Negative space analysis for human action recognition

Shah Atiqur Rahman

2012

Shah, A. R. (2012). Negative space analysis for human action recognition. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50706>

<https://doi.org/10.32657/10356/50706>

NEGATIVE SPACE ANALYSIS FOR HUMAN ACTION RECOGNITION

Shah Atiqur Rahman

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Dedicated to my parents

Shah Md. Shahjahan

&

Amatullah Begum

ACKNOWLEDGEMENTS

I express my heartfelt gratitude to **Prof. Dr. Leung Karhang, Maylor** who was my supervisor for 3.5 years during my PhD candidature. His invaluable guidance, suggestions, constant encouragements and kind behavior brought the best out of me. I benefit a lot from his serious attitude to research work and insightful criticism. Apart from that, I would like to specially mention about his advice regarding my attitude towards doing research as a post graduate student. This helped me a lot in adapting to the research environment in this university.

I am deeply grateful to my 2nd supervisor **Dr. Siu-Yeung Cho, David** for his guidance and suggestions. His helpful behaviour encouraged me a lot during challenging situations.

I am very much thankful to my supervisor **Assoc. Prof. Ramakrishna Kakarala** for his guidance, suggestions and cooperation during the last stage of my candidature. I consider myself very lucky to have such supervisors.

I am thankful to **Dr. Li Liyuan** for his insightful discussion and providing me invaluable technical advice. I am also thankful to my laboratory mates and Bangladeshi friends, who are studying in Singapore, for their warm help and encouragements.

Finally, special thanks to my family back in my country for their continual love, support and wishes. They always encouraged me to do well in my studies. Without their support, I would not be what I am today.

Singapore 2011

Shah Atiqur Rahman

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 CHALLENGES AND THE PROPOSED APPROACH	4
1.2 OBJECTIVES	10
1.3 CONTRIBUTIONS	11
1.4 DATASETS	12
1.5 THESIS ORGANIZATION.....	14
2. PREVIOUS WORKS	15
2.1 FEATURE TRACKING BASED METHODS	15
2.1.1 2D Feature Tracking Approaches	16
2.1.2 3D Feature Tracking Approaches	19
2.2 INTENSITY AND GRADIENT BASED METHODS.....	22
2.3 SILHOUETTE BASED METHODS	29
2.3.1 Implicit Methods	30
2.3.2 Explicit Methods	33
3 SYSTEM OVERVIEW	38
4. PREPROCESSING	45
4.1 LONG SHADOW ELIMINATION	45
4.2 SPEED COMPUTATION	48

4.2.1 <i>Generating Fuzzy Membership Functions</i>	52
4.3 BOUNDING BOX CUTTING	56
4.4 SUMMARY	58
5. REGION PARTITION AND POSE DESCRIPTION	59
5.1 REGION PARTITION	59
5.1.1 <i>Partitioning Process</i>	60
5.1.2 <i>Fine Tuning:</i>	65
5.2 POSE DESCRIPTION	67
5.2.1 <i>Positional Feature</i>	67
5.2.2 <i>Shape Based Features</i>	68
5.2.2.1 Contour-based description.....	68
5.2.2.2 Region based shape representation	69
5.2.2.3 Moments.....	71
5.2.2.4 Area	78
5.2.2.5 Principle axis lengths	78
5.2.2.6 Orientation.....	79
5.2.2.7 Rectangularity	79
5.2.2.8 Horizontal and vertical side length	80
5.3 SUMMARY	82
6. SEQUENCE MATCHING	84
6.1 DISTANCE BETWEEN POSES	84

6.2 DISTANCE BETWEEN SEQUENCES	87
6.2.1 <i>Recognizing Action Type</i>	91
6.2.2 <i>Query for Complex Activity</i>	95
6.3 SUMMARY	98
7. EXPERIMENTAL RESULTS	100
7.1 ACTION CLASSIFICATION	100
7.1.1 <i>Recognition Using Less Information</i>	123
7.1.2 <i>Feature Extraction Time</i>	124
7.1.3 <i>Contribution of different features</i>	125
7.2 ROBUSTNESS TEST	125
7.3 SEARCHING FOR COMPLEX ACTIVITY	128
7.4 SUMMARY	131
8. CONCLUSION AND FUTURE WORK.....	133
8.1 SUMMARY	133
8.2 CONTRIBUTIONS	137
8.3 FUTURE WORKS	138
8.4 CONCLUSION	140
REFERENCES:	141

LIST OF FIGURES

Fig. 1.1: Example of a marker based system [3].....	3
Fig. 1.2: Images of a mug and a chair.....	6
Fig. 1.3: Adobe Acrobat logo	7
Fig. 1.4: Pose description by means of negative space regions. (a) Input pose, (b) After region partitioning process, (c) Negative space regions represented as triangle or quadrangle.	8
Fig. 1.5: Block diagram of the system.	10
Fig. 1.6: Example frames of datasets used in the system. (a) Weizmann dataset, (b) KTH dataset and (c) UIUC complex activity dataset	12
Fig. 2.1: Representation of human poses by different systems. (a) stick figure [35], (b) ribbons [36], (c) blobs [29] and (d) rectangular patches [37].	18
Fig. 2.2: The tracking pipeline of Menier et al. [51] method: (a) Color image (b) Silhouette (c) Visual hull (d) Medial axis points (e) Skeleton pose	20
Fig. 2.3: Three stages approach of Lee and Nevatia [30] method. (a) Input. (b) Blobs tracking. (c) 2D inference. (d) 3D inference.	21
Fig. 2.4: Example of Local posture representations: (a) Space-time interest points in [58] and (b) Spatio-temporal features in [59]	25
Fig. 2.5: Example of SIFT features [32]. (a) Gradient orientation, (b) computed orientation histograms for sub-regions	28
Fig. 2.6: Example of 3D SIFT features [32]. (a) Three Gradient orientations, between row and column (α), between row and time (β), between column and time (γ) (b) computed orientation histograms for sub-regions.	29

Fig. 2.7: Example images of (a) MHI [33] (b) MHV [103](c) space-time shapes [26].....	33
Fig. 2.8: Feature extraction process of Ikizler and Duygulu [34] method. (a) input image, (b) segmented image, (c) candidate rectangular patches for different limbs (d) dividing the pose with equal sized grid and (e) generated histogram.	35
Fig. 2.9: Example images from Vuuren [25] system. (a), (b), and (c): example input images. (d), (e), (f), and (g): color-coded features, extracted from negative space regions for some poses.....	37
Fig. 3.1: Detail system diagram	39
Fig. 3.2: Complex regions are partitioned into simple regions.....	41
Fig. 3.3: Positional features. 14 anchoring points are labeled along the bounding box	42
Fig. 3.4: Anchor point shifting for two similar poses	43
Fig. 4.1: Discarding shadows. For each column, input is shown in top row, segmented image with shadow is shown in the next row, histogram of foreground pixels of input image is shown in 3 rd row, and bottom image shows result after discarding the shadow.....	46
Fig. 4.2: Block diagram of the speed score computation process.....	48
Fig. 4.3: Viewpoint normalization for horizontal speed calculation. The dots represent centroids of silhouettes, solid line is the major principle axis along the centroids, dashed line is the X-axis, δ is the angle between action performing plane and image plane and obs_{disp} , ac_{disp} are the observed displacement and actual displacement respectively.....	50
Fig. 4.4: Flow chart of speed calculation.....	51
Fig. 4.5: Membership functions for Weizmann dataset. (a) 10 membership functions for 10 types of actions (b) the final membership functions after merging.....	53

Fig. 4.6: (a) Area under Gaussian curve is shown according to 3-sigma rule. (b) Overlapping of two Gaussian membership functions with their respective limits according to 3-sigma rule. Gray colored area is the overlapped area.....	55
Fig. 4.7: Bounding box cutting. (a) Background segmented image, (b) Bounding box containing the human silhouette only.	57
Fig. 4.8: Flipping of bounding box. (a) Bounding box of a frame of an action which direction is left to right. (b) Flipped bounding box, making action direction from right to left.....	57
Fig. 5.1: The necessity of region partitioning. (a) No gap appears near the hand, (b) gaps appear near the hands of the person. (c) after region partitioning of (b).	59
Fig. 5.2: Region partitioning process. (a) Input image, (b) partitioning process for vertical side of the region 'x' (c) Partition output of region 'x' (d) final output image with new region names.....	61
Fig. 5.3: Graph for selecting P_{min} to partition a region.	62
Fig. 5.4: Region partitioning process for two sides of bounding box. (a) Input image. (b) and (c) Partition result of region 'P' for vertical and horizontal side respectively. (d) Combined output.....	64
Fig. 5.5: Removal of small regions. (a) before and (b) After removal of small regions. ...	65
Fig. 5.6: Information loss with respect to area of empty regions.	66
Fig. 5.7: Region positioning with numbers representing the anchoring points and letters representing the region. '*' represents the location of the mid-point on the boundary.....	67
Fig. 5.8: (a) Empty regions can be viewed as triangle or quadrangle, (b) pose representation by simple shapes.....	77
Fig. 5.9: Major and minor axis for a single negative space region.....	78

Fig. 5.10: Orientation of a region (Solid line represents the major principle axis, dashed line is the X-axis).....	79
Fig. 5.11: Rectangularity. Region is shown in red boundary and bounding rectangle is shown in green.	80
Fig. 5.12: <i>ver_len</i> and <i>hor_len</i> of a region.....	81
Fig. 5.13: Combining positional and shape based features. (a) a pose is shown with anchoring points. (b) feature vector with block numbering.....	82
Fig. 6.1: Pose matching (a) Pose 1 (b) Pose 2, (c) The feature vectors matching from (a) to (b) with one cross matching of regions.	85
Fig. 6.2: Matching matrix PM, for the two feature vectors from Fig. 6.1 ($\text{Inf} = \infty$).....	86
Fig. 6.3: An example of a DTW matrix D . Dashed gray line is the warping path with starting and end indices as $(1, i_1)$ and (n, i_2) respectively	89
Fig. 6.4: Matching process of model sequence with test sequence containing single action type.	91
Fig. 6.5: Example of doubly matching scheme. Number represents the frame number within a sequence. Test sequence contains 121 frames and model sequence contains 36 frames. 1 st match is from frame 8 to 41 and second match is from frame 51 to 80 of test sequence.....	93
Fig. 6.6: Block diagram of searching complex activity	95
Fig. 6.7: Matching process for one action query $action_k$. D_l is the DTW matrix for model l , $sp(l, j)$ is the speed score difference between model l and input sequence.....	98

Fig. 7.1: Example frames of different types of actions. For each group, 1 st row: input, 2 nd row: background segmented image, 3 rd row: output of shadow discarding process, 4 th row: bounding box image and 5 th row: after region partition.	115
Fig. 7.2: Example of segmented image frames with leaks, holes and protrusions from correctly recognized sequences (a) Weizmann dataset and (b) KTH dataset.....	116
Fig. 7.3: Optimized Fuzzy membership functions. (a) for Weizmann dataset (b) for KTH dataset, LOO scheme and (c) for KTH dataset, spilt based scheme.....	117
Fig. 7.4: Confusion matrix of our system for the KTH dataset. (a) leave-one-out scheme (b) split based scheme. For both cases all scenes are taken as a single dataset.	119
Fig. 7.5: Example frames of noisy background segmentation.....	120
Fig. 7.6: Mean distance of nearest correct and incorrect choice of each action type. (a) Weizmann dataset, (b) KTH dataset LOO scheme and (c) KTH dataset split scheme.....	123
Fig. 7.7: Recognition rate of our system for less information	124
Fig. 7.8: Accuracy without different types of features for Weizmann dataset	125
Fig. 7.9: Example of image frames used in robustness experiment. Upper row is the actual image and lower row is segmented image for each frame. (a) Normal walk, (b) Swinging bag, (c) Carrying briefcase, (d) Walking with a dog, (e) Knees up (f) Limping man, (g) Moon-walk, (h) Occluded legs (i) Walking in a skirt and (j) Occluded by a pole.....	126
Fig. 7.10: Correct and nearest incorrect choice of each deformed action.....	127
Fig. 7.11: Example frames from robust view dataset. For each column first row is original image and second row is corresponding segmented image. The view	

angles are (a) 0 degree, (b) 18 degrees, (c) 36 degrees (d) 54 degrees and (e) 81 degrees.	128
Fig. 7.12: Correct and incorrect choice of each view changing action.....	128
Fig. 7.13: Searching result of 15 queries. (a) Our method. On each row the ranking of relevant sequences are shown (b) Comparison with Ikizler and Forsyth [7] method. Queries are numbered according to their position in (a), e.g. query ‘walk-carry’ is 1 and ‘walk-stand-wave-walk’ is 15.	131

LIST OF TABLES

Table 5.1: The relationship from μ_{11} , μ_{20} , μ_{02} , ϕ to θ [160].	75
Table 7.1: Accuracy of different methods for Weizmann dataset	118
Table 7.2: Accuracy of different methods for KTH dataset	119
Table 7.3: Accuracy of different methods for different scenes of KTH dataset	120
Table 7.4: Recognition accuracy for different scenes treated as different datasets (LOO)	121
Table 7.5: Recognition accuracy for different scenes treated as different datasets (Split)	122
Table 7.6: Different methods accuracy for robust deformation dataset	126
Table 7.7: List of videos in UIUC complex human activity dataset	129

ABSTRACT

In this thesis a novel region-based method is proposed to recognize human actions. Other region-based approaches work on the silhouette of the human body which is termed as the positive space according to art theory. On the contrary, we investigate and analyze regions surrounding the human body, termed as the negative space for human action recognition. This concept takes advantage of the naturally formed negative regions that can always be described/approximated using simple shapes, simplifying the job for action classification. Negative space is less sensitive to segmentation errors, overcoming some limitations of silhouette based methods such as leaks or holes in the silhouette caused by background segmentation. The proposed method also addresses the problem of long shadows which is one of the major challenges of human action recognition. Some systems attempted to suppress shadows during segmentation process but our system takes input of segmented binary images where the shadow is not suppressed or discarded. This makes our system less dependent on the segmentation process. Nonetheless, the system also tackles the problem of video sequences retrieval by query of complex human actions.

The system consists of some hierarchical processing blocks: histogram analysis on segmented input image, motion and shape feature extraction, pose sequence analysis by employing Dynamic Time Warping and at last classification by Nearest Neighbor classifier. We evaluated our system by most commonly used datasets and achieved higher accuracy than the state of the arts methods. The proposed system is found relatively robust to partial occlusion, variations of clothing, noisy segmentation, illumination change,

deformed actions etc. Furthermore, good accuracy can be achieved even when half of the negative space regions are ignored. This makes the proposed system robust with respect to segmentation error and distinctive from other approaches. In future, this system can be combined with positive space based methods to make an improved system in terms of accuracy and computational costs.

Chapter 1

INTRODUCTION

One of the general goals of artificial intelligence is to design machines which act intelligently or human-like. To fulfill this goal, natural language understanding, large knowledge bases, and sophisticated reasoning have made significant contributions which are supported by the Turing test. Nevertheless, they provide only a partial solution. For a machine to be truly intelligent and useful, it requires the ability to perceive the environment in which it is embedded. It needs to be able to extract information from its environment independently, rather than rely on information supplied to it externally by keyboard input. Perhaps the most relevant information to be retrieved for interaction is where and who are the humans in the environment and what their activities are. In everyday life, humans perceive the activities of surrounding people. Suppose, in our work place we are curious about the activities of our co-researcher whether they are doing research or meeting with other researcher; in school common room whether my friends are watching football match on TV or reading newspapers. For humans, recognition of different activities is easy due to his/her past experience but actually it requires complicated task of sensing, recognition and inference. For instance, to make the decision whether a person is smoking or not, we need to gather some evidences like seeing the cigarette, smelling the smoke etc. With the evidences humans can make the decision based on the past experience. All these tasks, gathering information from the environment,

recognizing the event based on experience, and reacting in response of the event in the environment are still great challenge for machines. In this regard, computer vision can play an important role. As one of the most active research areas in computer vision, visual analysis of human motion attempts to detect, track and identify people, and more generally, to interpret human behaviors, from image sequences involving humans [1]. An added benefit of such a capability is that it makes communication with machines easier for mankind, allowing input modalities such as gestures. For example, a person extends his hands to a robot to handshake. In reply the robot may do the same to make the handshake happen by recognizing the action of the person from his gesture and this will avoid the necessity of giving input from keyboard and eventually improve the communication mechanism with the robot.

To fulfill the goal of human action recognition, the machine should have two capabilities: sensing or input, and perception or recognition. First, the mechanism of taking input. Humans have several sensors to gather information from the environment like eyes, ears etc. Similarly, we need to set a mechanism for a machine to take input from the environment. Sensing of human action for machine can be generally divided into two methods: marker-based methods and marker-less methods. In marker based methods some markers are attached with predefined location of the human body and then cameras are used to track the markers only, not tracking the human body. These kinds of systems are widely used in the animation and film industry [2]. One example of marker based system is shown in Fig. 1.1 [3]. For recognition of activities, marker based techniques can estimate body models and pose estimation accurately. However, these systems can be

implemented only in controlled environments which prevent them to apply on some application area such as video surveillance. Moreover, the target model is determined in advance, therefore the final data can be used for one purpose only [2]. In marker-less methods, actors in the image sequences are dressed normally; no markers are attached which means these approaches are less controlled than marker based methods. Recently, many approaches have been proposed on marker-less approach [4]. The proposed method in this thesis used marker-less input method.

Fig. 1.1: Example of a marker based system [3]

Second, the machine must have a mechanism to recognize actions which is the main focus of this thesis. Many approaches have been proposed, for example, optical flow [5], bag of words [6], tracker based method [7] etc. All the methods have their own pros and cons. In the proposed system a region-based method is developed which works on the surrounding regions of the human body.

Traditionally, there has been keen interest in human movement from a wide variety of disciplines [8]. In psychology [9], there have been the classic studies on human perception. In kinesiology (i.e., biomechanics [10]), the goal has been to develop models of the human body that explain how it functions mechanically and how one might increase its movement efficiency. In choreography [11], there has been long-term interest in devising high-level descriptions of human movement for the notation of dance, ballet, and theatre. Computer graphics has dealt with the synthesis of human movement [12]. This has involved devising realistic models of the human body for applications in crash simulations, workplace assessment, and entertainment. Another important application domain is advanced user interfaces [13] in which human motion analysis is usually used to provide control and command. In addition, human motion analysis shows its importance in other related areas. For instance, typical applications in virtual reality include chat-rooms, games, virtual studios, character animations, teleconferencing, video surveillance etc. [14].

1.1 Challenges and the Proposed Approach

It's a challenging task to achieve robust action recognition due to cluttered background, camera motion, occlusion, viewpoint changes, and geometric and photometric variances of objects. These challenges are common to a broad range of computer vision tasks. A cluttered background introduces irrelevant information to the foreground, making the latter harder to get isolated. Camera motion creates ambiguities in the motion patterns that are observed in the image plane e.g. when an object is moving with the same speed and direction as the camera, it could make an object appear static. In addition, human

actions can also be observed only partially due to occlusions, thus the actual signal of interest can be dramatically reduced. Viewpoint changes as well as geometric and photometric variances produce very different appearances and shapes for the same category. Sometimes, long shadow becomes a problem in case of pose recognition. It may result in connecting up blobs that should not be associated with the silhouette of the person.

Another challenge in action recognition is the intra-class variations. For many actions there could be some variations during flight of activities. For example, walk can have different speed and stride lengths [15] or a person can limp during walking due to leg problem. Anthropometric differences between individuals and/or clothing variations can also create intra-class variations. A good human action recognition approach should accommodate variations within one class and distinguish differences between action types.

Despite the fact that good results have been achieved by some action recognition approaches, there are still some limitations. Many of them involve computation of optical flow [5, 16, 17], which is difficult to derive due to, e.g., aperture problems, smooth surfaces, and discontinuities. Others, [7, 18, 19] employ feature tracking and face difficulties in cases of self-occlusions, change of appearance, and problems of re-initialization. Local features are extracted in bags-of-words methods [6], lacking temporal co-relation between frames. Methods that rely on key frames or Eigen-shapes of foreground silhouettes (e.g., [20, 21]), lack information about the motion. Some

approaches are based on periodicity analysis (e.g., [20, 22, 23]) and are thus limited to cyclic actions.

The region-based approaches work reasonably well as described in [1]. Most region based approaches work with the regions that are part of the human body. We propose a novel approach focusing on regions surrounding the human body termed as negative space. In formal studies of art, negative space is defined as follows: 1) the space between an object and the edges of the canvas i.e. the space labeled by pencil sketch in Fig. 1.2; or 2), the space between objects (Fig. 1.3 where the red regions are the objects) [24]. In other words, the spaces surrounding objects, or within the objects, are defined as negative space. In Fig. 1.2 two examples of negative space are shown where the artist did not draw the edges of the mug or chair. Instead, the surrounding regions (negative space) are sketched by a pencil, which eventually make the objects. From the figures, it can be seen that an object can be represented by its surrounding regions.

Fig. 1.2: Images of a mug and a chair

One of the application areas of negative space drawing is logo design as shown in Fig. 1.3 where the logo of Adobe Acrobat is shown. In this logo the letter ‘A’ is created by the space between the red colored objects.

Fig. 1.3: Adobe Acrobat logo

In our proposed work, we employ the first definition of negative space (i.e. spaces between object and canvas). According to the definition in our case, a human silhouette is the object (since we use background segmented binary image as input) and the canvas is defined as the bounding box of the silhouette. The region between the silhouette and the edges of bounding box is defined as negative space (also sometimes called as empty regions in this thesis since these regions do not contain any foreground pixels) as illustrated in Fig. 1.4(a). By means of negative space, simple and naturally formed regions between the canvas and human body can be used for pose or action description as shown in Fig. 1.4(c). This avoids derivation of sophisticated (but possibly non-robust) scheme to obtain consistent region description from positive space that can be afflicted with leaks and holes inside the silhouette due to image segmentation problem. Since this method does not involve tracking of limbs, it does not face difficulty in case of self-occlusion. Also, the proposed system does not require initialization like the feature tracking methods [19]. In

this system the features are extracted first locally and then combined the local features to obtain global pose description. This gives the opportunity to take advantage of local pose description but still preserve the temporal co-relation between frames which is discarded in case of bag-of-words methods [6]. Moreover, the system utilizes the speed feature which gives the motion information of actions.

