HUMAN BEHAVIOR UNDERSTANDING USING VIDEO ANALYSIS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY
CELAL ONUR GÖKÇE

IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MARCH 2016

Approval of the thesis:

HUMAN BEHAVIOR UNDERSTANDING USING VIDEO ANALYSIS

submitted by **CELAL ONUR GÖKÇE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Director, Graduate School of Natural and Applied Sciences

Prof. Dr. Gönül Turhan Sayan _____
Head of Department, Electrical and Electronics Engineering

Prof. Dr. Uğur Halıcı _____
Supervisor, Electrical and Electronics Engineering Dept., METU

Examining Committee Members

Prof. Dr. Uğur Halıcı _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Gözde Bozdağı Akar _____
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Ferda Nur Alpaslan _____
Computer Engineering Dept., METU

Prof. Dr. Hakkı Gökhan İlk _____
Electrical and Electronics Engineering Dept., Ankara University

Assist. Prof. Dr. Tolga İnan _____
Electrical and Electronics Engineering Dept., TED University

Date: 07-March-2016

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : CELAL ONUR GÖKÇE

Signature :

ABSTRACT

HUMAN BEHAVIOR UNDERSTANDING USING VIDEO ANALYSIS

Gökçe, Celal Onur

Ph.D.,Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Uğur Halıcı

March 2016, 88 pages

In this study we proposed a new hierarchical architecture for solution of human behavior understanding problem. A new dataset, namely football video game (FVG) dataset, is generated which involves activities more complex than any other dataset present in the literature. Football ball is detected using multilayer neural networks trained with gradient descent algorithm and enhanced using learning with queries method and it is tracked using growing window algorithm. After region of interest is extracted around ball detected and tracked, primitive action is recognized using one of three types of approaches. First one is based on the well-known Dollar et.al. cuboid features.The second one is mixture of poses approach proposed in this study. Third is an extension to mixture of poses, where fisher vector is employed for the representation of mixture of poses in vectorel form. Primitive action sequences found sequentially in this layer are fed to higher layer activity recognition layer. In the activity recognition layer one of two types of approaches are used. One is well known Hidden Markov Model (HMM), known to work well on time series data and the other is Context Free Grammar (CFG) which can theoretically recognize more complex sequences that HMM can not. Another novelty of this study is new activity types are learnt using grammar induction with the Cook, Yunger and Kasami (CYK) algorithm. So, this way either pre-taught activity types can be recognized or new activity types can be learnt. The FVG dataset that we generated is tested with four combinations of approaches and encouraging results are obtained. For primitive action recognition, the proposed MoP algorithm has success rate of 69.3%, clearly exceeding widely referenced Dollar et. al. cuboid features which achieves success rate of 54.0%. Employing

Fisher vector with MoP slightly decreases performance to 67.0 % while achieving high speed. For complex activity recognition, MoP-CFG pair achieves success rate of 62.5%, same with MoP-HMM pair. They clearly exceed cuboid-CFG/HMM pairs which achieves success rate of 42.5%.There are activities those can be recognized by CFG but not by HMM. An example for these is passing the ball around activity.

Keywords: Human behavior understanding, context-free grammar, primitive action recognition, complex activity recognition, mixture of poses, football video game dataset.

ÖZ

VİDEO ANALİZİ İLE İNSAN DAVRANIŞI ANLAMA

Gökçe, Celal Onur

Doktora, Elektrik ve Elektronik Mühendisliği Bölümü

Tez yöneticisi: Prof. Dr. Uğur Halıcı

Mart, 2016 88 sayfa

Bu çalışmada insan davranışı anlama probleminin çözümü için yeni bir hierarşik yapı önerilmiştir. Futbol video oyunu (FVO) veriseti isimli yeni bir veriseti üretilmiştir. Bu veriseti literatürdeki verisetlerinden daha karmaşık davranışları içermektedir. Futbol topu bayır inişi metoduyla eğitilmiş ve sorgularla öğrenme tekniğiyle geliştirilmiş çok katmanlı yapay sinir ağı ile tespit edilmekte ve büyüyen pencere algoritması ile takip edilmektedir. Topun etrafındaki ilgili alan bu yöntemle çıkarıldıktan sonra atomik aksiyon iki yöntemden birisiyle tanınmaktadır. Birinci yöntem literatürde sıkça atıf alan Dollar ve arkadaşlarının küboid özniteliklerine dayanmaktadır. İkinci yöntem ise bu çalışmada önerilmiş olan pozların karışımı yaklaşımıdır. Pozların karışımının bir uzantısı olarak Fisher vektörleri Pozların karışımını vektörel formda temsil edebilmek üzere kullanılmıştır. Bu katmanda tanınan atomik aksiyonlar serisi bir üst katmana girdi olarak beslenmektedir. Bir üst katmanda karmaşık aktivite tanıması yapılmaktadır. Aktivite tanıma katmanında iki yaklaşımdan birisi kullanılmaktadır. Birinci yaklaşım literatürde sıkça kullanılan Gizli Markov Modeli (GMM), ikinci yaklaşım ise teorik olarak GMM'nin tanıyamadığı serileri tanıyabilen İçerikten Bağımsız Gramer'dir (İBG). Bu çalışmanın bir özgünlüğü de yeni aktivite türlerinin Cook, Yunger and Kasami (CYK) algoritması kullanılarak gramer tümevarım yöntemiyle öğrenilmesidir. Bu şekilde hem daha önce sisteme tanıtılmış olan aktiviteler tanınabilmekte hem de yeni gelen aktivite türleri öğrenilebilmektedir. Bu çalışmada özgün olarak üretilmiş FVO veriseti bu dört kombinasyon ile test edilmiş ve sonuçlar verilmiştir. Basit aksiyon tanıma için önerilen MoP algoritması %70 başarı sağlayarak %53,95 başarı

sağlayan Dollar ve arakadaşlarının küboid öznitelikler algoritmasını açık ara geride bırakmıştır. MoP'la beraber Fisher vektörü kullaımı performansı %67'ye düşürse de hız açısından büyük kazanç sağlanmaktadır. Kompleks aktivite tanımada MoP-CFG çifti, MoP-HMM çifti gibi %62.5 başarı sağlayarak küboid-CFG/HMM çiftlerini açık ara geride bırakmışlardır. CFG'nin tanıyıp HMM'in tanıyamadığı aktiviteler vardır. Bunlara bir örnek top çevirme aktivitesidir.

Anahtar kelimler: İnsan davranışı anlama, içerikten bağımsız gramer, aktivite tanıma, aksiyon tanıma, pozların karışımı, futbol oyunu veriseti.

to my wife and children

ACKNOWLEDGEMENTS

The author wishes to express his deepest gratitude to his supervisor Prof. Dr. Uğur Halıcı for her guidance, advice, criticism, encouragements and insight throughout the research.

