POSE SENTENCES: A NEW REPRESENTATION FOR UNDERSTANDING HUMAN ACTIONS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Kardelen Hatun

August, 2008

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

Assist. Dr. Pınar Duygulu(Advisor)

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

Assist. Prof. Dr. Çiğdem Gündüz Demir

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

Assoc. Prof. Dr. Aydın Alatan

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

POSE SENTENCES: A NEW REPRESENTATION FOR UNDERSTANDING HUMAN ACTIONS

Kardelen Hatun
M.S. in Computer Engineering
Supervisor: Assist. Dr. Pınar Duygulu
August, 2008

In this thesis we address the problem of human action recognition from video sequences. Our main contribution to the literature is the compact use of poses while representing videos and most importantly considering actions as pose-sentences and exploit string matching approaches for classification. We focus on single actions, where the actor performs one simple action through the video sequence. We represent actions as documents consisting of words, where a word refers to a pose in a frame. We think pose information is a powerful source for describing actions. In search of a robust pose descriptor, we make use of four well-known techniques to extract pose information, Histogram of Oriented Gradients, k-Adjacent Segments, Shape Context and Optical Flow Histograms. To represent actions, first we generate a codebook which will act as a dictionary for our action dataset. Action sequences are then represented using a sequence of pose-words, as pose-sentences. The similarity between two actions are obtained using string matching techniques. We also apply a bag-of-poses approach for comparison purposes and show the superiority of pose-sentences. We test the efficiency of our method with two widely used benchmark datasets, Weizmann and KTH. We show that pose is indeed very descriptive while representing actions, and without having to examine complex dynamic characteristics of actions, one can apply simple techniques with equally successful results.

Keywords: Human motion, action recognition, string matching, bag-of-words.

ÖZET

POZ CÜMLELERİ: İNSAN AKTİVİTLERİNİ ANLAMAK İÇİN YENİ BİR TANIM

Kardelen Hatun
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. Pınar Duygulu
Ağustos, 2008

Bu tezde, video bilgisinden insan hareketlerini tanımaya yönelik bir çalışma sunulmaktadır. Literatüre ana katkılarımız, videoları temsil ederken poz bilgisinin verimli kullanılması ve en önemlisi insan hareketlerinin poz cümleleri olarak değerlendirilmesi ve bu cümlelerin dizgi eşleme yöntemlerinden yararlanarak sınıflandırılmasıdır. Tek bir aktörün tek bir hareket gerçekleştirdiği videolar üzerine odaklanıyoruz. Her bir sözcük bir poza denk gelecek biçimde, aktiviteleri sözcükler içeren dökümanlar olarak temsil ediyoruz. Poz bilgisinin hareketi tanımlarken çok güçlü bir kaynak olduğunu düşünmekteyiz. Pozları en iyi şekilde betimlemek için literatürde başarılı sonuçlar vermiş dört farklı yöntemi kullanıyoruz; gradyan yön histogramları, k-bitişik kesim, şekil konteksti ve optik akım histogramları. Hareketleri temsil etmek amacıyla, ilk önce aktivite veritabanımız için sözlük niteliği taşıyacak bir kod rehberi oluşturuyoruz. Hareketleri poz sözcüğü dizileri, poz cümleleri olarak temsil ediyoruz. Aktiviteler arası benzerlikeri bulurken, dizgi eşleştirme yöntemlerinden yararlanıyoruz. Ayrıca poz kümeleri yöntemini de karşılaştırma amaçlı uygulamaktayız ve poz cümleleri tekniğimizin üstün olduğu göstermekteyiz. Metodumuzun verimini ölçmek için sıklıkla kullanılan iki video veritabanı, Weizmann ve KTH, üzerinde testlerimizi gerçekleştirdik. Pozun hareket tanımında çok açıklayıcı olduğunu ve hareketlerin karışık dinamiklerini inceleyen yöntemler yerine, daha basit tekniklerle de başarılı sonuçlar alınabileceğini göstermekteyiz.

Anahtar sözcükler: İnsan hareketi, hareket tanıma, dizgi eşleştirme, bag-of-words.

Acknowledgement

First and foremost I would like to thank my supervisor, Dr.Pınar Duygulu, for her patience and guidance. It is a great privilege to work with her, her hard-working, understanding and kind nature always leads her students to the best possible state they can reach.

I would also like to pay my respect to Assoc.Prof.Aydın Alatan and Assist. Prof. Dr. Çiğdem Gündüz Demir for sharing their invaluable insight with me. Without their input this thesis would have been incomplete.

My many thanks to RETINA members, who were always there for cheering me up. My special thanks to Nazlı Ikizler for her kindness and much needed help.

I want to express my love and gratitude to my family, they made me the person that I am today.

Contents

1 Introduction **1**

1.1 Organization of the Thesis 3

2 Related Studies **5**

2.1 Action Representations 5

2.2 Bag-of-Words Approaches 6

2.3 Sequence Matching Approaches 7

2.4 Discussion 8

3 Pose Sentences **9**

3.1 Approach 9

3.2 Pose Words Representation 11

3.2.1 Histogram of Oriented Gradients 11

3.2.2 Shape Context 13

3.2.3 k-Adjacent Segments (kAS) 14

3.2.4 Optical Flow Histograms 17

3.2.5 Codebook Generation 18

3.3 Pose Sequence Matching Techniques 19

3.3.1 Bag-of-Poses 21

3.3.2 String Matching 21

3.3.3 Classification 26

4 Experiments **28**

4.1 Datasets 28

4.1.1 Weizmann Dataset 29

4.1.2 KTH Dataset 29

4.2 Experiments on Weizmann Dataset 30

4.2.1 Experiments with HOG 30

4.2.2 Experiments with Shape Context 33

4.2.3 Comparisons 34

4.3 Experiments on KTH Dataset 37

4.3.1 Experiments with HOG 37

4.3.2 Experiments with Shape Context 39

4.3.3 Experiments with kAS 39

4.3.4 Experiments with Optical Flow 45

4.3.5 Comparisons 45

4.4 Experimental Discussion 47

5 Summary and Conclusions **52**

5.1 Summary of Contributions 52

5.2 Discussion Summary 53

5.3 Future Work 54

List of Figures

3.1 An overall view of our system 10

3.2 The step-by-step system for action recognition 11

3.3 The feature extraction process for Histogram of Gradients feature 12

3.4 Calculating shape context of a pose 13

3.5 The similarity between two poses using shape context 15

3.6 Example figures for kAS descriptor 16

3.7 Optical flow histogram computation 18

3.8 An overview of codebook generation 19

3.9 Matching methods 20

3.10 Example Bag-of-Poses representations 22

3.11 An example of how pose-sentences approach correctly classifies an action, bag-of-words misclassifies. 24

4.1 Weizmann dataset sample frames 29

4.2 KTH dataset's four different environmental settings 31

4.3 Parameters for HOG on Weizmann 32

4.4 Success rates for varying K values 32

4.5 Confusion matrices for three matching methods for Histogram of Oriented Gradients on Weizmann dataset 35

4.6 Confusion matrices for three matching methods for Shape Context on Weizmann dataset 36

4.7 Sample clusters from KTH dataset for two different pose descriptors 38

4.8 Example KTH silhouettes 40

4.9 Confusion matrices for three matching methods for HOG on KTH dataset 41

4.10 Confusion matrices for three matching methods for shape context on KTH dataset 42

4.11 Confusion matrices for three matching methods for 2AS one by one matching on KTH dataset 43

4.12 Confusion matrices for three matching methods for Optical Flow on KTH dataset 44

List of Tables

4.1 Pairings of *pose descriptor - matching method* for Weizmann Dataset 34

4.2 Comparison with related studies (Weizmann Dataset) 34

4.3 Pairings of *pose descriptor - matching method for KTH dataset* . . 45

4.4 Results for OF + Matching Methods for KTH dataset's all shooting conditions. 46

4.5 KTH dataset's all shooting conditions, compraison with other studies. 46

4.6 Comparison with related studies (KTH Dataset) 47

Chapter 1

Introduction

Understanding human motion from a video sequence is a widely studied yet still challenging problem of computer vision. There are many application areas which can make use of an action recognition system including security surveillance, human computer interaction and social analysis. Considering the vast amount of video data accumulated every day, processing hundreds hours of video by hand is not possible. We need automated systems to aid us in this quest.

However, building automatic action recognition systems is difficult due to many facts. First of all, the area in motion should be found as the initial step. However, most of the current methods suffer from working only on limited conditions, causing the action recognition systems to start with an imperfect input. Secondly, even for simple actions, each person has his/her style in performing actions resulting in large variations. Also, creating an action recognition system robust to environmental factors; such as complex backgrounds or illumination changes, is very challenging. Last but not least, it is an open problem to build recognition systems that can work for everyday activities where the actions are random and complex.

Representation of an action is a crucial point for recognition. Similar to phonemes for speech recognition, and textons for texture analysis, one way to represent actions is to use primitive action units. However, building of primitives,

and representing the actions as collection of these primitives is critical.

The use of codebook representations in object and scene recognition, which are usually referred as visual words, also lead some studies in action recognition to use codebooks for building primitive action units. In some recent approaches, spatio-temporal codebooks, usually as generalization of 2D interest points, are proposed as primitive action units.

We argue that pose is a very important cue for understanding actions. Therefore, in this study, we used poses as the primitive action units and consider actions as collections of poses. We use the observation that, pose is essentially a shape, and we describe poses using shape features extracted from single action frames. Then, we quantize these features to obtain a *pose codebook*. We then represent the actions in the form of primitives from the codebook, which we refer as *pose words*.

The other crucial point is how to use the collection of primitives to represent action sequences. A recent direction is to use bag-of-words approaches, where the action sequences are represented in an unordered collection, referred as a bag, of primitives. However, such an approach disregards the temporal characteristics of actions. For example, consider the actions *running* and *walking* which are very similar in terms of poses. Even though sets of poses are similar, the sequence of these poses would be different since they have different temporal characteristics. If we incorporate sequence information with a good pose descriptor then we can build a successful action recognition system which will not confuse walking and running.

In this study, we capture the temporal characteristics by representing actions as sequences of poses, which we refer as *pose sentences*. We then propose a novel action recognition method based on the matching of pose sentences as if they are strings.

Our overall method consists of the following steps. First we extract pose shape information from each frame of an action, then we quantize these features

to obtain our primitives, which are then used to code actions and obtain pose-word representation. After converting each action to this representation, actions are matched as strings, preserving their temporal information.

To obtain pose shape information in the best way possible, we utilize three shape descriptors which have shown successful results in the shape matching area, from simple to complex: Histogram of Oriented Gradients [9], Shape Context [1], k-Adjacent Segments [13]. Although we consider action as sequence of poses, we think transitions between poses can also be very descriptive. To investigate this idea, we utilize an optical flow pose descriptor[18], which describes the transitive characteristics in-between poses.

For matching pose-word representations we utilize two string matching methods Edit Distance [24] and Longest Common Subsequence (LCS) [16]. We also apply the bag-of-words approach for comparison purposes and present experimental results regarding the effectiveness of these three methods.

We present two important contributions with this thesis; the first one is the use of pose for representing an action. Instead of examining complex action dynamics, we make use of shapes of poses, which is a powerful source of information for understanding actions. Our second contribution is representation and matching of action sequences by string matching techniques. With this method we also use temporal information while classifying actions, which is a very important cue when describing an action.

1.1 Organization of the Thesis

In Chapter 2 we present a literature review on action recognition. We mention well-known studies and studies that are similar to ours.

In Chapter 3 we explain the methods we use in detail. We first give an overall description of our system, then we explain the pose descriptor used in the system. Finally we talk about our matching approach and give algorithms. In Chapter 4

we present our experimental results and we conclude with a discussion in Chapter 5.

In Chapter 5 we give an overall discussion of our system and conclude the thesis. Future work is also a part of this chapter.

Chapter 2

Related Studies

Action recognition became a very popular research topic over the last decade and the literature on the subject is broad (see [15, 17, 22] for recent surveys). In this chapter, we first discuss the related studies based on action representations. Then we present the studies that use bag-of-words representation in detail. Finally, we will review other studies that view action recognition as a sequence matching problem.

2.1 Action Representations

The studies in action recognition can be grouped in terms of how they represent an action: in some group of studies the entire action sequence is combined into a single spatio-temporal representation [2, 3, 4, 26], while in another group the actions are represented in the form of basic action units or action primitives [5, 7, 8, 14, 20, 30].

Spatio-temporal features are used to recognize actions in many studies. In [3], actions are recognized using spatio-temporal patterns. Bobick and Wilson [4] presented a gesture recognition system, which makes use of a state based method, representing gestures as strings of observation segments. [26] presents

a new optical flow based motion descriptor using spatio temporal volumes. A well-known study by Blank et al. [2] models actions as space-time volumes.