Fig. 1.4: Pose description by means of negative space regions. (a) Input pose, (b) After region partitioning process, (c) Negative space regions represented as triangle or quadrangle.

Another advantage of negative space based pose description is robustness to partial occlusion and small shadow. In our system, when partial occlusion or small shadow occurs, it will affect only 1 or 2 negative space regions inside the bounding box, not all the negative space regions. This implies that during partial occlusion pose description will be affected partially which means intra-class variations will be low in presence of partial occlusion. Moreover, we also propose a method to handle long shadows which is one of the major challenges of human action recognition. Some other methods [21] discard shadows during segmentation process which makes those methods much dependent on segmentation. Further, to the best of our knowledge there is no such segmentation algorithm which can segment shadows perfectly from all scenarios. Hence, in our system during segmentation we do not suppress the shadows but feed directly the segmented images as input to other processing blocks. It is found that the negative space feature can

compliment the positive space approach. This can be exploited to build a better action recognizer in future.

Vuuren [25] had derived an approach similar to our proposed method. However, that system only color code the negative spaces, which actually give the positional information of the regions, not the shape information. As a result that system can give similar result for different shapes depending on the color of the negative space. Vuuren's approach was very sensitive to noise, as one pixel can split a single region into two color coded region. Also, they had not used any motion information of the person in their system which makes their system difficult to recognize moving actions (e.g. running). In our proposed system all positional, shape and motion information of the negative space regions are extracted. Moreover, region partitioning process is applied on negative space regions to make complicated shapes into simpler shapes. Further, the proposed system can recognize actions in presence of long shadows.

The block diagram of the proposed work is illustrated in Fig. 1.5. Input to this system is a video which consist the human action. The video is converted into image frames and the background is segmented from each frame. In our case, during segmentation no shadow suppression techniques are applied, which implies that our input segmented images may contain shadows if shadow is present in the original video. In the shadow elimination step shadows are detected and discarded from the segmented images. Speed of the person and direction of the action are then calculated. Meanwhile, a bounding box (i.e. the canvas) is cut from the image, containing the whole silhouette of the person. Inside the bounding box attention is focused on empty regions (i.e. negative space regions)

which are not part of the human silhouette. Complex empty regions are partitioned in the “region partitioning” block to simplify the pose description process, at which features of the empty regions are extracted. Two types of features are extracted, one is the positional feature and the other is the region-shape features. Finally, pose sequence comparison is done by DTW and input action is recognized by Nearest Neighbor classifier based on the speed and DTW score.

Fig. 1.5: Block diagram of the system.

1.2 Objectives

The objectives of the work are listed below

- 1) Identify and capture negative spaces
- 2) Derive a pose description based on the negative space regions
- 3) Develop a technique to compare poses

- 4) Recognize actions based on the extracted features
- 5) Make the system robust so that it can adapt with shadow, partial occlusion and deformable actions, and
- 6) Devise a frame work hence the system is able to search complex human activities in a single video.
- 7) Compare the accuracy of the proposed system with other state of the art methods.

1.3 Contributions

Main contributions of the proposed system are

- 1) Building on the work of [25], we propose in this work a novel way to employ negative space for describing human poses and recognizing actions based on that description. New feature descriptor has been proposed to describe human poses that support recognizing human actions effectively.
- 2) A novel system based on the negative space concept has been successfully built with performance comparable or better than other systems.
- 3) The proposed system addresses some of the major challenges of human action recognition and proposes novel techniques to overcome them. The system is found robust to partial occlusions, shadows, clothing variations, deformable actions etc.
- 4) The system also supports complex activity searching (content based video retrieval). A complex activity can be for example a ‘walking’ action followed by a ‘running’ etc.

1.4 Datasets

Fig. 1.6: Example frames of datasets used in the system. (a) Weizmann dataset, (b) KTH dataset and (c) UIUC complex activity dataset

To evaluate the performance of the proposed system, publicly available datasets are employed. First dataset used in the system is known as Weizmann human action dataset [26]. Weizmann human action dataset contains 9 persons performing 10 natural actions, to provide a total of 90 videos. This dataset provides a good testing bed to investigate the performance of the algorithm in the presence of relatively large number of action types. This dataset also contain other two small datasets for testing the performance under viewing angle change, partial occlusions, clothing variations, shadows and deformed action. Second dataset is known as the KTH human action dataset [27] which is the most commonly used dataset for human action recognition. This dataset is more challenging than the Weizmann dataset due to considerable amount of camera movement, long shadows, different clothing and scale and viewpoint changes. It contains 6 action types performed by 25 persons in 4 different scenarios. To test the system performance for complex activity, UIUC complex human activity dataset [7] is employed. In this dataset 5 actors performed several composite actions in a single sequence. There are a total of 73 sequences of composite actions in this dataset. Each sequence is captured against a simple background. The main challenge of this dataset is to detect different action types without knowing the actions boundary in the sequence. Example frames of different datasets are shown in Fig. 1.6. More images are given in Fig. 7.1.

1.5 Thesis organization

The remainder of this thesis is organized as follows. In chapter 2 previous related works are reviewed. In chapter 3, we give an overview of our system. Shadow elimination and speed calculation are discussed in chapter 4. Chapter 5 describes the procedure to partition complex regions into simpler regions and extract features from regions. The matching process of the input sequence with training data is described in chapter 6. In chapter 7, we provide experimental results with analysis and discussion on the result. In the last chapter we conclude the thesis.

Chapter 2

PREVIOUS WORKS

This chapter reviews the relevant research done in the field of human pose, action and gesture recognition. Automatic capture and analysis of human motion is a highly active research area due to the number of potential applications and its inherent complexity. Much research has been done in this field [1, 4, 8, 14, 15, 28]. Some of the related works are discussed in this section. Many criteria could be used to classify previous works: for example low level feature cues, type of models, dimensionality of the tracking space etc. We have chosen the first criteria since we mainly focused on proposing simple and efficient features for pose description. Existing related works can be broadly divided into three major categories based on the low level feature cues [14, 28]. They are *Feature tracking based methods*, *Intensity and gradient based methods*, and *Silhouette based methods*. In the following subsections, some of the related works [29-34] of each category are highlighted.

2.1 Feature Tracking Based Methods

Many methods recognize actions based on tracking features in either 2D or 3D space. Hence body segments can be approximated as 2D ribbons and 3D volumes accordingly.

2.1.1 2D Feature Tracking Approaches

Some methods built a model first and then recognize actions by tracking the changes of different limbs. These models are represented by stick figure [35], ribbons [36], blobs [29], rectangular patches [37] etc. as shown in Fig. 2.1. Guo et al. [35] proposed a 2D stick figure by obtaining the skeleton of the silhouette of the walking human and matching it to a model stick figure. They used a combination of link orientations and joint positions of the obtained stick figure as features for a subsequent action recognition step. In the work by Leung and Yang [36] the subject's outline was estimated as edge regions represented by 2D ribbons which were U-shaped edge segments. The 2D ribbon model was used to guide the labeling of the image data. Wren et al. [29] developed a system called as Pfinder, which is a real-time system for tracking people and interpreting their behavior. They used small blob features to track the single human in an indoor environment. Both human body and background scene were modeled with Gaussian distributions. The pixels belonging to the human body were assigned to different body parts of blobs using the log-likelihood measure. Ju et al. [37] proposed a cardboard people model, in which the limbs of human were represented by a set of connected planar patches. The parameterized image motion of these patches was constrained to enforce the articulated movement, and was used to deal with the analysis of articulated motion of human limbs. Deva et al. [38] learned a global body part configuration model based on conditional random fields to simultaneously detect all body parts where the constraints were aspect ratio, scale, appearance, orientation and connectivity. This approach demonstrates impressive results for pose estimation in cluttered scenes such as sports images. Bregler et al. [18] presented a comprehensive framework using the statistical

decomposition of human body dynamics at different levels of abstractions. In the low-level processing, the small blobs were estimated as a mixture of Gaussian model based on motion and color similarity, spatial proximity, and prediction from the previous frames. During the intermediate-level processing, those regions with coherent motion were fitted into simple movements of dynamic systems. Finally, HMMs were used as mixture of these intermediate-level dynamic systems to represent complex motion.

Sometimes tree structure model representation is used for pose estimation [39, 40]. In these models body parts are represented by nodes and edges of graph where nodes represent limbs and edges represent kinematics constraints between connected limbs. Forsyth and Fleck [41] introduced the notion ‘body plans’ to represent human as a structured assembly of parts. Motivated from [41], Falzenwalb et al. [39] represented the visual appearance of human as a composition of rigid parts which were arranged in a deformable configuration. In this method they introduced an efficient global matching algorithm which was able to find multiple good matches of a model to an image. Ioffe and Forsyth [42] method sequentially drew body parts and made the prediction by matching the predicted configuration with image observations. Hua et al. [40] modeled body configuration by Markov network and the estimation of body parts was done based on pose parameters from image cues such as appearance, shape edge and color. Instead of using rectangular patches of [42] they employed quadrangles to represent body parts which facilitated inference of pose parameters. Mikolajczyk et al. [43] introduced robust AdaBoost body part detectors to locate different parts of a human. These parts modeled the joint likelihood of a body parts configuration. Micilotta et al. [44] extended this approach

by using RANSAC to assemble body parts configuration with pose constraints which were learned previously. Since the descriptions of body limbs are local, these methods can fail to track different limbs with same image data (e.g. two legs of a person). Part-based and tree structured 2D model based human tracking has been proposed in [45, 46]. Sigal and Black [45] used a graphical model where each node of the graph represents different limbs and edges represents kinematics constraints between different limbs. Their method could recover 2D poses even in presence of occlusion but the method needed manual initialization. Later Ferrari et al. [46] proposed very similar method to [45] where the initialization was done automatically.

2D feature tracking approaches are relatively computationally simple, usually work on the positive space and provide semantic level description of poses. However, these methods are sensitive to the details of humans shape variations, noise and segmentation errors.

Fig. 2.1: Representation of human poses by different systems. (a) stick figure [35], (b) ribbons [36], (c) blobs [29] and (d) rectangular patches [37].

2.1.2 3D Feature Tracking Approaches

Besides 2D feature tracking, some methods used 3D information of body parts to classify actions. An early work by Rohr [47] made use of 14 elliptical cylinders to model human body in 3D 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 2D projections were fit to the 3D human model using a similarity distance measure. The constraint propagation scheme of Chen and Lee [48] starts at the human head and continues via the torso to the limbs. An interpretation tree was built, to account for the spatial ambiguity, which was pruned later for physically implausible poses. After reconstruction, tracking was done by ellipsoid matching and their system can track motion in real time. Their assumption of six known feature points on the head to start the procedure and the overhead of the interpretation tree made their approach somewhat unappealing for practical applications. Chueng et al. [49] reconstructed 3D poses by voxel-based reconstruction algorithm where tracking was done by ellipsoid matching. Sminchisescu and Triggs [50] proposed a method for monocular 3D human body tracking by optimizing a cost metric which worked on the model-image matching. The optimization was performed using covariance scaled sampling, an extension of particle filters which increases the covariance in the direction of maximum uncertainty by approximately an order of magnitude. By using a gradient descent approach, samples were then optimized to find the local minima. Menier et al. [51] recognized motion by using 3D skeletal model. 3D poses were estimated from medial axes of the visual hull which were obtained from multiple views foreground silhouettes (Fig. 2.2). Hofmann and Gavrila [52] estimated upper body pose from 3 cameras where candidate pose were generated from one camera

image and verified by other two camera images. However, this method cannot handle occlusion.

Fig. 2.2: The tracking pipeline of Menier et al. [51] method: (a) Color image (b) Silhouette (c) Visual hull (d) Medial axis points (e) Skeleton pose

Ikizler and Forsyth [7] proposed a tracker based method where complex queries were searched by Finite State Automata. At first a compositional model for each body part was constructed by using a motion capture dataset. After forming the compositional models over 3D data, they tracked the 2D video with a state-of-the-art full body tracker and lifted 2D tracks to 3D by matching the snippets of frames to motion capture data. By this lifting procedure view invariance was achieved, since their body representation is in 3D. Finally, they wrote text queries to retrieve videos. However, if the tracker failed to track different limbs correctly, which can occur during self-occlusion, the system performance will be degraded. Lee and Nevatia [30] proposed a three-stage approach which enables a hierarchical estimation of 3D poses (Fig. 2.3). They addressed various issues including automatic initialization, self and inter-occlusion etc. Conversely, computation speed of their system was high.

Fig. 2.3: Three stages approach of Lee and Nevatia [30] method. (a) Input. (b) Blobs tracking. (c) 2D inference. (d) 3D inference.

Horaud et al. [53] proposed a method for tracking human motion based on fitting an articulated implicit surface to 3D points and normals. They exploited the analogy between the distance from a set of observations to the implicit surface and the negative log-likelihood of a mixture of Gaussian distributions. However, their system performance decreases for improper initialization of the algorithm. Shen and Foroosh [54] proposed a view invariant method which worked on fundamental ratios, derived from the ratios of fundamental matrix of different frames captured by a fixed camera. Peursum et al. [55] proposed a full 3D human tracking system with high dimensional body model where some important issues were addressed such as partial occlusion, long-term tracking, generalization to different people tracking etc. In generative method of [56], 3D pose is inferred from 2D poses. In this method 2D poses were estimated by a bottom-up approach i.e. searching different limbs first then construct the whole pose by using kinematics constraints. One of the major drawbacks of 3D approaches is the initialization problem. Learning based approaches (e.g. [57]) address this problem and initialize automatically. These methods are fast and conceptually appealing. However, learning based methods may fail due to wide variations of pose and external parameters.

3D feature tracking techniques are accurate and invariant under viewing angle changes but the computational cost is high. Multiple cameras may be required for reconstruction of the 3D body configuration. Hence it is difficult to apply for human action recognition on video from real world scenes.

In general, the Feature tracking approach is complex due to large variability of shape and articulation of the human body, fast motions, self occlusion, changes of appearance etc.

2.2 Intensity and Gradient Based Methods

These methods employed gradient or intensity based features to recognize actions. In most of the cases, first the space-time interest points are detected from the input pose and then actions are recognized by modeling the interest points.

‘Bag-of-words’ approaches originated from text retrieval research are being adapted for action recognition. These studies are mostly based on the idea of forming codebooks of ‘spatio-temporal’ features. Laptev and Lindeberg [58] first introduced ‘space-time interest points’ features and used SVM to classify actions. By using linear filters Dollàr et al. [59] extracted cubes from poses and recognized actions by forming histogram from these cubes whereas Niebles et al. [60] used pLSA on the extracted cubes for recognition. The algorithm automatically learned the probability distributions of the spatial-temporal words and the intermediate topics corresponding to human action categories. Their approach can handle noisy feature points arisen from dynamic

background and moving cameras due to the application of the probabilistic models. Bregonzio et al. [61] refined the approach of [59] by using Gabor filters in both spatial and temporal dimensions. Wang et al. [6] employed hierarchical Bayesian models (the Latent Dirichlet Allocation (LDA) mixture model, the Hierarchical Dirichlet Processes (HDP) mixture model, and the Dual Hierarchical Dirichlet Processes (Dual-HDP) model) to connect three elements in visual surveillance and was able to complete many challenging visual surveillance tasks including activity recognition without tracking and human labeling effort. However, “hard” division is a problem in their system as they divided long sequence into short clips.

Han et al. [62] recognized actions in movies by extracting features based on Histograms of Gradients (HoG) and Histograms of Flow (HoF) around space-time interest points. They proposed a Gaussian process classifier named Multiple Kernel Gaussian Process Classifier (MKGPC) which learnt all parameters and hyper-parameters by maximizing Bayesian marginal likelihood measure. Laptev et al. [31] extracted HoG and HoF descriptors around detected space-time interest points, and use k-means in order to construct a codebook. Similar work was done in [63], where SIFT features were also used. Zhang et al. [64] proposed a boosted multi-class semi-supervised learning algorithm for human action recognition. They extracted interest points by [31] and as features they employed HoG and HoF. By combining adaboost.MH and co-EM they could extract information from un-labeled data. Though their system achieved good accuracy, their system computational complexity is high.

By extending the spatial salient points of [65] in time, Oikonomopoulos et al. [66] detected spatiotemporal salient points in image sequences. By measuring the variations in the information content of pixel neighborhoods in space and time dimensions, the spatiotemporal salient points were detected. A distance metric, which was based on the chamfer distance and an iterative linear time-warping technique, was proposed to compare two sets of spatiotemporal salient points that dealt with time expansion or time-compression issues. Jhuang et al. [67] used a hierarchy of Gabor filters in order to construct their C-features. With a set of Gabor filters they extracted dense local motion information. The responses of the filter were pooled locally, and converted to higher-level responses by comparing to more complex templates which were learnt from examples and fed into a discriminative classifier. Schindler and Gool [68] extended the method of [67] by combining both shape and optical flow responses. Ning et al. [69] proposed similar method of [67] which employed responses of 3D Gabor filter banks in order to build their descriptors. Modified spatial correlogram [70] were utilized to capture the spatial-temporal structural information of video words. Then they applied k-means clustering algorithm to form a code book and performed Maximization of Mutual Information (MMI) to make a compact representation of the codebook. At last SVM was used to classify actions. Savarese et al. [71] proposed spatial-temporal correlations to capture the underlying distribution of video words. Hu et al. [72] proposed the descriptive video phrases and descriptive video cliques in spatial-temporal domain which helped to capture and understand the semantic meaning of a video word.

Fig. 2.4: Example of Local posture representations: (a) Space-time interest points in [58] and (b) Spatio-temporal features in [59]

Leibe et al. [73] proposed an implicit shape model for object detection which consists a codebook of visual words, maintaining the relative position of each word with respect to the object center. Their method combined the capabilities of object categorization and segmentation in one common probabilistic framework. Oikonomopoulos et al. [74] extended the work of Leibe et al. [73] by proposing a voting scheme in the space-time domain that allows both the temporal and spatial localization of activities. The voting nature of [74] allowed spatiotemporal localization and classification in sequences. A similar voting scheme was implemented by Mikolajczyk and Uemura [75] for activity recognition and localization. However, this work was restricted only to the spatial localization of the subjects at each frame. In [76] Mikolajczyk and Uemura extended their earlier work [75] where local features were detected by four different detectors (Maximally Stable Extremal Regions (MSER) [77], Pairs of Adjacent Segments (PAS) [78] Harris-Laplace and Hessian-Laplace [79]), described by Gradient Location-

Orientation Descriptor then the dimensionality of descriptors was reduced by local discriminant projection [80, 81] and the features were tracked to obtain motion vectors. Shao et al. [82] method extracted interest points by using separable linear filters. They employed three transform based techniques, Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), to describe the detected interest points and found superior accuracy for DWT. Yuan et al. [83] extracted spatial-temporal features proposed by [58] and for classification they extended the naïve Bayes nearest classifier to recognize the actions. Willems et al. [84] extended the exemplar-based object detection approach employing local features into spatiotemporal domain. Rapantzikos et al. [85] proposed a 3D salient points approach to solve the energy minimization problem which involved a set of spatiotemporal constraints. Fathi and Mori [86] developed a method based on mid-level features which were extracted from low level optical flow features. AdaBoost classifier was then used to recognize actions from these mid-level features. However this method needs some parameters to be manually initialized. Yao et al. [87] proposed a spatiotemporal voting framework where voting was performed in two steps for dimensionality reduction. Seo and Milanfar [88] proposed a feature representation which was derived from space-time local regression kernels (3D LSKs). These 3D LSKs captured the underlying data quite well even in case of significant distortion. They also proposed a training-free non-parametric detection scheme based on matrix cosine similarity measure. The bag of words approaches were successful for simple periodic actions [4].

Scale Invariant Feature Transform (SIFT), which was introduced by Lowe [89], has been used in a variety of applications including object (e.g., [90]) and scene (e.g., [91] [92]) classification. SIFT descriptor is extracted based on Gaussian space. At each scale, gradient magnitude and gradient orientation is computed. To achieve rotation invariance the neighborhood of the interest point is rotated for each dominant orientation. Then the neighborhood is divided into multiple sub-regions. In each sub-region an orientation histogram is produced. One example of 2D SIFT feature descriptor is shown in Fig. 2.5. Uemura et al. [93] employed SIFT feature with a motion model based on optical flow in order to accurately classify multiple actions. Their system was able to successfully recognize actions on a sequence containing large motion and scale changes. Zhou et al. [94] proposed SIFT-bag based generative-to-descriptive framework for event recognition. In the generation stage they extracted bag of SIFT features which distribution was described by Gaussian Mixture Models (GMM). In the discriminative stage they designed the SIFT-bag kernel based on the upper bound of the Kullback-Leibler divergence of the specialized GMMs of two video clips. SIFT-bag representation had the capability to implicitly match the content of two video which were different in spatial position, scale and temporal position. Liu et al. [95] proposed a system where multiple features were used to recognize actions. As features they used cuboids, which were extracted using the method of [59], and SIFT features [89]. To classify the actions they employed hyper-space multi-class SVM called OC-K-SVM [96]. Partly inspired from [89], the Speeded Up Robust Features (SURF) [97] utilized second order Gaussian filters and the Hessian matrix in order to obtain a fast scale and rotation invariant interest point detector and descriptor. Some author employed 3D SIFT features to recognize actions [32, 98]. In 3D SIFT feature

time dimension is utilized along with the spatial (xy-plane) dimension (shown in Fig. 2.6(a)). Scovanner et al. [98] extracted 3D SIFT feature from video sequence and employed SVM classifier for recognition purpose. Similar to 3D SIFT descriptor, 3D SURF was proposed by Willems et al. [99] which is a direct extension of SURF. In comparison with 3D SIFT descriptor, 3D SURF is faster, relied on integral video representation and Haar wavelet responses. Mattavi et al. [100] experimented on KTH human action dataset employing both 3D SIFT and 3D SURF features. They found that 3D SIFT gave slightly better result than 3D SURF but 3D SURF is approximately 35 times faster than the 3D SIFT descriptors. Sun et al. [32] proposed a method where local and template based features were combined and then SVM classifier was used to recognize actions. As a local feature they extracted 2D and 3D SIFT descriptors. For template based features, they employed frame differencing technique and extracted features by Zernike moments. Generally, the high dimensionality of the descriptor is a drawback of SIFT descriptor.