The author would also like to thank İlknur Gökçe, Naciye Gökçe, Halim Sözbilir, Barış Gökçe, Muzaffer Yiğit and Nadire Yiğit for their patience, support and goodwill.

The technical critisism of of juri members Prof. Dr. Gözde Bozdağı Akar, Prof. Dr. Ferda Nur Alpaslan, Prof. Dr. Gökhan İlk and Assist. Prof. Dr. Tolga İnan are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table 5.10 Confusion matrix for whole system with HMM in activity recognition layer and manual feed in atomic action recognition layer

CHAPTER 1

INTRODUCTION

Given a video containing human/humans performing some action/actions, how to find the class of actions performed using computer automatically? This is human behavior understanding problem and has attracted several computer vision researchers around the world in the last two decades.

1.1 Motivation

The problem may be attacked with several approaches which may be classified variously. One taxonomy of approaches may be done considering the features used in recognition. The features may be broadly classified into two as shape and motion features. Shape features are global or local shapes of body or body parts in each frame. This may be seen as information in frames independent of each other. On the other hand motion features encode some kind of information about difference between frames. These may be based on either optical flow or some kind of spatio-temporal filter applied to whole or parts of video.

A comprehensive survey of human behavior understanding literature is given in [1]. In İkizler and Duygulu [2], histogram of oriented rectangular patch features are fed to several classifiers. In Bobick and Davis [3], background subtracted images are aggregated to motion energy and motion history images where Hu moments are computed as feature descriptors and Mahalanobis distance is used as classifier. In Yilmaz and Shah [4], 2D tracked object contours are stacked to form 3D spatio-temporal (ST) object of action and Gaussian and mean curvature of the surfaces of ST shapes are used to categorize their surface type. Types of surfaces of the 3D ST

shapes are used as action descriptors. In Gorelick et.al. [5], 3D ST shape is formed by stacking foreground frames and using solution of Poisson equation several features are extracted. In Veeraraghavan et.al. [6], a sparse statistical shape descriptor invariant to scale and translation is used to extract features from binarized silhouette. Autoregressive model and DTW are used as classifiers.

In Kepenekçi [7], 3D Gabor filters of various scale and orientation are used as features. A variation of Hidden Markov Model (HMM) namely Profile HMM is used as the classifier to recognize the behavior. In Schuldt et.al. [8], ST jets of order four are used as feature descriptor. Two types of classifiers are tested, nearest neighbor classifier (NNC) and support vector machines (SVM). In Dollár et.al. [9], a new ST interest point detector with 1D Gabor filter for temporal dimension is proposed. As feature descriptor, several are tested and concatenated vector of brightness gradient was found to be most successful. In Niebles et.al. [10], a codebook of Dollar et. al.'s cuboids is formed from samples using k-means clustering with Euclidean distance metric. Two types of generative classifiers are used: probabilistic Latent Semantic Analysis and Latent Dirichlet Allocation. In Niebles and Fei-Fei [11], a hierarchical model is used as a constellation of parts approach. For learning the parameters of the hierarchical model Expectation Maximization (EM) algorithm is used. Note that the details of each algorithm summarized here are given in Chapter 2 in details. In [55], an easy-to-implement representation for human actions based on skeletal joint selection was proposed. The actions were represented as sequence of most informative joints. Selection of joints was based on interpretable measures like mean or variance of joint angle trajectories. In [56], an action descriptor based on differences of skeletal joints was proposed. An informative frame selection scheme was used to reduce computational cost. A non-parametric Bayes Classifier is used for classification. In [57], a representation of interaction of a subset of human joints was used as a model. A 3D appearance feature that describes the depth appearance near a joint was proposed. Additionally, a discriminative temporal representation robust to noise was used.

1.2. Contribution of the thesis

Human actions range from simple breathing to the whole history of humanity. Most of the studies in literature are based on simple human actions like walking, running, handwaving, etc. In order to work on more complex human activities, football game is chosen in this study. A new dataset, namely football video game (FVG) dataset is constructed. The FVG dataset contains activities that are more complex than any existing in the literature. Details of the dataset are given in Chapter 3.

Recognizing more complex activities require a different approach from recognizing simple actions. For this reason, a novel hierarchical architecture for complex activity recognition is proposed in this thesis. There are two novelties in the architecture proposed. One is for recognizing primitive actions, a novel algorithm, namely Mixture of Poses (MoP) algorithm, is proposed in the middle layer of the architecture. Second novelty is, for recognizing complex activities grammar induction is employed. Each of these algorithms are explained in Chapter 4 in detail.

1.3. Organization of the thesis

In Chapter 1, a brief introduction to the study is made.

In Chapter 2, literature survey and background information is given.

In Chapter 3, newly constructed dataset is explained in detail.

In Chapter 4, proposed architecture and each layer of the architecture is explained in detail.

In Chapter 5, experimental results are given.

In Chapter 6, conclusion is made and some possible future work is mentioned.

CHAPTER 2

LITERATURE SURVEY AND BACKGROUND

In this chapter, literature survey and some background information is given. Literature survey is given in section 2.1. Background information is given in 2.2.

2.1 Literature Survey

A good classification of approaches to human behavior understanding is given in [1] and shown in Figure 2.1 below.