Action primitives are used to describe a temporal sequence by using few time instances. In [14] for recognition only a few primitives are used as opposed to using the entire sequence (possibly divided into sub-trajectories). This study focuses on one-arm gestures. Here each primitive is classified using Mahalanobis distance and converted into a string. The actions are represented by these collections of strings and matched using edit distance as comparison metric. [20] presents a data-driven method for deriving perceptual-motor action and behavior primitives from human motion capture data. They separate the body parts into significant kinematic substructures. The data for these substructures are then clustered, forming action units. The action units arising from the intrinsic embedding and clustering are generalized into action primitives using motion interpolation. Other methods include using movemes [5], atomic movements[8], states [7] and dynamic instants [30] as action primitives.

2.2 Bag-of-Words Approaches

The bag-of-words approach is originally introduced in information retrieval area, to represent text documents as an unordered distribution of code-words and became popular in computer vision with object [10, 32, 33] and scene categorization area [6, 35].

In recent years the bag-of-words approach has been employed by many action recognition studies. In the application of this approach in action recognition, actions are represented as a collection of visual words which are the codebooks of spatio-temporal features.

In [23] Laptev et al. introduced space-time interest points and in [31] they used it to recognize action using Support Vector Machines (SVM). Dollar et al. use histogram of cuboids to recognize actions [12], whereas Niebles et al. used cuboids with probabilistic Latent Semantic Analysis (pLSA) to create a

unsupervised system [27]. Wong et al.'s study involved a pLSA approach with an implicit shape model to understand actions from spatio-temporal codebooks. In [36] actions are represented as histograms of code-words, and recognized with a novel method Semi-Latent Dirichlet Allocation (SLDA). In this study, similar to ours, poses/frames are used as code-words unlike other studies which often use primitives within a frame. k-medoids clustering is used for codebook generation. Actions are converted into bag-of-words representation and classified using SLDA. In [34], Thurau uses Histogram of Oriented Gradients (HOG) to track humans and uses the same feature to describe poses. To classify actions, bag-of-ngrams are used as behavior histograms.

2.3 Sequence Matching Approaches

Using sequence matching techniques in action recognition is a new technique; the authors of these studies stress the importance of ordering in an action.

[28] proposes a subsequence mining approach which is an extension of PrefixSpan([29]) subsequence mining algorithm with LPBoost ([11]). In this study Dollar's spatio-temporal detector is used to find interesting points within a sequence [12]. The feature vector provided by this method is a 3211-dimensional vector, which is then reduced to a 25-dimensional vector using PCA. The code book generation is done via k-means clustering algorithm. The coded video sequence is then binned to overlapping subsequences. For classification LPBoost algorithm is used to construct the classification function as a linear combination of weak hypothesis functions. To find the maximum-gain for the hypothesis functions Nowozin et al. propose a generalization of PrefixSpan called Discriminative Prefix Span (DPrefixSpan). The methods presented in this study are sophisticated methods that need to adopt optimization techniques.

2.4 Discussion

Our approach have many advantages over similar approaches. In [36], Wang et al. used bag-of-words representation to represent actions, but this causes the loss of temporal characteristics. We use string matching techniques which mind the ordering of poses. Also our classification scheme is simple, yet we obtain satisfactory results, on the other hand Wang et al. use an adapted version of Latent Dirichlet Allocation as the classification model.

Another similar study by Thurau [34], uses Histogram of Oriented Gradients(HoG) as the pose descriptor. For representing actions this study exploits n-grams, and each action is described as histograms of n-grams. This is a simple representation which can lose very important information about the action sequence. In our study, more complex string matching methods are used giving us a more accurate comparison among actions. Our approach is a sequence matching approach, but unlike the studies listed above, we use very simple techniques for matching actions.

Chapter 3

Pose Sentences

In this chapter, we describe the details of our proposed approach for action recognition based on a new representation for actions. In the following sections, we first present the overview of the proposed approach and then give detailed information about the generation of pose-words and the matching of pose-sentences for recognition.

3.1 Approach

In our proposed representation, each frame in an action sequence is coded as a *pose-word*, meaning each pose in a frame is replaced by a representative pose. Then the entire sequence is represented as a *pose-sentence* formed by a sequence of pose-words. By this way, every video sequence is represented with the same poseword dictionary. Then each video's coded representation, in the form of pose sentences, is compared with the representation of a query video to find the matching action. The overall approach is summarized in Figure 3.1. In the following paragraph, formal overview of the process is explained.

Each video $F_i = f_{i1}, f_{i2}, \ldots, f_{iN}$ is a sequence of frames $f_{ij}, j \in \{1 \ldots N\}$, where N is the number of frames in the F_i^{th} video. Then all frames in the set of

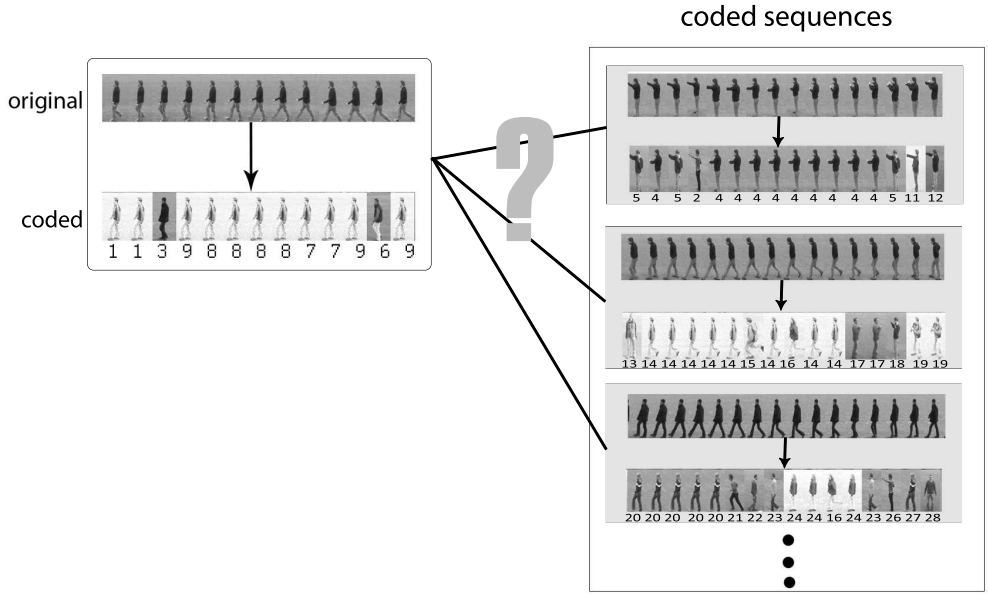


Figure 3.1: An overall view of our system

all videos, $F_1, F_2, \ldots, F_L$, where L is the number of videos in the dataset, goes through a feature extraction process to represent the poses in each frame. All of the extracted features, that is pose descriptors, which now represent frames in the data set are vector quantized to form a codebook of poses; $P = \{p_1 \ldots p_K\}$, where K, is the number of clusters. The clustering algorithm is applied to a frame-by-frame similarity matrix S. The entries of this matrix are calculated by using the distance corresponding to each feature type. After codebook generation each frame is coded with a pose-word a_{in} corresponding to one of p_k. The coded sequence is then represented as $A_i = a_{i1}, a_{i2}, \ldots, a_{iN}$. Following this step every frame in an action sequence is represented *pose-word*. Than the entire sequence is represented as a *pose-sentence* consisting of a sequence of pose-words, which is the coded representation of the features extracted from the frames. After obtaining the pose-word representations for each action in the dataset, we perform matching between pose-sentences.

The step-by-step process is illustrated in Figure 3.2.Following sections give the details of our pose-word representation.

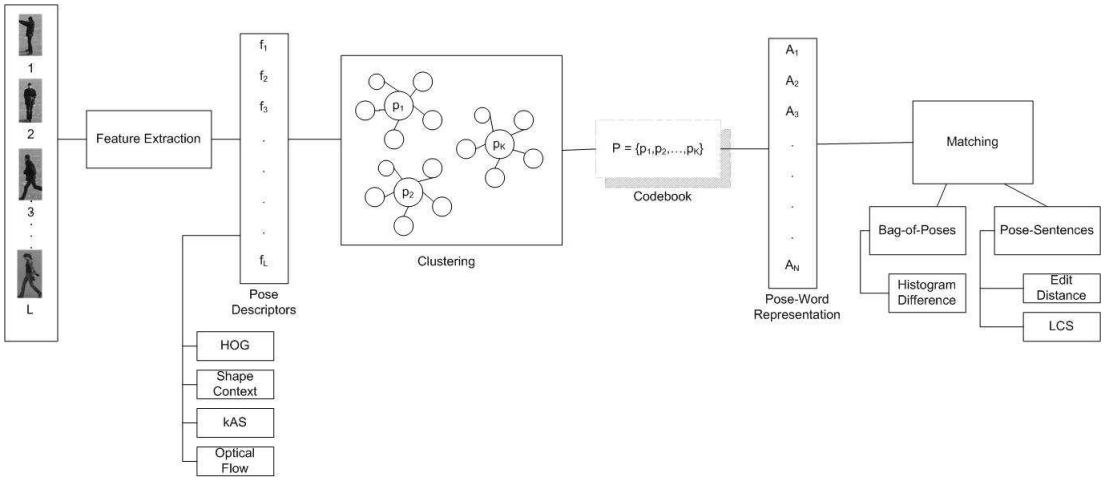


Figure 3.2: The step-by-step system for action recognition

3.2 Pose Words Representation

In this section, we present our method used to represent poses. Since pose is essentially a shape, we think pose can be represented with shape descriptors. But also describing the transition between poses is very important, to accomplish that we used an optical flow based descriptor.

The first three parts of this section explains the shape descriptors, from simple to complex. The fourth part is concerned with the optical flow based features. The last subsection explains the codebook generation process.

3.2.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) was reintroduced by Dalal and Triggs [9] in their study for human detection. HOG captures the edge direction information by histogramming orientations of gradients on the image. In our study HOG is utilized as a pose descriptor. It is a simple feature which provides fair shape information. Figure 3.3 summarizes the feature extraction process for HOG based pose descriptor.

In the first step, the gradients in a frame are obtained by applying 1-D filters

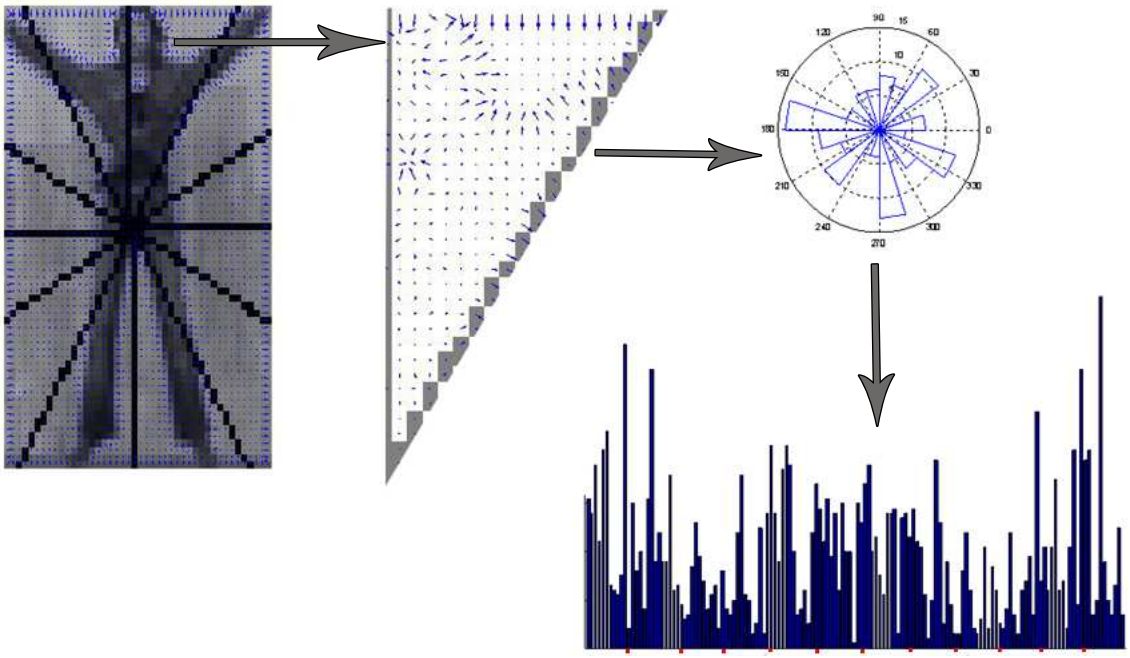


Figure 3.3: This figure shows the steps of the feature extraction process for HOG based descriptor. First an image is divided into n spatial bins radially, and then histogram of orientated gradients for each bin is constructed. After merging the histograms of all bins a feature vector of length $n \times m$ is obtained.

$[-1\ 0\ 1]$ and $[-1\ 0\ 1]^T$ (which is shown to be best in [9]) on the gray level image of the frame to obtain x and y directions gradients, G_x and G_y for each pixel.