Fig. 2.5: Example of SIFT features [32]. (a) Gradient orientation, (b) computed orientation histograms for sub-regions

Fig. 2.6: Example of 3D SIFT features [32]. (a) Three Gradient orientations, between row and column (α), between row and time (β), between column and time (γ) (b) computed orientation histograms for sub-regions.

In general, the performance of intensity and gradients based methods depends on the reliable and sufficient detection of stable interest points which can be difficult to derive in case of noisy and dynamic backgrounds.

2.3 Silhouette Based Methods

Silhouette based methods are becoming popular due to their robustness to noise and easy extraction of region of interest [101]. Silhouette based methods first obtain foreground silhouette by background segmentation and then extract features from the segmented image to recognize actions. Proposed methods on silhouette based action recognition differ in the way of description of poses and the modeling of pose sequence. Based on the dynamics of the actions, proposed methods can fall into two categories: *implicit* and *explicit*.

2.3.1 Implicit Methods

In case of implicit model, features are extracted from combination of all the silhouettes of an action sequence. Motion History Image (MHI) and Motion Energy Image (MEI) is one of the way to describe action as temporal template [33, 102] where matching was done by Hu moments [33] or SVM classifier [102]. First, motion images in a sequence were extracted by frame differencing and these motion images were accumulated in time to form MEI. Then the MEI was enhanced into MHI which was a scalar-valued image. Taken together the MEI and MHI were considered as a two-component version of a temporal template, a vector-valued image in which each component of each pixel was some function of the motion at that pixel position. Finally, these view-specific templates were matched against the stored models of views of known actions during the recognition process. 3D extension of temporal templates was proposed by Weinland et al. [103] where Motion History Volume (MHV) was constructed by using images of multiple cameras and classification was done by Fourier analysis in cylindrical co-ordinate. Gait Energy Image (GEI) and Motion Intensity Image (MII) was proposed where actions were recognized based on Eigen space methods and dimensionality reduction was done by either PCA [104] or LDA [21]. Lam et al. [105] proposed two similar gait representations for recognition: Motion silhouette contour template (MSCT), for capturing the motion characteristics of a gait, and static silhouette template (SST) for capturing the static characteristics of a gait. However, the generation of an MSCT mainly used the silhouette contour which was more sensitive to noise than the binary silhouette and SST could only extract the static information from the gait sequence, which was not able to capture the motion information. To overcome these limitations they proposed another method [106] called gait flow image

(GFI) which was very similar to GEI but achieved better accuracy in gait recognition. Zhang et al. [107] proposed an active energy image (AEI) which had the advantage of retaining the dynamic characteristics of gait for recognition. Bazin et al. [108] method was a fusion of a dynamic feature and two static features in a probabilistic frame-work where the dynamic signatures were derived from bulk motion and shape characteristics of the subject. The first static signature was a vector, yielded from an average silhouette, and the second static vector was obtained from a block-based silhouette averaging method. They proposed a process for determining the probabilistic matching scores using intra-class and inter-class variance models together with the Bayesian rule. Kumar et al. [109] proposed a LDA tree based classifier to recognize actions. They represented an action by thinning the human silhouette of each frame and then formed histogram for each grid of bounded ellipse of the thinned silhouette. However, their system is not scale invariant and non-robust to partial occlusion. Veres et al. [110] proposed a two stage data fusion rule to fuse the dynamic feature and the first static signature mentioned in [108]. Shechtman and Irani [111] proposed a behavior-based correlation to compute the similarity between space-time volumes which allowed to find similar dynamic behaviors and actions. At each pixel, the space-time gradients of the corresponding video patch was computed and summarized in a matrix. The eigen-values of the resulting matrices were used to compute similarity between two spatial-temporal patches. Therefore, this method required significant computation due to the correlation procedure between every patch of the testing sequence and the video database. Gorelick et al. [26] represented actions as space-time shapes and extracted space-time features for action recognition, such as local space-time saliency, action dynamics, shape structures and orientation. These features were extracted by using

properties of the solution to the Poisson equation. The advantages are it does not need video alignment and it can handle low quality video. Yilmaz and Shah [112] extended space time shape into space time volume (STV) and extracted differential geometric properties to describe actions. A combination of STVs of silhouettes and flow was used by Ke et al. [113]. A 4D action feature model (4D-AFM) was proposed by Yan et al. [114] where 3D STV [112] was mapped into 4D action shape of ordered 3D visual hulls. Recognition was performed by matching STV of observed video with 4D-AFM. Sometimes Radon Transform was used as a descriptor for human actions [115-117]. Boulgouris et al. [115] used Radon Transform for gait recognition. Singh et al. [116] used Radon Transform for pose recognition, but their work was restricted on hands or feet position. Chen et al. [117] used Radon Transform to select key postures in an action sequence. Then key postures were combined to construct an action template for each sequence. Later, LDA was used for dimensionality reduction. Lin et al. [118] described actions by a combination of GMMs which were extracted from Category Feature Vector (CFV) ignoring the temporal ordering of the silhouettes.

The implicit modeling approaches have the advantage of simple action recognition and can work under small number of training data. However, action dynamics are weakly described. Further, periodic and cyclic actions should be handled differently in these methods [101].

(a)

(b)

(c)

Fig. 2.7: Example images of (a) MHI [33] (b) MHV [103] (c) space-time shapes [26].

2.3.2 Explicit Methods

In explicit models, actions are treated as a composition of a sequence of poses. These models consist of two components: pose description and modeling of the dynamics of poses. Ikizler and Duygulu [34] described poses by histogram of oriented rectangles and used DTW, NN, SVM to recognize actions. Yamato et al. [119] made use of the mesh

features [120] of 2D moving human blobs such as motion, color and texture, to identify human behavior. Action dynamics were modeled by HMMs which were trained to generate symbolic patterns for each action class, and the optimization of the model parameters was achieved by forward-backward algorithm. Moments were used to describe poses and dynamics were modeled by HMM in [121]. Wang and Suter [122] used Kernel Principle Component Analysis (KPCA) based feature extraction and Factorial Conditional Random Field (FCRF) for modeling. Lao et al. [123] proposed a layered framework that enabled multi-level human motion analysis. For pose recognition they extracted features called HV-PCA where horizontal and vertical projection of human pose was taken first and then PCA was applied to get a compact representation of each pose. Modeling of human actions was done by Continuous Hidden Markov Model (CHMM) with left-right topology. Lv and Nevatia [124] modeled the dynamics by un-weighted directed graph where each node represents the key postures and edges represent allowed transition between key poses. Li et al. [125] represented actions by weighted graph where salient poses were represented by nodes and edges are numbered by the transitional probabilities. They modeled poses by GMM and salient poses were captured by clustering. Jia and Yeung [126] proposed a manifold embedding method called as Local Spatio-Temporal Discriminant Embedding (LSTDE) to find both spatial and temporal discriminant structure of the human silhouette. Abdelkader et al. [127] described poses by 2D closed elastic curves, parameterized by square-root parameterization and pose dynamics were modeled by template and graphical based HMM. Wu et al. [128] developed Increment Discriminant-Analysis of Canonical Correlation (IDCC) which could update discriminant transformation matrix over time. This gives the advantage of adaptation with high

irregularities quickly and recognizes actions effectively. Ali and Aggarwal [129] identified knee and hip by calculating curvature of the silhouette. Then three angles surrounding the hip and knee point were used as features. Their system fails due to large number of points of curvature and false knee and hip detection.

Fig. 2.8: Feature extraction process of Ikizler and Duygulu [34] method. (a) input image, (b) segmented image, (c) candidate rectangular patches for different limbs (d) dividing the pose with equal sized grid and (e) generated histogram.

The dynamic texture (DT) model [130] is an efficient method for modeling the dynamics of image sequence [131-133]. Mazzaro et al. [131] described poses by the angles of the shoulder, elbow, hip and knee joints and employed the angle vector to get a nominal model. Bissacco and Soatto [132] introduced a representation with the projection features, which encoded the distance of the points on the silhouette from lines passing through its centroid. The auto regressive and moving average (ARMA) model in [133] was constructed on the tangent space projections of a shape sequence extracted from the binary images by uniform sampling or uniform arc-length sampling. In [134] a hybrid autoregressive model of human motion and novel algorithms was proposed to estimate switches and model parameters. A set of joint-angle trajectories on the skeletal model were extracted to train the hybrid model for dynamic discrimination. Chan and Vasconcelos [135-137] further developed the modeling capability of DT by KPCA [135], mixture of

DT [136] and layered DT [137] approaches. In [138] an improved dynamic Bayesian network called the Layered Time Series Model (LTSM) was proposed to directly model binary image sequence. Their method can tackle the limitations hindering the application of HMM, DT/logistic DT to gait recognition.

Most of the explicit model based techniques needs to model each actions individually. Both implicit and explicit model face difficulties in case of long shadow, partial occlusion and noisy segmentation.

Our proposed system is an explicit model based technique where poses are described by extracting features from the negative space and the action dynamics is modeled by DTW. Moreover, a technique is proposed which can discard long shadow from segmented image effectively. Unlike other explicit model based techniques, our system is relatively robust to partial occlusion and noisy segmentation since negative space based methods extracts features, first locally and then combines the features to give global description of a pose. Vuuren [25] proposed a method similar to ours where poses were described by negative space and action recognition was done by string matching technique. In Vuuren's method, negative spaces were extracted using horizontal and vertical scanning processes. The color-coded negative space representation was formed using the four binary images produced by the scanning processes. Features were then extracted from the color-coded images which were based on the percentage of area occupied by distinct colored regions as well as the bounding box proportions. Subsequent images were classified using a simple Euclidean distance measure and action recognition was done by detecting

temporal order of the poses that characterizes the action. However, according to that system, the color-code could describe the shape of the negative space regions and its position but same color can be assigned to different shapes and same color can occur in different positions as shown in Fig. 2.9 (bottom row). For example, yellow color can take place between two legs and the space between arms and torso (Fig. 2.9(d)). Moreover, no motion analysis was performed in that system which makes it difficult to recognize moving actions (e.g running, walking). Also, the system can be significantly affected by noise. In our system, we have employed positional information, shape-based information and motion analysis of the person which makes our system capable of recognizing actions effectively. We also propose a method to handle long shadow which is one of the major challenges of human action recognition. Our proposed system is relatively robust to noise, partial occlusion, variations of clothing, long shadow etc.

Fig. 2.9: Example images from Vuuren [25] system. (a), (b), and (c): example input images. (d), (e), (f), and (g): color-coded features, extracted from negative space regions for some poses.

Chapter 3

SYSTEM OVERVIEW

Based on the discussion of previous chapter, we found that silhouette based methods are popular due to their robustness to noise and easy extraction of region of interest. Among the two categories of silhouette based methods, implicit methods have the limitation that the action dynamics are weakly described, and periodic and cyclic actions are handled differently whereas explicit methods do not have those limitations. However, explicit methods face difficulties in case of shadows, partial occlusion and noisy segmentation. In this thesis, we propose an explicit silhouette based method that extracts features from negative space. The extracted features are shown to be robust to partial occlusion and noisy segmentation. We also propose a shadow elimination technique to recognize actions effectively in the presence of shadows in the input sequence.

The proposed system block diagram is shown Fig. 3.1. Input of the system is a video file containing the human action performed by a person. Presently, it is assumed that each video contains only one person doing some activity. Multi-person activity recognition is left as the future work of current study. The video file is converted into image sequences by a frame grabber and then the foreground is segmented from the background. In the current proposed system no segmentation algorithm is developed. It is assumed that all the image sequences are pre-segmented. If the image sequences are not pre-segmented, we employed other background segmentation algorithm to segment the background.

Fig. 3.1: Detail system diagram

Many background segmentation approaches have been proposed [139]. Among them background subtraction is widely used [140-142]. We employed one of the background subtraction algorithms [142] in our work. Two versions of this segmentation are available: one with shadow detection and the other without. In this work, the without-shadow-detection version is employed. After segmentation, binary images are obtained, with each containing one human silhouette.

During segmentation no shadow discarding algorithm is applied and hence the segmented binary images may contain shadow. Existence of shadows in segmented images may alter the pose description. Some other methods [21] discard shadows during segmentation process which makes those methods much dependent on segmentation. Further, to the best of our knowledge there is no such segmentation algorithm which can segment shadows perfectly from all scenarios. Hence, in our system during segmentation we do not suppress the shadow. Negative space based methods have the advantage of robustness to the small shadows since the system first gather the local information and then combine them to construct a global description of the pose. This implies that small shadow may partially alter the pose description, not the global pose description in case of negative space based methods. However when long shadow exists in the segmented image, it could be problematic. Hence, in the *long shadow elimination* step a histogram based technique is applied to detect and discard long shadow from segmented binary images. We refer to shadows as small shadows if their existence do not change the size of the bounding box of the human body. Otherwise, we term them as long shadows.

Humans perform different actions in different speeds (e.g. running is performed faster than walking). Hence, speed can be an important cue in human action recognition which is calculated in the next step. The inputs of this step are the frame rate of the video and all the segmented binary images of a sequence, obtained after the *long shadow elimination* step. In the proposed system, speed is calculated based on the horizontal displacement of the human body centroid. In case of testing sequences, speed score is determined based on Fuzzy functions. The Fuzzy functions of different type of actions are generated automatically from the training sequences. Next, a bounding box containing the human body is captured which actually extract the negative space regions. Some action directions are reverted such that we do not need to generate two action models for same actions performed in opposite directions. More details on *Long shadow elimination*, *speed* and *bounding box computation* are given in chapter 4.

Fig. 3.2: Complex regions are partitioned into simple regions

In the region partitioning step, each empty region is separated for further processing. In particular, complex empty regions are partitioned into simple regions, so that each region can be described by small and simple set of feature values. Region

partitioning is done by connecting suitable peninsula of the body to the bounding box as shown in Fig. 3.2.

In the *pose description* step, suitable features are extracted to describe the input pose. Two types of features are extracted in this step. One is the positional information of the region and the other is region-shape features. For extracting the positional information, the bounding box is labeled with 14 anchoring points as shown in Fig. 3.3. Each region is then assigned to the nearest anchoring point. For shape description of each negative space region, six region-based features are extracted. They are *orientation*, *eccentricity*, *area*, *rectangularity*, *horizontal length* and *vertical length*. Next, these two types of features are combined to generate feature vectors for each pose. The details of region partitioning and pose description are given in chapter 5.

Fig. 3.3: Positional features. 14 anchoring points are labeled along the bounding box

Testing data are compared with the model data in *Sequence matching* step. Here two hierarchical comparisons are done. In the first level two poses are compared and in the second level two sequences are compared. In case of pose matching we consider the

scenarios where shifting of regions can occur (i.e. positional information can vary for same type of pose) as shown in Fig. 3.4 where for two similar regions ‘A’ and ‘B’ the anchor point has shifted from 5 to 4. Hence, a distance metric needs to be developed which allows negative space regions to shift at most one position without any penalty and calculate the distance between poses effectively. In this regard, a distance calculation is proposed to determine the distance between two poses in presence of region shifting.

Fig. 3.4: Anchor point shifting for two similar poses

The temporal duration of an action performed by a subject can vary at different scenarios. Therefore, time warping should be applied during matching of two sequences. For calculating the distance between two sequences Dynamic Time Warping (DTW) is applied. After performing the DTW matching, simple Nearest Neighbor classifier is applied by taking input of speed score and DTW score to recognize the type of the input video.

Humans tend to perform composite activities on temporal dimension. For example, in a video one person first starts walking and then running. To search for such composite activities in a single video, a framework is developed in this system. The proposed framework employed all the processing steps of single action recognition with an exception in sequence matching. More details on template matching and searching of composite activities are given in chapter 6.

Chapter 4

PREPROCESSING

This chapter first describes the technique to eliminate long shadows from the segmented binary images. Afterward, it gives a detail description of speed calculation based on the shadow eliminated image sequence. After calculating the speed, extraction of negative space regions is stated. To simplify the recognition process, some actions performed in opposite direction are reverted, which is described in the last part of this chapter.

4.1 Long Shadow Elimination

Shadow is one of the major challenges in human action recognition. Existence of shadows in segmented images may alter the pose description since the shadow could be treated as parts of the human silhouette. Researchers have proposed many methods to detect and eliminate shadows from color images [143-149]. These approaches, which worked on color images, detected shadow using color-related cues. However, in our method shadow elimination is performed on binary images since we tackle shadows after the segmentation process. Hence, we proposed a shadow elimination technique which works on the binary images instead of color image by employing the contextual information of the shadow to tackle it. This makes our computation simpler than other methods.

Fig. 4.1: Discarding shadows. For each column, input is shown in top row, segmented image with shadow is shown in the next row, histogram of foreground pixels of input image is shown in 3rd row, and bottom image shows result after discarding the shadow.

Long shadows have the characteristics that they are connected to the lower part of the body and are projected on the ground as shown in Fig. 4.1 where three scenarios of shadow discarding process is shown. Motivating from these characteristics of long shadows, a histogram based analysis is proposed to isolate shadows from the body and then discard it. In Fig. 4.1 input images are shown in top row and their corresponding

segmented images are shown in the 2nd row where the shadow is connected at the lower part of the body. Next a histogram is generated based on the foreground pixels (black pixels in Fig. 4.1) of segmented binary image. Number of bins in the histogram is same as the number of columns of segmented image and frequency of each bin is the pixel count of foreground pixels along the corresponding column of the input image. Empirically we found that the lower foreground pixels (lower than 10% of the silhouette height) of columns (bins) that satisfy equation (4.1) belong to the shadows.

$$\frac{|c_i|}{\max_{k=1}^n |c_k|} \leq 0.20 \quad (4.1)$$

In (4.1) $|c_i|$ is the pixel count of foreground pixels of the i^{th} column and n is the number of columns in the input image. Hence, we remove lower pixels of those columns and then take the biggest blob (the biggest group of connected pixels of foreground) from the image. Eventually shadows are discarded from the input image as shown in Fig. 4.1 bottom row. Observe that the hands of the person in left and middle images are thin and satisfy equation (4.1), but are not discarded, since they are not within the lower region of the silhouette. Note that this shadow elimination technique will also remove the small shadows in the frame. This shadow detection algorithm may fail in case of multiple shadows in one side of the body (e.g. players' shadows in a football match) if the accumulated bin values of the histogram corresponding to the shadow are higher than the specified threshold.

4.2 Speed Computation

Human actions differ in their performing speed. For example, the walking action is slower than the running action, during handclapping the body displacement is approximately zero whereas during jogging there is some body displacement. Hence, action performing speed can be an important cue to distinguish different types of actions. Inclusion of speed feature in our feature set improves the system performance in terms of accuracy. However, without speed our system performance is degraded but still, the accuracy is significant, which is shown by means of experiments in Section 7.1.3.

Fig. 4.2: Block diagram of the speed score computation process

The block diagram of speed calculation process is shown in Fig. 4.2. Input of this process is the image sequence which is obtained after the *long shadow elimination* step. For each image frame in the sequence centroid of the human body is calculated. In the next step, displacement change of the human body centroid in two consecutive frames is

calculated and then speed is calculated by means of displacement change and the frame rate of the sequence. In case of training data, Fuzzy membership functions are generated. For testing data speed score is calculated by means of Fuzzy membership functions generated from the training data and the speed of the testing sequence. In more details, speed of a person could be calculated as

$$hor_{sp} = \frac{ds}{dt} = \frac{|x_i - x_{i-1}|}{1/fr_rate} = |x_i - x_{i-1}| \times fr_rate \quad (4.2)$$

where x_i is the x-axis coordinate of the centroid of the human body of the i^{th} frame and fr_rate is the frame rate of the sequence. To remove the scaling effect, we normalize equation (4.2) by the height of the bounding box of the person. Expressing with respect to the average value, we have

$$hor_{sp} = \frac{t_hor_{disp}/(n-1) \times fr_rate}{t_height/(n-1)} \quad (4.3)$$

where $t_hor_{disp} = \sum_{i=2}^n |x_i - x_{i-1}|$, n is the total number of frames in the sequence and t_height

is the sum of all bounding box heights excluding the first frame.

The height value is used as the normalization factor because in most actions, height of the bounding box remains unchanged. Height changing actions, for example bending-in-front (see Fig. 7.1), will not be affected too much as the average height is used. Simplifying equation (4.3), we get

$$hor_{sp} = \frac{t_hor_{disp} \times fr_rate}{t_height} \quad (4.4)$$

Equation (4.4) can calculate speed effectively when the action is performed in parallel to the image plane but in case of actions which have body displacement not performed in parallel with the image plane (i.e. viewpoint is changed), it calculates the observed speed rather than the actual speed of the body as shown in Fig. 4.3.

Fig. 4.3: Viewpoint normalization for horizontal speed calculation. The dots represent centroids of silhouettes, solid line is the major principle axis along the centroids, dashed line is the X-axis, δ is the angle between action performing plane and image plane and obs_{disp} , ac_{disp} are the observed displacement and actual displacement respectively.