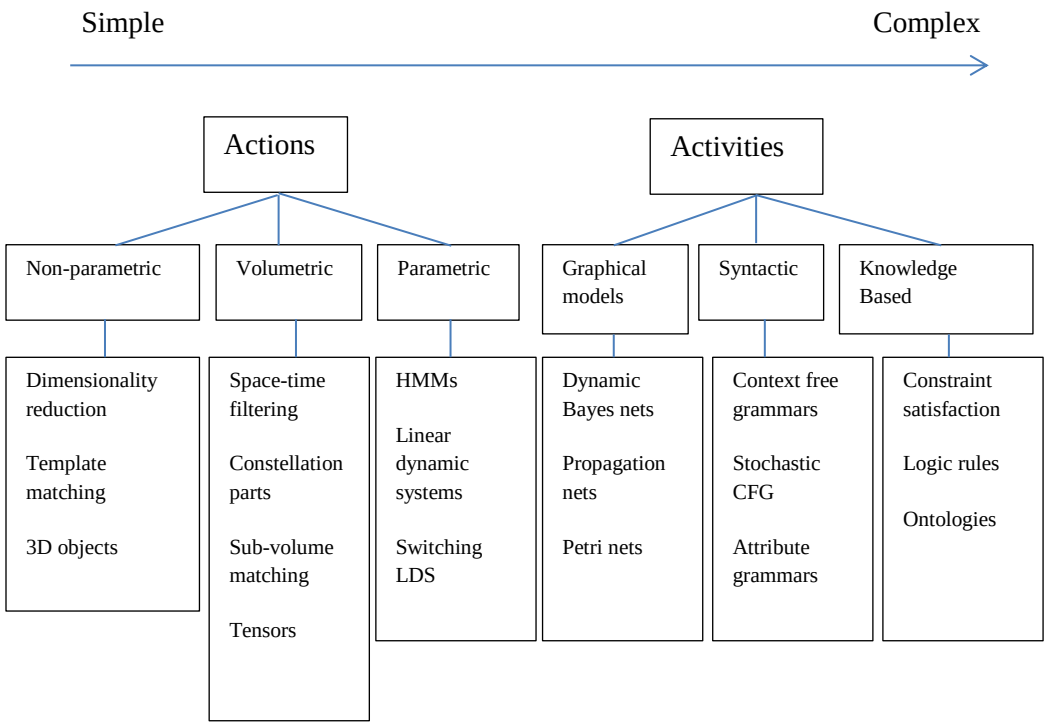


Figure 2.1: Classification of approaches to human behavior understanding [1]

Here the behavior understanding problem is divided into two broad categories, namely action recognition and activity recognition. Actions are defined as relatively short and simple movements of single human while activities are long, complex movements composed of several actions and possibly involve more than one human.

This section on literature survey lists some of the work done in literature with subsections divided according to the categorization above.

2.1.1) Solutions to action recognition problem

Solutions to action recognition problem is categorized into three main categories, namely non-parametric, volumetric and parametric approaches.

2.1.1.1 Non-parametric approach

In non-parametric approach, input video is considered as a set of frames and features from each frame are extracted and matched to a learned template. Non-parametric approaches are further divided into three subcategories: 2D template matching, 3D object based and dimensionality reduction based approaches.

2.1.1.1.1 2D template matching

In this approach no 3D structure is considered.
Polana & Nelson [12] used simple motion features for detecting periodic activities. As detector they used low level motion information for tracking and keeping the human in the center of image frame. This was refined using a periodicity analysis. After this analysis a normalized spatio-temporal (ST) volume containing the periodic action to be classified was found and used to compute the features for classification. Motion field was computed between successive frames of the volume and used to build feature vector. The normalized ST volume is divided into 4x4x6 cells, in x, y and t dimensions, respectively. Three types of local statistics for motion is computed for each cell: First, using only flow magnitude, second dominant motion direction and third summed motion magnitude in the dominant

motion direction. This is used as a feature vector. As classifier simple nearest centroid method is used to show the effectiveness of the features. Experiments are done on seven types of activities: walk, exercise, jump, swing, run, ski and swimming frog. Dataset was formed by the authors themselves. Without any motion clutter 100% classification rate is reported with the 3rd feature vector type. It is also reported that motion clutter up to one half of the activity motion magnitude can be compensated.

In Bobick and Davis [3], background subtracted images are used to detect the regions of motion. Thresholding results in binary images of motion where pixels of motion are one and static pixels are zero. Two types of aggregated images are used for recognition. First type, namely motion energy image (MEI), is the simple aggregation of motion images. In MEI, pixels having motion are one while static pixels are zero. Second type, motion history image (MHI) gives more weight to more recent frames. This results in a gray image with brighter pixels of recent motion. In order to have view invariance Hu moments of the MEI and MHI images are extracted as features. As a classifier simple Mahalanobis distance is used to measure the similarity of test samples to stored templates. System is tested on an aerobics dataset consisting of 18 aerobic exercises and 67% success is reported.

In Thurau and Hlavac [13], they used posed based action recognition. As basic descriptor they used Histogram of Oriented Gradient (HOG) descriptor of [14]. They extended this descriptor to cope with problems in pose matching. They used non-negative matrix factorization (NMF) basis representation of HOG descriptors. By applying NMF to training set of poses and background independently, they find parameters <Wpose,Hpose>, and <Wbg,Hbg>. They claim that around 40-80 basis vectors are sufficient for pose. Reconstruction is of the form

V=[Wpose Wbg][Hpose Hbg]

For estimating pose from a new image V^{new} they find H^{new} corresponding to $W = [W_{pose}\ W_{bg}]$ which is in the form $H^{new} = [H_{pose}\ H_{bg}]$. H^{new}_{pose} is compared against

example poses in the training set (pose primitives) and matching is done with minimum Euclidean distance. Note that they cluster pose primitives using Euclidean distance and standard agglomerative clustering method of [15] during training and claim that 30-80 pose primitives are good enough. For classification they use KL divergence of histogram of poses. They also add different weightings to different poses since each pose has different information content: *"... a person in a simple upright position could indicate almost any behavior, where as we can immediately spot the distinctive pose if someone is waving his arm…".* Additionally they also used n-gram approach: grouping sequential poses $A = [a_1, a_2, \dots a_n]$ results in bi-gram of $A_{bi\text{-}gram} = [\{a_1a_2\},\{a_2a_3\},\dots,\{a_{n-1}a_n\}]$.

2.1.1.1.2 3D object models

This approach is based on 3D object recognition methods.

In Syeda-Mahmood et.al. [16] a simple moving object is assumed and thus background subtraction is found to be sufficient for detection and tracking. For each frame edges are detected with an edge detector and curves are formed from edges. The points having sharp change in curvature are decided as corners. These curves and corners are used to form an action cylinder, a 3D geometric object, which is proposed to be a discriminative representation for each action. Apart from action cylinder, salient corners are tracked with a corner detection algorithm as additional features. Comparison and matching of test cylinder and model cylinder are made on these corner features. It is shown that eight points are theoretically enough to compare two generalized cylinders. If the similarity is above a certain threshold recognition of the action is declared. Proposed solution is tested on various objects including articulated hand and recognition rate between 85% to 100% is reported.

In Yilmaz and Shah [4], 2D object contour tracking is used for detection and tracking. After contours of the tracked object are obtained for each frame in the video, correspondences between contour points are found with a graph theoretical approach. 3D spatio-temporal object of action is formed by stacking the contours. Action descriptors are extracted using differential geometry features from the 3D

ST shape. Gaussian and mean curvature of the surfaces of ST shape are used to categorize the surface type. These action descriptors are used as set of points, thought as interest points, to match test shape to model shape by least squares fitting to a homogeneous equation of homograhpy between set of points. The experimental results seem not detailed enough to compare the success of the approach to others.