Then, each frame is divided into n cells using a radial grid structure. In each cell, for m directions over the interval $[-\pi, \pi]$, the gradient magnitudes of the pixels in that direction are summed. Then, n histograms are attached to each other to obtain a $n \times m$ dimensional feature vector for each frame, describing the shape of the pose.

The similarity between HOGs are found by calculating the Euclidian distance between feature vectors.

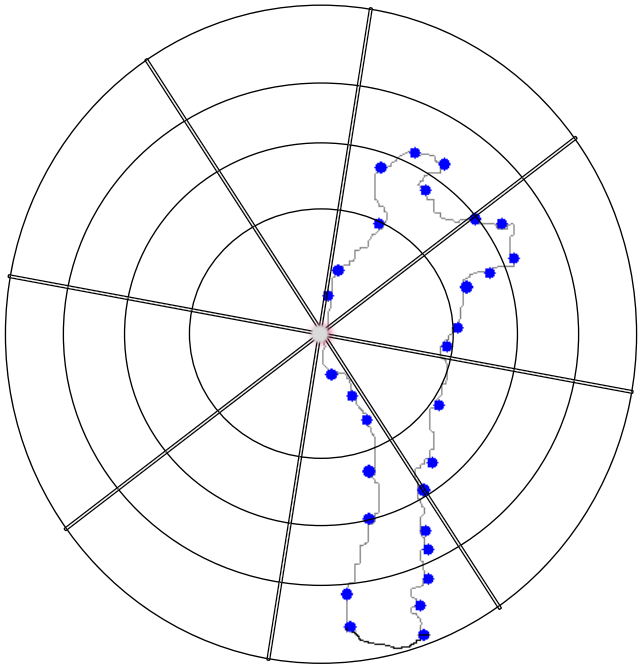


Figure 3.4: In shape context a circular grid is positioned over the sample points of an image. Sample points are obtained by randomly sampling over contour of the pose. There are 4 radial bins and 8 orientation bins in this figure.

3.2.2 Shape Context

Shape context is introduced by Belongie et al. in [1] originally for object recognition. In this approach a shape is represented by a discrete set of points sampled from internal or external contours on the shape. The contour information can be obtained by an edge detector. Then a sampling algorithm is applied to the contour, giving a set of n points: $U = \{\alpha_1, \alpha_2, \ldots \alpha_n\}, \alpha_i \in \mathbb{R}^2$. These points are not necessarily curvature or maxima points. The *shape context* is the coarse distribution of the rest of the shape with respect to a given point, α, on the shape. This is done by positioning α at the center of a circular grid which has r radial bins and θ orientation bins. In Figure 3.4 an example circular grid is shown. This circular grid covers the rest of the $n-1$ points, from where the distance and direction of distance to the point α is recorded.

Matching with shape context is mainly finding the best matching α_q in shape Q for each point α_w in shape W. Figure 3.5 shows application of shape context

matching between different frames of a *boxing* video. In Figure 3.5(a) two dissimilar frames of the video is matched, although the person's torso and legs are in very similar positions, the arms are positioned differently, giving a shape context cost of 0.24. In Figure 3.5(b) the matching of two consecutive frames are shown, the frames are very similar hence the cost is much lower; only 0.15.

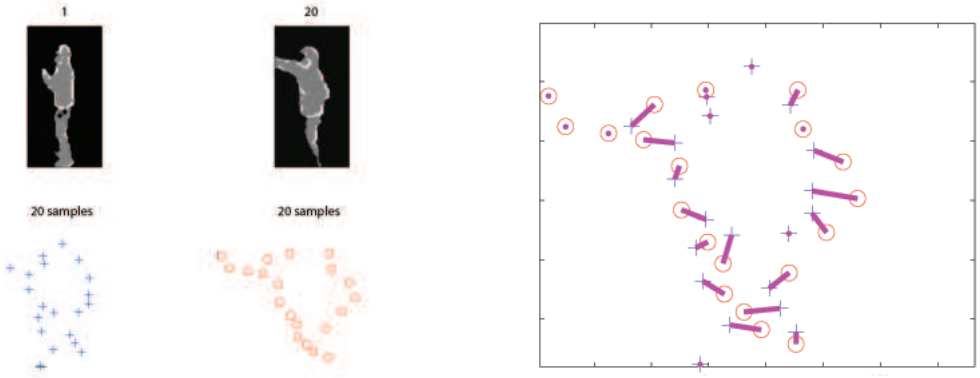
The similarity between shape contexts of two poses is obtained by the sum of distances between corresponding points of two poses.

3.2.3 k-Adjacent Segments (kAS)

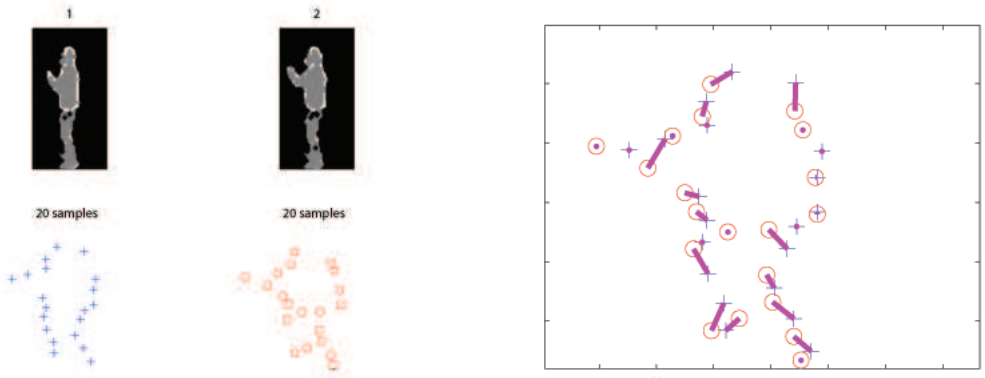
k-Adjacent Segments (kAS) descriptor is introduced by Ferrari et al. in [13] and is becoming popular in the object recognition area. kAS can directly be applied to graylevel images, so when there is not an accurate contour information this feature is more usable.

kAS computation has multiple steps. First edgels, edge pixels, of a gray level image is detected using Berkeley natural boundary detector[25]. Edgels are then chained by using closeness and orientation information. The edgel-chains are partitioned into roughly straight contour segments. This chained structure is used to construct a *contour segment network* (CSN).

A group of k segments is a kAS iff they can be ordered so that the i^{th} segment is connected in the CSN to the $(i+1)^{th}$ one, for $i \in \{1, \ldots, k-1\}$. As k grows, kAS can form more and more complex local shape structures: individual segments for $k = 1$; L shapes and 2-segment T shapes for $k = 2$; C, Y, F,Z shapes, 3-segment T shapes, and triangles for $k = 3$. In our study we used 2AS features. To devise kAS descriptor first the k segments forming kAS is ordered so that similar descriptor can have the same order. The ordered kAS is a list of segments; $\Phi = \{s_1, s_2, \ldots s_k\}$. Let $r_i = (r_i^x, r_i^y)$ be the vector going from midpoint of s_1 to the midpoint of s_i. Furthermore, let $\theta_i, l_i = \|s_i\|$ be the orientation and length of s_i. The descriptor of Φ is has $4k-2$ values, in Equation 3.1 we show the descriptor for 2AS:



(a) Dissimilar frames are matched, $cost = 0.24$



(b) Similar frames are matched, $cost = 0.15$

Figure 3.5: Two different cases of matching with shape context, + markers represent the samples points for the first pose and **o** markers are for the second pose. In (a) frames are dissimilar and cost is higher, but in (b) similar frames are matched so cost is much lower.

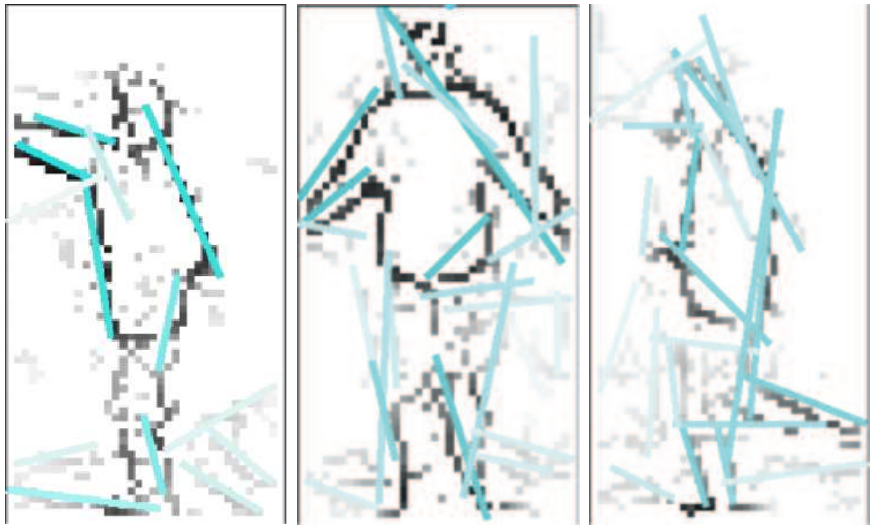


Figure 3.6: Example segments from kAS shape descriptor for actions *boxing*, *handclapping* and *running*

$$(\frac{r_2^x}{N_d}, \theta_1, \theta_2, \frac{l_1}{N_d}, \frac{l_2}{N_d}) \tag{3.1}$$

where N_d is a distance between the two farthest midpoints used as a normalization factor.

The dissimilarity measure $D(a, b)$ between two 2AS Φ^a, Φ^b is as follows ([13]):

$$D(a,b) = w_r \sum_{i=2}^{k} \|r_i^a - r_i^b\| + w_\theta \sum_{i=1}^{k} D_\theta(\theta_i^a, \theta_i^b) + \sum_{i=1}^{k} |\log(l_i^a, l_i^b)| \tag{3.2}$$

Here w_r and w_θ are weighting factor, which are chosen as 4 and 2 respectively, these values are shown to be best in [13].

In Figure 3.6 we present some example poses. Here we drew the segments on the edge image; darker colored segments' strength is larger than the lighter colored ones. The strengths are determined according to the boundary detector.

Originally in object recognition kAS is used the following way, after extracting kAS a codebook is generated and every object is represented as a bag-of-kAS. We exploit kAS information in a different manner.

The similarity between two frames containing multiple kAS is calculated as follows:

- *kAS one-by-one matching:* Since every pose is composed of multiple kAS we used a one-by-one matching scheme to calculate the similarity between two poses. Suppose pose a_{in} is composed of Δ_i number of kAS, and pose a_{jn} is composed of Δ_j number of kAS. Then the similarity between these poses is computed as follows: for each kAS $\Phi_{i\delta}, \delta \in \{1 \dots \Delta_i\}$ and $\Phi_{j\delta}, \delta \in \{1 \dots \Delta_j\}$ we compute the pair wise similarity using Equation 3.2 and put it in the similarity matrix S^{ij}, which has dimensions of $\Delta_i \times \Delta_j$. The similarity between a^{in} and a^{jn} is not the same as the similarity between a^{jn} and a^{in}, i.e. the similarity matrix of poses is not symmetric. Since our clustering method does not work on such matrices, we took the average of a^{jn} and a^{in}, and obtained symmetric matrix from these average values.

3.2.4 Optical Flow Histograms

Optical Flow Histograms is shown to be a promising pose descriptor in [18]. In this study, we adapt this feature to code the transitions in-between the frames as another pose descriptor.

For this descriptor first dense block based optical flow of each frame is computed by matching it to the previous frame ([18]). Then orientation histograms of optical flow values are computed. A similar approach can be found in [26]. Different from [26], here spatial ($M \times M$ grid) and directional binning (over $\theta = \{0, 90, 180, 270\}$) is used instead of the whole template. Also the smoothing step is skipped and optical flow values are used as they are. For each i^{th} spatial bin $i \in \{1 \dots M \times M\}$ the optical flow histogram $\kappa_i(\theta)$ is defined as

$$\kappa_i(\theta) = \sum_{j \in B_i} \psi(u_\theta \cdot F_j) \tag{3.3}$$

where F_j represents the flow value in each pixel j, B_i is the number of pixels in the spatial bin i, u_θ is the unit vector in the θ direction and ψ is defined as

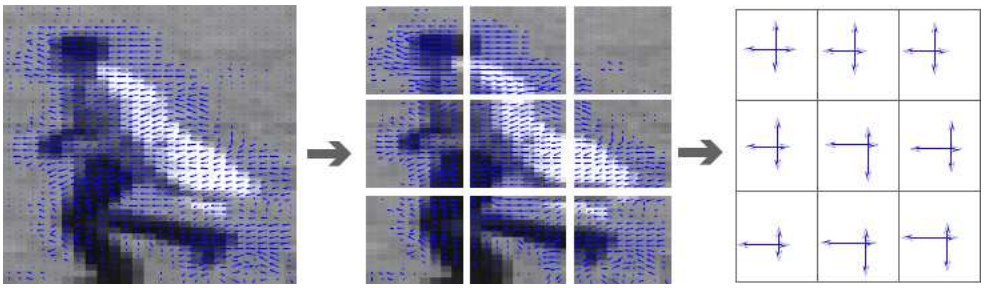


Figure 3.7: Inside each bin, the total amount of optical flow in four perpendicular directions is used as the motion descriptor.(Image taken from [18])

$$\psi(x) = \{ \begin{matrix} 0 \; if \; x \leq 0 \\ x \; if \; x \; 0 \end{matrix} \} \tag{3.4}$$

The process is illustrated in Figure 3.7.