In Fig. 4.3, action is performed in a plane which makes an angle (δ) with the image plane. In this scenario, equation (4.4) will calculate the observed speed. To calculate the actual speed we need to compute the actual displacement of the person as

$$t_hor_{disp} = t_obs_{disp} / \cos \delta \quad (4.5)$$

Here δ is calculated as [150]

$$\delta = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right] \quad (4.6)$$

where, μ_{ij} is the i, j^{th} order centralized moment of silhouette centroids (the black dots in Fig. 4.3) of the human in the sequence. Details of centralized moments are given in Chapter 5 during shape based feature extraction (specially, equation (5.6)). The actual speed with viewpoint normalization is

$$ac_hor_{sp} = \frac{\frac{t_obs_{disp}}{\cos \delta} \times fr_rate}{t_height} = \frac{hor_{sp}}{\cos \delta} \quad (4.7)$$

During speed calculation, we first calculate the speed hor_{sp} , using equation (4.4). If the action has significant body movement (e.g. walking, running), we apply equation (4.7) to normalize the speed with respect to viewpoint change and compute the actual speed ac_hor_{sp} . Otherwise (e.g. waving, clapping) ac_hor_{sp} is same as hor_{sp} since for non-moving actions viewpoint change does not affect the speed calculation. Hence, the *calculate speed of the sequence* blocks of Fig. 4.2 can be shown as Fig. 4.4 in further detail:

Fig. 4.4: Flow chart of speed calculation

4.2.1 Generating Fuzzy Membership Functions

In case of training sequences, for each type of actions fuzzy membership functions are generated which are then employed to calculate speed score for testing sequences. As a shape of the membership functions, Gaussian membership function is chosen since it requires only two parameters i.e. mean and standard deviation. These two parameters can be calculated from the training data of each action type. First, speed is calculated for each training sequence. Next, to avoid noisy data, 5% data of both highest and lowest boundary in terms of speed are truncated from each action type for calculating the membership function parameters. To avoid over-partitioning of feature space, membership functions are merged depending on the overlapped area of the membership functions. If one membership function is overlapped with more than 60% of its area with another membership function, both membership functions are replaced by a new membership function. The parameters of the new membership function are determined by employing the data of both the overlapped membership functions. This merging process goes on until no new membership function is generated. One example is shown in Fig. 4.5 where membership functions of 'skip', 'side' and 'jump' of Fig. 4.5(a) are merged into one membership function in Fig. 4.5(b). Same merging occurs for bend, jack, pjump, wave1 and wave2 membership functions.

To determine the overlap amount of two membership functions, the intersection area of two membership functions has to be determined. The intersection area of two curves can be found by integrating the difference between the curves up to the initial and final limits of the intersection points between the two curves. For example, let $f(x)$ and $g(x)$

be two curves with $x = a$, and $x = b$, be the initial and final intersection points respectively of the two curves, so, the area between the curves is

$$ins_area = \int_a^b (f(x) - g(x)) dx \quad (4.8)$$

(a)

(b)

Fig. 4.5: Membership functions for Weizmann dataset. (a) 10 membership functions for 10 types of actions (b) the final membership functions after merging.

Now, for two Gaussian curves with mean μ_1 and μ_2 , and standard deviation σ_1 and σ_2 , the intersection area will be

$$ins_area = \int_a^b \left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} - \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}} \right) dx \quad (4.9)$$

At this moment the task is to determine the value of two limits a and b . Usually, Gaussian distribution ranges from $-\infty$ to $+\infty$ but to reduce calculation three-sigma rule is employed. Three-sigma rule or empirical rule states that approximately 68% of values of a normal distribution lie within one standard deviation (σ) away from the mean; approximately 95% of the values lie within two standard deviations; and approximately 99.7% are within three standard deviations as shown in Fig. 4.6(a). To determine the lower and upper limit of equation (4.9) (i.e. a , b) three sigma rule is employed. Suppose two overlapped Gaussian membership functions g_1 and g_2 are given which parameters are μ_1 , σ_1 and μ_2 , σ_2 respectively. Hence the lower limit and upper limit of equation (4.9) will be

$$a = \max(\mu_1 - 3\sigma_1, \mu_2 - 3\sigma_2) \quad (4.10)$$

$$b = \min(\mu_1 + 3\sigma_1, \mu_2 + 3\sigma_2) \quad (4.11)$$

(a)

(b)

Fig. 4.6: (a) Area under Gaussian curve is shown according to 3-sigma rule. (b) Overlapping of two Gaussian membership functions with their respective limits according to 3-sigma rule. Gray colored area is the overlapped area.

One example of overlapping Gaussian membership functions is shown in Fig. 4.6(b) where the lower limit of equation (4.9) will be $a = \mu_2 - 3\sigma_2$ and upper limit will be $b = \mu_1 + 3\sigma_1$.

After calculating the *ins_area*, if the ratio $\frac{ins_area}{\min(area_g_1, area_g_2)}$ is greater than or equal to 0.6 (i.e. at least 60% overlap), the two membership functions will be merged.

4.3 Bounding Box Cutting

In this step, the human body is separated from the whole image frame. This can be done by cutting an up-right bounding box, which contains entire person, from the image. The bounding box is defined automatically by taking the minimum and maximum from the positive space, in both dimensions. Real images will have noisy segmentations. Noise can change the size of the bounding box, and furthermore, if the noise is connected with one side of the silhouette, it could change regions, which could partially alter the negative space based pose description. However, our system can recognize actions effectively even with the 50% negative space regions are corrupted due to noise or some other reasons as shown in experimental result section. For different frames, bounding box size could be different for a single video sequence but this does not affect the recognition process since we employed scale invariant features.

By cutting the bounding box, we capture the negative spaces which are the regions of interest inside the bounding box to recognize the input action. Fig. 4.7 shows one example of bounding box cutting from an image frame. In the later processing, only this bounding box image is used as input image, not the whole image frame and in later writings, image means only the image cut by the bounding box.

Fig. 4.7: Bounding box cutting. (a) Background segmented image, (b) Bounding box containing the human silhouette only.

Human can perform action in both directions (i.e. moving to the left or right of the image frame, boxing to the right or left side of the body). To make computation easier, we alter all moving action sequences (e.g. walk, run) into one direction (i.e. move to the left of the image frame). This is done by flipping pixels of all the bounding boxes of sequences moving from left to right about the Y-axis. A movement is seen as from left to right, if the X-coordinate of the human body centroid increases over time. Fig. 4.8 shows one example of the original and flipped images.

Fig. 4.8: Flipping of bounding box. (a) Bounding box of a frame of an action which direction is left to right. (b) Flipped bounding box, making action direction from right to left

For non-moving actions, if the limbs movement is asymmetric (e.g. box, one-hand-wave), we employed two sets of training data for a single sequence: the flipped and non-flipped images of the sequence. For symmetric movement actions (e.g. handclapping, two-hand-wave) we employed the training data as it is since for symmetric actions flipped image will be same as non-flipped image.

If we do not make all action direction into one direction we need to add actions with both directions in the training set (like asymmetric non-moving actions) which will increase the computation time eventually. Hence for reduced training set we perform the task of direction change.

4.4 Summary

In this chapter, at first, the process of long shadow elimination is described. A histogram based analysis is performed on each background segmented image to detect long shadows and then discard it. Next, the procedure of speed score calculation is described. The speed of a sequence is calculated based on the displacement of the centroid of the human body. This speed is invariant to viewpoint change. Then a fully automatic process is described to generate the fuzzy membership functions from training data. Gaussian membership functions are generated which parameters are computed from the training data and then significantly overlapped membership functions are merged to avoid over-segmentation of the feature space. Afterwards, extraction of negative space region are described which is done by cutting an up-right bounding box from the segmented image. At last, a technique is described for handling different directed actions.

Chapter 5

REGION PARTITION AND POSE DESCRIPTION

In this chapter, first a technique is described to partition complex negative space regions into simple regions. This process helped to describe each pose with simple shapes like triangle or quadrangle instead of more complex shapes. Later, features are extracted to describe each pose numerically. Two types of features are extracted to describe each pose: positional feature which describes the position of a negative space region inside the bounding box, and shape based features which describe the shape of the region. Next, these two types of features are combined to construct feature vectors.

5.1 Region partition

Fig. 5.1: The necessity of region partitioning. (a) No gap appears near the hand, (b) gaps appear near the hands of the person. (c) after region partitioning of (b).

Due to continuous movement, same pose of same person but captured in different time, may not share the same number of empty regions. Fig. 5.1 (a) and (b) show one example of this scenario with two bounded image taken from same pose and same person. In Fig. 5.1(a) both hands of the person has touched the bounding box which create five empty regions inside the bounding box. But in Fig. 5.1(b) none of the hands touches the side of the bounding box. As a result, regions ‘A’ and ‘B’ in Fig. 5.1(a) are merged in Fig. 5.1(b). Also, regions ‘D’ and ‘E’ of Fig. 5.1(a) are merged in Fig. 5.1(b). Consequently, when we compare the empty regions from these two images, they are likely to be classified as not similar. To overcome this and simplify the matching process, we propose to partition a region if a peninsula can be found to point to the bounding box. For example in Fig. 5.1(b), if we partition regions ‘X’ and ‘Z’ using the arms as peninsulas as shown in Fig. 5.1(c), it would be easier for the system to compute the similarity between Fig. 5.1(a) and (c). The partition detail will be illustrated in the next section.

5.1.1 Partitioning Process

The partition process is illustrated using Fig. 5.2. Fig. 5.2(a) is the input image with 3 empty regions ‘x’, ‘y’ and ‘z’. Region partitioning is performed in each region separately. For region ‘x’, the boundary includes two sides from the bounding box. For each side of bounding box, the top of the nearest peninsula is searched as shown in Fig. 5.2(b) by scanning along the bounding box boundary in a layer by layer (line by line) manner. We define a silhouette part as peninsula if that part is intersected by a line in the scanning process. When one of the lines intersects a peninsula (the silhouette), the scanning process stops.

Fig. 5.2: Region partitioning process. (a) Input image, (b) partitioning process for vertical side of the region 'x' (c) Partition output of region 'x' (d) final output image with new region names.

To examine validity of the peninsula, a protrusive measure is computed from three distances, max_distLT , $intersect_dist$ and max_distRB , with respect to the bounding box as shown in Fig. 5.2(b). Here max_distLT is the maximum distance from the bounding box side to a silhouette pixel belong to the top half of 'x' with respect to the peninsula (if bounding box side is one of the vertical lines) or to the left half of 'x' (if bounding box side is one of the horizontal lines). max_distRB is defined similarly to a silhouette pixel at bottom or to the right of the peninsula. $intersect_dist$ is the distance between the peninsula and the bounding box side. The protrusive measure is then defined as

$$protrusive = max_dist / intersect_dist \quad (5.1)$$

with $max_dist = \min(max_distLT, max_distRB)$. A minimum threshold P_{min} is set. If $protrusive \geq P_{min}$, the negative region is divided into two regions. This is done by

connecting the peninsula to the nearest pixel to the bounding box. Fig. 5.2(c) gives one example of the partition where region ‘x’ is partitioned into ‘x1’ and ‘x2’.

Fig. 5.3: Graph for selecting P_{min} to partition a region.

The threshold, P_{min} is set dynamically according to Fig. 5.3. For example, P_{min} is 3 and 1.75, if the max_dist value is 5 and greater than or equal 15 respectively. If the P_{min} value is big, no partition may occur which means the complex negative space regions remain, and it will not be properly represented by simple shapes (e.g. triangle or quadrangle) in the feature extraction step. This may result in using more than 4-sided polygons to represent the complex negative space region, which will increase the dimension of extracted feature vector, and eventually increases the computational complexity of the system. On the other hand, if the P_{min} value is small, the original negative space regions may be partitioned into several small negative space regions, whereas without partitioning the original regions might have been properly represented by simple shapes. This introduces overhead of computation for the newly formed regions. Hence, an appearance-based experiment was performed where the P_{min} value is set by visually observing the resultant pose. This experiment is performed on 300 images of the

Weizmann dataset. Later, in all experiment using the full dataset of Weizmann and other two datasets, it is found that the values set in the graph give optimal action recognition result. Incorrect selection of this value will result in either over partitioning, or no partition in the region.

After each partition, the same procedure is repeated on the newly formed regions. For regions with two sides from the bounding box, the partition process will be applied on each side separately and partition results will be combined using an *OR* operation. For example, Fig. 5.4 shows partition result for a region ‘P’ where the region is partitioned for both bounding box sides. Finally, partition result of each region is combined. One example of the final result is shown in Fig. 5.2(d). Algorithm 5.1 describes the region partitioning process.

Algorithm 5.1

Function reg_partition(*input_im*)

Begin

$[n, m] = \text{size}(\text{input_im});$

Output_im = zeros(*n, m*);

for each region *reg_i* in *input_im* create an image *a_i*, which contain only the region *reg_i*

o_reg = *a_i*

for each side of bounding box *s_j* included in *reg_i*

o_sj = *a_i*;

Start line growing from nearest line and parallel to *s_j*

if line intersect with silhouette //peninsula found

Calculate *intersect_dist* and *max_dist*

if (*max_dist* / *intersect_dist*) >= *P_{min}* //a valid peninsula for partition

Partition *reg_i* and make two images *a_{i_1}* and *a_{i_2}* each containing newly formed region

R_i = reg_partion(*a_{i_1}*);

```

     $R_2 = \text{reg\_partion}(a_i, 2);$ 
     $o\_s_j = R_1 \parallel R_2;$ 
    end
  end
   $o\_reg = o\_reg \& o\_s_j;$ 
  end
   $\text{output\_im} = \text{output\_im} / o\_reg;$ 
  end
  return output_im;
end

```

where *input_im* is the input pose with silhouette pixels=0 and other pixels=1, P_{min} is the threshold which is set based on the graph shown in Fig. 5.3, *size(im)* returns the size of image in terms of rows and columns, *zeros(n, m)* returns a n×m matrix of 0's, *max_dist* and *intersect_dist* are computed as shown in Fig. 5.2 and '&' and '|' are the AND and OR operations respectively.

Fig. 5.4: Region partitioning process for two sides of bounding box. (a) Input image. (b) and (c) Partition result of region 'P' for vertical and horizontal side respectively. (d) Combined output.

5.1.2 Fine Tuning:

Fig. 5.5: Removal of small regions. (a) before and (b) After removal of small regions.

It is observed that small regions can form inside the bounding box due to noisy segmentation, loose clothing of the person etc. These small regions do not provide vital information for pose recognition. By discarding these small regions, unnecessary complexity for recognition can be avoided and the system can be made faster. One example of the same pose with/without noise removal is shown in Fig. 5.5.

A region is seen as small, if its area is less than $1/50$ of the total empty area of the bounding box ($empty_bb_area$). The threshold value 50 is set according to observations (Fig. 5.6) on 2000 images of Weizmann dataset. On the X-axis of Fig. 5.6, we record the normalized size of empty regions as $empty_region_area/empty_bb_area$, on the Y-axis, the normalized accumulative area of the empty regions of sizes larger than or equal to X. For example, the graph indicates that the total area of empty regions with area greater than or equal to 0.16 of the empty bounding box area, will occupy 60% space of the total

empty area. The Y-axis value can be seen as the amount of information of the empty regions. From Fig. 5.6, it can be seen that about 98.5% of pose information is contained in the regions whose $empty_region_area/empty_bb_area$ value is greater than 0.02. Thus, if we remove the regions with area less than 1/50 (or 0.02) of the bounding area, we can avoid unnecessary calculation without losing much information. If this threshold value is set bigger than 50, then small negative space regions are not removed. This indicates that all the subsequent processing would be performed on those small regions which will increase the processing time, whereas these regions do not provide vital information for pose description.

Fig. 5.6: Information loss with respect to area of empty regions.

Though the threshold values used during *region partitioning* and *fine tuning* showed good result for the three datasets used in this thesis, there is a possibility that they may not give optimal results for other datasets. Optimal estimate of these thresholds is a complex topic, which we leave for future work.

5.2 Pose Description

To compare input poses, discriminative features should be extracted from each empty region to model a pose. We extract two types of features here. One is the positional information and other is the shape information of the regions.

5.2.1 Positional Feature

To extract positional information of regions, the bounding box is first labeled with 14 anchoring points as shown in Fig. 5.7. Afterwards each empty region is associated with one of these points by first computing the mid-point (*) that bisects the boundary on the bounding box and then finding the nearest anchoring point to the mid-point. For example, in Fig. 5.7, the anchoring points of regions A, B, C, D and E are points 1, 12, 9, 5 and 7 respectively. The employment of large number of anchoring points on the bounding box can make the region's location more accurate, but it can increase unnecessary computation. On the other hand, if we use less number of anchoring points, the location of the region will not be specific enough. Empirically we found that 14 anchoring points are good enough since we do not expect to find many large empty regions.

Fig. 5.7: Region positioning with numbers representing the anchoring points and letters representing the region. '*' represents the location of the mid-point on the boundary.

5.2.2 Shape Based Features

To describe the shape of the extracted negative space regions, some features have to be extracted. In this sub-section the shape based feature extraction process is described. In general, descriptors are some set of numbers that are produced to describe a given shape. A suitable shape representation should have the following criteria [151]:

- The representation of a shape should be invariant under different transformation (e.g. scaling, translation, rotation)
- The description should be good enough to find similarity between perceptually similar shapes
- The representation of shapes should be compact and easy to derive such that the finding similarity between shapes is efficient.

Many shape representation and description techniques have been proposed in the past [152]. Proposed method for shape description can be generally categorized into two classes: contour-based description and region-based description. Each of these classes can be subdivided into two approaches: structural approach and global approach. Brief description of shape description methods are given below.

5.2.2.1 Contour-based description

These methods describe shapes based on the boundary pixels of the shapes. Contour based shape description is divided into structural and global approach shape description.

Structural contour shape description: In the structural approach, shapes are described by broken the shape into boundary segments called primitives. These methods differ in the selection of primitives and the primitives' organization to represent the shape. Common

methods of this approach include chain codes [153], polygonal approximation [154], curve decomposition [155] etc. Since the structural description of shapes is local in nature, it has the advantage of handling occlusion problem. However, the demerits of this approach include computationally expensive, selection of appropriate primitives, sensitive to noise etc.

Global contour shape description: These techniques usually calculate a multi-dimensional numeric feature vector from the shape boundary information. Common simple global shape descriptors are eccentricity, principle axis orientation etc. Some other shape descriptor in this category include shape signature (e.g. curvature [156], boundary moments [157], spectral transform (e.g. Fourier descriptor, Wavelet transform [82]) etc. Simple global shape descriptors are compact and have perceptual meaning but they are not accurate enough to describe shapes perfectly. They need to be combined to represent shapes accurately. Wavelet transform suffers due to primitive selection [152] and Fourier descriptor cannot analyze local or transient property of a shape [82].

5.2.2.2 Region based shape representation

In region based shape representation, all the pixels in the shape are employed to represent the shape, rather than only using the boundary information. Same as the contour-based description, this category can also be divided as structural approach and global approach.

Structural region shape approach: Like the contour structural approach, this approach also decomposes the shape region into small parts (primitives) for shape description.

Example of this approach is convex hull, Medial axes [152] etc. Same as the contour structural approach this approach has the disadvantage of selection of primitives, computational complexity etc. Apart from these disadvantages, the matching technique is also an issue in this approach.

Global region shape approach: In this approach, all the pixels are used as a whole to extract some property to represent the shape. Common shape descriptors of this approach are geometric moments, shape matrix, orthogonal moments, generic Fourier descriptor etc. This approach makes the effective use of all the pixels in the shape and is less likely affected by noise and other variations. Most commonly used region-based global approach feature descriptions are moment based methods [152]. However, higher order moments are sensitive to noise and the higher order moments are difficult to associate with physical interpretation.

Among the two categories of shape descriptor (contour-based and region-based) we preferred region based shape descriptor since contour based shape descriptors are sensitive to noise and variations [158]. Among the two approaches of region based methods, global approach is chosen, since structural approaches are too complex to implement [152]. Hence, as a commonly used global region-based shape description we select moment based shape description. To avoid the limitation of moment based feature description (higher orders are sensitive to noise) only lower order moments (up to order 2) are employed which have physical meaning also. In the next section, we describe moment based feature description in more details.

5.2.2.3 Moments

Moments numerically describe the shape of a set of points. Moments are commonly used in statistics to characterize the distribution of random variables, and, similarly, in mechanics to characterize bodies by their spatial distribution of mass. The use of moments for image analysis is straightforward if we consider a binary or grey level image segment as a two-dimensional density distribution function.

Moments are applicable to many different aspects of image processing, ranging from invariant pattern recognition and image encoding to pose estimation. When applied to images, they describe the image content (or distribution) with respect to its axes. The general form of a moment of order $(p+q)$, evaluating over the complete image plane ξ is

$$M_{pq} = \int_{\xi} \int \psi_{pq}(x, y) f(x, y) dx dy \quad (5.2)$$

where $p, q = 0, 1, 2, \dots, \infty$. The *weighting kernel* or *basis* function is ψ_{pq} . The choice of basis may introduce constraints including limiting the x and y range, different kind of transformation etc.

Non-Orthogonal moments:

In case of non-orthogonal moments the basis set is non-orthogonal monomials. These monomials increase rapidly in range as the order increases, producing highly correlated descriptions. This can result in important descriptive information being contained within small differences between moments, which lead to the need for high computational precision. Two non-orthogonal moments are described below:

Cartesian moments: The two-dimensional Cartesian moment, m , of order $p + q$, of a density distribution function, $f(x, y)$, is defined as

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (5.3)$$

The discrete version of the Cartesian moment for an image consisting of pixels P_{xy} , replacing the integrals with summations, is

$$m_{pq} = \sum_{x=1}^M \sum_{y=1}^N x^p y^q P_{xy} \quad (5.4)$$

where M and N are the image dimensions and the monomial product $x^p y^q$ is the basis function. For binary image, m_{00} is the area of the region and the two first order moments, (m_{10}, m_{01}) , are used to locate the center of mass (COM) or centroid of the object. In terms of moment values, the coordinates of COM are

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \bar{y} = \frac{m_{01}}{m_{00}} \quad (5.5)$$

Centralized moments: The definition of a discrete centralized moment as described by Hu [159] is

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - \bar{x})^p (y - \bar{y})^q P_{xy} \quad (5.6)$$

Here $\bar{x}$ and $\bar{y}$ are the centroid of the region. This is essentially a translated Cartesian moment, which means that the centralized moments are invariant under

translation. By using centralized moments we will derive equations for different features. Hu [159] defines seven values, computed from central moments through order three, that are invariant to object scale, position, and orientation but those invariant moments are difficult to correlate with physical meaning.

