International Journal of Computer Vision*, vol. 30, no. 3, pp. 337–357, 2008.

[154] B. Chakraborty, O. Rudovic, and J. Gonzàlez, “View-invariant human body detection with extension to human action recognition using component-wise HMM of body parts”, In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition (FGR’08)*, pp. 1-6, 2008.

[155] F. Lv, and R. Nevatia, “Recognition and segmentation of 3-D human action using HMM and multi-class adaBoost”, In: *Proceedings of the European Conference on Computer Vision (ECCV’06)*, vol. 4, no. 3953, *Lecture Notes in Computer Science*, pp. 359–372, 2006.

[156] P. Peursum, S. Venkatesh, and G. West, “Tracking-as-recognition for articulated full-body human motion analysis”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’07)*, pp. 1-8, 2007.

[157] F. Caillette, and A.Galata, and T. Howard, “Real-time 3-D human body tracking using learnt models of behavior, *Computer Vision and Image Understanding*, vol. 109, no. 2, pp.112–125, 2008.

[158] P. Natarajan, and R. Nevatia, “Online, real-time tracking and recognition of human

actions", In: *Proceedings of the Workshop on Motion and Video Computing (WMVC'08)*, pp. 1-8, 2008.

[159] L. Zelnik-Manor, and M. Irani, "Statistical analysis of dynamic actions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1530–1535, 2006.

[160] E. Shechtman, and M. Irani, "Matching local self-similarities across images and videos", In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'07)*, pp. 1–8, 2007.

[161] P. Matikainen, M. Hebert, R. Sukthankar, and Y. Ke, "Fast motion consistency through matrix quantization. In: *Proceedings of the British Machine Vision Conference (BMVC'08)*, pp. 1055–1064, 2008.

[162] R. D. Green and L. Guan, "Quantifying and recognizing human movement patterns from monocular video images-part i: a new framework for modeling human motion", *IEEE Transactions Circuits System Video Technology*, vol. 14, no. 2, pp. 179–190, 2004.

[163] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[164] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle, "Indexing large human-motion databases", In: *Proceedings of Very Large Data Bases*(*VLDB) Conference*, pp. 780-791, 2004.

[165] I. Bouchrika, and M. S. Nixon, "Model-based feature extraction for gait analysis and recognition", *Mirage: Computer Vision / Computer Graphics Collaboration Techniques and Applications*, 2007.

[166] T. Shiraki, H. Saito, Y. Kamoshida, K. Ishiguro, R. Fukano, T. Shirai, K. Taura, M. Otake, T. Sato, and N. Otsu, "Real-time motion recognition using chlac features and cluster computing", In: *Proceedings of the IFIP International Conference on Network and Parallel Computing* (*NPC* 2006), pp. 43-49, 2006.

[167] D. P. Huttenlocher, R. H. Lilien, and C. F. Olson, "Object recognition using subspace methods", In: *Proceedings of European Conference on Computer Vision* (*ECCV*), pp. 536-545, 1996.

[168] A. Pentland, B. Moghaddam, and T. Starner", View-based and modular eigenspaces for face recognition", In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR'94*), pp. 84-91,1994.

[169] J. Krumm, "Eigenfeatures for planar pose measurement of partially occluded objects", In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR'96*), pp.55-66, 1996.

[170] A. Leonardis, and H. Bischof, "Dealing with occlusions in the eigenspace approach", In:

Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (*CVPR'96*), pp. 453-458, 1996.

[171] C. Ellis, "Concurrent search and insertion in AVL trees", *IEEE Transactions on Computers*, vol. C-29, no. 9, pp.811-817, 1980.

[172] R. Bayer, and E. McCreight, "Organization and maintenance of large ordered indexes", *Acta Informatica*, vol. 1, no. 3, pp. 173-189, 1972.

[173] A. Guttman, "R-Trees: A Dynamic Structure for Spatial Searching", In: *Proceedings of Annual Meeting of SIGMOD'84*, pp. 47-57, 1984.

[174] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos, "The R+ -tree: a dynamic index for multi-dimensional objects", In: *Proceedings of Very Large Data Bases (VLDB) Conference*, pp. 507-518, 1987.

[175] W. Wang, J. Yang, and R. Muntz, "PK-tree: a dynamic spatial index structure for large data sets", *UCLA Computer Science Department Technical Report #970039*, 1997.