This descriptor provides feature vectors for each pose, which are compared by using Euclidian distance.

3.2.5 Codebook Generation

After describing the poses in each frame using one of the descriptors and constructing a similarity measure, we group similar poses in K centroids with a clustering technique. We chose to use k-medoids because some pose descriptor we use cannot be represented with a feature vector, but rather only the similarities between points are known.

In this clustering technique, set of N objects, $O = \{o_1, o_2, \ldots o_N\}$ are partitioned into K clusters, where cluster centroids(medoids) $P = \{p_1, p_2, \ldots p_K\}, p_k \in O$. The aim is to minimize the overall cost, which is described by the cost of $o_n, i = \{1 \ldots N\}$ belonging to the group with the cluster centroid $p_k, k = \{1 \ldots K\}$.

The input of k-medoids algorithm is the similarity matrix S, an $M \times M$ matrix. Every element S_{ij} of matrix S, is the similarity between frame i and frame j. For small datasets we use the whole dataset to form this matrix, but for

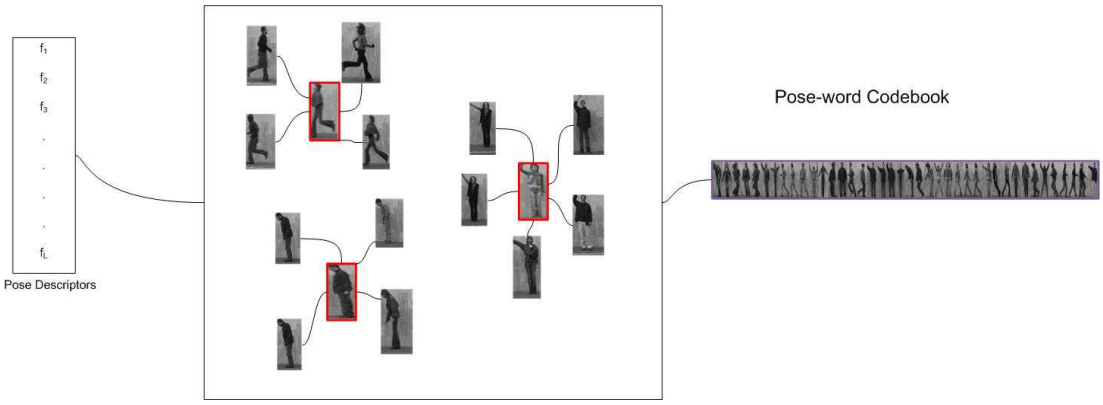


Figure 3.8: An overview of the codebook generation process. After the obtaining pose-descriptors, pose-words are generated using clustering method and a codebook for poses is obtained.

larger datasets only a collection frames are used.

The output of this algorithm,medoid set P, is used as the codebook of pose-words. The overview of the codebook generation process can be seen in Figure 3.8.

3.3 Pose Sequence Matching Techniques

In this section we explain the methods we used for pose sequence matching, after representing each action in terms of pose words. Some recent approaches use bag-of-words idea to obtain histograms of pose words. However, this causes loss of temporal information, since the actions are represented only as a distribution of poses . We consider pose sequences as strings and use string matching techniques to compare actions. This allows us to capture the temporal characteristics of actions and distinguish between the subtle temporal differences between two actions.

First we explain the bag-of-poses approach for simulating the bag-of-words approach. Then we present a novel method, which considers actions as pose sentences and uses string matching techniques to classify actions. An overview of matching methods can be seen in Figure 3.9.

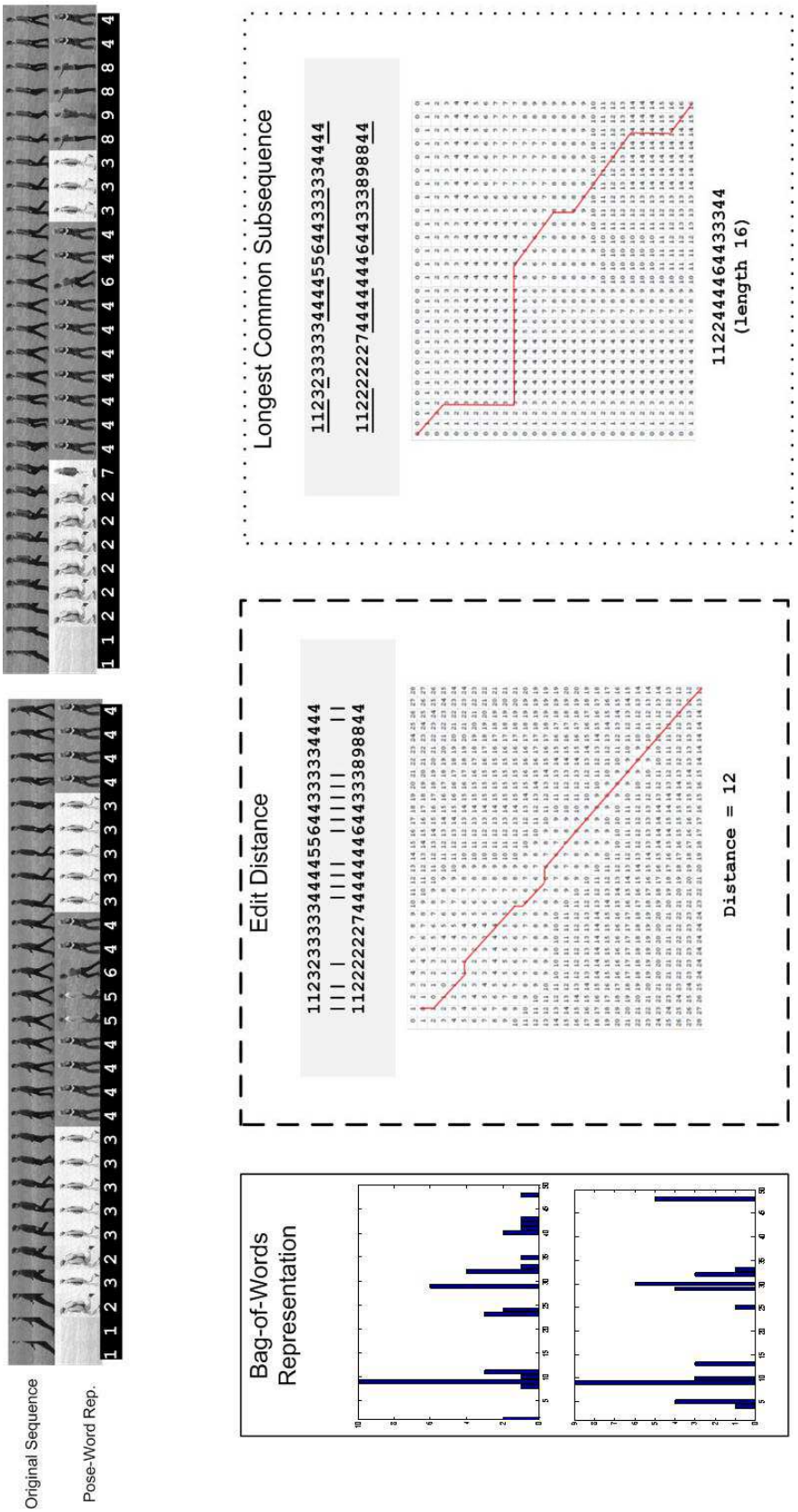


Figure 3.9: An overview of the matching methods, in pose-word representation we showed frames instead of pose-word numbers for easy visualization. We have three matching techniques; bag-of-poses, edit distance and longest common subsequence.

3.3.1 Bag-of-Poses

Bag-of-words was originally used in information retrieval and can be defined as representing a document by an unordered collection of words. To simulate the bag-of-words approaches in the simplest way, we represent the action sequences as histograms of pose-words. Let A_i be an action sequence and K be the number of pose words. In the bag-of-poses method, we represent A_i by a $1 \times K$ bins histogram $H_i = h_{i1} \dots h_{iK}$, where each bin h_{ik} corresponds to the number of frames represented with the pose word p_k.

The similarity between two actions' bag-of-poses, H_i and H_j, is defined using the Chi-square distance as

$$\chi^2(H_i, H_j) = \frac{1}{2} \sum_n \frac{(h_{in} - h_{jn})^2}{h_{in} + h_{jn}} \tag{3.5}$$

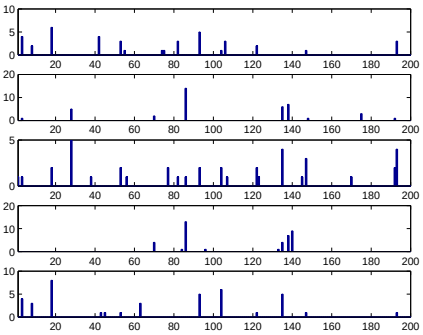
where H_i and H_j are the histogram representations of A_i and A_j.

Figure 3.10 shows the bag-of-poses comparisons of actions in KTH dataset performed by 5 different people, we used optical flow as the feature. If we look closely to Figure 3.10(e) and Figure 3.10(f) we can see that the histograms of these actions are quite similar, which may lead to misclassification.

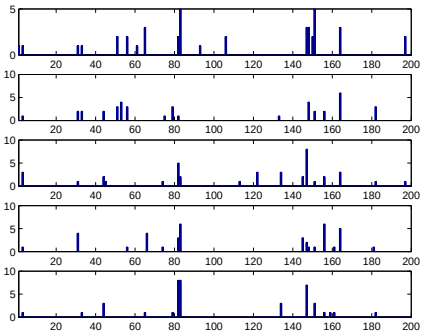
3.3.2 String Matching

In order to capture the temporal characteristics of actions, we represent the actions in the form of ordered sequences rather than simply using bag-of-poses. That is we represent an action A_i as a pose sentence $a_{i1}a_{i2} \dots a_{iN}$, where $N = |A_i|$ and each a_n is a pose-word $p_k \in P$. Then we utilized two well known string matching techniques to find the similarity of two pose sentences: namely, edit distance and longest common subsequence. In the following we explain the details of these methods.

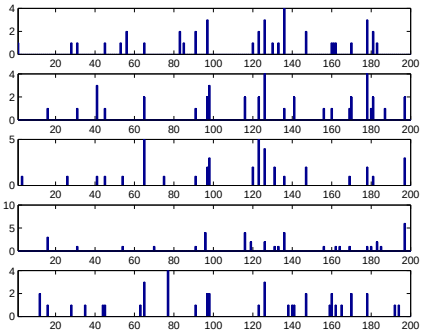
To illustrate the advantage of pose-sentences approach, in Figure 3.11 we



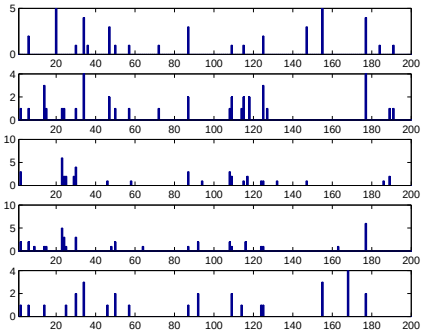
(a) Boxing



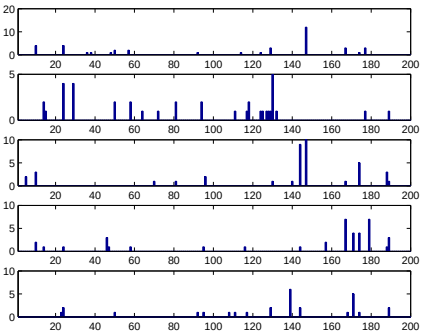
(b) Hand Clapping



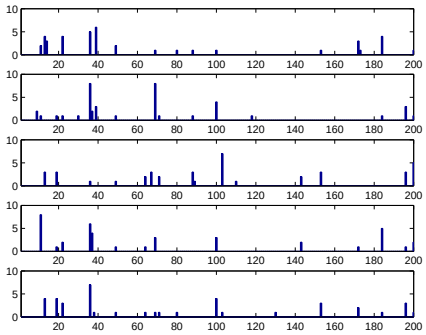
(c) Hand Waving



(d) Jogging



(e) Running



(f) Walking

Figure 3.10: Example Bag-of-Poses representations for each action in KTH dataset, $K = 200$

show similar actions being misclassified with bag-of-poses, but being correctly classified with pose-sentences approach. Here we have three action sequences, two of them being walk and one of them being move sideways. If we were to classify these with bag-of-poses approach, then the walk video would be misclassified as move sideways, since their histograms are more similar. But with pose sentences approach, we look at the sequence information and find that, two walk sequences are more similar.