In Gorelick et.al. [5], detection and tracking is done using median background subtraction and simple thresholding in color space. The resulting 2D shapes in each frame are stacked to form 3D spatiotemporal shape. Using solution of Poisson equation several features are extracted. One of the extracted features are space-time saliency which is high for fast moving parts. Spatio-temporal orientations and aspect ratio of different parts are also extracted. This is done by constructing a Hessian matrix of Poisson solution and solving for three eigenvalues of the matrix. By looking at the eigenvalues of the matrix it is easy to classify the part's shape into one of three forms:

$\lambda_1 \approx \lambda_2 >> \lambda_3$ corresponds to space-time stick shape, $\lambda_1 >> \lambda_2 \approx \lambda_3$ corresponds to space-time plate shape and $\lambda_1 \approx \lambda_2 \approx \lambda_3$ corresponds to space-time ball shape.

These features are mixed with a weighted integral into global feature form. Classification is done using simple nearest neighbor with Euclidean distance metric. The popular Weizmann dataset is collected in this work. Experiments are performed using this dataset. Upto 99.6% recognition rate is achieved.

2.1.1.1.3 Dimensionality reduction

This approach tries to find some discriminative subspace by manifold learning techniques.

In Yacoob and Black [17], an assumption of initial segmentation of body into parts is made and tracking is done using parametrized optical flow. A set of activity parameters from each frame, quantatively six vectors of temporal measurements, which were not stated clearly, mentioned as translation, rotation, etc, were concatenated to form initial feature vector. Principal Component Analysis (PCA)

was applied to this representation to find the informative subspace. For a new sequence, to find the components on the principal activity basis together with affine transformation coefficients to improve matching, an error minimization with a gradient descent scheme was used. Classification was done using a simple nearest neighbor (NN) with Euclidean distance metric. A recognition rate of 82% was reported on a dataset collected by the authors.

2.1.1.2 Volumetric approaches

Volumetric approaches consider the video as a 3D volume. They are subdivided into three categories: spatio-temporal filtering, part based approaches and sub-volume matching.

2.1.1.2.1 Spatio-temporal filtering

This approach process video with a large filter bank and the responses of the filters are used to derive discriminative features.

In Chomat and Crowley [18], a Gabor filter bank of four orientation and three motion scales were applied to each pixel of the spatio-temporal volume. In training, density estimation for each action was done using a large training set. An extension of quadtree representation was used to handle computation of histogram over that 12-d large space. For recognition, Bayes rule was used to vote for the action category by each pixel. The class decision was done by simply averaging the vote of each pixel.

2.1.1.2.2 Part-based approaches

These approaches consider video as a collection of local parts.
In Schuldt et.al. [8], 3D-Harris-like corner detector was used for interest point detection. By finding the local maxima of a response function involving second gradient of spatio-temporal volume, corner points with movements of high inflection were detected and used as interest points. As feature descriptor spatio-temporal jets of order four were used. Histograms of these descriptors were also

used as an alternative feature descriptor but it was reported that they give better results in their raw form. Two types of classifiers were tested, nearest neighbour classifier (NNC) and support vector machines (SVM). SVM was reported to give higher level of success in classification. A popular dataset, namely KTH dataset, was introduced in this paper and used to demonstrate the success of the approach. It was reported that recognition rate of 72% is achieved.

In Dollár et.al. [9], a new ST interest point detector was proposed. It consisted of two parts: Spatial part sensitive to variations in local image intensities and a temporal 1D Gabor filter part having high response to complex motion especially if periodic. The local maxima of the response function composed of convolution of video with smoothing spatial Gaussian and temporal Gabor was used to detect interest points. Around interest points a feature descriptor was computed in a cuboid (Fig2.2). Note that this is not the final behaviour descriptor to be used in behaviour recognition but local feature descriptor to categorize the cuboid around interest point. As feature descriptor several were tested and concatenated vector of brightness gradient was found to be most successful. A library of cuboid prototypes was learnt using k-means algorithm. For a test video the cuboid features were extracted and a histogram of cuboid prototypes was built as a feature vector.

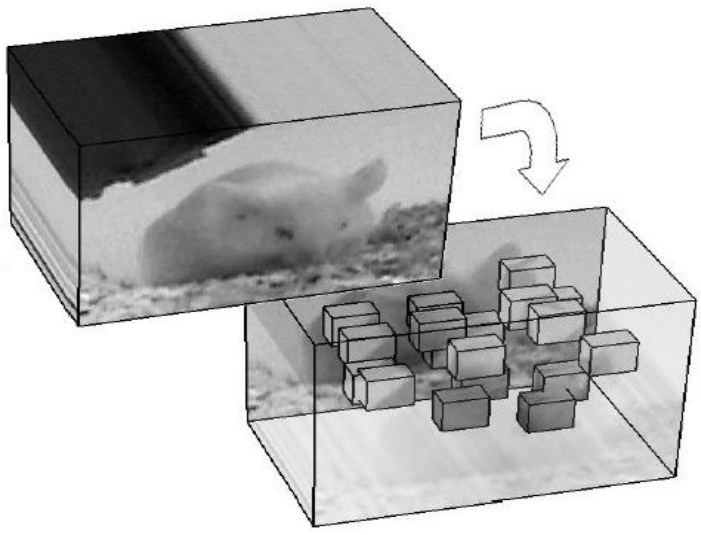


Figure 2.2 Visualization of cuboid based behavior recognition [9]

For recognition two types of classifiers were used: NN with Euclidean distance and SVM. A recognition rate of over 80% was achieved on KTH dataset.

Niebles et.al. [10] were inspired from solutions of latent topic discovery problem in text classification literature and they formed analogy between two problems by drawing similarity lines between spatio-temporal words and text words and between mixture of actions and mixture of topics.

At feature extraction stage, they used the same framework of Dollar et al. First they determined spatio-temporal interest points by using maxima of linear separable filters and then computed feature descriptors inside a cuboid centered around these points using concatenated vector of brightness gradient. Afterwards PCA was used to reduce dimension. An unsupervised learning framework was proposed. In the learning stage a codebook was formed from samples using k-means clustering with Euclidean distance metric. Two types of generative classifiers were used: probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). The parameters of pLSA model was learnt using an EM procedure. In classification using pLSA mixing coefficients of posterior was found using EM algorithm by minimizing KL divergence between measured empirical distribution and distribution computed by generative model. In LDA model a variational inference approach was used as an approximation. KTH, Weizmann and skating datasets were used to demonstrate the approach. Recognition rates of 83.3%, 90% and 80.7% were reported for the datasets, respectively.