[176] S. M. A. Eftakhar, J. K. Tan, H. Kim, and S. Ishikawa, "Human motion recognition employing large motion-database structure", *International Journal on Advanced Computer Engineering*, vol. 2, no. 1, pp. 17-23, 2009.

[177] S. M. A. Eftakhar, J. K. Tan, H. Kim, and S. Ishikawa, "Improvement of a structured motion database for high accuracy human motion recognition", *International Journal of Biomedical Soft Computing and Human Sciences*, vol.17, no.1, pp. 19-28, 2011.

[178] J. K. Tan, K. Kouno, S. Ishikawa, H. S. Kim, and T. Shinomiya, "High speed human motion recognition employing a motion database", *Journal of Image & Electronic Society*, vol. 36, no. 6, pp. 110-118, 2007. (*Japanese*)

[179] S. M. A. Eftakhar, J. K. Tan, H. Kim, and S. Ishikawa, "Multiple Persons' Action Recognition by Fast Human Detection", In: *Proceeding of SICE Annual Conference*, pp. 1639-1644, 2011.

[180] K. Yamane, Y. Yamaguchi, and Y. Nakamura, "Human motion database with a binary tree and node transition graphs", *Autonomous Robots*, vol. 30, no. 1, pp. 87-98, 2011.

[181] E. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle, "Indexing large human-motion databases", In: *Proceeding of Very Large Data Bases (VLDB) Conference*, pp. 780-791, 2004.

[182] C. Li, and B. Prabhakaran, "Indexing of motion capture data for efficient and fast similarity search", *Journal of Computers*, vol. 1, no. 3, pp. 35-42, 2006.

[183] M. A. Nascimento, E. Tousidou, V. Chitkara, and Y. Manolopoulos," Image indexing and retrieval using signature trees", *Data & Knowledge Engineering*, vol. 43, issue 1, pp. 57-77, 2002.

[184] P. Punitha, Naveen, and D. S. Guru, "Indexing of Document Images based on Triangular Spatial Relationships", In: *Proceeding of International Conference of Computing: Theory*

and Applications (ICCTA ‘07), pp. 533-537, 2007.

[185] F. Liu, Y. Zhang, F. Wu, and Y. Pan,” 3D motion retrieval with motion index tree”, *Computer Vision and Image Understanding*, vol. 92, issue 2-3, pp. 265-284, 2003.

[186] J. Ben-Arie, and Z. Wang, “Human activity recognition using multi-dimensional indexing”, *International Journal of Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1091-1104, 2002.

[187] D.Comer, “the ubiquitous b-tree”, acm computing surveys”, vol. 11, no. 2, pp. 121-137, 1979.

[188] C.-C. Yu, J.-N. Hwang, G.-F. Ho, and C.-H. Hsieh, “Automatic human body tracking and modeling from monocular video sequences”, In: *Proceeding of International Conference on Acoustics, Speech and Signal Processing*, pp. 917-920, 2007.

[189] N. R. Howe, “Silhouette lookup for automatic pose tracking”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition Workshops*, pp. 15-22, 2004.

[190] P. Peursum, S. Venkatesh, and G. West, “Tracking-as-recognition for articulated full-body human motion analysis”, In: *Proceeding of International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[191] J. Ben-Arie, Z. Wang, “Human activity recognition using multi-dimensional indexing”, *International Journal of Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1091-1104, 2002.

[192] I. A. Kakadiaris, and D. Metaxas, “Three-dimensional human body model acquisition from multiple views”, *International Journal of Computer Vision*, vol. 30, n. 3, pp. 191-218, 1998.

[193] M. Tominaga, H. Hongo, H. Koshimizu, Y. Niwa, and K. Yamamoto, “Estimation of human motion from multiple cameras for gesture recognition”, In: *Proceeding of International Conference on Pattern Recognition*, vol. 1, pp. 401-404, 2002.

[194] G. K. M. Cheung, S. Baker, and T. Kanade, “Shape-From-Silhouette of articulated objects and its use for human body kinematics estimation and motion capture”, In: *Proceeding of International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 77-84, 2003.

[195] B. Wu, R. Nevita, “Detection and segmentation of multiple partially occluded objects by grouping, merging, assigning part detection responses”, *International Journal of Computer Vision*, pp. 185-204, 2009.