Properties of Low-Order Moments:

To illustrate the properties of moments and show the applicability to object representation, we can consider the moment values of a distribution function as binary and contiguous, i.e. a silhouette image of a segmented object.

Zero-th order moment (area): The definition of the zero-th order moment, m_{00} , of the distribution, $f(x,y)$

$$m_{00} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy \quad (5.7)$$

represents the total mass of the given distribution function or image. When computed for binary image of segmented object (pixels of object are given in 1), the zero-th moment represents the total object area.

Second Order Moments: The second order moments, $(\mu_{20}, \mu_{11}, \mu_{02})$, known as the moments of inertia, may be used to determine several useful object features. Some are described below:

Image Ellipse: The first and second order moments also define an inertially equivalent approximation of the original image, referred to as the image ellipse [150]. The image

ellipse is a constant intensity elliptical disk with the same mass and second order moments as the original image. If the image ellipse is defined with semi-major axis, α , along the x axis and semi- minor axis, β , along the y axis, then α and β may be determined from the second order moments using following equations [160]

$$\alpha = \left(\frac{2 \left[\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right]}{\mu_{00}} \right)^{1/2} \quad (5.8)$$

$$\beta = \left(\frac{2 \left[\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2} \right]}{\mu_{00}} \right)^{1/2} \quad (5.9)$$

Orientation: The 2nd order moments can be used to calculate the orientation of the shape. In terms of moments, the orientation of the principal axes, ϕ is given by [159] (which is same as equation (4.6))

$$\phi = \frac{1}{2} \tan^{-1} \left[\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right] \quad (5.10)$$

In equation (5.10), ' ϕ ' is the angle of the nearest principal axis (either major or minor) of the shape to the positive X-axis and is in the range $-\pi/4 \leq \phi \leq \pi/4$. To compute the orientation of the shape specifically, we determine the angle of the major principle axis to the positive X-axis. The angle of major principal axis, denoted by, θ can be determined from the values of μ_{11} , $(\mu_{20} - \mu_{02})$ and ϕ from Table 5.1 [160].

Table 5.1: The relationship from μ_{11} , μ_{20} , μ_{02} , ϕ to θ [160].

$\mu_{20} - \mu_{02}$	μ_{11}	ϕ	Relation of ϕ and θ	Ranges of θ
Positive	Positive	$\pi/4 > \phi > 0$	$\theta = \phi$	$\pi/4 > \theta > 0$
Positive	Negative	$0 > \phi > -\pi/4$		$0 > \theta > -\pi/4$
Negative	Zero	0	$\theta = \phi + \pi/2$	$\pi/2$
Negative	Positive	$0 > \phi > -\pi/4$		$\pi/2 > \theta > \pi/4$
Negative	Negative	$\pi/4 > \phi > 0$	$\theta = \phi - \pi/2$	$-\pi/4 > \theta > -\pi/2$
Zero	Positive	0	$\theta = \phi + \pi/4$	$\pi/4$
Zero/Positive	Zero	0	$\theta = \phi$	0
Zero	Negative	0	$\theta = \phi - \pi/4$	$-\pi/4$

Higher order moments: The two third order central moments, (μ_{30}, μ_{03}) , describe the skewness of the image projections and two of the fourth order central moments, (μ_{40}, μ_{04}) , describe the kurtosis of the image projections. Other high order moments does not have physical meanings.

Orthogonal moments:

Replacing the monomials in equation (5.3) with an orthogonal basis set results in an orthogonal moment set. The orthogonal condition simplifies the reconstruction of the original function from the generated moments [160]. Two such well established orthogonal moments are Legendre and Zernike moments.

In general, in terms of sensitivity to additive random noise, high order moments are the most sensitive to noise [160]. It was concluded that Legendre moments are most severely affected by noise. To apply Zernike moments, the image coordinate space must be

transformed to the domain where the orthogonal polynomial is defined (unit circle for the Zernike polynomial). On the other hand, centralized moments are computationally simple and inherently translation invariant. Low order moments are not very sensitive to noise. Also, lower order moments can be used to describe some physical properties of a shape. As a result centralized moments are employed to extract features in the proposed system.

In general, descriptors should be compact with invariance to different types of transformations and should have perceptual meaning. Good descriptors should have the ability to reconstruct the shape that is described by the features values. Moreover, the reconstructed shape should be an approximation such that similar shapes can give same approximation and hence similar values of the features or descriptors. In the proposed system, after region partitioning, most empty regions are simple in shape. The system proposed to model each empty region by a triangle or quadrangle as shown in Fig. 5.8 where an empty region is modeled by a triangle if it faces only one side of the bounding box. A quadrangle is used if it faces two sides otherwise.

At this moment, we performed a vision based experiment where 2 sequences of each action type are randomly selected from Weizmann human action dataset (see Fig. 7.1). Then each pose of a sequence is represented by the bounding box where the negative space regions inside the bounding box are represented by simple shapes (either triangle or quadrangle as shown in Fig. 5.8(b)). Afterwards, 10 persons are requested individually to watch the approximated image sequences (bounding box images like Fig. 5.8(b)) and recognize the action. Before watching the approximated image sequence each person

watched the original image sequences to familiar with the actions. Note that the approximated image sequence only shows the shape information of each sequence. It does not show the motion information of the actor (i.e. speed of the person). Among the 10 persons, 2 persons made confusions in two sequences (one place jumping and other is jumping). From this experiment, it is evident that human can distinguish different types of actions when the poses are represented by negative space regions which are modeled by simple shapes.

Fig. 5.8: (a) Empty regions can be viewed as triangle or quadrangle, (b) pose representation by simple shapes.

The extracted shape features are expected to facilitate reconstruction of either triangle or quadrangle. To reconstruct a triangle, the base and the top point of a triangle is needed. The base is known if the length and position of the side on the bounding box are known. The top point can be estimated approximately by area and the principle orientation of the triangle. Similarly, a quadrangle can be reconstructed if the lengths of the two sides on the boundary, the area and orientation are known. Furthermore, two additional features are employed to enhance the description. All these features are described in detail in subsequent sub-sections.

5.2.2.4 Area

As stated earlier, for binary image m_{00} denotes the area of the regions. We normalize each empty region area by the total empty area (discarding the silhouette area) in the bounding box. As a result, area of the region ranges from 0 to 1.

5.2.2.5 Principle axis lengths

The simplest eccentricity is the ratio of the major and minor axes of an object [157]. This ratio can be computed as β / α where α and β can be computed from equation (5.8) and (5.9) respectively. Fig. 5.9 shows one example of the major and minor axes for an empty region of a single pose of walk sequence. Value of this feature is in the interval of 0 to 1.

Fig. 5.9: Major and minor axis for a single negative space region.

5.2.2.6 Orientation

Orientation computation of shape is shown in section 5.2.2.3 . Fig. 5.10 shows the orientation for a single region of a pose. The range of θ is $-\pi/2$ to $\pi/2$. To make θ 's range from 0 to 1, $\pi/2$ is added with θ and normalized by π .

Fig. 5.10: Orientation of a region (Solid line represents the major principle axis, dashed line is the X-axis)

5.2.2.7 Rectangularity

Rectangularity is a measure of how close a shape to a rectangle. A simple rectangularity measure is the ratio of region area over the bounding rectangle area of that region, as shown in equation (5.11). This value is 1 for a rectangle and for other shapes it will be in the range 0 to 1. One example of rectangularity for a region is shown in Fig. 5.11.

$$rectangularity = reg_area / bounding_rec_area; \quad (5.11)$$

Fig. 5.11: Rectangularity. Region is shown in red boundary and bounding rectangle is shown in green.

5.2.2.8 Horizontal and vertical side length

For each region there must be at least one side of the bounding box and at most two sides are included in the region. The length and position of each side are used as two features of the region description. The positional feature has been described in detail in section 5.2.1. Here, we record the length of each side as the desirable feature. The lengths along the horizontal and vertical sides are recorded as *hor_len* and *ver_len*. *hor_len* and *ver_len* have been normalized by the width (*bb_width*) and height (*bb_height*) of the bonding box respectively to give scale invariant measures. These two lengths are calculated as follows

$$hor_len = (sign) \frac{reg_hor_len}{bb_width} \quad (5.12)$$

$$ver_len = (sign) \frac{reg_ver_len}{bb_height} \quad (5.13)$$

where *sign* is negative for bottom or left sides of the bounding box, and positive otherwise. For regions with corner point, both *hor_len* and *ver_len* will be present, but in case of

regions without a corner, one of the lengths will be zero. Both of the values range from 0 to 1. One example of *ver_len* and *hor_len* is shown in Fig. 5.12.

Fig. 5.12: *ver_len* and *hor_len* of a region.

Up to this point, the extraction of shape based features and the positional features are described. Now, the two types of features have to be combined to generate a feature vector. To do this, the feature vector of each pose is divided into 14 blocks representing the 14 anchoring points. Each of these blocks is further divided into 6 cells to represent the shape based features. For a negative space region, all the 6 feature values are assigned to the block corresponding to the anchor point of that region. For instance, in Fig. 5.13(a) region ‘A’'s anchor point is 1, hence the 6 feature values of region ‘A’ is assigned in block 1 (Fig. 5.13(b)). The dimensionality of the feature space of the system is $6 \times 14 = 84$. Feature vector of each pose look like as follows

$$f_{1,1}f_{1,2}f_{1,3}f_{1,4}f_{1,5}f_{1,6} \quad f_{2,1}f_{2,2}f_{2,3}f_{2,4}f_{2,5}f_{2,6} \quad f_{3,1}f_{3,2} \dots \dots \dots f_{14,1}f_{14,2}f_{14,3}f_{14,4}f_{14,5}f_{14,6}$$

where each $f_{i,j}$ represents a feature value, $i=1$ to 14 represents 14 anchoring points and $j=1$ to 6 represents orientation, eccentricity, area, rectangularity, horizontal side length and vertical side length respectively.

Feature vector for one pose can be represented as a gray scale image as shown in Fig. 5.13(b). A fully black block means that all the 6 features values of that block (i.e. associated anchor point) are zero. Hence, a fully black block in Fig. 5.13(b) corresponds to an anchoring point not assigned to any negative space region.

Fig. 5.13: Combining positional and shape based features. (a) a pose is shown with anchoring points. (b) feature vector with block numbering

5.3 Summary

The region partitioning process and pose description are described in this chapter. First, the necessity of region partitioning process is described. Then the proposed region partitioning process is explained. For partitioning a negative space region a peninsula has to be identified which is done by line scanning process. Next, the testing of validity of

identified peninsula for partitioning the region is described. After partitioning the regions, some fine tuning is performed by removing the small regions which have very low influence on recognition process. Also the process for selecting two threshold values that are used in the region partitioning and region removal process are described.

In section 5.2 two types of features are extracted. For positional feature 14 location points are assigned along the bounding box. Then each empty region is allocated to one of these points. Next, some features are extracted to describe the shape of the region. Six region-based features are extracted. They are *orientation*, *eccentricity*, *area*, *rectangularity*, *hor_len* and *ver_len*. Most of these features are calculated by means of statistical moments. At the end of the chapter, combining process of the two types of features is narrated.

Chapter 6

SEQUENCE MATCHING

This chapter describes the matching process of model sequence with input sequence. First, the input poses are compared with the stored feature vectors which are generated from the training poses. For pose matching, one can run into the situation that the anchor point of a region get shifted to a neighboring point for two similar poses. Hence, a technique has to be developed to match poses effectively even in the presence of anchor point shifting. This is described in section 6.1. Next, input image sequence is compared with training sequences by employing Dynamic Time Warping (DTW) algorithm which is described in the subsequent sections. This system can be extended as a search engine to search complex activities in a single image sequence by modifying the sequence matching technique which is illustrated in the last part of this chapter.

6.1 Distance between poses

For pose description, the association of an anchoring point can be shifted by one position for two similar poses, as shown in Fig. 6.1(a) region ‘A’ and Fig. 6.1(b) region ‘B’. A technique is proposed for matching which can adapt with point shifting and can calculate distance between feature vectors efficiently. In Fig. 6.1, the matching of the two feature vectors from Fig. 6.1(a) to (b) should match block 5 in (a) to block 4 in (b) as shown in Fig. 6.1(c).

Fig. 6.1: Pose matching (a) Pose 1 (b) Pose 2, (c) The feature vectors matching from (a) to (b) with one cross matching of regions.

Based on this observation, a matching matrix PM can be constructed, where one feature vector $v1$, is placed along the columns and another feature vector $v2$ is along the rows, with 14×14 elements computed as

$$PM(i, j) = \begin{cases} r_{dist}(v1(i), v2(j)) & \text{if } |i - j| \leq 1 \text{ or } |i - j| = 13 \\ \infty & \text{otherwise} \end{cases} \quad (6.1)$$

$$r_{dist}(v1(i), v2(j)) = \sqrt{\left(\min(|1 - |v1(i)_1 - v2(j)_1||, |v1(i)_1 - v2(j)_1|) / 0.5 \right)^2 + \sum_{k=2}^6 (v1(i)_k - v2(j)_k)^2} \quad (6.2)$$

where $i, j=1$ to 14 , $v1$ and $v2$ are two pose vectors, k being the index variable of different features for each region except orientation. $v1(i)_1$ or $v2(j)_1$ are orientations of $v1$ and $v2$ respectively and $\min(a,b)$ returns the minimum between a and b . Since, the anchor point can shift only for one position, hence for one anchor point we compute r_{dist} value only for the neighboring and same anchor points. Note that for anchor point 1 (or 14) one of the neighboring anchor points will be 14 (or 1). Hence, in equation (6.1) we calculated r_{dist} value in the PM matrix where the difference between anchor points is 0, 1 or 13. It can be seen from equation (6.2) that r_{dist} is the Euclidean distance between different features with maximum orientation difference is 0.5 (i.e. $\pi/2$). To keep the distance calculation simple, we employed Euclidean distance without giving weights to different features. At this moment, we have not investigated the effect of various distance metrics on the recognition result. Hence, as a distance metric in equation (6.2) we employed Euclidean distance which is one of the simple distance metrics. Uses of other distance metrics are left as a future work of this current work.

Fig. 6.2: Matching matrix PM, for the two feature vectors from Fig. 6.1 (Inf = ∞)

The matching can be done in two steps. In the first step we match the two poses by assuming that no anchor point is shifted, which means each anchor point block of one pose will be compared with corresponding anchor point's block of the other pose. Thus the matching distance is just the sum of the diagonal elements of PM, as indicated by the dotted line in Fig. 6.2. In the second step, we check for anchor point shifting. The intuitive idea here is, if an anchor point of one pose is shifted, the corresponding anchor point of the other pose would be empty and the distance between two non-empty anchor points is lower than the distance between one empty and one non-empty block (here, by non-empty block we mean that one of the negative space regions of the pose is assigned to that anchor point block and the block contains some non-zero feature values). For example, in Fig. 6.2, in feature vector $v1$ anchor point 5 is shifted to anchor point 4 in the feature vector $v2$ and distance between the blocks of anchor point 5 of $v1$ and 4 of $v2$ (i.e. $PM(5,4)$) is lower than the distance between the blocks of corresponding anchor points (i.e. $PM(5,5)$ and $PM(4,4)$) which is shown in Fig. 6.2. Hence, we minimize the distance by permitting two neighbouring pairs of matrix elements to shift horizontally as shown in Fig. 6.2 (see the shaded cell region). This shifting process will lower the diagonal sum. The greedy shifting process continues until no further minimization of the diagonal sum. In addition, each shifted element is not allowed to get shifted again. Eventually, this process will detect the anchor point shifting and will calculate the optimal distance between poses.

6.2 Distance between Sequences

The temporal duration of an action performed by a subject at different scenarios can be different. Suppose in two video sequences “walking” action is performed by same

person but in different speeds. In those two videos, the poses would be same but the temporal positions of the poses will be different. Hence time warping should be applied to determine the distance between two sequences. By the warping process, we are trying to compute the optimal distance between two sequences even in the presence of temporal variations of poses.

According to [161], Dynamic Time Warping (DTW) shows known best performance for time series problems in various fields which includes bioinformatics [162], medicine [163] engineering [164] etc. Some of the previous systems on human action recognition also achieved good accuracy by employing DTW (e.g. [34]). DTW not only find the similarity of two sequences regardless of non-linear variation in the time dimension, it also maintains the sequential order of poses between two sequences. Therefore in this system DTW is employed to compute the distance between two sequences.

Let M (model) and T (test) be two sequences of poses of lengths n and m , respectively. The goal of DTW is to find a mapping path (or alignment) $[(p_1, q_1), (p_2, q_2), \dots, (p_k, q_k)]$ where the path length is k , according to a recurrence relation (see equation (6.3)

below), such that the distance $\sum_{b=1}^k d(M_{p_b}, T_{q_b})$ on this mapping path is minimized [165].

In the proposed system, the following recurrence relation is employed (also see Fig. 6.3).

$$D(i, j) = d(i, j) + \min \begin{cases} D(i-1, j) + w_v(i, j) \\ D(i-1, j-1) \\ D(i, j-1) + w_h(i, j) \end{cases} \quad (6.3)$$

Initialization:

$$D(1, t) = d(1, t), \quad D(i, 1) = \infty$$

where D is the DTW matrix, $i=2$ to n , $j=2$ to m , $t=1$ to m , $d(i, j)$ being the distance between poses i and j (or the cost function) calculated as in previous section, m and n are number of poses in test and model sequences respectively, and $w_h(i, j)$ and $w_v(i, j)$ are slope weights which are described below.

In the proposed system, test sequence (T) is placed along the horizontal side and model sequence (M) is placed along the vertical side of D as shown in Fig. 6.3. In our system, model sequences contain only poses of one cycle of one action type. However, the test sequence can contain several cycles of either one action type or several action types, as stated in the next two sub-sections. For action sequences of training data, each sequence is divided into sub-sequences such that each sub-sequence contain only one cycle of an action type. These sub-sequences are then used as model sequences (M) in our system.

Fig. 6.3: An example of a DTW matrix D . Dashed gray line is the warping path with starting and end indices as $(1, i_1)$ and (n, i_2) respectively

DTW finds an optimal match between two given sequences with user defined constraints. In the proposed system, the following constraints are used.

Boundary constraint: For alignment of test and model sequences, warping path starts from $D(1,1)$ and ends at $D(n,m)$. In general, the input and model sequences might not start from the same pose (i.e. sequences are not aligned). Hence we match the whole model sequence with arbitrary starting point from the test sequence. This is done by the initialization part of equation (6.3) for the warping path to start from $D(1,i_1)$ with $i_1 = 1$ to m . At the end the path must ends at $D(n,i_2)$ with $i_2 = 1$ to m and $i_1 \leq i_2$. One example of a warping path is shown in Fig. 6.3.

Slope constraints: Ideal mapping path should traverse diagonally in the DTW matrix. Since same action can be performed in different speeds by different persons, some vertical and horizontal movements are allowed in our system. Empirically it is found that speed variation could cause a pose in one sequence to occur at most 2 frames before or after in another sequence, provided that both sequences frame rates are same. Hence, we add a slope penalty of $cons_m_v$ ($cons_m_h$) if the mapping path makes more than l_v (l_h) consecutive moves in strictly vertical direction with $l_v = (2 \times \text{test_frame_rate} / \text{train_frame_rate})$ [or horizontal direction with $l_h = (2 \times \text{train_frame_rate} / \text{test_frame_rate})$] where $cons_m_{v(h)}$ denotes the number of consecutive moves in vertical (horizontal) direction. The penalties are captured by the $w_h(i,j)$ and $w_v(i,j)$ in equation (6.3) as

$$w_h(i, j) = \begin{cases} cons_m_h(i, j) & \text{if } cons_m_h(i, j) > l_h \\ 0 & \text{otherwise} \end{cases} \quad (6.4)$$

$$w_v(i, j) = \begin{cases} cons_m_v(i, j) & \text{if } cons_m_v(i, j) > l_v \\ 0 & \text{otherwise} \end{cases} \quad (6.5)$$

$$cons_m_h(i, j) = \begin{cases} cons_m_h(i, j-1) + 1 & \text{if } D(i, j) = d(i, j) + D(i, j-1) \\ 0 & \text{otherwise} \end{cases} \quad (6.6)$$

$$cons_m_v(i, j) = \begin{cases} cons_m_v(i-1, j) + 1 & \text{if } D(i, j) = d(i, j) + D(i-1, j) \\ 0 & \text{otherwise} \end{cases} \quad (6.7)$$

After calculating the warping distance between the test and model sequences, the distance is normalized by the warping path length which is the number of elements of D traversed by the path.

6.2.1 Recognizing Action Type

Fig. 6.4: Matching process of model sequence with test sequence containing single action type.

The proposed system can work as either action recognizer, which attempts to recognize a human action (e.g. walking) in the input sequence, or as a search engine which can search for several human actions in a certain order (e.g. walking followed by running) in a single input sequence. The two approaches differ in the matching process. This section describes the matching process of action recognition. Next section describes the matching process of the system when it works as a search engine. During action recognition, our assumption is that the input video sequences contain numerous cycles of one type of action which is true for the available human action datasets (e.g. Weizmann and KTH dataset). In case of video sequences which contain multiple cycles of one action type, doubly matching scheme is performed as shown in Fig. 6.4 where two matching of the model with test sequence are done consecutively. The system can still recognize the action type of input video with single matching which is shown in experimental result chapter. Doubly matching is necessary because certain action types, e.g. bending forward ('bend'), can contain two parts with one part having perfect match with another short action type, e.g. place jumping ('pjump'). Hence, if we perform single matching, 'bend' could be recognized as 'pjump'. To avoid this situation, doubly matching is needed.