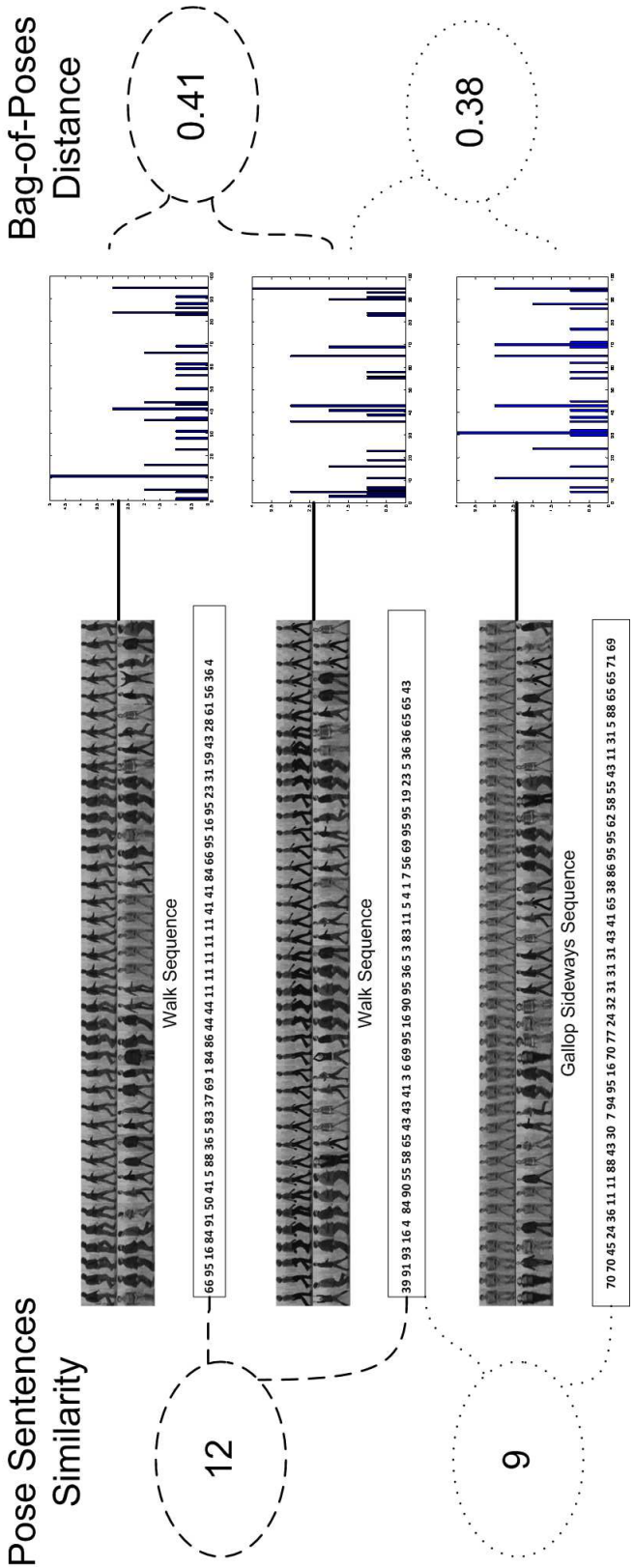


Figure 3.11: An example of how pose-sentences approach correctly classifies an action, bag-of-words misclassifies.

3.3.2.1 Edit Distance

To find the similarity of two actions A_i and A_j represented in the form of pose-sentences, we use a very simple string matching algorithm, *edit distance* [24]. With the edit distance algorithm, distance between two strings is defined as the minimum number of steps to be taken to convert A_i to A_j. The input of the algorithm is A_i and A_j and the output is the last, i.e. mn^{th}, entry of the dynamic programming matrix, which is the distance between A_i and A_j(see Algorithm 1).

```
Input: A_i (1..m), A_j (1..n)
Output: d(m,n)
foreach k from 0 to m do
    d(k,0) = k
end
foreach l from 0 to n do
    d(0,l) = l
end
foreach k from 1 to m do
    foreach l from 1 to n do
        if A_i (k) = A_j (l) then
            cost = 0
        else
            cost = 1
        end
        d(k,l) = minimum ( d[k-1, l] + 1, d[k, l-1] + 1, d[k-1, l-1] + cost)
    end
end
```

Algorithm 1: Algorithm for Edit Distance. Computes the distance between two strings, A_i and A_j

3.3.2.2 Longest Common Subsequence (LCS)

LCS finds the longest common subsequence between two sequences, the subsequence doesn't need to be contiguous ([16]). There can be wrongly interpreted poses, which may break the substring structure; the advantage this algorithm provides us comes from its ability to ignore these faults. (see Algorithm 2). The

input of the algorithm is A_i and A_j and the output is the last, i.e. mn^{th}, entry of the dynamic programming matrix, which is length of the longest common subsequence between A_i and A_j. Unlike Edit Distance LCS provides us with a similarity value not a dissimilarity value.

```
Input: A_i (1..m), A_j (1..n)
Output: d(m,n)
foreach k from 0 to m do
 |  d(k,0) = 0
end
foreach l from 0 to n do
 |  d(0,l) = 0
end
foreach k from 1 to m do
 |  foreach l from 1 to n do
 |  |  if A_i (k) = A_j (l) then
 |  |  |  d(i,j) = d(i-1,j-1) + 1
 |  |  else
 |  |  |  d(i,j) = maximum(d(i,j-1), d(i-1,j))
 |  |  end
 |  end
end
```

Algorithm 2: Algorithm for Longest Common Subsequence. Gives the length of the longest common subsequence of A_i and A_j

3.3.3 Classification

At this point we have the similarity values of action sequences. The next step is to find the corresponding action from the training set given a query video.

In this study, we focus on representation rather than classification and therefore choose a very simple classification scheme, nearest neighbor classification, to classify actions. Nearest neighbor classification is performed in the following way: one sequence whose label is unknown is picked, then from the rest of the sequences the least distant sequences label (i.e nearest neighbor's label) is assigned to that sequence. If the picked sequence's label is the same with the least

distance ones then we count this as a match.

Chapter 4

Experiments

This chapter presents our experimental results and comparisons with other related studies. We test various methods to understand how good they cohere with our compact action representation. Also our results shed light on many weak and strong points of the features used. In addition we compare our success rates to well-known studies in this chapter. In the following we explain the datasets we used, under separate sections, we will present the results of the experiments for each dataset.

4.1 Datasets

We perform our experiments in two different datasets, Weizmann and KTH. These are the benchmark datasets in action recognition research.

All of the experiments are done in MATLAB, without any optimization methods. We used 2.4 GHz Pentium Dual Core processor with 3Gbs of physical memory for most of the experiments.

Figure 4.1: Example poses from Weizmann dataset From left to right, upper row: bend, jump forward, gallop sideways and wave one hand, lower row: first two wave both hands, last two walk.

4.1.1 Weizmann Dataset

Weizmann dataset is introduced by Blank et al.[2]. It consists of 9 actions: walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place and jumping jack, performed by 9 different actors, a total of 81 videos. Video sizes vary in this dataset and there are a total of 5098 frames. Example poses can be seen in Figure 4.1.

4.1.2 KTH Dataset

KTH dataset, introduced in [31], is a more difficult dataset than Weizmann dataset in terms of actions involved and shooting conditions. It contains six actions: boxing, hand clapping, hand waving, jogging, running and walking. These actions are performed by 25 people in 4 different shooting conditions. In the first shooting condition **SC1** subjects perform the action on a grass background with stable camera except for the occasional zoom. Second shooting condition **SC2** is with the same grass backgorund but there are sots from different viewpoints and zoom effects. In third shooting condition **SC3** actors carry bags or wear baggy clothes and fourth shooting condition **SC4** is indoors with varying illumination. In our experiments we mainly used the first shooting condition. Example frames

from KTH dataset can be seen in Figure 4.2.

4.2 Experiments on Weizmann Dataset

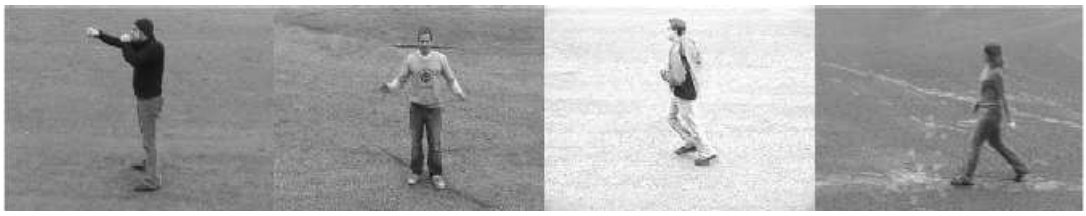
We performed experiments on Weizmann dataset using HOG and shape context features as pose descriptors. In the following sections we present the detailed experiments for choosing the parameters regarding these representations.

4.2.1 Experiments with HOG

In the following we present our experiments using HOG as a pose descriptor. First, we analyze the performance of HOG descriptor using different parameters. There two parameters of HOG which needs to be optimal to get successful results. First one is **n**, number of pieces in the radial grid and second one is **m** orientation bin size. Based on the observations it is seen that number of different action types is best represented when $K = 47$ and therefore for the experiments on choosing n and m, a fixed medoid size is set to $K = 47$ manually and the values of n and m vary. We tried the following values; $n = 4, 8, 12, 16, 20, 24$ and $m = 4, 8, 12, 16, 24, 36$, a total of 36 experiments. We also tried higher values but they gave poor success rates due to redundancy. Graphs in Figure 4.3 show the results of these experiments for varying m and n values.

Another parameter we have to determine is the choice of K, the number of posewords. In order to understand the choice of K in a randomly initialized k-medoids clustering algorithm, we choose $K = 30$, 40, 50, 60 values, and record the performance as shown in Figure 4.4 for fixed values m=24 and n=24. The results show that, although the choice of K affects the performance, the results are still in a similar level, and even with random initialization K around 50 is an acceptable choice. Note that this result is overlapping with the manual choice for $K = 47$ for representing different pose types.

The results suggest higher values for m and n, and show that the orientation



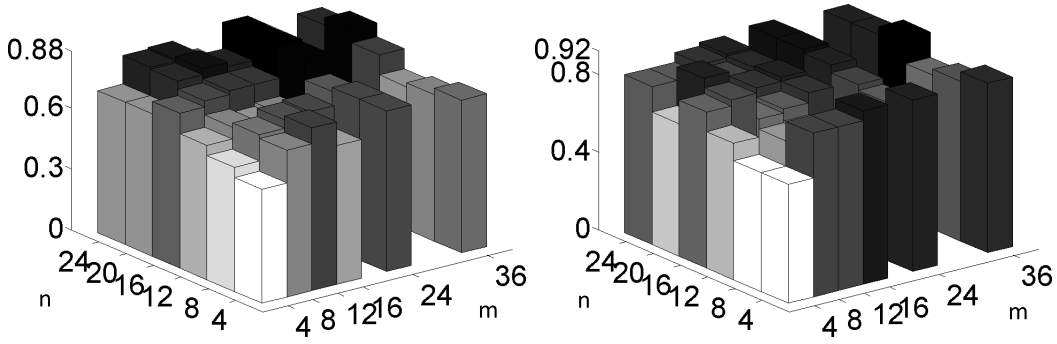
(a) Outdoor, SC1, left to right: boxing, hand clapping, jogging and walking

(b) Outdoor with zoom and different viewpoints, SC2, left to right: boxing, hand waving, running and walking

(c) Outdoor with clothing changes, SC3, left to right: boxing, hand clapping, walking and walking

(d) Indoor, varying illumination, SC4, left to right: boxing, hand clapping, running and walking

Figure 4.2: KTH dataset's four different environmental settings



(a) Different values of m,n for *Bag-of-Words* approach

(b) Different values of m,n for *Pose Sentences* approach

Figure 4.3: Determining m(orientation bin size) and n(number of pieces in the radial grid) parameters for HOG descriptor on Weizmann dataset

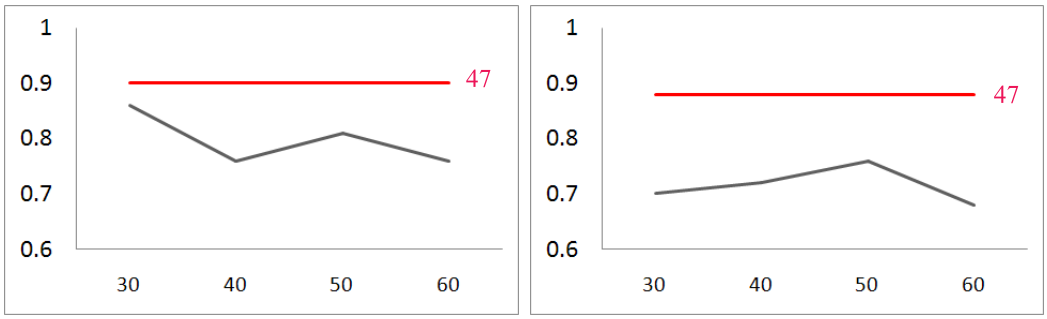


Figure 4.4: Success rates for varying K values using *Pose Sentences* and *Bag-of-Words*, these graphs show that a balanced initial set is crucial for k-medoids algorithm

bin size is more important. These parameters are specific to Weizmann dataset, for different datasets these experiments are repeated and best values are determined accordingly.