[196] H. J. Lee and Z. Chen, “Determination of 3D human body posture from a single view”, *International Journal of Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 148-168, 1985.

[197] H. J. Lee and Z. Chen, “Knowledge-guided visual perception of 3-D human gait from a

single image sequence”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 336-342, 1992.

[198] C. I. Attwood, G. D. Sullivan, and K. D. Baker, “Model-based recognition of human posture using single synthetic images”, In: *Proceeding of Fifth Alvey Vision Conference*, pp. 25-30, 1989.

[199] O. Munkelt, C. Ridder, D. Hansel, and W. Hafner, “A model driven 3D image interpretation system applied to person detection in video images”, In: *Proceeding of International Conference on Pattern Recognition*, vol. 1, pp. 70-73, 1998.

[200] I. A. Karaulova, P. M. Hall, A. D. Marshall, “A hierarchical model of dynamics for tracking people with a single video camera”, In: *Proceeding of British Machine Vision Conference (BMVC)*, pp. 352–361, 2000.

[201] Y. Guo, G. Xu, and S. Tsuji, “Tracking human body motion based on a stick Agure model”, *Visual Communication and Image Representation*, vol. 5, pp. 1-9, 1994.

[202] Y. Guo, G. Xu, and S. Tsuji, “Understanding human motion patterns”, In: *Proceedings of the International Conference on Pattern Recognition*, pp. 325–329, 1994.

[203] J. C. Niebles, and L. Fei-Fei, “A hierarchical model of shape and appearance for human action classification”. In: *Proceeding of the International Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2007.

[204] Y. Iwai, K. Ogaki, and M. Yachida, “Posture estimation using structure and motion models”, In: *Proceedings of the International Conference on Computer Vision*, pp. 214-219, 1999.

[205] Y. Luo, F.J. Perales, J. Villanueva, “An automatic rotoscopy system for human motion based on a biomechanic graphical model”, *International Journal of Computer Graphics*, vol. 16, no. 4, pp. 355-362, 1992.

[206] C. Yaniz, J. Rocha, and F. Perales, “3D region graph for reconstruction of human motion”, In: *Proceedings of the Workshop on Perception of Human Motion at ECCV*, 1998.

[207] M. Silaghi, R. Plänkers, M. L. Silaghi, R. Boulic, P. Fua, and D. Thalmann, “Local and global skeleton fitting techniques for optical motion capture”, In: *Proceedings of the Workshop on Modeling and Motion Capture Techniques for Virtual Environments*, pp. 26-40, 1998.

[208] C.-C. Yu, J.-N. Hwang, G.-F. Ho, and C.-H. Hsieh, “Automatic human body tracking and modeling from monocular video sequences”, In: *Proceeding of International Conference on Acoustics, Speech and Signal Processing*, pp. 917-920, 2007

[209] A. Sundarsesan, R. Chellappa, “Acquisition of articulated human body models using multiple cameras”, In: *Proceeding of International Conference on Articulated Motion and Deformable Objects (AMDO)*, 2006.

[210] J.-C. Cheng, J. M. F. Moura, “Capture and representation of human walking in live video

sequences”, *IEEE Transactions on Multimedia*, vol. 1, no. 2, pp. 144-156, 1999.

[211] J. Gao, R. T. Collins, A. G. Hauptmann, and H. D. Wactlar, “Articulated motion modeling for activity analysis”, In: *Proceeding of International Conference on Image and Video Retrieval, Workshop on Articulated and Nonrigid Motion*, 2004.

[212] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: spatial pyramid matching for recognizing natural scene categories”, In: *Proceeding of the International Conference on Computer Vision and Pattern Recognition (CVPR’06)*, pp. 2169 - 2178, 2006.

[213] M. K. Leung, and Y.-H. Yang, “First Sight: A human body outline labeling system”, *International Journal of Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 359-377, 1995.

[214] L. Mündermann, S. Corazza, and T. P. Andriacchi, “Accurately measuring human movement using articulated ICP with soft-joint constraints and a repository of articulated models”, In: *Proceeding of International Conference on Computer Vision and Pattern Recognition (CVPR’07)*, pp. 1-6, 2007.

[215] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. P. Andriacchi, “Markerless motion capture through visual hull, articulated icp and subject specific model generation*”, International Journal of Computer Vision*, vol. 87, pp. 156–169, 2010.