The first match is determined by DTW matching from model to test sequence which can occur in any position of test sequence. Let the number of input frames matched be k . The second match is determined by performing DTW matching on the subsequent $1.1 \times k$ frames right after the first match if the later part (frame p_{j+k+1} to p_n in Fig. 6.4) is longer than the former part (frame p_1 to p_{j-1} in Fig. 6.4). If former part is longer than the later part, the second match is done on the preceding $1.1 \times k$ frames. If both parts (former

part and later part) is smaller than the $1.1 \times k$ frames, no 2nd matching is done. Theoretically, if the input and model sequences are from same action type, the neighboring (either left or right) k frames of 1st match should be a perfect match with the model sequence but a person can vary his/her speed within a sequence, hence $1.1 \times k$ frames are employed instead of k frames for the second match. The average of the two matching scores is then taken as the distance between the model and test sequence. The computation of doubly matching score can be shown as equation (6.8).

$$d_s = \left\langle dtw(X, Y) + \begin{cases} \text{no second match} & \text{if } (n - j - k) < 1.1k \text{ \& } j < 1.1k \\ dtw(X, Y(j - 1.1k : j - 1)) & \text{elseif } j > (n - j - k) \\ dtw(X, Y(j + k + 1 : j + 2.1k)) & \text{Otherwise} \end{cases} \right\rangle \quad (6.8)$$

where X and Y are model and input sequence respectively; $Y(i:l)$ denotes frame i to l of Y sequence; $\langle . \rangle$ denotes the average; $dtw(x, y)$ is the normalized DTW score calculated as in section 6.2 ; n, j and k represent the same variables as denoted in Fig. 6.4.

Fig. 6.5: Example of doubly matching scheme. Number represents the frame number within a sequence. Test sequence contains 121 frames and model sequence contains 36 frames. 1st match is from frame 8 to 41 and second match is from frame 51 to 80 of test sequence

One example of doubly matching scheme is shown in Fig. 6.5 where the test sequence and model sequence length is 121 and 36 frames respectively. The first DTW match is found from frame 8 to 41. Some example matched frames are shown in Fig. 6.5. The second DTW matching will be performed between frame 42 to 80 since the later part of test sequence (frame 42 to 121) is longer than former part (frame 1 to 7) and the first match length (frame 8 to 41) is 34 (i.e. $42+(34 \times 1.1) \approx 80$). The second match is found from frame 51 to 80.

For recognition, the DTW matching scores are computed between the test sequence and all the model sequences. Next, the speed score is added to the individual matching score and then the Nearest Neighbor classifier is employed to recognize the action. Given an input sequence T , the speed of the sequence T and the warping distance of T from each of the model sequence R_j is calculated. Let speed of the input sequence T be sp_{input} and it is matched with a model of action type ‘walk’, then the speed score S_C of T is the likelihood score of ‘walk’ membership function for speed sp_{input} . Ideally, if T and R_j are from the same action category, S_C should give high score. Since the likelihood score is a similarity measure, it’s converted to a dissimilarity measure to combine with DTW distance, by subtracting S_C from 1. Hence, the total distance of input sequence T from a model sequence R_j can be calculated as equation (6.9). Afterwards, Nearest Neighbor classifier is applied to recognize the action type of the input sequence.

$$t_dist(T, R_j) = d_s(T, R_j) + (1 - S_c) \quad (6.9)$$

6.2.2 Query for Complex Activity

In real life, human activities can be meaningfully composed, e.g. walking followed by running followed by walking. In this section, method for searching complex human activity (composition of different human activity sequentially) in a single video is described which could be applied in different application e.g. content based video retrieval [165]. The system processing is just a little bit different (shown in Fig. 6.6) from the process shown in Fig. 1.5 during searching for complex human activity. All processes are same as the *action recognition* method except the speed calculation and the matching step. In speed calculation there is an interaction between sequence matching block and speed calculation block for input data whereas in case of *action recognition* speed calculation block does not depends on the sequence matching step. Also the sequence matching is different from the process described in section 6.2.1.

Fig. 6.6: Block diagram of searching complex activity

In current scenario, query activity can contain multiple action type (e.g. walk, run) and the input video is a single video sequence containing multiple types of activities which are performed sequentially. The goal in this situation is to search for the action types in the input video, in the same order as given in query activity (e.g. walk followed by run) and rank that input video based on the matching distance from the query. Here we are interested to find the action types regardless of their temporal position in the input sequence. Model data and input data are processed separately which share some common processing steps. Among these steps *Negative region extraction*, *region partitioning* and *pose description* processes are same as the processes described in chapters 4 and 5. Computation of speed score is slightly different. Previously, all the frames in the input sequence are used to calculate the speed but in this case, since there are multiple types of actions with different speeds, only the matched frames of the input sequence, which matched with a certain model sequence, are employed for speed calculation purpose. Next the Euclidean distance between the model sequence speed and matched frames speed is employed as speed score.

The sequence matching process for action query is described below. For a query of combination of action types, such as $action_1$ followed by $action_2, \dots$ followed by $action_q$, the matching scheme is shown by equation (6.10) for q consecutive actions as

$$D_l(i, j) = d_l(i, j) + \min \begin{cases} D_l(i-1, j) + w_v(i, j) \\ D_l(i-1, j-1) \\ D_l(i, j-1) + w_h(i, j) \end{cases} \quad (6.10)$$

Initialization:

$$D_l(1, t) = d_l(1, t) + ini_k(t), \quad D_l(i, 1) = \infty, \quad ini_k = l_row_{k-1}$$

$$l_row_k(j) = \min_{l=1}^p (D_l(n, j) + sp(l, j)), \quad l_row_0 = [0, 0, 0, \dots, 0]_{m \times 1}$$

where D_l is the DTW matrix for model l ,

$$i=2 \text{ to } n, j=2 \text{ to } m, k=1 \text{ to } q, t=1 \text{ to } m, l=1 \text{ to } p,$$

n and m are the number of poses in model and input sequences respectively,

q is the number of query action types,

p is the number of model sequences for query $action_k$,

$d_l(i, j)$ is the distance between pose i of model l and input pose j , (as in equation (6.1))

$sp(l, j)$ is the difference between speed score of model l and the speed score of input poses r to j where r is the starting pose of a warping path which ends at pose j of input sequence.

The final matching distances for one action type $action_k$ are recorded in l_row . The scheme for matching one action type is shown in Fig. 6.7. l_row vector is formed by taking the minimum of all the last row of p DTW matrices. This l_row vector is then used as *ini* vector for next action matching. Finally, matching score of whole query with the input sequence is computed as equation (6.11)

$$m_score = \min_{j=1}^m (l_row_q(j)) \quad (6.11)$$

Fig. 6.7: Matching process for one action query $action_k$. D_l is the DTW matrix for model l , $sp(l,j)$ is the speed score difference between model l and input sequence.

6.3 Summary

This chapter described the matching process of input sequence with training data. First the matching of poses is described. A technique is proposed which can calculate distance between two poses effectively in the presence of anchor point shifting which can occur due to the flexible movement of the human body. Next, the DTW matching technique is described which employed the pose distance as the cost function. The proposed system can work either as an *action recognizer* or as a *search engine*. During

working as an *action recognizer* doubly matching scheme is employed which actually match the model sequence with input sequence two times consecutively. In case of *search engine* a query is given, constructed by composing different actions, the system searches the query in all the input sequences and ranks them according to their distance with the query.

Chapter 7

EXPERIMENTAL RESULTS

In this chapter, our experimental results are presented. Setting for each experiment is described and the results are compared with appropriate previous systems. First the recognition power of the proposed system is tested by two publicly available datasets i.e. Weizmann [26] and KTH [27] dataset. Next, the robustness of the system in case of partial occlusion, deformed actions and viewing angle change is evaluated. At last, as a human action search engine, the searching effectiveness of the system is assessed by another publicly available dataset, the UIUC complex human activity dataset [7]. Apart from these experiments, other investigations such as feature extraction time, recognition using less information are performed. All the parameters in the system are set based on the training data of the Weizmann dataset and applied on all the datasets except the generation of the fuzzy membership functions which are generated automatically from the training data of each dataset.

7.1 Action Classification

Weizmann dataset: In this dataset there are 9 persons performing 10 actions (90 sequences in total). The actions are “bending-in-front” (or “bend”), “jumping-jack” (or “jack”), “jump-forward-on-two-legs” (or “jump”), “jump-in-place-on-two-legs” (or “pjump”), “run”, “gallop-sideways” (or “side”), “Jump-forward-on-one-leg” (or “skip”),

“walk”, “one-hand-wave” (or “wave1”), and “two-hands-wave” (or “wave2”). Example frames are shown in Fig. 7.1 (a) to (j) row one of each action sequence. This dataset provides a good test bed to investigate the performance of the algorithm in presence of relatively large number of action types. Background segmented images are provided by the author of the dataset and those images are employed directly in our system. Since the number of video sequences is low, each training sequence is divided into sub-sequences. Each sub-sequence contains only one cycle of an action and we use these sub-sequences as the model sequences. Leave-one-out (LOO) testing scheme is employed for this dataset as most of the other systems used this scheme.

KTH dataset: This dataset is more challenging than Weizmann dataset due to considerable amount of camera movement, long shadow, different clothing and scale and viewpoint changes. There are 25 persons performing 6 actions (box, hand-clap, hand-wave, jog, run and walk) in 4 different scenarios (s1-outdoors, s2-outdoor with scale variations, s3-outdoors with clothing variations and s4-indoors). Each video is sub-divided into 4 sequences, each containing several cycles of one action type and there are 2391 sequences in the dataset. Example frames are shown in Fig. 7.1 (k) to (t). To extract the silhouette, the algorithm by Li et al. [142] is employed without discarding the shadows. Some methods first cut a bounding box either manually [166] or empirically [34] to reduce noise and then perform background segmentation on the bounding box image but in the proposed system we directly feed the input sequence to the segmentation algorithm. This makes our system less dependent on segmentation process compared with those methods. Since the number of video sequences is high enough, we did not take all the cycle in a

training video sequence as our model sequence. Instead, only one cycle of an action from each video is taken to avoid unnecessary calculation. Other systems treat this dataset either as a single dataset (all scenarios in one) or as four different datasets (individual scenario is treated as a separate dataset, trained and tested separately). Our system is evaluated on both these settings. For testing, some system used Leave-one-person-out (LOO) scheme. Others used split based methods where the video sequences are split into training and testing data according to the author of this dataset [27]. We report results for both testing schemes.

For both datasets, the segmented silhouettes contains “leaks”, “holes” and “protrusion” due to imperfect subtraction, shadows, and color similarities with the background but our system can cope with all these complexities well. Example frames of these segmented images, taken from the correctly recognized image sequences are shown in Fig. 7.2.

To calculate the speed score, Fuzzy membership functions are generated automatically according to the process described in chapter 4. For each dataset Fuzzy membership functions are generated separately as the type and number of actions in two datasets is not same. The optimized Fuzzy membership functions for Weizmann and KTH dataset are shown Fig. 7.3.

(a) Weizmann dataset, *bending-in-front*

(b) Weizmann dataset, *jumping-jack*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(c) Weizmann dataset, *jump-forward-on-two-legs*

(d) Weizmann dataset, *jump-in-place-on-two-legs*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(e) Weizmann dataset, *run*

(f) Weizmann dataset, *gallop-sideways*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(g) Weizmann dataset, *jump-forward-on-one-legs (or skip)*

(h) Weizmann dataset, *walk*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(i) Weizmann dataset, *one-hand-wave (or wave1)*

(j) Weizmann dataset, *two-hand-wave (or wave2)*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(k) KTH dataset, *s1-box*

(l) KTH dataset, example of *s1-handclap*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(m) KTH dataset, example of s1-*handwave*

(n) KTH dataset, example of s1-*jogging*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(o) KTH dataset, example of s1- *running*

(p) KTH dataset, example of s1- *walking*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(q) KTH dataset, example of s2: non-moving action (*handclapping*)

(r) KTH dataset, example of s2: moving action (*walking*)

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(s) KTH dataset, example of s3 (*handclapping*)

(t) KTH dataset, example of s4 (*handclapping*)

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(u) UIUC dataset, *crouch-run*

(v) UIUC dataset, *run-jump-run*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(w) UIUC dataset, *run-jump-reach*

(x) UIUC dataset, *run-pickup-run*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(y) UIUC dataset, *walk-jump-carry*

(z) UIUC dataset, *walk-jump-walk*

Fig. 7.1: Example frames of different types of actions. For each group, 1st row: input, 2nd row: background segmented image, 3rd row: output of shadow discarding process, 4th row: bounding box image and 5th row: after region partition.

(a)

(b)

Fig. 7.2: Example of segmented image frames with leaks, holes and protrusions from correctly recognized sequences (a) Weizmann dataset and (b) KTH dataset

Example frames of different actions from each dataset are shown in Fig. 7.1. For each group of image, the first row shows the input image, second row the background

segmented image. In case of Weizmann dataset almost all the silhouette based methods used same pre-segmented data which are supplied by the author of this dataset, but in case of KTH and UIUC dataset different system used different segmentation algorithm to get the segmented images, since these two datasets do not provide pre-segmented sequences. For KTH and UIUC datasets, we employed Li et al. [142] algorithm to obtain segmented images. The third row of each action sequence in Fig. 7.1 shows the output images after the *shadow removal* process where significant amount of foreground pixels are removed if the segmented image contain shadow (Fig. 7.1(i)). Otherwise very few foreground pixels are removed. The forth row shows the bounding box images which are obtained by cutting bounding box from the images of third row. The last row shows output of the region partitioning process from which shape based features are extracted.

Fig. 7.3: Optimized Fuzzy membership functions. (a) for Weizmann dataset (b) for KTH dataset, LOO scheme and (c) for KTH dataset, spilt based scheme

Table 7.1: Accuracy of different methods for Weizmann dataset

Method	Without skip	With skip
Our Method	100%	100%
Fathi and Mori [86] (2008)	100%	100%
Ikizler and Duygulu [34]	100%	***
Wu et al. [128] (2010)	***	98.9%
Li et al. [125] (2008)	97.8%	***
Bregonzio et al. [61] (2009)	***	96.66%
Junejo et al. [167] (2011)	***	95.3%

Accuracy of different systems for Weizmann dataset is shown in Table 7.1. Note that in case of Weizmann dataset almost all the silhouette based methods used same pre-segmented sequences which are supplied by the author of this dataset. Some authors evaluated their systems with a subset of this dataset (discarding the ‘skip’ action type) as shown in Table 7.1. Our system achieved 100% accuracy on this dataset. Fathi and Mori [86] method also achieved perfect accuracy on full dataset but their system requires some parameters to be manually initialized.

Comparison of our method with other methods for KTH dataset is shown in Table 7.2. Our accuracies for this dataset are 95.49% for LOO and 92.13% for split testing scheme which indicates that the amount of training data does not affect too much in our system. In case of KTH dataset Our system performance is better than others for ‘all scenes in one’. In case of ‘average of all scenes’ our system accuracy outperforms most of the other systems. Lin et al. [168] method achieved the highest accuracy in this case which is comparable to ours. Lin et al. method is a prototype (key frame) based method which described poses by combination of shape (of the silhouette) and motion (of different limbs) descriptors. They need a 256-dimensional shape descriptor whereas our pose descriptor uses only 84 dimensions. Moreover, their system did not employ global motion (speed) of

the body to distinguish actions which could significantly improve the accuracy. Hence, by combining Lin et al.'s method (positive space) with our method (negative space) including speed feature, a new improved system could be obtained in terms of accuracy and computational cost as well.

Table 7.2: Accuracy of different methods for KTH dataset

Method	LOO/ Split	Recognition rate (%)	
		Average of all scenes	All scenes in one
Our method	LOO	94.00	95.49
Our method	Split	91.67	92.13
Lin et al. [168] (2009)	LOO	95.77	93.43
Seo and Milanfar [88] (2011)	Split	95.1	N/A
Kellokumpo et al. [166] (2010)	LOO	N/A	95.4
Schindler and Gool [68] (2008)	Split	90.72	92.7
Yao et al. [87] (2010)	Split	N/A	90.0
Fathi and Mori [86] (2008)	Split	N/A	90.5
Ikizler and Duygulu [34] (2009)	Split	N/A	89.4
Liu and Shah [70] (2008)	LOO	94.15	N/A
Ahmad and Lee [169] (2008)	Split	87.63	88.83
Niebles et al. [60] (2008)	LOO	N/A	83.33

(a)

(b)

Fig. 7.4: Confusion matrix of our system for the KTH dataset. (a) leave-one-out scheme (b) split based scheme. For both cases all scenes are taken as a single dataset.

The Confusion matrices for both testing scheme of KTH dataset are shown in Fig. 7.4 where all sequences of all the scenes are taken as a single dataset. Most of the misclassifications occur due to the noisy segmentation of background (Fig. 7.5) which indicates that if the segmentation process was controlled like other methods (e.g. [166]), our accuracy could be further improved. The confusion matrix for Weizmann dataset is not shown here since the accuracy for this dataset is 100%.

Fig. 7.5: Example frames of noisy background segmentation.

Table 7.3: Accuracy of different methods for different scenes of KTH dataset

<i>Methods</i>	<i>Recognition rate (%)</i>			
	Outdoor (S1)	Outdoor + scale change (S2)	Outdoor + clothing variations (S3)	Indoor (S4)
Our method	98.67	94	93.31	90
Lin et al. [168]	98.83	94	94.78	95.48
Schindler and Gool [68]	93	81.1	92.1	96.7
Ahmad and Lee [169]	90.17	84.83	89.83	85.67

Next, the proposed system's accuracy is compared with other methods (in Table 7.3) where the sequences of each scene are treated as an individual dataset (training and testing are done on individual scene). Our system accuracy is comparable with the state of the arts methods in this setting as shown in Table 7.3

Recognition rates of proposed system for different actions are shown in Table 7.4 and Table 7.5 where the sequences of each scene are treated as an individual dataset. Result of LOO scheme is shown in Table 7.4 whereas Table 7.5 shows results for split scheme. From these two tables it can be seen that the proposed system gives lower recognition rate in both the scheme for *indoor* scene since most of the bad segmentation results occur in this scene.

Table 7.4: Recognition accuracy for different scenes treated as different datasets (LOO)

<i>Action types</i>	<i>Recognition rate (%)</i>			
	Outdoor (S1)	Outdoor + scale change (S2)	Outdoor + clothing variations (S3)	Indoor (S4)
Box	100	88	84	88
Clap	100	100	96	100
Wave	96	92	84	56
Jog	100	100	100	100
Run	96	84	96	96
walk	100	100	100	100
	98.67	94	93.33	90
<i>Average: 94.00%</i>				

Mean distance differences between nearest correct and incorrect choices are shown in Fig. 7.6. For each action type k , distances of nearest correct and incorrect choices are computed by equation (7.1) and (7.2) respectively.

$$mean_c^k = avg \left(\min_{j=1}^M (t_dist(s_i^k, m_j^k)) \right) \quad \text{where } s_i \neq m_j \quad (7.1)$$

$$mean_ic^k = avg \left(\min_{j=1}^M (t_dist(s_i^k, m_j^p)) \right) \quad \text{where } p \neq k \quad (7.2)$$

where, superscripts (k and p) represent type of actions, T is the number of input sequences of type k , M is the total number of model sequences, s_i and m_j are input and model sequences respectively and $t_dist()$ is calculated as equation (6.9). For both the datasets most of the action types have significant differences between correct and incorrect choice which indicates that the selected features have the ability to distinguish between different types of actions. In case of KTH dataset, the difference between two choices of ‘wave’ action type is low since most of the misclassification occurs in this action type, as shown in confusion matrices (Fig. 7.4).

Table 7.5: Recognition accuracy for different scenes treated as different datasets (Split)

<i>Action types</i>	<i>Recognition rate (%)</i>			
	Outdoor (S1)	Outdoor + scale change (S2)	Outdoor + clothing variations (S3)	Indoor (S4)
Box	100	100	88.89	77.78
Clap	88.89	100	100	100
Wave	77.78	88.89	77.78	66.67
Jog	100	88.89	100	100
Run	88.89	77.78	88.89	88.89
walk	100	100	100	100
	92.59	92.59	92.56	88.89
<i>Average: 91.67%</i>

Fig. 7.6: Mean distance of nearest correct and incorrect choice of each action type. (a) Weizmann dataset, (b) KTH dataset LOO scheme and (c) KTH dataset split scheme

Vuuren et al. [25] method is based on negative space analysis and it would be interesting if we could compare the two methods. Unfortunately, [25] has performed experiment on a different dataset which is not publicly available. Hence, we could not show the experimental comparison with that system.

7.1.1 Recognition Using Less Information

Under the assumption that the input can be partly corrupted, we employed reduced but more robust information from each pose. For each negative space region of input pose, we calculate the distance from the corresponding negative space regions of model pose,

and rank the input pose regions. The ranking is done among the input pose regions, based on the distances where region with lowest distance is the top ranked region. Then some portions of top regions (e.g. 75%) are chosen for action recognition. By this method the recognition rates for the full Weizmann and KTH datasets are shown in Fig. 7.7. These results indicate that even if 50% regions of a pose are corrupted due to partial occlusion, segmentation error etc. our system can still recognize the action effectively.

Fig. 7.7: Recognition rate of our system for less information

7.1.2 Feature Extraction Time

Our descriptors are computationally inexpensive. Feature extraction time of our un-optimized Matlab[®] code is 4.31s for a $(180 \times 144 \times 50)$ segmented sequence on a Pentium 4, 3.4 GHz machine. This is approximately 12 frames per second ($\approx 50/4.31$). The speed can be further improved using C++. Gorelick et al. [26] takes about 30 seconds to extract features for same sequence in similar machine.

7.1.3 Contribution of different features

Fig. 7.8: Accuracy without different types of features for Weizmann dataset

Contributions of different types of features are shown in Fig. 7.8, which is generated based on the Weizmann dataset. According to the graph, shape and positional information provide vital information for our system, whereas speed information is needed for only few action sequences to provide distinguishable information since, without speed feature, the accuracy is 97.78%.