In Figure 4.5, confusion tables for each of the three matching methods are shown. The most similar actions for this dataset can be grouped as follows, A = {*move sideways*, *walk*, *run* and *jump forward*}, B = {*wave two hands*, *jumping jack*}. Figure 4.5(a) shows the results for bag-of-poses approach, which couldn't distinguish the group A actions on multiple occasion, on the other hand edit distance (Figure 4.5(b)) managed to reach a higher accuracy. Longest Common Subsequence failed to classify half of the jump forward videos, but successfully classified the rest of the group A actions.

On the average 92% performance as the best result is obtained for Weizmann dataset using HOG with edit distance.

4.2.2 Experiments with Shape Context

In addition to Histogram of Oriented Gradients we also used shape context to understand the effects of a more powerful shape descriptor. As mentioned before, Weizmann dataset has come with clean silhouettes which are essential for this particular shape feature. While testing on this dataset, we used 5 radial bins and 12 orientation bins.

We also tested for different values of K for this pose descriptor, according to our results $K \approx 100$ gives the best results for shape context.

For shape context pose descriptor we have achieved 100% success rate using LCS. Figure 4.6 shows the confusion matrices for each matching method.

	HOG	**SC**
Bag-of-Poses	90%	95%
Edit Distance	93%	92%
LCS	89%	100%

Table 4.1: Pairings of *pose descriptor - matching method* on Weizmann dataset, HOG:Histogram of Oriented Gradients, SC:Shape Context

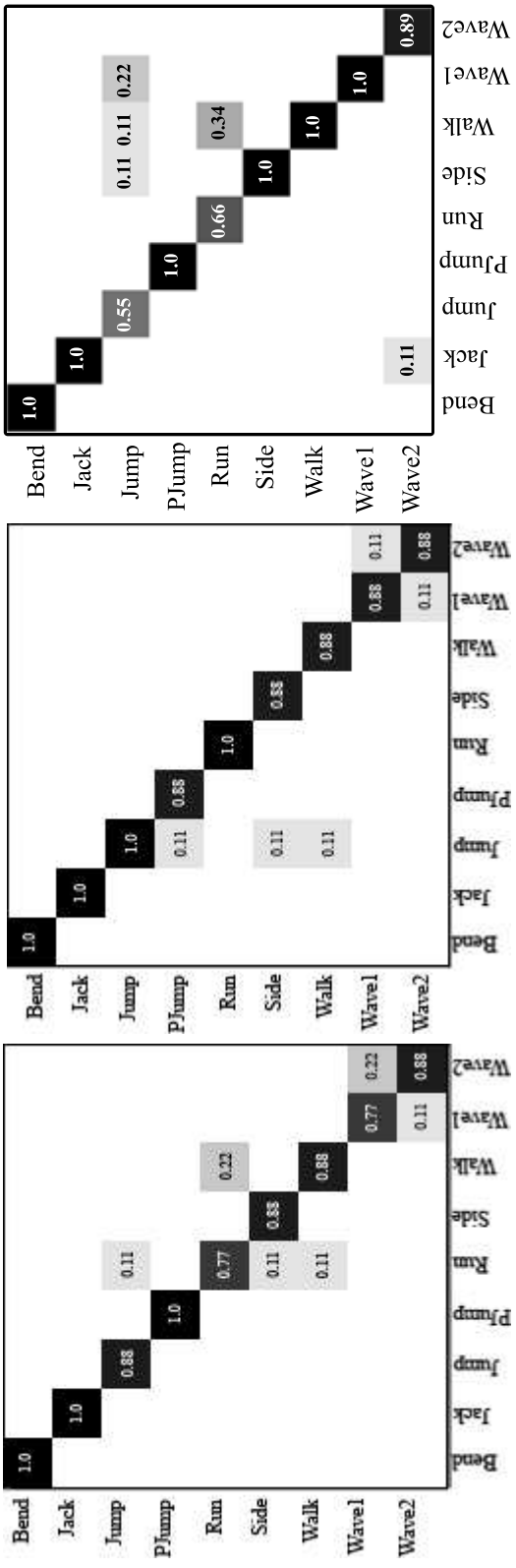
Matching Method	Success Rate
OurApproach	**100%**
Ikizler[19]	100%
Blank[2]	99%
Thurau[34]	87%
Niebles[27]	73%

Table 4.2: Comparison with related studies (Weizmann Dataset)

4.2.3 Comparisons

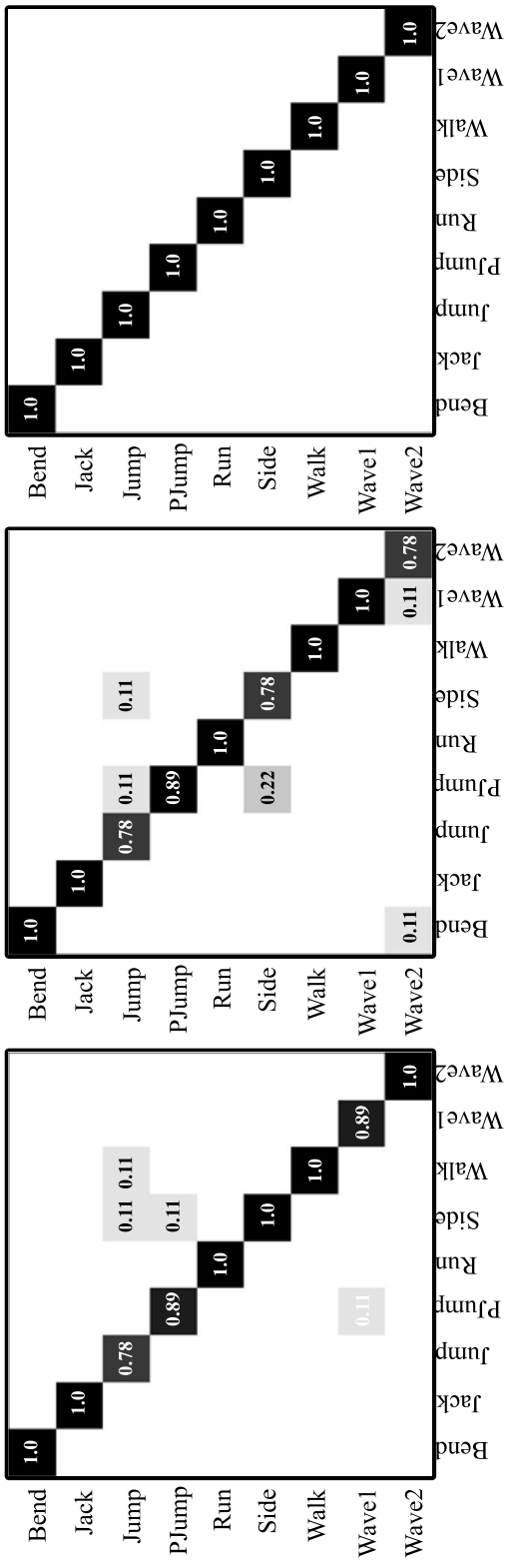
There are a number of studies which were also tested on Weizmann dataset. In Table 4.2 comparisons of our method with these methods are shown.

Results show that using shape context on Weizmann dataset, we achieve the perfect performance.



(a) Bag-of-poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.5: Confusion matrices for three matching methods for Histogram of Oriented Gradients on Weizmann dataset



(a) Bag-of-poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.6: Confusion matrices for three matching methods for Shape Context on Weizmann dataset

4.3 Experiments on KTH Dataset

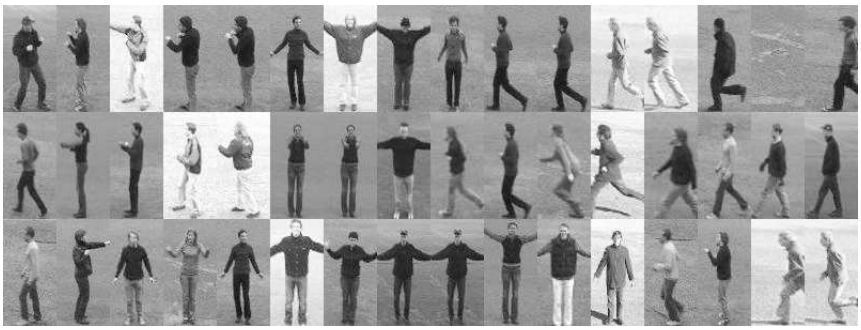
We used the 150 action videos from the first shooting condition of the KTH dataset, which is less complex than the other three shooting condition KTH dataset includes. We chose this part of the dataset because we had fewer irregularities to worry about when evaluating results.

KTH dataset is a large dataset, sequences contain more frames and each person performs an action different than another person which affects, so we tried the following values to understand which is a good vocabulary size: $K = \{50, 100, 200, 300, 400\}$. Previously Wang et al. found that 350 code-words was best for their system [36]. For different pose descriptors the ideal value of K changes. But from our experiments we have concluded that the value of K should be around 200-300 for KTH dataset. Some sample clusters can be seen in Figure 4.7.

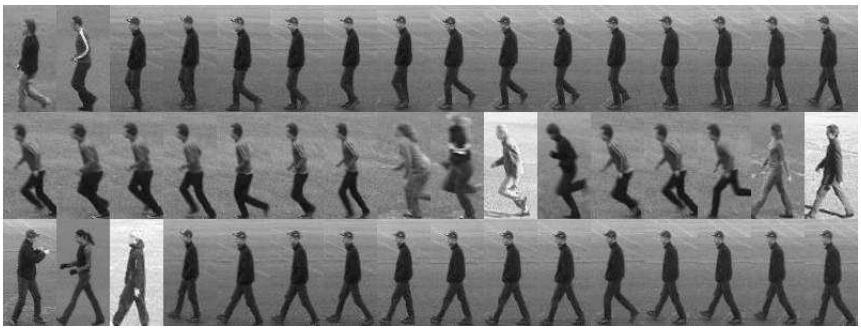
4.3.1 Experiments with HOG

We have used $n = 24$ and $m = 24$ with HOG feature on KTH dataset. Since we have tested with Weizmann dataset on determining m and n, we have not tested with values smaller than 12 for both of these parameters on KTH dataset.

In Figure 4.9 the confusion matrices for classification with HOG descriptor with each matching technique can be seen. If we look closely to each confusion table, we can see the advantages of string matching techniques over bag-of-poses. The results for HOG descriptor are moderately successful. For all methods misclassification can be considered in two chunks, hand actions = {boxing, handclapping, handwaving} and feet actions = {jogging, running, walking}. The misclassification of hand actions are mostly because the HOG descriptor is not an accurate enough shape descriptor when trying to discriminate similar poses. Also with hand action the actual ordering is not very important, even misleading in this case. If we look at the confusion tables, in fact with bag-of-poses representation we have acquired a higher success rate with hand actions. But with



(a) Sample clusters from kAS



(b) Sample clusters from HOG

Figure 4.7: Sample clusters from KTH dataset for two different pose descriptors

feet actions we have a different set of rules. Mainly because the actions *jogging* and *running* are very similar. For all of the matching techniques these actions are confused for most of the videos. But for action *running* and *walking*, which are also considered very similar, we see that Edit Distance performs much better than Bag-of-Poses. The same is true for LCS, although with LCS running and jogging have lower recognition rates than edit distance. Here, even though the success rates are not very high, the advantage of matching pose-sentences over bag-of-poses is very clear. The overall highest succes rate for this feature is 61% with LCS. Since this success rate is not satisfactory, thus we moved on to using shape context feature on this dataset, which gave us very good results with Weizmann dataset.

4.3.2 Experiments with Shape Context

Since we have seen the effectiveness os shape context on Weizmann dataset, we also used this descriptor on KTH dataset too. We used the following shape context parameters while testing on KTH, radial bin size 5, orientation bin size 12. We chose 20 as the number of points to be sampled over the contour.

Although this descriptor gave excellent results for Weizmann dataset, the success rates are very low for KTH dataset. As we mentioned before, shape context's accuracy is highly dependent on the silhouette information. In Figure 4.8 he silhouettes extracted from KTH dataset are shown. These silhouettes are not accurate and this affects this descriptors performance significantly. The poor success rates we obtained with shape context forced us to take another shape feature, kAS, into consideration and used it to represent frames.