[216] K. Rohr, “Towards model-based recognition of human movements in image sequences”, *CVGIP: Image Understanding*, vol. 59, no. 1, pp. 94–115, 1994.

[217] S. Wachter, and H.-H. Nagel, “Tracking persons in monocular image sequences”, *Computer Vision and Image Understanding*, vol. 74, no. 3, pp. 174–192, 1999.

[218] J. M. Rehg, and T. Kanade, “Model-based tracking of self-occluding articulated objects”, In: Proceedings of International Conference on Computer Vision, pp. 612–617, 1995.

[219] I. A. Kakadiaris, and D. Metaxas, “Model-based estimation of 3-D human motion with occlusion based on active multi-viewpoint selection”, In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 81–87, 1996.

[220] N. Goddard, “Incremental model-based discrimination of articulated movement from motion features”, In: *Proceedings of IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 89–94, 1994.

[221] C. Bregler, and J. Malik, “Tracking people with twists and exponential maps”, In: *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR’98)*, 1998.

[222] Q. Delamarre, and O. Faugeras, “3D articulated models and multi-view tracking with silhouettes”, In: *Proceedings of the International Conference on Computer Vision*, 1999.

[223] J. P. Luck, D. E. Small, C. Q. Little, “Real-time tracking of articulated human models using a 3d shape-from-silhouette method”, in: *Proceedings of the Robot Vision*

Conference, 2001.

[224] R. Kehl and L. J. van Gool, “Markerless tracking of complex human motions from multiple views”, *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–209, 2006.

[225] I. Haritaoglu, D. Harwood, and L. S. Davis, “W4s: A real-time system detecting and tracking people in 2½D”, In: *Proceedings of the European Conference on Computer Vision (ECCV’98)*, vol. 1, no. 1406, *Lecture Notes in Computer Science*, pp. 877–892, 1998.

[226] Y. Huang, and T. S. Huang, “Model-based human body tracking” In: *Proceedings of the International Conference on Pattern Recognition (ICPR’02)*, vol. 1, pp. 552-555, 2002.

[227] A. Agarwal, and B. Triggs, “Tracking articulated motion using a mixture of autoregressive models”, In: *Proceedings of the European Conference on Computer Vision (ECCV’04)*, vol. 3, no. 3024, *Lecture Notes in Computer Science*, pp. 54-65, 2004.

[228] T.-J. Cham, and J. M. Rehg, “A multiple hypothesis approach to figure tracking”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’99)*, vol. 2, pp. 239–245, 1999.

[229] C. Bregler, J. Malik, and K. Pullen, “Twist based acquisition and tracking of animal and human kinematics”, *International Journal of Computer Vision*, vol. 56, no. 3, pp. 179–194, 2004.

[230] I. A. Kakadiaris, and D. N. Metaxas, “Model-based estimation of 3D human motion”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1453–1459, 2000.

[231] D. D. Morris, and J. M. Rehg, “Singularity analysis for articulated object tracking”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR’98)*, pp. 289–297, 1998.

[232] L. Sigal, M. Isard, B. Sigelman, and M. J. Black, “Attractive people: Assembling loose-limbed models using non-parametric belief propagation”, *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, pp. 1539–1546, 2003.

[233] M. Yamada, K. Ebihara, and J. Ohya, “A new robust real-time method for extracting human silhouettes from color images”, In: *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pp. 528-533, 1998.

[234] G. Mori, and J. Malik, “Recovering 3d human body configurations using shape contexts”, *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 28, no. 7, pp. 1052-1062, 2006.

[235] M. W. Lee, and R. Nevatia, “Human pose tracking using multi-level structured models”, In: *Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science*, vol. 3, no. 3953, pp. 368– 381, 2006.

[236] M. W. Lee, and I. Cohen, “Proposal maps driven mcmc for estimating human body pose in static images”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 334–341, 2004.

[237] C. Sminchisescu, A. Kanaujia, and D. Metaxas, “Learning joint top–down and bottom–up processes for 3D visual inference”, In: *Proceedings of the Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1743–1752, 2006.

[238] A. Bottino, and A. Laurentini, “A silhouette based technique for the reconstruction of human movement”, *Computer Vision and Image Understanding*, vol. 83, pp. 79-95, 2001.