In Savarese et al. [19], Niebles et al.'s approach of modified Dollar et.al.'s features was used to extract spatio-temporal features. A data structure called ST-colleraton was used to represent the number of co-occurances of label pairs in ST kernels of various sizes. After forming vocabulary of ST words with K-means clustering, spatio-temporal correlations between ST pairs were formed using K-means clustering of ST-correlaton vectors. These collegram words were used together with normal video words in learning and classification. The model used was pLSA similar to Niebles et al.'s approach and was trained using EM algrotihm as well.

The difference from Niebles et al.'s work lied in the extra type of words used, words corresponding to ST-colletan vector labels. The proposed approach was tested on KTH dataset and a recognition accuracy of 86.8% was reported.

In Nowozin et al. [20], Niebles et al.'s approach of modified Dollar et al.'s features was used to extract spatio-temporal features. Different from Niebles et al.'s approach, information of temporal ordering of features was also used. This was done using a temporal binning scheme with overlapping windows. Linear Programming Boosting (LPBoost) algorithm was used as a classifier which uses linear combination of hypothesis functions as weak classifiers. The parameters of the classifier were found using constraint generation technique. KTH dataset was used to demonstrate the success of the approach and a recognition rate of 84.7% was reported.

In Wong et al. [21], for feature extraction the same approach of Dollar et al.'s seminal paper was used: Maxima of separable linear filters for interest point detection and concatenation of brightness gradient in x,y and t dimensions. The paper extended parts based approach based on generative pLSA models by introducing an extra geometric information: centroids of parts. Inspired by a work on object recognition which proposed to use an implicit shape model (ISM) to infer object location in static images, the graphical model of pLSA was extended. EM was used to infer the pLSA model parameters. These two generative models were used in the recognition and localization of the activity. KTH, facial expression and hand gesture datasets were used to demonstrate the approach and recognition rates of 83.9%, 83.3% and 91.9% were reported, respectively.

In Song et al. [22], Lucas-Tomasi-Kanade corner features were detected and tracked. These features were used to find position and velocity of human body parts. A decomposable triangulated graph model was used to represent human body where each node represents a part of the body and vertices represent conditional dependencies. An EM-like algorithm was used to find optimal graph from unlabeled training data. Structure of the solution enabled to deal with occlusion.

In Niebles and Fei-Fei [11], a hierarchical model was used as a constellation of parts approach.

They used a hybrid of two types of features. First type was spatial shape features, which were extracted using Canny edge detector and a shape descriptor with 3 spatial and 8 angular bins was used. Second type was motion features of Dollar et al. Interest points were found using linear separable filters and feature descriptors were constructed using concatenation of brightness gradients on space and time dimensions. The dimension of both feature types were reduced using PCA.

For learning the parameters of the hiearchical model EM algorithm was used. Likelihood of data being generated from each of C classes was computed from the generative models learnt and a feature vector of dimension C was formed from these likelihoods. This feature vector was used to train a discriminative SVM classifier using a validation data set. Weizmann dataset was used to test the approach and a recognition rate of 72.8% is achieved.

In İkizler and Duygulu [2], they used rectangular patch features extracted using zero padded rectangular Gaussian filter convolution. They used filters of different scales, positions and orientations. After extracting rectangles (of about 1000 for each frame) they used histograms of oriented rectangles as pose descriptors. They used 12 orientation bins of 15 degrees and location grid of 3x3 for best performance. As recognizer they used 4 methods. Simplest was frame-by-frame voting. Each frame was assigned to a trained example using min X^2 distance. Frame-by-frame voting was done afterwards. Second method was similar but done on global temporal scale. All frames' histograms were summed to a single global histogram and X^2 distance was used as classification. As a third classifier they used SVM. Fourth classifier was Dynamic Time Warping (DTW). Since different videos can have different temporal scales of activity, from fast to slow, and may also be nonlinear, DTW was used as activity speed normalization tool.

In Chen and Hauptmann [23], they proposed a new kind of feature inspired from the famous SIFT features. They found interest points in two steps. First, they found SIFT points as candidate interest points in all frames and then they added motion constraints to select the ones with sufficient amount of optical flow around them. As a descriptor they used a 256 dimensional vector: 128 from SIFT and another 128 from optical flow. In order to add some kind of structure constraint they used bigram model of video codewords together with bag-of-words representation. Bag-of-words video codebook was constructed using K-means clustering. Classification was done using SVM with X^2 kernel.

In Fathi and Mori [24], they used two-layered classification scheme and two types of features, one for each layer. In the low layer, they used motion features similar to [25]: first, they used Lucas and Kanade [26] algorithm to compute optical flow. Then, they obtained 5 non-negative channels and then blurred this with a Gaussian to reduce negative effects of noise and spatial shifts in figure centric volume. These features were computed at every point. They claimed that these features were not much better than random classification. So, they used AdaBoost to obtain midlevel features. Midlevel features were around a small spatio-temporal cuboid and each covered certain amount of low level features. For each midlevel cuboid each low level feature inside it was considered as a part of weak classifier. At each iteration of AdaBoost a new low level feature was added in the AdaBoost framework to form weak classifiers. After all T iterations, T weak classifiers were formed. For decision for cuboid, the outputs of weak classifiers were combined in a weighted way to establish a midlevel classifier. For each cuboid this kind of midlevel classifier was formed. Each of these midlevel classifiers had a confidence value apart from classification result. These values were defined as midlevel features. As a final classifier, similar AdaBoost framework above was used with previous midlevel classifier of each cuboid was now weak classifier of final AdaBoost. Final classifier was trained with AdaBoost framework. Note that this was binary classifier and to apply it to multiclass problem they used Hamming decoding scheme among others.

Ke et al. [27] used optical flow decomposed into horizontal and vertical components. They used two types of descriptors: one-box: sum of all pixels in feature volume, two-box: difference of sum. They used rectangular features similar to [28] and extended it to 3D volume. Smallest feature size of 4x4x4 with smallest search cube of 64x64x40 was used. They used integral video for computational efficiency. There were millions of features and feature selection was done by direct forward feature selection mechanism of [29]. To train a cascade classifier they used the cascaded training mechanism of [29] where misclassified samples of previous stage were added to training set. Decision output was majority vote of the ensemble. Each threshold was found independently by minimizing error rate of corresponding feature. For detection, a volume was scanned over all locations in space and time and scale was also changed. In true positives this resulted in multiple detections in neighbors and # of detections was used in classification. Likelihood of an event occurring to non-occurring was computed and used in classification. The entire sequence was classified by summing likelihood ratios that passed the threshold and choosing the action whose detector reported the highest sum.