7.2 Robustness Test

In this experiment, robustness of the system is evaluated in cases of high irregularities of action and viewing angle changes. In this regard the *Weizmann robust deformation* dataset and *Weizmann robust view* dataset are employed [26]. The robust deformation dataset contains 10 video sequences (normal walking, walking in a skirt, carrying briefcase, limping man, occluded Legs or no legs, knees Up, walking with a dog, sleepwalking or moon walk, swinging a bag, occluded by a pole) of a walking person in various complex scenarios in front of different non-uniform backgrounds. The Robust view dataset contains 10 videos of walking action, captured from 10 different viewing angles (from 0 to 81 degrees angles relative to the image plan with steps of 9 degrees).

Example frames are shown in Fig. 7.9 and Fig. 7.11. In each case, all the sequences of normal action of Weizmann dataset are employed as model sequence (i.e. number of model action type is 10 which are stated in section 7.1 : *Weizmann dataset*).

Fig. 7.9: Example of image frames used in robustness experiment. Upper row is the actual image and lower row is segmented image for each frame. (a) Normal walk, (b) Swinging bag, (c) Carrying briefcase, (d) Walking with a dog, (e) Knees up (f) Limping man, (g) Moon-walk, (h) Occluded legs (i) Walking in a skirt and (j) Occluded by a pole.

In case of robust deformation dataset all the input sequences are recognized correctly and most of them have significant difference with the nearest incorrect choice which is shown in Fig. 7.10. Table 7.6 shows that the proposed system accuracy is comparable with other methods.

Table 7.6: Different methods accuracy for robust deformation dataset

Method	Accuracy
Our method	100%
Gorelick et al. [26]	100%
Bregonzio et al. [61]	90%
Li et al. [125]	80%

Fig. 7.10: Correct and nearest incorrect choice of each deformed action.

In case of robust view dataset the proposed system accuracy is 70% (3 misclassifications). The proposed system fails to recognize actions with viewing angle equal or larger than 63 degrees. The reason behind is that for larger viewing angle the visibility of the region of interest significantly reduced. Since all the data in the training set is captured in 0 degree angle, for larger viewing angle the intra-class similarity is decreased. Hence, the system failed to recognize actions, captured in a viewing plane which makes a large angle with the image plane. Accuracy could be further improved by incorporating other viewing angle data in the training set or by employing multi-camera system (one approach is stated in future works). The differences between correct and incorrect choices for each sequence are shown in Fig. 7.12. All the misclassified actions are classified to place-jumping. The reason is minimum visibility of the negative space regions, and the displacement of the body centroids which makes walking pose look similar to place-jumping pose.

Fig. 7.11: Example frames from robust view dataset. For each column first row is original image and second row is corresponding segmented image. The view angles are (a) 0 degree, (b) 18 degrees, (c) 36 degrees (d) 54 degrees and (e) 81 degrees.

Fig. 7.12: Correct and incorrect choice of each view changing action.

7.3 Searching for Complex Activity

In this experiment, the proposed system's effectiveness is tested as a human activity searcher. For this experiment the UIUC complex human activity dataset [7] is employed which contains 73 video sequences of complex human activities as listed in Table 7.7. The goal of the system in this scenario is to query for sequences containing

composite activities specified by user (e.g. find sequences which contain ‘walk’ followed by ‘run’ [those listed at rows 11 and 13 of Table 7.7]). Same as Ikizler and Forsyth [7] method, we are also interested to find the presence of action types in the same order given by the user, rather than the precise location of actions in the input video sequence.

Table 7.7: List of videos in UIUC complex human activity dataset

context	No. of videos	context	No. of videos
crouch-run	2	run-backwards-wave	2
jump-jack	2	run-jump-reach	5
run-carry	2	run-pickup-run	5
run-jump-walk	2	walk-jump-carry	2
run-wave-walk	2	walk-jump-walk	2
stand-pickup-walk	5	walk-pickup-walk	2
stand-reach-walk	5	walk-stand-wave-walk	5
stand-wave-crouch	2	crouch-jump-run	3
walk-carry	2	walk-crouch-walk	3
walk-run	3	walk-pickup-carry	3
run-stand-run	3	walk-jump-reach-walk	3
run-backwards	2	walk-stand-run	3
walk-stand-walk	3		

There are 5 actors in the dataset. Ikizler and Forsyth [7] employed the video sequences of actor 1 as training data. Same arrangement is used in the proposed system. To create the model data, each training video sequence is divided into sub-sequences so that each sub-sequence contain only one cycle of one action type. Performance over a set of queries is evaluated by mean average precision (MAP), which is the area under precision recall curve of a query. Mathematically, average precision P_{avg} over a dataset, S is defined as

$$P_{avg} = \frac{\sum_{r=1}^N (p(r) \times rel(r))}{\text{number of relevant sequence in } S} \quad (7.3)$$

where, r is the rank of a video sequence (lowest distance video is rank 1) $rel(r)$ is the binary relevance vector (relevant=1, or 0), $p(r)$ is the precision at rank r and N is the number of retrieved sequence. For example, the average precision ‘*walk-stand-run*’ query (row 14 in Fig. 7.13(a)) is $\left(\frac{1}{1} + \frac{2}{5} + \frac{3}{11}\right)/3 = 0.56$. For multiple queries the mean value of P_{avg} of all queries is the MAP value. Large MAP value implies better searching power of a system.

We did not generate fuzzy membership function for this dataset since there are few model sequences (for some action type there is only one model sequence). In this case, the speed score is the Euclidean distance between model sequence and matched portion of input sequence. Ikizler and Forsyth [7] performed 15 queries on whole dataset where the queries are shown on the left of Fig. 7.13(a). We also perform these queries and reported our result in Fig. 7.13(a). The proposed system MAP value is 0.58 which is higher than Ikizler and Forsyth method (MAP: 0.56). Among 15 queries, 9 queries give higher average precision than [7] (Fig. 7.13(b)) which implies that our system can effectively search for complex activity in a dataset. Ikizler and Forsyth [7] method is a tracker based method which faced difficulties due to self-occlusion whereas in our method self-occlusions is not a problem. Main challenge faced by our system in this dataset is insufficient model data. The P_{avg} values of our system are usually lower as compared to [7] for those queries containing an action type which does not has enough model data. For example, the number of model data of ‘pickup’ is very low (only 3 models), hence most of the queries containing ‘pickup’ action type give lower P_{avg} value (query 7 and 13 in Fig. 7.13).

(a)

(b)

Fig. 7.13: Searching result of 15 queries. (a) Our method. On each row the ranking of relevant sequences are shown (b) Comparison with Ikizler and Forsyth [7] method. Queries are numbered according to their position in (a), e.g. query ‘walk-carry’ is 1 and ‘walk-stand-wave-walk’ is 15.

7.4 Summary

In this chapter several experimental results are reported, denoting the performance of the proposed system in different scenarios. First, the system is evaluated for its action classification capability by two commonly used datasets, Weizmann and KTH action datasets. In this experiment the system performance is found as good as the state of arts

methods. Next, two other experiments were performed on these datasets: *recognizing using less information* and *feature extraction time*. By these two experiments it is implied that the proposed system can effectively classify actions even when 50% of the negative space regions are corrupted. Further, the feature extraction can be done in real time. In the next experiment the robustness of the system was tested in case of high irregularities of actions and viewing angle changes. The proposed system has been found robust to deformed actions which are deformed by variations of clothing, partial occlusion etc. In case of viewing angle change experiment, the system could not recognize actions captured in a viewing plane which makes an angle with the image plane equal or larger than 63 degree. In the last experiment the system's efficiency is tested as a search engine of complex human activity. From the experiment it is evident that the system can efficiently search complex human activity compared with state of the art method.

Chapter 8

CONCLUSION AND FUTURE WORK

This chapter summarizes the research work done with obtained achievements and gives some possible direction of future works.

8.1 Summary

A system is proposed to recognize human action in video sequences where Novel feature description technique is proposed. Unlike other methods this system extract features from the surrounding regions of the human silhouette termed as negative space in art theory, whereas other methods extract features from the silhouette of the person termed as positive space. It has been shown that each human pose can be described by simple shapes implying simple pose description, if one employs negative space to describe poses. Additionally, negative space based pose description is robust to partial occlusion, deformed actions, variations of clothing etc. Moreover, a technique is proposed in the system to detect and discard long shadows from each pose. Further, Fuzzy membership functions, employed during speed calculation, are generated automatically, which facilitate the system to step forward towards automatic human action recognition. Furthermore, the proposed system can work as an activity searcher which is capable of searching complex activities in a single video sequence.

The proposed system is a composition of a hierarchical layout of actions. In the first step the foreground in the input video is segmented from the background. Long shadow which can modify the pose description are detected and discarded from each frame in the next step. A histogram based method is proposed which discard shadows from the segmented binary image. Other shadow detection algorithm works on either color or gray scale image but the proposed method works on binary images. This makes the shadow detection algorithm computationally simpler.

Next, the speed of the detected human body is computed. The speed of a person can be a distinguishing feature for human action recognition. Speed is calculated based on the displacement of the human body centroid in consecutive frames divided by the time required for that displacement. For calculating the time, frame rate of the video is employed which makes the system robust to variations of video capturing frame rate implying that the proposed speed calculation will give same speed for same actions type even if the videos are captured in different frame rate. Moreover, the speed is normalized to viewing angle change by calculating the angle between the image plane and viewing plane. Finally, the speed score is computed employing Fuzzy membership functions which are generated automatically based on the speeds of the training data. At the end of this step negative space regions are extracted by cutting a bounding box containing the human body.

Complex negative space regions are partitioned into simple regions in the next step. Regions are partitioned by identifying all the peninsulas of a silhouette by line scanning

process. Then the validity of a peninsula for partition is checked and the region is partitioned into two regions at the tip of the peninsula based on this checking. Afterwards, all the small regions yielded due to noise, variations of clothing etc. are removed to simplify the pose description. Later poses are described by identifying the position inside the bounding box and extracting shape information for each negative space region. Position of a region is identified by determining the mid point of that region. Six region based features are extracted to describe the shape of a region. At last feature vector is formed by combining both positional and shape based information.

In case of sequence matching two distances are calculated: distance between poses and distance between sequences. Since the position of a sequence can shift, a matrix based distance metric is proposed to effectively compute distance between poses even in the presence of position shifting. Next, sequences are matched by employing basic DTW algorithm. During recognition of action type, model sequences are matched twice with the test sequence consecutively. Then this matching score is combined with speed score and Nearest Neighbor classifier is used to make the final decision. Main purpose of this system is to recognize action type. Additionally, this system is capable of searching complex activities in a video. During activity searching the basic DTW algorithm is modified to facilitate the matching of multiple model action type instead of single action type. By this matching the system rank the input video based on the distance for a particular complex human action query.

It has been found that the proposed system action classification capability is better than most of the previous works and comparable to other state of the arts methods on most commonly used datasets for human action recognition. The experiment also shows that the features can be extracted in real time. In other experiment, the robustness of the system is tested in case of variations of clothing, partial occlusion etc. which shows that the system is robust under these deformed actions. In case of viewing angle change, the system is capable of recognize actions up to certain angle (less than 63 degree) where the training data contains actions performed parallel to the image plane (viewing angle 0 degree) only. One possible solution for this limitation could be multi-camera system. One approach is described in the next section. During searching for complex activity, the proposed system outperforms other state of the art method on a publicly available dataset which indicates that the proposed system can retrieve video sequences from a dataset effectively.

However, our long shadow detection algorithm may fail in case of multiple shadows in one side of the body (e.g. players' shadows in a football match). Also, our system is not fully viewpoint invariant. The system fails to recognize actions with viewing angle equal or larger than 63 degrees because of the significant reduction of visibility of negative space regions. Similar things can occur in case of positive space based methods [122], since the movement of different limbs are less visible in case of large viewing angle. In our system this limitation could be overcome by using multi-camera setup. One possible solution is stated in the 'future work' section.

8.2 Contributions

We have made the following contributions in this investigation

- 1) A novel concept of employing negative space has been proposed to model and recognize human actions. A prototype system has been implemented to show/prove the feasibility of this concept of negative space description.
- 2) The proposed system has been implemented and fine tuned to give recognition rate comparable or better than other systems depending on the types of action to be tackled. It is found the recognition rate of our system is same as other state of the art methods for Weizmann dataset (accuracy 100%) and is comparable for KTH dataset (accuracy 95.50%). Furthermore, our system can recognize more than 80% actions correctly even when half of the feature regions are corrupted. The system is found robust to partial occlusions, clothing variations, deformable actions etc. which is evident from the result shown in Table 7.6.
- 3) To make a working system, we have addressed some of the challenges for action recognition with novel techniques. They are
 - a) A method is proposed to detect and eliminate long shadows from segmented images by employing contextual information of the shadow.
 - b) Fuzzy membership functions are generated automatically from training data which can compute speed score successfully.
 - c) Computationally simple features have been proposed for the system. The system can extract features from 12 frames (approximately) in one second. Potentially, our system can run in real time in concept.

d) The system is further developed to work as a human activity searcher which is able to search different combinations of human activity (e.g. *walk* followed by *run* followed by *jump*) in a single video. Our system performance is higher than the state of the art method. In most of the cases, our system has better accuracy in terms of individual queries (Fig. 7.13).

8.3 Future works

Though promising results are achieved, many extensions of the system are possible. In this section, some of the future directions of work are discussed.

In the current system no background segmentation algorithm is developed. For segmentation purpose, the current proposed system employs algorithm contributed by other. One of the future directions of this work could be the development of a segmentation algorithm to separate the human silhouette from the background and then capturing negative space regions. In this system, main target of using background segmentation is to facilitate the extraction of negative space regions. One can choose not to use background segmentation algorithm in the system for extraction of the negative space regions. In that case, some other method has to be developed to find the bounding box of the human body and the human body itself to extract the negative space regions. A refined method of Ikizler and Duygulu [34] method, which proposed a technique to calculate bounding box automatically, could be used for this purpose.

It is shown in the experimental result that the proposed system faces difficulties to recognize actions which have viewing angle equal or larger than 63 degree. The system failed to recognize actions captured in larger viewing angle because the visibility of the area of interest significantly reduced which actually reduce the inter-class dissimilarity and the intra-class similarity. The system could be made robust to viewing angle change by using multi-camera setup where same action is captured by two cameras simultaneously but only one camera's image sequence is employed for recognizing the action. The cameras should be setup in such a way that the image plane of two cameras will be perpendicular to each other. In this setup if the visibility of negative space regions is reduced in images captured by one camera due to large viewing angle change, the regions visibility will increase in other camera images. Now the decision have to make, for a particular viewing angle which cameras image sequence should be used as input of the current system. The decision could be made by calculating the rate of change of negative regions area for both sequences captured by the two cameras. The image sequence of that camera could be fed as input of the current system which gives larger rate of change of negative regions area.

The pose distances are calculated without giving different weights to different features, perhaps such a weighting could improve the recognition accuracy of the system. Also, Euclidean distance is employed to calculate the distance between two regions since it is one of the simplest distances metric. Using alternative distance metric can be tested against the recognition accuracy. Furthermore, we used a few thresholds during developing the algorithms in the system. These values showed optimal results for the three datasets

used in our system. However, there is a possibility that these values may not give optimal results for other datasets. These values can be fine tuned by using other datasets and more improved performance could be obtained. We leave that subject for future work.

Experimental results shows that the system can extract features in real time (12 frames per second) but the matching time is not real time since DTW matching algorithms are computationally expensive. To make the matching process fast one can use global constraints on DTW matrix to reduce calculation or use faster version of DTW algorithms like FTW [170], FastDTW [171] etc. Another approach could be building up prototype of training sequences by clustering and use the cluster centre as a representative for that prototype. During classification the input sequence could be matched only with the prototypes, not the whole training data which will improve the matching time.

8.4 Conclusion

We have proposed a Negative space based feature extraction technique to recognize human actions. It has been shown that the recognition power of negative space based method is as good as positive space based methods with the advantage of simple pose description. It is also stated that an improved human action recognition system could be achieved by combining negative space based method with positive space based methods. Despite the fact that good result has achieved by the system, there are still some scope for future work by which more robust and reliable system could be attained.

References:

- [1] L. Wang, W. Hu, and T. Tan, "Recent Developments in Human Motion Analysis," *Pattern Recognition*, vol. 36, pp. 585-601, 2003.
- [2] P. Azad, T. Asfour, and R. Dillmann, "Toward an Unified Representation for Imitation of Human Motion on Humanoids," in *International Conference on Robotics and Automation* 2007, pp. 2558-2563.
- [3] T. Beth, I. Boesnach, M. Haimerl, J. Moldenhauer, K. Bös, and V. Wank, "Characteristics in Human Motion – From Acquisition to Analysis," in *International Conference on Humanoid Robots*, 2003, pp. 56-74.
- [4] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys*, vol. 43, pp. 1-43, 2011.
- [5] I. Saleemi, L. Hartung, and M. Shah, "Scene Understanding by Statistical Modeling of Motion Patterns," in *Computer Vision and Pattern Recognition*, 2010, pp. 2069-2076.
- [6] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 539-555, 2009.
- [7] N. Ikizler and D. A. Forsyth, "Searching for Complex Human Activities with No Visual Examples," *International Journal of Computer Vision*, vol. 80, pp. 337-357, 2008.
- [8] D. M. Gavrila, "The Visual Analysis of Human Movement: A Survey," *Computer Vision and Image Understanding*, vol. 73, pp. 82-98, 1999.
- [9] T. Hohmann, N. F. Troje, A. Olmos, and J. r. Munzert, "The influence of motor expertise and motor experience on action and actor recognition," *Journal of Cognitive Psychology*, vol. 23, pp. 403-415, 2011.
- [10] Y. Tagawa, N. Shiba, S. Matsuo, and T. Yamashita, "Analysis of Human Abnormal Walking Using a Multi-body Model: Joint Models for Abnormal Walking and Walking Aids to Reduce Compensatory Action," *Journal of biomechanics*, vol. 33, pp. 1405-1414, 2000.
- [11] K. Nahrstedt, R. Bajcsy, L. Wymore, R. Sheppard, and K. Mezur, "Computational model of human creativity in dance choreography," in *AAAI Spring Symposium*, 2008, pp. 53-60.
- [12] C. Stoll, J. Gall, E. d. Aguiar, S. Thrun, and C. Theobalt, "Video-based reconstruction of animatable human characters," *ACM Trans. Graphics*, vol. 29, pp. 1-10, 2010.

- [13] J. García-Rodríguez and J. M. García-Chamizo, "Surveillance and human-computer interaction applications of self-growing models," *Applied Soft Computing*, vol. *article in press*, 2011.
- [14] 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, pp. 90-126, 2006.
- [15] R. Poppe, "A Survey on Vision-based Human Action Recognition," *Image and Vision Computing*, vol. 28, pp. 976-990, 2010.
- [16] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing Action at a Distance," in *International Conference on Computer Vision*, 2003, pp. 726-733.
- [17] M. J. Black, "Explaining Optical Flow Events with Parameterized Spatio-temporal Models," in *Computer Vision and Pattern Recognition*, 1999, pp. 326-332.
- [18] C. Bregler, "Learning and Recognizing Human Dynamics in Video Sequences," in *Computer Vision and Pattern Recognition*, 1997, pp. 568-574.
- [19] Y. Yacoob and M. J. Black, "Parameterized Modeling and Recognition of Activities," in *International Conference on Computer Vision*, 1998, pp. 1-8.
- [20] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, "Behavior Classification by Eigen Decomposition of Periodic Motions," *Pattern Recognition*, vol. 38, pp. 1033-1043, 2005.
- [21] A. Diaf, R. Ksantini, B. Boufama, and R. Benlamri, "A Novel Human Motion Recognition Method Based on Eigenspace," in *Lecture Notes in Computer Science*, 2010, pp. 167-175.
- [22] R. Polana and R. C. Nelson, "Detection and Recognition of Periodic, Nonrigid Motion," *International Journal of Computer Vision*, vol. 23, pp. 261-282, 1997.
- [23] S. M. Seitz and C. R. Dyer, "View-Invariant Analysis of Cyclic Motion," *International Journal of Computer Vision*, vol. 25, pp. 231-251, 1997.
- [24] M. J. V. Vuuren and G. Jager, "Art and Image Processing," in *Pattern Recognition Association of South Africa*, South Africa, 2001, pp. 23-28.
- [25] M. J. V. Vuuren, "Human Pose and Action Recognition Using Negative Space Analysis," PhD, Department of Electrical Engineering, University of Cape Town, 2004.
- [26] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 2247-2253, 2007.

- [27] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," in *International Conference on Pattern Recognition*, 2004, pp. 32-36.
- [28] D. Weinland, R. Ronfard, and E. Boyer, "A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition," *Computer Vision and Image Understanding*, vol. 115, pp. 224-241, 2011.
- [29] C. R. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time Tracking of the Human Body," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 780-785, 1997.
- [30] M. W. Lee and R. Nevatia, "Human Pose Tracking in Monocular Sequence Using Multilevel Structured Models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 27-38, 2009.
- [31] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [32] X. Sun, M. Chen, and A. Hauptmann, "Action Recognition via Local Descriptors and Holistic Features," in *Computer Vision and Pattern Recognition* 2009, pp. 58-65.
- [33] A. F. Bobick and J. W. Davis, "The Recognition of Human Movement Using Temporal Templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, pp. 257-267, 2001.
- [34] N. Ikizler and P. Duygulu, "Histogram of Oriented Rectangles: A New Pose Descriptor for Human Action Recognition," *Image and Vision Computing*, vol. 27, pp. 1515-1526, 2009.
- [35] Y. Guo, G. Xu, and S. Tsuji, "Understanding Human Motion Patterns," in *International Conference on Image Analysis and Pattern Recognition*, 1994, pp. 325-329.
- [36] M. K. H. Leung and Y. H. Yang, "First Sight: A Human Body Outline Labeling System," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, pp. 359-377, 1995.
- [37] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard People: A Parameterized Model of Articulated Image Motion," in *International Conference on Automatic Face and Gesture Recognition*, 1996, pp. 38-44.
- [38] D. Ramanan and C. Sminchisescu, "Training Deformable Models for Localization," in *Computer Vision and Pattern Recognition*, 2006, pp. 206-213.
- [39] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," *International Journal of Computer Vision*, vol. 61, pp. 55-79, 2005.