[239] I. Kakadiaris, and D. Metaxas, “Model-based estimation of 3D human motion”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, 2000.

[240] J. Ben-Arie, Z. Wang, P. Pandit, and S. Rajaram, “Human activity recognition using multidimensional indexing”, *IEEE Transactions on Pattern Analysis And Machine Intelligence*, vol. 24, no. 8, 2002.

[241] D. M. Gavrila, “The visual analysis of human movement: a survey”, *Computer Vision and Image Understanding*, vol. 73, no. 1, pp. 82–98, 1999.

[242] H. Fujioka, H. Kano, and X. Chen, “Motion recovery under perspective stereo vision”, *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol.5, no.1, pp.167-182, 2009.

[243] T. B. Moeslund, A. Hilton, and V. Kruger, “A survey of advances in vision-based human motion capture and analysis”, *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90-126, 2006.

[244] D. Gavrila, and L. Davis, “3D model-based tracking of humans in action: A multi-view approach”, In: *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR’96)*, pp. 73-80, 1996.

[245] M. Isard, and A. Blake, “Condensation - conditional density propagation for visual tracking”, *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[246] X. He, and P. Niyogi, “Locality preserving projections”, *Advances in Neural Information Processing Systems*, vol. 16, 2004.

[247] D. J. Berndt, and J. Clifford, “Finding patterns in time series: a dynamic programming approach”, *Advances in Knowledge Discovery and Data Mining*, pp. 229-248, 1996.

[248] H. Murase, and M. Lindenbaum, “Spatial adaptive method for partial eigenstructure decomposition of large images”, *NTT Technical Report No. 6527*, 1992.

[249] K. Kouno, J. K. Tan, and S. Ishikawa, “High-speed data retrieval in an eigenspace employing a b-tree structure”, In: *Proceedings of the SICE-ACCAS International Joint Conference*, pp. 2717-2720, 2006.

[250] S. S. Beauchemin, and J. L. Barron, “The computation of optical flow”, *ACM Computing Surveys*, vol. 27, issue 3, 1995.

[251] B. D. Lucas, and T. Kanade, “An iterative image registration technique with an application to stereo vision”, In: *Proceedings of Imaging Understanding Workshop*, pp. 121-130, 1981.

[252] J. J. Verbeek, N. Vlassis, and B. Kröse, “Efficient greedy learning of gaussian mixture models”, *Neural Computation*, vol. 15, no. 2, pp. 469-485, 2003.

[253] S. M. A. Eftakhar, J. K. Tan, H. Kim, and S. Ishikawa, “Robust human motion recognition employing adaptive database structure”, *ICROS-SICE International Joint Conference*, pp. 3989-3994, 2009.

[254] C. Papageorgiou, and T. Poggio, “A trainable system for object detection”, *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15-33, 2000.

[255] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”, In: *Proceeding of the International Conference on Computer Vision and Pattern Recognition (CVPR’05)*, pp. 886-893, 2005.

[256] D. M. Gavrila, and V. Philomin, “Real-time object detection for smart vehicles”, In: *Proceeding of the International Conference on Computer Vision*, vol. 1, pp. 87-93, 1999.

[257] P. Viola, M. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance”, In: *Proceeding of the International Conference on Computer Vision*, vol. 2, pp. 734-741, 2003.

[258] P. Felzenszwalb, and D. Huttenlocher, “Pictorial structures for object recognition”, *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[259] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, “Fast human detection using a cascade of histograms of oriented gradients”, In: *Proceeding of the International Conference on Computer Vision and Pattern Recognition*, pp. 1491-1498, 2006.

[260] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance”, In: *Proceeding of European Conference on Computer Vision*, pp. 428-441, 2006.

[261] Y. Pang, Y. Yuan, X. Li, and J. Pan, “Efficient HOG human detection”, *Signal Processing*, vol. 91, pp. 773-781, 2011.

[262] X. Wang, T. X. Han, and S. Yan, “An HOG-LBP human detector with partial occlusion handling”, In: *Proceeding of the International Conference on Computer Vision*, pp. 32-39, 2009.

[263] L. Fei-Fei, and P. Perona, “A bayesian hierarchical model for learning natural scene categories, In: *Proceeding of IEEE Computer Vision and Pattern Recognition (CVPR’05)*, pp. 524–531, 2005.