2.1.1.2.3 Sub-volume matching

This approach is based on matching sub-volumes between a video and a template.
In Shechtman and Irani [30], correlations of space-time brightness gradients was used as a measure of similarity between a small query video and a target video being searched. Computations were done on small patches in both template query video and target video segment. Correlations were computed using a proposed continuous rank increase computation. For this purpose, eigenvalues of concatenated space-time gradient matrices of template patch and video patch were calculated. A small rank increase corresponded to more correlation while a high rank increase corresponded to less correlation between patches. Overall decision was given by averaging correlations of all spatio-temporal pixels within segment.

2.1.1.3 Parametric approaches

In this approach the structure of the classifier is usually preset but its parameters are learnt through statistical machine learning techniques. Variations of HMMs are the most popular of this approach.

In Kepenekçi [7], Gaussian mixture models were used for background separation. 3D Gabor filters of various scale and orientation were used as features. A genetic algorithm was used to select the most discriminative features from large feature set. Finally, a variation of Hidden Markov Model (HMM) namely Profile HMM was used as the classifier to recognize the behavior.

2.1.2 Solutions to activity recognition problem

Solutions to activity recognition problem is categorized into three main categories, namely graphical models, syntactic approaches and knowledge based approaches. In this study we are aiming to work mostly on syntactic approach so emphasis is given to this part.

2.1.2.1 Graphical models

Graphical models are based on some kind of graph with probabilistic links. Nodes of the graph correspond to random variables, hidden and observed, and edges correspond to dependency relations between these random variables. Difference from HMM is in the number of hidden variables. Graphical models encode dependencies between several hidden variables while in traditional HMM just one hidden variable is inferred. Dynamic belief networks, Petri nets and other forms of graphical models are studied [1].

In Wang and Mori [31], they used motion features of [25]. Using [26] they found optical flow and four nonnegative channels with Gaussian blurring. The comparison of two different frames' motion descriptors was done by normalized correlation. To construct codebook, they used k-medoid clustering. Codewords were defined as centers of obtained clusters. A new video was converted to bag-of-words

representation by replacing each frame by its corresponding codeword without temporal information. So, each video was $w = (w_1, w_2, \ldots, w_N)$, w_i: motion word representing i^{th} frame. As models they used 2 different probabilistic graphical models: 1) sLDA: semi-Latent Drichlet Allocation 2) sCTM: semi-latent Correlated Topic Model. Learning of α and β was done using (not variational EM method) but an easier method. First β was learnt by counting of each word appearing together with topic z_i: $\beta_i = n_{ij} / n_i$. Then α was estimated using Newton-Rhapsody iterations. It was claimed this method is much easier than training of original LDA and gives better recognition accuracy. β was estimated like in sLDA. μ and Σ were estimated using EM. For classification both sLDA and sCTM used variational inference.

In Suter and Wang [32], they used features based on silhouettes. After extracting silhouettes, they normalized and centered them. They used block sized features for computational efficiency. Each subblock was used to calculate normalized value as $N_i = b(i)/mv$ where b(i): # of foreground pixels, mv: max value. The resulting descriptor was a vector (row scan): $F_t = [N_1\ N_2\ \ldots\ N_{hxw}]^T$. They believed that the activities lie on nonlinear manifolds in subspace. So, they made a nonlinear dimensionality reduction with KPCA. As classifier they used Factorial CRF. They used Quasi-Newton optimization methods to train FCRF. For inference they used Viterbi decoding. They used two types of hidden variables: key poses and activities. To determine # of key poses they used MDL rule and to find key poses $KP = \{P_1, P_2, \ldots, P_K\}$ they used K-means clustering. To use long range dependencies (in temporal axis) they used a window parameter w designating number of past and future frames used in predicting current state. They used an adapted limited-memory BFGS optimizer to learn parameters and loopy belief propagation for approximate inference.

Mori and Greg [33] used motion features based on optical flow similar to [31] and [25]. There were two kinds of motion features: s_0: global motion feature of 4 channels for the whole frame I, and $s_1, s_2, \ldots, s_m$: local motion features of salient patches of frame I. So, the feature vector was $x=(s_0, s_1, \ldots, s_m)$. the model: each frame

was composed of salient patches $I = \{I_1, I_2, \ldots, I_n\}$ and each patch corresponded to a hidden part $h:\{z_1,z_2, \ldots, z_m\}$. z_i takes on of a possible part labels. Part label meant a certain motion pattern corresponding to certain actions: "moving down", "handwaving 1", … The model had a tree structure where each node corresponded to a z_i and each edge corresponded to a constraint between z_i and z_j. Structure was formed using minimum spanning tree algorithm over salient patches. Model had parameters $w = \{w_0, w_1, w_2, w_3\}$. w_0: root filter vector measuring compatibility of each class label and global motion feature. w_1: vector where each component measures compatibility of feature s_j and part label z_j. w_2: vector measuring compatibility of class label y and part label z_j. w_3: vector measuring compatibility of class label y and pair $<z_j,z_k>$. (x,y) was scored by $f_w(x,y)=max_h w^T \varphi(x,y,h)$. Once model parameters were learnt classification was achieved by first finding the best labeling of the hidden part for each action and then picking the action label with the highest score. Model parameters were learnt by optimization.

2.1.2.2 Syntactic approach

This approach is based on a grammar defining the structure of the activity. An analogy to natural language processing is made by modeling activities instead of sentences using production rules. In activity recognition systems short time actions are used instead of words used in natural language understanding systems [1].

A hierarchical architecture is used in this approach. In lower and mid levels statistical classifiers and HMMs are used to detect and recognize short and simple actions. These actions are used as terminal symbols in the grammar in the higher level. Either context-free grammar (CFG) or its stochastic version is used. This approach deals with the curse of dimensionality in long activities by dividing the activity into temporal sequence of short duration actions and parsing the structure using parsing algorithms.

In Ryoo and Aggarwal [34], a hierarchical architecture with four levels was proposed. The lowest level was the body-part extraction layer. Previously

developed framework of [35] was adopted here. Human body was divided into three parts, head, upper-body and lower-body and parameters for each part were extracted in terms of ellipses and convex hulls. These continuous parameters were converted to discrete values before feeding to upper layer, pose layer. In the pose layer, a static pose for each body part was inferred using Bayesian networks. Pose of each part was discrete taking one of small number of states. The whole pose of human was defined by the pose of each body part. The example given in the paper was a person standing still, facing left with arm fully raised and stretched. The pose variable corresponding to this pose was [head=left, upper-body=<high, stretched>, lower-body=<low, withdrawn>]. This process was done for each frame involving human and produces series of poses for upper layer, gesture layer. Gesture was defined as an elementary movement of a body part. So, they were relatively short in duration and simple like stretching hand for handshaking. To detect and recognize gestures HMMs were used. One HMM for each gesture was trained and an additional noise HMM was used to find gestures that are none of the predefined ones. Output of this gesture layer was several gestures with gesture class, start time, end time, performer and target of action. These parameters were used in higher level parsing mechanism in order to classify composite activities and interactions. To define gestures with these parameters interval temporal logic for actions of [36] was used. Gesture together with performer (agent) and target made an atomic action. At the highest level context-free grammar (CFG) was used to represent composite actions and interactions. A composite action was composed of several atomic actions together with possibly other composite actions. In CFG proposed, composite actions and interactions were non-terminals while atomic actions were terminals. CFG production rules for each composite action and interaction were defined manually, which was one drawback of the system. Composite actions and interactions were recognized by using a tree traversal algorithm with some modifications and constraints mentioned but not explained very explicitly. They tested the approach in dataset they collected. Although this approach seems quite successful it did not deal with the uncertainty in the low level detectors. A fault in