- [40] G. Hua, M. H. Yang, and Y. Wu, "Learning to estimate human pose with data driven belief propagation," in *Computer Vision and Pattern Recognition*, 2005, pp. 747-754.
- [41] D. A. Forsyth and M. M. Fleck, "Body plans," in *Computer Vision and Pattern Recognition*, 1997, pp. 678-683.
- [42] S. Ioffe and D. Forsyth, "Finding people by sampling," in *International Conference on Computer Vision*, 1999, pp. 1092-1097.
- [43] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Lecture Notes in Computer Science*, 2004, pp. 69-82.
- [44] A. Micilotta, E. Ong, and R. Bowden, "Detection and Tracking of Humans by Probabilistic Body Part Assembly," in *British Machine Vision Conference*, 2005.
- [45] L. Sigal and M. J. Black, "Measure locally, reason globally: Occlusion-sensitive articulated pose estimation," in *Computer Vision and Pattern Recognition*, 2006, pp. 2041-2048.
- [46] V. Ferrari, M. Marín-Jiménez, and A. Zisserman, "2D Human Pose Estimation in TV Shows," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2009, pp. 128-147.
- [47] K. Rohr, "Towards Model-based Recognition of Human Movements in Image Sequences," *CVGIP: Image Underst.*, vol. 59, pp. 94-115, 1994.
- [48] Z. Chen and H. J. Lee, "Knowledge-guided Visual Perception of 3-D Human Gait from a Single Image Sequence," *Systems, Man and Cybernetics*, vol. 22, pp. 336-342, 1992.
- [49] G. K. M. Cheung, K. Takeo, B. Jean-Yves, and H. Mark, "A Real Time System for Robust 3D Voxel Reconstruction of Human Motions," in *Computer Vision and Pattern Recognition*, 2000, pp. 714-720.
- [50] C. Sminchisescu and B. Triggs, "Estimating Articulated Human Motion with Covariance Scaled Sampling," *International Journal of Robotics Research*, vol. 22, pp. 371-391, June 1, 2003 2003.
- [51] C. Menier, E. Boyer, and B. Raffin, "3D Skeleton-Based Body Pose Recovery," in *International Symposium on 3D Data Processing, Visualization and Transmission*, 2006, pp. 389-396.
- [52] M. Hofmann and D. Gavrilu, "Single-Frame 3D Human Pose Recovery from Multiple Views," in *DAGM Symposium on Pattern Recognition*, 2009, pp. 71-80.

- [53] R. Horaud, M. Niskanen, G. Dewaele, and E. Boyer, "Human Motion Tracking by Registering an Articulated Surface to 3D Points and Normals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 158-163, 2009.
- [54] Y. Shen and H. Foroosh, "View-invariant Action Recognition Using Fundamental Ratios," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-6.
- [55] P. Peursum, S. Venkatesh, and G. West, "Tracking-as-Recognition for Articulated Full-body Human Motion Analysis," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [56] D. Ramanan, D. A. Forsyth, and A. Zisserman, "Tracking People by Learning Their Appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 65-81, 2007.
- [57] A. Agarwal and B. Triggs, "Recovering 3D human pose from monocular images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 44-58, 2006.
- [58] I. Laptev and T. Lindeberg, "Space-time Interest Points," in *International Conference on Computer Vision*, 2003, pp. 432-439.
- [59] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65-72.
- [60] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *International Journal of Computer Vision*, vol. 79, pp. 299-318, 2008.
- [61] M. Bregonzio, G. Shaogang, and X. Tao, "Recognizing Action as Clouds of Space-time Interest Points," in *Computer Vision and Pattern Recognition*, 2009, pp. 1948-1955.
- [62] D. Han, L. Bo, and C. Sminchisescu, "Selection and context for action recognition," in *International Conference on Computer Vision*, 2009, pp. 1933-1940.
- [63] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," in *Computer Vision and Pattern Recognition*, 2009, pp. 2929-2936.
- [64] T. Zhang, S. Liu, C. Xu, and H. Lu, "Boosted multi-class semi-supervised learning for human action recognition," *Pattern Recognition*, vol. 44, pp. 2334-2342, 2011/11// 2011.
- [65] T. Kadir and M. Brady, "Scale Saliency: a novel approach to salient feature and scale selection," *International Conference on Visual Information Engineering*, vol. 2003, pp. 25-28, 2003.

- [66] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *Systems, Man and Cybernetics*, vol. 36, pp. 710-719, 2006.
- [67] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A Biologically Inspired System for Action Recognition," in *International Conference on Computer Vision*, 2007, pp. 1-8.
- [68] K. Schindler and L. V. Gool, "Action Snippets: How Many Frames Does Human Action Recognition Require?," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [69] H. Ning, Y. Hu, and T. S. Huang, "Searching Human Behaviors using Spatial-Temporal words," in *International Conference on Image Processing*, 2007, pp. 337-340.
- [70] J. Liu and M. Shah, "Learning Human Actions via Information Maximization," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [71] S. Savarese, A. DelPozo, J. C. Niebles, and L. Fei-Fei, "Spatial-Temporal correlations for unsupervised action classification," in *IEEE Workshop on Motion and video Computing*, 2008, pp. 1-8.
- [72] Q. Hu, L. Qin, Q. Huang, S. Jiang, and Q. Tian, "Action Recognition Using Spatial-Temporal Context," in *International Conference on Pattern Recognition*, 2010, pp. 1521-1524.
- [73] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *European Machine Vision Conference*, 2004, pp. 17-32.
- [74] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal Localization and Categorization of Human Actions in Unsegmented Image Sequences," *IEEE Trans. Image Processing*, vol. 20, pp. 1126-1140, 2011.
- [75] K. Mikolajczyk and H. Uemura, "Action Recognition with Motion-Appearance Vocabulary Forest," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [76] K. Mikolajczyk and H. Uemura, "Action Recognition with Appearance-motion Features and Fast Search Trees," *Computer Vision and Image Understanding*, vol. 115, pp. 426-438, 2011.
- [77] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, pp. 761-767, 2004.

- [78] V. Ferrari, F. Jurie, and C. Schmid, "Accurate Object Detection with Deformable Shape Models Learnt from Images," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [79] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1615-1630, 2005.
- [80] K. Mikolajczyk and J. Matas, "Improving Descriptors for Fast Tree Matching by Optimal Linear Projection," in *International Conference on Computer Vision*, 2007, pp. 1-8.
- [81] C. Hongping, K. Mikolajczyk, and J. Matas, "Learning Linear Discriminant Projections for Dimensionality Reduction of Image Descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 338-352, 2011.
- [82] L. Shao, R. Gao, Y. Liu, and H. Zhang, "Transform based spatio-temporal descriptors for human action recognition," *Neurocomputing*, vol. 74, pp. 962-973, 2011.
- [83] J. Yuan, Z. Liu, and Y. Wu, "Discriminative Subvolume Search for Efficient Action Detection," in *Computer Vision and Pattern Recognition*, 2009, pp. 2442 - 2449.
- [84] G. Willems, J. H. Becker, and L. V. Gool, "Exemplar-based Action Recognition in Video," in *British Machine Vision Conference*, 2009, pp. 1-11.
- [85] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense Saliency-based Spatiotemporal Feature Points for Action Recognition," in *Computer Vision and Pattern Recognition*, 2009, pp. 1454-1461.
- [86] A. Fathi and G. Mori, "Action Recognition by Learning Mid-level Motion Features," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [87] A. Yao, J. Gall, and L. Van Gool, "A Hough Transform-based Voting Framework for Action Recognition," in *Computer Vision and Pattern Recognition*, 2010, pp. 2061-2068.
- [88] H. Seo and P. Milanfar, "Action Recognition from One Example," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 867 - 882, 2011.
- [89] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [90] A. Agarwal and W. Triggs, "Hyperfeatures - Multilevel Local Coding for Visual Recognition," in *European Machine Vision Conference*, 2006, pp. 30-43.

- [91] L.-J. Li and L. Fei-Fei, "What, where and who? Classifying events by scene and object recognition," in *International Conference on Computer Vision*, 2007, pp. 1-8.
- [92] J. Sivic and A. Zisserman, "Video Google: Efficient Visual Search of Videos," in *Toward Category-Level Object Recognition*, 2006, pp. 127-144.
- [93] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition," in *British Machine Vision Conference*, 2008, p. .
- [94] X. Zhou, X. Zhuang, S. Yan, S. Chang, M. Johnson, and T. Huang, "SIFT-Bag kernel for video event analysis," in *ACM international conference on Multimedia*, Vancouver, British Columbia, Canada, 2008, pp. 229-238.
- [95] J. Liu, J. Yang, Y. Zhang, and X. He, "Action Recognition by Multiple Features and Hyper-Sphere Multi-class SVM," in *International Conference on Pattern Recognition*, 2010, pp. 3744-3747.
- [96] X.-Y. Yang, J. Liu, M.-Q. Zhang, and N. Ke, "A New Multi-class SVM Algorithm Based on One-Class SVM," in *international conference on Computational Science*, Beijing, China, 2007, pp. 677-684.
- [97] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.
- [98] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *international conference on Multimedia*, Augsburg, Germany, 2007, pp. 357-360.
- [99] G. Willems, T. Tuytelaars, and L. V. Gool, "An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector," in *European Conference on Computer Vision*, Marseille, France, 2008, pp. 650-663.
- [100] R. Mattivi and L. Shao, *Robust Spatio-Temporal Features for Human Action Recognition* vol. 346: Springer Berlin / Heidelberg, 2011.
- [101] L. Wang and D. Suter, "Learning and Matching of Dynamic Shape Manifolds for Human Action Recognition," *IEEE Trans. Image Processing*, vol. 16, pp. 1646-1661, 2007.
- [102] H. Meng, N. Pears, and C. Bailey, "A Human Action Recognition System for Embedded Computer Vision Application," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-6.
- [103] D. Weinland, R. Ronfard, and E. Boyer, "Free Viewpoint Action Recognition Using Motion History Volumes," *Computer Vision and Image Understanding*, vol. 104, pp. 249-257, 2006/12// 2006.

- [104] J. Han and B. Bhanu, "Human Activity Recognition in Thermal Infrared Imagery," in *Computer Vision and Pattern Recognition*, 2005, pp. 17-17.
- [105] T. H. W. Lam, R. S. T. Lee, and D. Zhang, "Human gait recognition by the fusion of motion and static spatio-temporal templates," *Pattern Recognition*, vol. 40, pp. 2563-2573, 2007.
- [106] T. H. W. Lam, K. H. Cheung, and J. N. K. Liu, "Gait flow image: A silhouette-based gait representation for human identification," *Pattern Recognition*, vol. 44, pp. 973-987, 2011.
- [107] E. Zhang, Y. Zhao, and W. Xiong, "Active Energy Image Plus 2DLPP for Gait Recognition," *Signal Processing*, vol. 90, pp. 2295-2302, 2010.
- [108] A. I. Bazin, L. Middleton, and M. S. Nixon, "Probabilistic fusion of gait features for biometric verification," in *International Conference on Information Fusion*, 2005, pp. 1211-1217.
- [109] S. Kumar, B. Raman, and N. Sukavanam, "Human action recognition in a wide and complex environment," in *SPIE - The International Society for Optical Engineering*, 2011.
- [110] G. V. Veres, M. S. Nixon, L. Middleton, and J. N. Carter, "Fusion of dynamic and static features for gait recognition over time," in *International Conference on Information Fusion*, 2005, pp. 1204-1210.
- [111] E. Shechtman and M. Irani, "Space-Time Behavior Based Correlation," in *Computer Vision and Pattern Recognition*, 2005, pp. 405-412.
- [112] A. Yilmaz and M. Shah, "Actions Sketch: A Novel Action Representation," in *Computer Vision and Pattern Recognition*, 2005, pp. 984-989.
- [113] Y. Ke, R. Sukthankar, and M. Hebert, "Spatio-temporal Shape and Flow Correlation for Action Recognition," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [114] P. Yan, S. M. Khan, and M. Shah, "Learning 4D action feature models for arbitrary view action recognition," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-7.
- [115] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait Recognition: A Challenging Signal Processing Technology for Biometric Identification," *IEEE Signal Processing Magazine*, vol. 22, pp. 78-90, 2005.
- [116] M. Singh, M. Mandai, and A. Basu, "Pose Recognition Using the Radon Transform," in *Midwest Symposium on Circuits and Systems*, 2005, pp. 1091-1094.

- [117] Y. Chen, Q. Wu, and X. He, "Human Action Recognition by Radon Transform," in *International Conference on Data Mining Workshops.*, 2008, pp. 862-868.
- [118] W. Lin, M.-T. Sun, R. Poovandran, and Z. Zhang, "Human Activity Recognition for Video Surveillance," in *International Symposium on Circuits and Systems*, 2008, pp. 2737-2740.
- [119] J. Yamato, J. Ohya, and K. Ishii, "Recognizing Human Action in Time-sequential Images Using Hidden Markov Model," in *Computer Vision and Pattern Recognition.*, 1992, pp. 379-385.
- [120] M. Umeda, "Recognition of Multi-Font Printed Chinese Characters," in *International Conference on Pattern Recognition*, 1982, pp. 793-796.
- [121] J. W. Davis and A. Tyagi, "Minimal-latency Human Action Recognition Using Reliable-inference," *Image and Vision Computing*, vol. 24, pp. 455-472, 2006.
- [122] L. Wang and D. Suter, "Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [123] W. Lao, J. Han, and H. N. d. W. Peter, "Multi-level Human Motion Analysis for Surveillance Applications," in *Visual Communications and Image Processing*, San Jose, CA, USA, 2009, pp. 72570-72578.
- [124] F. Lv and R. Nevatia, "Single View Human Action Recognition Using Key Pose Matching and Viterbi Path Searching," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-8.
- [125] W. Li, Z. Zhang, and Z. Liu, "Graphical Modeling and Decoding of Human Actions," in *IEEE Workshop on Multimedia Signal Processing*, 2008, pp. 175-180.
- [126] K. Jia and D.-Y. Yeung, "Human Action Recognition Using Local Spatio-Temporal Discriminant Embedding," in *Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
- [127] M. F. Abdelkader, W. Abd-Elmageed, A. Srivastava, and R. Chellappa, "Silhouette-based Gesture and Action Recognition via Modeling Trajectories on Riemannian Shape Manifolds," *Computer Vision and Image Understanding*, vol. 115, pp. 439-455, 2011.
- [128] X. Wu, Y. Jia, and W. Liang, "Incremental Discriminant-Analysis of Canonical Correlations for Action Recognition," *Pattern Recognition*, vol. 43, pp. 4190-4197, 2010.
- [129] A. Ali and J. K. Aggarwal, "Segmentation and Recognition of Continuous Human Activity," in *IEEE Workshop on Detection and Recognition of Events in Video*, 2001, pp. 28-35.

- [130] G. Doretto, A. Chiuso, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, pp. 91-109, 2003.
- [131] C. Mazzaro, M. Sznajder, O. Camps, S. Soatto, and A. Bissacco, "A model (In) validation approach to gait recognition," in *3D Data Processing Visualization and Transmission*, 2002, pp. 700-703.
- [132] A. Bissacco, A. Chiuso, and S. Soatto, "Classification and Recognition of Dynamical Models: The Role of Phase, Independent Components, Kernels and Optimal Transport," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1958-1972, 2007.
- [133] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, pp. 1896-1909, 2005.
- [134] A. Bissacco and S. Soatto, "Hybrid Dynamical Models of Human Motion for the Recognition of Human Gaits," *International Journal of Computer Vision*, vol. 85, pp. 101-114, 2009.
- [135] A. B. Chan and N. Vasconcelos, "Classifying Video with Kernel Dynamic Textures," in *Computer Vision and Pattern Recognition*, 2007, pp. 1-6.
- [136] A. B. Chan and N. Vasconcelos, "Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 909-926, 2008.
- [137] A. B. Chan and N. Vasconcelos, "Layered Dynamic Textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, pp. 1862-1879, 2009.
- [138] C. Chen, J. Liang, and X. Zhu, "Gait recognition based on improved dynamic Bayesian networks," *Pattern Recognition*, vol. 44, pp. 988-995, 2011.
- [139] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, and C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms," in *International Conference on Pattern Recognition*, 2008, pp. 1-4.
- [140] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of Gaussians-Based Background Subtraction for Bayer-Pattern Image Sequences," *Circuits and Systems for Video Technology*, vol. 21, pp. 365-370, 2011.
- [141] M. Heikkilä and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 657-662, 2006.
- [142] L. Li, W. Huang, I. Y.-H. Gu, and T. Qi, "Statistical Modeling of Complex Backgrounds for Foreground Object Detection," *IEEE Trans. Image Processing*, vol. 13, pp. 1459-1472, 2004.

- [143] V. Ar'evalo, J. Gonz'alez, and G. Ambrosio, "Shadow detection in colour high-resolution satellite images," *International Journal of Remote Sensing*, vol. 29, pp. 1945 - 1963, 2008.
- [144] G. D. Finlayson, S. D. Hordley, L. Cheng, and M. S. Drew, "On the removal of shadows from images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 59-68, 2006.
- [145] Y. Shor and D. Lischinski, "The Shadow Meets the Mask: Pyramid-Based Shadow Removal," *Computer Graphics Forum*, vol. 27, pp. 577-586, 2008.
- [146] Y. Wang, K.-F. Loe, and J.-K. Wu, "A dynamic conditional random field model for foreground and shadow segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, pp. 279-289, 2006.
- [147] N. Martel-Brisson and A. Zaccarin, "Learning and Removing Cast Shadows through a Multidistribution Approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1133-1146, 2007.
- [148] S. Nadimi and B. Bhanu, "Physical models for moving shadow and object detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 1079-1087, 2004.
- [149] A. J. Joshi and N. P. Papanikolopoulos, "Learning to Detect Moving Shadows in Dynamic Environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 2055-2063, 2008.
- [150] M. R. Teague, "Image Analysis via the General Theory of Moments," *Journal of the Optical Society of America*, vol. 70, pp. 920-930, 1980.
- [151] G. Lu and A. Sajjanhar, "Region-based shape representation and similarity measure suitable for content-based image retrieval," *Multimedia Systems*, vol. 7, pp. 165-174, 1999.
- [152] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, pp. 1-19, 2004.
- [153] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *IRE Trans. Electronic Computers*, vol. EC-10, pp. 260-268, 1961.
- [154] F. Krolupper and J. Flusser, "Polygonal shape description for recognition of partially occluded objects," *Pattern Recogn. Lett.*, vol. 28, pp. 1002-1011, 2007.
- [155] S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval by shape similarity with perceptual distance and effective indexing," *IEEE Trans. Multimedia*, vol. 2, pp. 225-239, 2000.

- [156] R. Huang, V. Pavlovic, and D. N. Metaxas, "Shape Analysis Using Curvature-Based Descriptors and Profile Hidden Markov Models," in *International Symposium on Biomedical Imaging*, 2007, pp. 1220-1223.
- [157] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing Analysis and Machine Vision*, second ed.: Thomson-Engineering, 1998.
- [158] Y. Huang, T. S. Huang, and H. Niemann, "A Region-based Method for Model-free Object Tracking," in *International Conference on Pattern Recognition*, 2002, pp. 592-595.
- [159] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IEEE Trans. on Information Theory*, vol. 8, pp. 179-187, 1962.
- [160] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *CVGIP: Graphical Models and Image Processing*, vol. 54, pp. 438-460, 1992.
- [161] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *SIAM International Conference on Data Mining*, 2005, pp. 506-510.
- [162] J. Aach and G. Church, "Aligning Gene Expression Time Series with Time Warping Algorithms," *Bioinformatics*, vol. 17, pp. 495-508, 2001.
- [163] E. G. Caiani, A. Porta, G. Baselli, M. Turiel, S. Muzzupappa, F. Pieruzzi, C. Crema, A. Malliani, and S. Cerutti, "Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume," in *Computers in Cardiology*, 1998, pp. 73-76.
- [164] Y. Zhu and D. Shasha, "Warping indexes with envelope transforms for query by humming," in *ACM SIGMOD international conference on Management of data*, San Diego, California, 2003, pp. 181-192.
- [165] Z. Huang, Y. Li, J. Shao, H. T. Shen, L. Wang, D. Zhang, X. Zhou, and X. Zhou, "Content-Based Video Search: Is there a Need, and Is it Possible?," in *Information-Explosion and Next Generation Search*, 2008, pp. 12-19.
- [166] V. Kellokumpu, G. Zhao, and M. Pietikinen, "Dynamic Textures for Human Movement Recognition," in *ACM International Conference on Image and Video Retrieval*, Xi'an, China, 2010, pp. 470-476.
- [167] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-Independent Action Recognition from Temporal Self-Similarities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 172-185, 2011.
- [168] Z. Lin, Z. Jiang, and L. S. Davis, "Recognizing actions by shape-motion prototype trees," in *International Conference on Computer Vision*, 2009, pp. 444-451.

- [169] M. Ahmad and S.-W. Lee, "Human Action Recognition Using Shape and CLG-motion Flow from Multi-view Image Sequences," *Pattern Recognition*, vol. 41, pp. 2237-2252, 2008.
- [170] Y. Sakurai, M. Yoshikawa, and C. Faloutsos, "FTW: fast similarity search under the time warping distance," in *ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2005, pp. 326-337.
- [171] S. Salvatore and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *Workshop on Mining Temporal and Sequential Data*, 2004, pp. 70-80.