4.3.3 Experiments with kAS

Originally in object recognition kAS is used by constructing a kAS codebook, and representing each object as the bag-of-words representation of that codebook[13]. In our methods, instead of using bag-of-words approach on kAS, we calculated

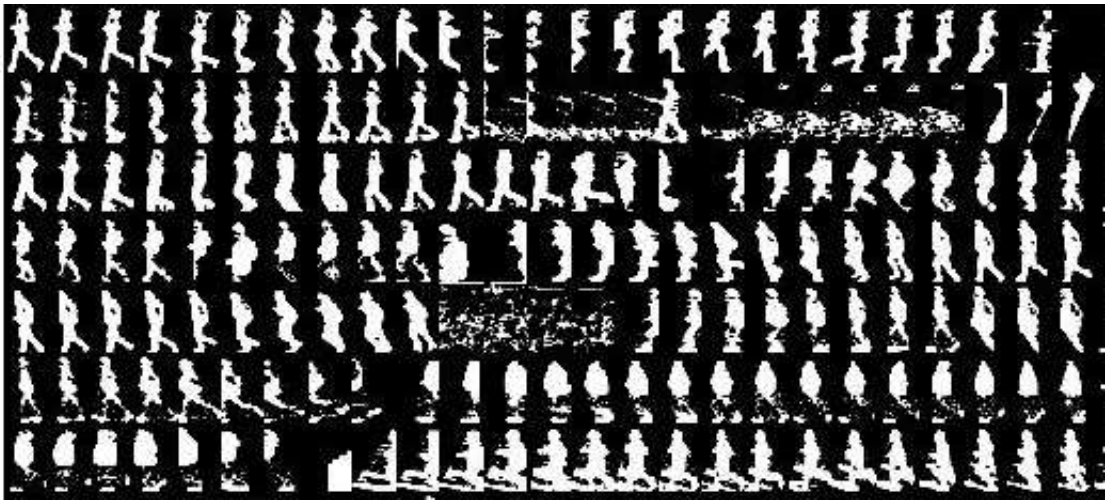


Figure 4.8: A portion of KTH silhouettes, these silhouettes are not sufficient for an accurate computation of shape context descriptor

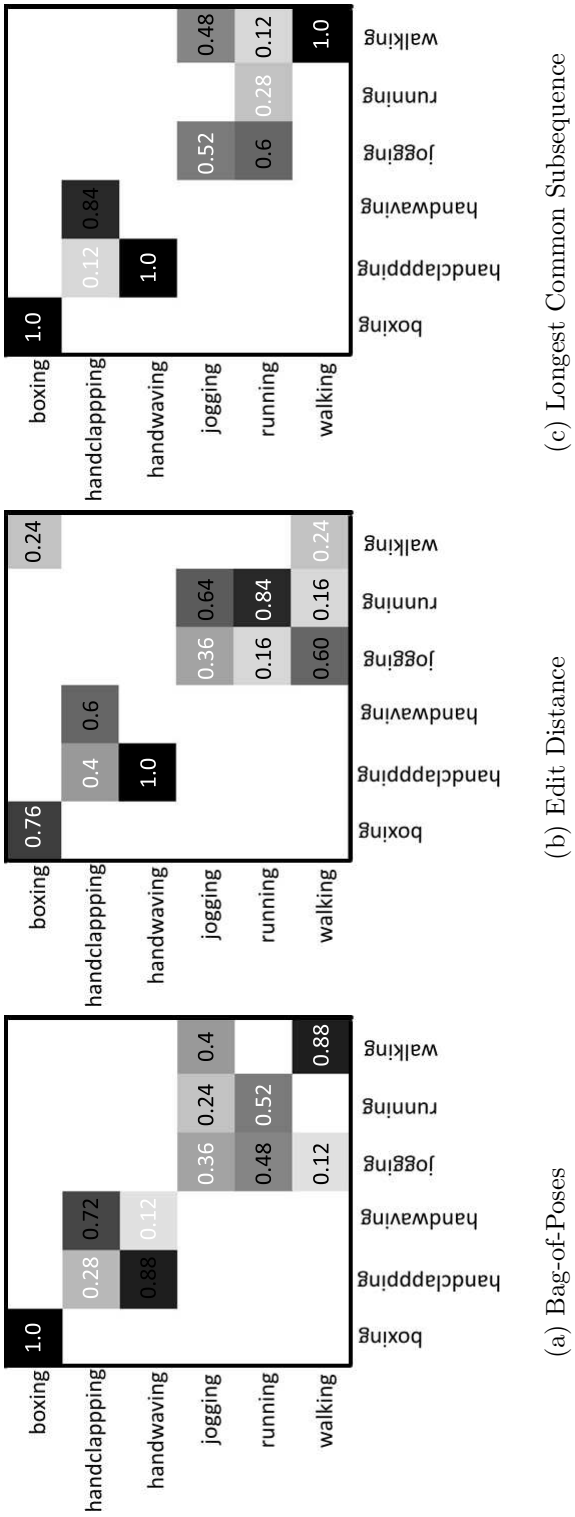
the total distance between 2AS of two poses.

We've obtained ta better recognition rate on KTH dataset with kAS feature an average performance of 74%. Although this is not successful, it is still an indication of how well shape descriptors perform while representing actions. Also as mentioned before, KTH is a very challenging dataset. kAS depends highly on edge detection, which may perform poorly with complex conditions of KTH dataset.

Although shape is very important descriptor as we have seen, we also tested the optical flow based feature to see the effect of transitions. Note that the used OF based feature does not only code the temporal information but since it uses a grid partitioning it considers the localizations and therefore it is a spatio-temporal feature.

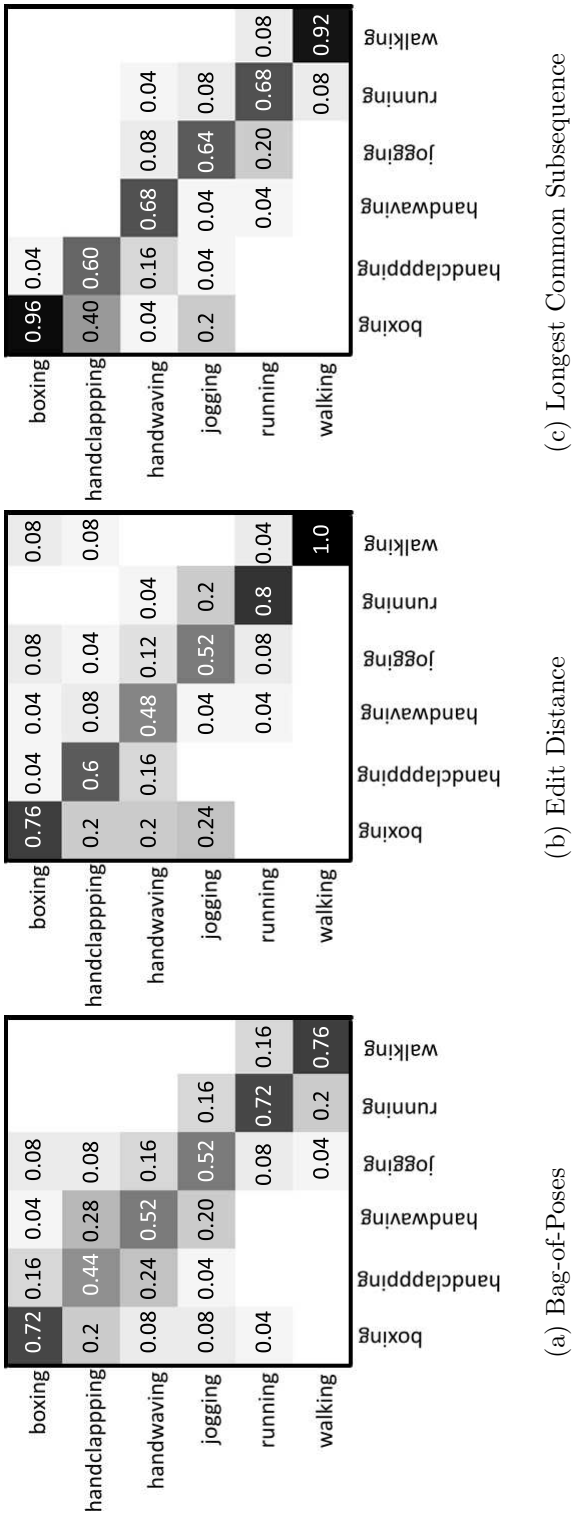
(a) Bag-of-Poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.9: Confusion matrices for three matching methods for HOG on KTH dataset



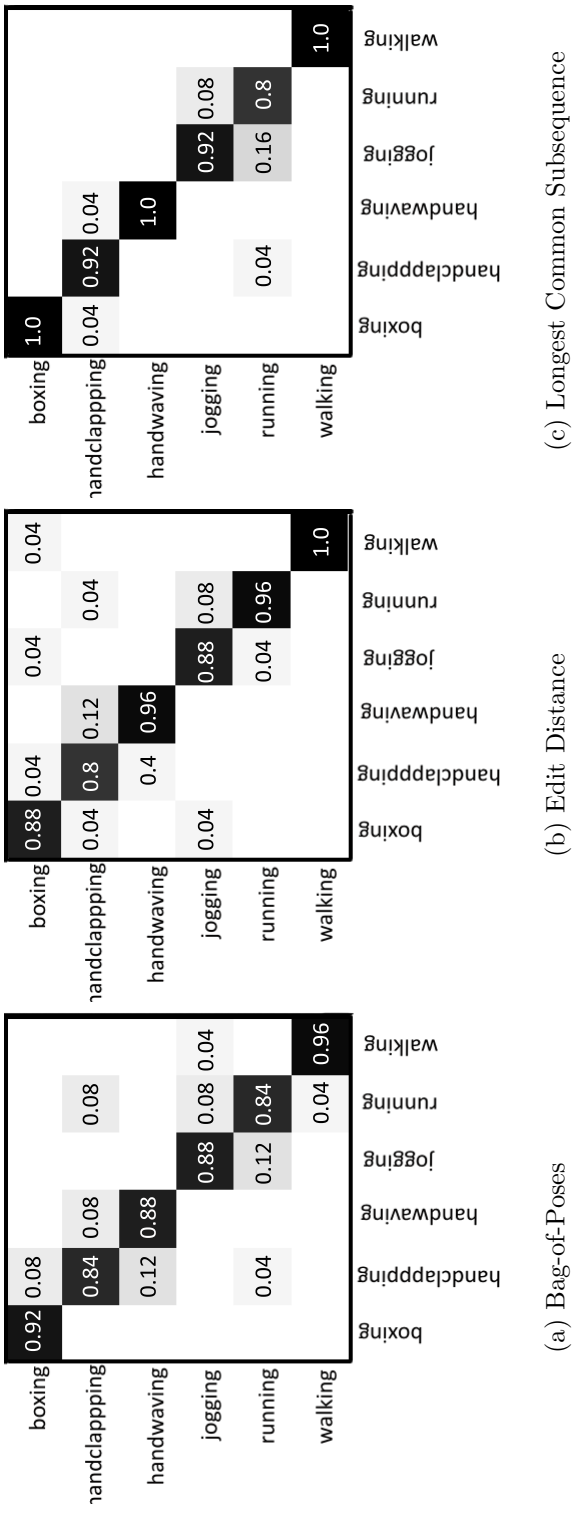
(a) Bag-of-Poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.10: Confusion matrices for three matching methods for shape context on KTH dataset



(a) Bag-of-Poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.11: Confusion matrices for three matching methods for 2AS one by one matching on KTH dataset



(a) Bag-of-Poses (b) Edit Distance (c) Longest Common Subsequence

Figure 4.12: Confusion matrices for three matching methods for Optical Flow on KTH dataset

	HOG	SC	2AS	OF
Bag-of-Poses	58%	56%	61%	89%
Edit Distance	60%	44%	70%	92%
LCS	61%	47%	74%	93%

Table 4.3: Pairings of *pose descriptor* - *matching method* on first shooting condition of KTH dataset, HOG:Histogram of Oriented Gradients, OF:Optical Flow, 2AS: 2AS matching, SC:Shape Context

4.3.4 Experiments with Optical Flow

We used the following setting for this descriptor, 5×5 block size and window size is 3. We used L_1 as the distance measure.

The recognition rate for optical flow is high, very close to best in literature. The confusion tables are given in Figure 4.12. The first confusion table in Figure 4.12(a) belongs to the results we have obtained by bag-of-poses approach. Compared to other tables in Figure 4.12(b) and Figure 4.12(c), the misclassification of actions running, jogging and walking is worse. However with Edit distance and LCS we don't confuse the action walking with others at all. Also we only confuse running and jogging in a few occasions, which is expected, since some of the jogging and running videos is KTH dataset are hard to distinguish, even with naked eye.

With optical flow feature we've obtained an average of 89% recognition set over the whole KTH dataset.

4.3.5 Comparisons

In Table 4.3 show the highest results obtained with the combinations of pose descriptors and matching methods. The values in the table are for SC1 of KTH dataset. We obtain the highest results with optical flow - LCS combination and a moderately successful result with 2AS - LCS. These results are discussed in Section 4.4.

	Bag-of-Poses	Edit Distance	LCS
sc1	89%	92%	93%
sc2	79%	81%	84%
sc3	84%	82%	85%
sc4	90%	92%	95%
average	85.5	86.75	89.25

Table 4.4: Results for OF + Matching Methods for KTH dataset's all shooting conditions.