low levels was not considered in higher levels since CFGs require deterministic inputs and does not consider probability of error or noise in terminal symbols. This seems to be a good improvement over this approach as claimed in the conclusions as future work.

In Ivanov and Bobick [37], the vulnerability of statistical pattern recognizers to even a seemingly simple gesture was mentioned. It was claimed that a hand gesture of drawing square in the air results in two quite distinct patterns when performed in clockwise and counterclockwise. Especially, when the structure of the behavior is difficult to learn with statistical machine learning techniques but easy to define verbally by humans, syntactic techniques have advantage. In this case it was advantageous to divide the problem in two parts: recognition of action primitives and recognition of activity structure. A hierarchical architecture of two levels was used. In the lower level, HMMs trained for short sequences were used to detect action primitives. One HMM per atomic gesture was trained. HMMs recognized action primitives which corresponded to words of gesture vocabulary by using the algorithm in [38]. Outputs of HMMs were passed to higher level. Stochastic Context Free Grammar (SCFG) was used in higher level for activity representation. The difference of SCFG from CFG is in the production rule: each production rule has a probability associated with it. In order to deal with the uncertainty in the input symbols, a modified version of [39] which is based on [40] was used. They examined their method on three experiments. First was the "hand drawing square in air" example mentioned above. Second was recognizing complex gestures of a musical conductor. Third was a surveillance setting in a parking lot.

2.2 Background information

2.2.1 Context Free Grammars: Inference and induction

2.2.1.1 Context Free Grammar basics

A language is defined as a set of strings, where strings are formed by concatenating several symbols from an alphabet. Languages can be finite, i.e. number of strings

within an language can be finite, or infinite. Languages are classified in terms of their complexity within a Chomsky language hierarchy framework. Lowest in the Chomsky are regular languages, which can be generated by regular expressions and recognized by finite state machines (FSM), the simplest mathematical model of a computer.

The probabilistic version of FSMs are Hidden Markov Models (HMM), which are used extensively in pattern recognition. HMMs, by their capacity, are able to generate and recognize patterns only in the lowest level of Chomsky hierarchy.

Up in the next level of Chomsky hierarchy is context free languages (CFL). A CFL can be generated by context free grammar (CFG) and recognized by an extended version of FSM which is called push down automata (PDA). A PDA is a device similar to a FSM with an added data structure called stack.

A context free grammar (CFG) is a 4-tuple G = (N,T,P,S), [42], where:

N: a finite set of non-terminal symbols

T: a finite set of terminal symbols

P: a finite set of production rules of the form A->β, where A Є N, β Є (N U T)*

S: S Є N is the starting symbol.

CFGs are used to define most of programming languages and some part of natural languages. Designing a CFG is more intuitive for humans. Understanding a learnt recognizer is also more intuitive.

In theory, a PDA can be used to recognize a CFL. The problem with the PDA approach to language recognition is in the nondeterministic nature of PDAs. With their naïve forms, they cannot be converted to standard computer programs which are deterministic in nature. Although any nondeterministic FSM (NFSM) has a corresponding deterministic FSM (DFSM) and a standard procedure exists to convert NFSM to DFSM, this is not the case with PDAs. There may be languages

that can only be recognized with nondeterministic PDAs. So, other techniques then straight implementation of PDAs are used to recognize CFLs.

Parsing: Language recognition problem can be defined informally as deciding whether given a string is in the language or not. During recognition, it may also be possible to infer how a string is derived with the language rules. Inferring the derivation of the string is called parsing.

2.2.1.2 Cook, Younger and Kasami (CYK) Algorithm

There are several parsing algorithms for CFGs. The one used in this study is called CYK.

The CYK algorithm [43] used in this study uses a special form of CFG, namely revised Chomsky normal form, where all the production rules are of the form:

$$A \rightarrow \alpha_1\alpha_2 \ , \ \alpha_1,\alpha_2 \in (N \cup T)$$

The pseudo-code of the CYK algorithm [44] shown in Fig2.3.

```
function cyk(w:string, P:set of rules): Boolean;
% cyk(w,P) returns true, if S derives w.
begin
        var T: array [1..n, 1..n] of set of symbols;
        var i,j,k: integer
% Initialization of the array. It is assumed that w=a1a2…an.
        for i ← 1 until n-j+1 do
                begin
                        T[i,j] ← Ø;
                        for k ← 1 until j-1 do
                                T[i,j] ← T[i,j] U apply(P,T[i,k],T[i+k,j-k])
                end
        return (S ∈ T[1,n])
end
```

Figure 2.3 CYK algorithm for revised Chomsky normal form

Here, the function apply(P,B,C) is used to find the set of non-terminal symbols in P such that they are on the left side of production and they derive ->BC on the right side of production.

The fundamental data structure used in the CYK algorithm is the T[i,j] which stores the partial bottom-up derivation of substring $a_i a_{i+j}$. Illustration of how algorithm works is done below with an example. [45]

2.2.1.3 Example of CYK algorithm:

Assume that we are given the grammar:

S->bS|bC, C->cs and the string ‘bbcs’ and asked to find whether the string belongs to the grammar. CYK works the following way:

First, the lowest of the T table is filled with the terminal symbols (Fig2.4):

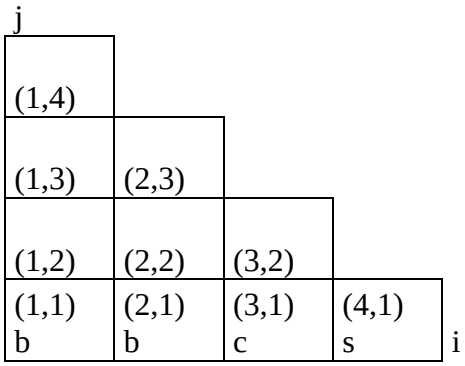


Figure 2.4 T table filled with the terminal symbols

Note that T[i,j] corresponds to the symbols that can derive the sub-string of length j starting from position i. For example T[2,1] is b since that is the only symbol that can derive the substring ‘b’ of length 1 and starts from position 2 in ‘bbcs’. The algorithm systematically finds the set of non-terminal symbols that can derive part of the input string. At the end, T[1,4] corresponds to the only substring of length 4 starting from position 1, which is the string itself, and if that cell of the table includes S, the starting symbol, this means that the string can be derived by starting from the starting symbol which proves it is in the language.