Condition	Our Approach	Ikizler[18]	Jhuang [21]
s1	93%	98.2%	96.0%
s2	84%	90.7%	86.1%
s3	85%	88.9%	89.8%
s4	95%	98.2%	94.8%

Table 4.5: KTH dataset's all shooting conditions, compraison with other studies.

The successful results we've obtained with optical flow descriptor encouraged us into further testing with much more complex shooting conditions of KTH dataset. The results of these experiments can be seen in Table 4.4. With simple backgrounded videos in SC1 and SC4 we obtain very high recognition rate. As expected when camera motion is involved in SC2 our recognition rate drops, and for SC3 where actors carry bags or wear baggy clothes, recognition rate is not as high as in SC1 and SC4. Table 4.5 shows our results for KTH dataset's all shooting conditions with other studies'. Although our results are slightly lower than compared studies, we want to note that we use a simpler classification scheme.

Table 4.6 presents a comparison with other studies, our **OF + LCS** method is the fifth successful one on this list. This is a remarkable result since our method is simple, yet captures the essence of the actions and classifies them successfully. Using a shape descriptor with KTH dataset is a problematic issue due to the dataset's changing conditions. The results for shape features will get better if they're provided with more accurate silhouettes and contours.

Matching Method	Success Rate
Ikizler[18]	94%
Wang[36]	92.43%
JHuang [21]	91.7%
Wong [37]	91.6%
Our Approach	**89.25%**
Nowozin[28]	84.72%
Niebles[27]	81.5%
Dollar[12]	81.2%
Schuldt[31]	71.7%

Table 4.6: Comparison with related studies (KTH Dataset)

4.4 Experimental Discussion

Throughout our studies we were compelled with various challenges of action recognition. Our experiments shed light on many issues regarding our approach. The major conclusion we drew from our results is action recognition does not have to rely on complex representations and classification techniques. Simple methods may outperform more complex methods in terms of both success and performance. This is also validated with experiments. We think that an action recognition system should respect the sequence structure, ignoring temporal characteristics of an action usually leads to misclassification of similar actions, which may be distinguished by their temporal property. The first part of this discussion is concerned with how well pose descriptors worked with the datasets we used, the second part discusses only pose descriptor in terms of their strong and weak points and the last part discusses the pros and cons of the matching methods used.

In our study, pose descriptors had major importance. We experimented with four features; Histogram of Oriented Gradients (HOG), k-Adjacent Segments(kAS), Shape Context and Optical Flow histograms. We had the highest results with shape context and LCS combination on Weizmann dataset and optical flow histograms and LCS combination on KTH dataset.

On Weizmann dataset our idea of representing poses in terms of their shapes

proved to be very successful. With HOG descriptor our results were very successful, which encoured us to move on to a more descriptive shape feature, shape context. This feature gave us 100% recognition rate on this particular dataset.

HOG feature performed very well for Weizmann dataset, which is a homogenous dataset with no illumination changes. Also the creators of this dataset provide us with the silhouettes, which make it trivial to find tight bounding boxes. This is not the case for KTH dataset. For this dataset we didn't have tight bounding boxes. Also HOG is computed over gray-level images, in KTH dataset some of the actors tops or bottoms were nearly identical in color with the background. This causes confusion in the following nature, when a person is jogging, his hands move forward and backwards but still close to his body. If the legs of a jogging person could not be described by the feature accurately, then this action may seem like boxing. Also if we take a look at clusters when HOG is the pose descriptor, we see that clusters mainly consist of very similar frames, usually frames from the same video. This causes every video to be coded with very different codewords, even if they include the same action. Mainly, HOG descriptor was not descriptive enough to stress subtle differences in very similar poses of KTH dataset.

Encouraged by the successful results we got from Weizmann dataset with shape context feature, we also applied this to KTH datset. But unfortunately we didn't have clear silhouettes, which affected the performance for shape context. Also the bounding box information we used to crop the person form the video was not accurate, sometimes cropping the bottom half of the person. This also posed a disadvantage for us since we mainly tested with shape features. Due to poor performance of shape context, we employed another shape descriptor, k-Adjacent Segments (kAS). kAS feature is affected by people whose clothes' colors are very similar to the background, in those cases Berkeley's Edge Detector sometimes failed to find the edges for that part of the person, which caused inconsistent kAS information. For the KTH dataset the most successful feature was optical flow histograms, which is calculated in small windows so it is not affected by the fluctuations of shape. Its a fact that extracting the perfect silhouette or finding the tightest bounding box is not always possible. In this case we need a robust

descriptor, which will endure these imperfections. Most of the shape descriptors rely solely on extracting a good contour of the shape, which is clearly not always the case.

At this point we employed optical flow histograms, which contain both the spatial and temporal information. This property makes this descriptor robust and very powerful in terms of representation. With this descriptor we have achieved 89% recognition rate for the whole KTH dataset.

During our experiments we examined the weak and strong points of the shape descriptors we have used. HOG is a compact descriptor yet it preserves shape data very roughly, which causes low success rates in complex contditions. Shape Context is a silhouette based method, to extract the best information, we need an accurate contour. Shape context performed very well with Weizmann dataset, where clean silhouettes were available. Unfortunately for KTH dataset this was not available. Another disadvantage of shape context is comparing two shape contexts takes too much time, which contradicts with our initial idea of having a simple and efficient action recognition system. kAS feature provides very good results for the object recognition area. So we included kAS as a pose descriptor, to see the how it will affect the results. We applied two types of similarity measures for kAS which are explained in Section 3.2.3. kAS' accuracy and success is based solely on the goodness of edge detection. Although the edge detector authors included with kAS detector is a successful one due to clothing colors or illumination changes, some parts of the contour of the person has a lower strength than other parts, which effects the kAS detection. This poses a problem since we apply tresholding to kAS to decrease the numbers and eliminate background segments; which sometimes eliminates body segments as well. But if we don't apply tresholding then we have to match twice as much kAS, which has very large computational overhead. Another important reason why kAS was less successful is because human body is too articulated for this feature. There are many poses in an action sequence, and considering different kind of forms actions are performed by different people, adds up to a very versatile object dataset in this case.

Another factor that has a significant effect on the recognition rate is k-medoids

clustering algorithm. This algorithm chooses a random initial set and since partitioning around medoid is an NP-hard problem, it tries to find an approximate lowest cost. But with a bad initial set, the clusters are not evenly weighted, which causes an unhealthy representation of the action sequences. For Weizmann dataset we were able to manually select a number of frames, which provided a good initial set for the algorithm, but since KTH is a very large dataset we couldn't specify a balanced initial set.

During our experiments on each dataset we saw that we get better results with pose-sentences approach than bag-of-poses. When we look at the actions which are misclassified in both approaches, we saw that bag-of-words representation particularly confuses actions *walk* and *run*. This is because we don't include temporal information when we're deciding on the action label. Since the pose information of these actions is very similar, the distribution of poses do not differ enough to distinguish them from each other. Weizmann dataset provides a good example of how temporal characteristics help us to make a better decision. Shape context, which is more descriptive shape feature than HOG, gave us the best results on this dataset. This supports our claim on shape of the pose being a very powerful source of information in terms of describing an action and when a good pose descriptor is used with a system which also uses temporal characteristics of an action for discrimination results are very satisfactory.

As mentioned before we used three different algorithms to compare action sequences. The advantages of string matching approaches over bag-of-poses are discussed, but we should also discuss which string matching algorithm is superior. Edit distance is one of the most simple string matching algorithms in the literature. It strictly uses sequence information meaning even if two pose-sentences are very similar, it does not oversee a few pose-words which may be coded differently and considers them as discontinuities distancing the compared sequences. On the other hand Longest Common Subsequence(LCS) hops over these faulty coded poses and continues with the next matching pose-word, adding it to the common subsequence list. What may seem as LCS' flexibility also fails this algorithm in some cases. Because it may find that a *jump forward* action may have a longer subsequence with a *run* action than another jump forward action, which

is a misclassification.

Experimental results presented in this thesis show that pose is a powerful primitive for actions and temporal information is a crucial part of action recognition. Further conclusions we drew from our studies will be discussed in Chaper 5.

Chapter 5

Summary and Conclusions

In this chapter we will first summarize our contributions and then state the conclusion we have drawn from our study.

5.1 Summary of Contributions

In this thesis we presented another aspect on how to understand human motion in videos. We proposed a novel approach for representing actions, pose sentences. This representation is beneficial in various angles, first of all it provides a compact representation of data, and secondly it makes use of sequential information which preserves the temporal characteristics of actions.

Our other main contribution is the use of pose information. We utilize shape and shape transition information of the pose as action primitives. Through our results we have shown that pose information is indeed very appropriate for representing actions.

Overall we have presented an action recognition system, which offers a novel and simple action representation scheme. We have reached a 100% percent success rate on Weizmann dataset and 89% success rate on KTH dataset. Our results

show that for successful action recognition complex systems are not needed, simple yet effective techniques can give the same recognition results also.

5.2 Discussion Summary

While constructing our system, we have been challenged by various problems. Some points regarding these and generally action recognition are:

- Bag-of-words representations are not sufficient for representing action sequences, they lose temporal information.
- Although pose is a very good primitive for actions, it is not easy to extract pose information.
- Pose transition information is as important as pose shape information.
- HOG is a fair pose descriptor, although when shooting conditions get more complex it fails to extract pose information effectively.
- kAS descriptor, despite being successful in object recognition, did not match our expectations. This descriptor works better with objects which always have the same shape, human body is too articulated for kAS to capture characteristics of poses fully.
- Shape Context is a powerful shape descriptor if the data is very noise free. Since it solely relies on good contour data, this feature may not be very desirable with cases where extracting a good contour is not possible.
- Optical flow descriptor is successful at capturing pose transitions and robust to shooting conditions.

5.3 Future Work

The study we presented in this thesis is the starting point of our approach. In the future we plan to bring the following improvements:

- Using more complex classification schemes is very important. We mainly focused on the representation aspect of the system and used a very simple classification technique. We plan to utilize SVM as a starting point.
- The pose descriptors we used in this thesis gave us a great insight on the weak and strong points of shape features. Using this knowledge we plan to propose a new descriptor.
- In this thesis we performed single action recognition, in the future we will improve our system for recognizing composite actions.
- We plan to utilize substring matching approaches to extend our system to a searching mechanism which takes a video as a query and brings up the similar ones.
- Hidden Markov Models or Dynamic Time Warping can also be an option while examining sequence information, in the future we plan to exploit this technique for classification.

Bibliography

[1] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, pages 831–837, 2000.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.

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

[4] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1325–1337, 1997.

[5] A. F. Bobick and A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(12):1325–1337, 1997.

[6] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via plsa. In *Proceedings of the European Conference on Computer Vision*, 2006.

[7] C. Bregler. Learning and recognizing human dynamics in video sequences. *cvpr*, 00:568, 1997.

[8] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *ICCV*, pages 624–630, 1995.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[10] C. Dance, J. Willamowski, L. Fan, C. Bray, and G. Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[11] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Mach. Learn.*, 46(1-3):225–254, 2002.

[12] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[13] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):36–51, 2008.

[14] P. Fihl, M. Holte, and T. Moeslund. Motion primitives for action recognition. In *Workshop on Gesture in Human-Computer Interaction and Simulation*, 2007.

[15] D. Forsyth, O.Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion i: Tracking and animation. *Foundations and Trends in Computer Graphics and Vision*, 1(2/3), 2006.

[16] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Commun. ACM*, 18(6):341–343, 1975.

[17] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3), 2004.

[18] N. Ikizler, R. G. Cinbis, and P. Duygulu. Human action recognition with line and flow histograms. *ICPR*, 2008.

[19] N. Ikizler and P. Duygulu. Human action recognition using distribution of oriented rectanguar patches. In *Human Motion Workshop, (with ICCV)*, 2007.

[20] O. Jenkins and M. Mataric. Deriving action and behavior primitives from human motion data. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[21] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8, 2007.

[22] V. Kruger, D. Kragic, A. Ude, and C. Geib. The meaning of action: A review on action recognition and mapping. *Advanced Robotics*, 21(13):1473–1501, 2007.

[23] I. Laptev and T. Lindeberg. Space-time interest points, 2003.

[24] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, February 1966.

[25] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, 2004.

[26] G. Mori, A. Efros, A. Berg, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[27] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[28] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV 2007)*, pages 1919–1923. IEEE Computer Society, 10 2007.

[29] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440, 2004.

[30] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *Int. J. Comput. Vision*, 50(2):203–226, 2002.

[31] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[32] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV 2005)*, October 2005.

[33] E. B. Sudderth, A. B. Torralba, W. T. Freeman, and A. S. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, pages 1331–1338, 2005.

[34] C. Thurau. Behavior histograms for action recognition and human detection. In *Human Motion Workshop, (with ICCV)*, 2007.

[35] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *Int. J. Comput. Vision*, 72(2):133–157, 2007.

[36] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Human Motion Workshop, (with ICCV)*, 2007.

[37] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, June 2007.