T[3,2] is the set of non-terminals that can derive the sub-string cs. This is found with application of the function apply(P,B,C). Here, P is the set of rules in the grammar, B is ‘c’ and C is ‘s’. Note that the only non-terminal that is in the left of the rules with ->cs in the right is C. So, a C is added to the cell T[i,j] with the line T[i,j]<-T[i,j] U apply(P, T[i,k],T[i+k,j-k]).

Since there are no non-terminals that can derive substrings bb and bc, corresponding cells of the table is empty.

T[2,3] is the set of non-terminals that can derive the substring bcs. This can be devided into substrings in two ways: T[2,1]T[3,2] or T[2,2]T[4,1]. The first corresponds to (b)(cs) and the second corresponds to (bc)(s). Note that the first way takes B=b and C=C as arguments to the apply function. The only non-terminal that has ->bC in the right side of production rules is S, so an S is added to the cell

T[2,3]. So, when we come to the last cell the T table has the following information (Fig2.5):

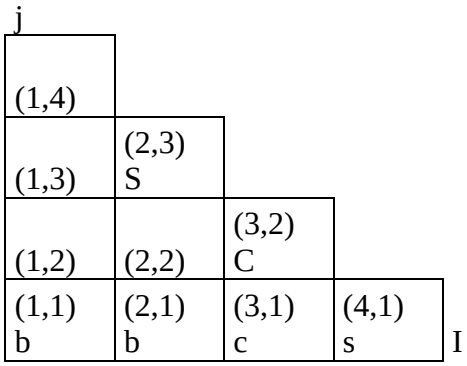


Figure 2.5 T table when the last cell is approached

T[1,4], which is the string itself can be divided into two substrings three different ways:

T[1,1]T[2,3], T[1,2]T[3,2] and T[1,3]T[4,1]

Note that the first way corresponds to bS. If we input this as input to the apply(P,'b','S'), the function returns S, since it is the only non-terminal that is in the left side of production rules that has ->bS in the right side. We add 'S' to T[1,4]. So, at the end since S is element of T[1,4] the CYK function returns 'true' meaning the given string can be derived with the given grammar. This is true and the derivation of the string can be done the following way:

S->bS->bbC->bbcs

The Matlab code for the CYK algorithm, along with Matlab code for other algorithms explained below, is given in the appendix.

2.2.1.4 Learning structure of grammars (Grammar induction) :

The CYK algorithm requires a grammar and a string to compare whether the string belongs to the grammar. It is also possible to learn the grammar from samples strings that do and do not belong to the grammar. This problem is known as

“grammar induction” in literature and for CFGs it is mentioned to be a hard problem. It is shown theoretically [46] that learning an arbitrary CFG from only positive samples is not possible. But natural languages are higher than CFGs in the Chomsky hierarchy and humans can learn from only positive samples [47].

In this study we are assuming that we have negative samples, too. And this is the case for most of pattern recognition problems. For some problems, like face detection, number of negative samples are generally much more than those of positives. Here, in some examples, we show this is not to be the case.

The grammar induction algorithm used in this study is so called Synapse which has the inductive CYK algorithm at its core [44]. Below both are explained in detail and algorithms and codes are given.

Note that Synapse and Inductive CYK are nondeterministic algorithms, which have to be converted to the deterministic versions in order to code in real computers [48]. Writing programs in nondeterministic way makes them shorter and more intuitive but this requires a systematic and careful conversion procedure.

2.2.1.5 Inductive CYK algorithm

The most intuitive way to derive a grammar that can recognize positive samples and reject negative samples is to search for all the production rules until a solution is achieved. The approach taken here is to find the production rules that can recognize all positives first and then test whether they give any false alarms. The most important question is how to make a systematic search. So, we have to find what to do in the case of a false alarm when the system finds a rule set with zero missed detections. The system must backtrack to some previous point and try adding some other rules so that it recognizes all positives with no false alarm. This requires keeping track of system states when a choice is made so that the backtracking can be done correctly. This is done using a stack that stores the full system state in choice points and retrieves the last system state when a failure occurs.

Inductive CYK algorithm is an extension of CYK algorithm with an additional function to add new rules to the grammar when the existing rules do not derive a given string. The algorithm takes the input string w, the existing rules P, number of nonterminal symbols K and number of maximum rules allowed R_{max} as input and returns a new set of rules P in revised Chomsky normal form. The pseudo-code of the Inductive CYK algorithm is given in next page (Fig 2.6).

Note that the **nondeterministic goto** and **failure** are two operations that do not exist in conventional deterministic programs and have to be dealt with accordingly. In every nondeterministic goto the system decides one of two ways: to **goto** or to **not goto**. The system first tries **goto** and then in case of failure chooses **not goto**. This assures minimum length grammar by adding rules if and only if it is necessary. In the **failure** operation, system goes to the last choice point, which corresponds to the last **not goto**. The Matlab code of Inductive CYK algorithm is given in appendix.

Inductive CYK algorithm is called repetitively from an outside system, Synapse, which is explained next.

```
function inductive_cyk(w:string, P:set of rules, K,R,Rmax: integer):
% K is the number of nonterminal symbols
% R is the number of rules. Rmax is the maximum number of rules.
begin
    var T: array [1..n, 1..n] of set of symbols;
    var i,j,k: integer, N_K: set of symbols
    N_K ← { K possible nonterminal symbols };
% Initialization of the array. It is assumed that w=a_1…a_n.
    for j ← 1 until n do T[i,1] ← a_i
    for j ← 2 until n do for i ← 1 until n-j+1 do
    begin
    T[i,j] ← Ø;
    for k ← 1 until j-1 do
    begin
        for each B ∈ T[i,k] do for each C ∈ T[i+k,j-k] do
            begin
            for each A ∈ N_K if (A → BC) not ∈ P then
                begin
                nondeterministic goto brake;
                if R < R_max then R=R+1 else failure;
                P ← P U {(A → BC)};
                brake: end
            T[i,j] ← T[i,j] U apply(P,{B},{C})
            end
    end
    end;
    if S ∈ T(1,n) then return (P) else failure
end
```

Figure 2.6 Inductive CYK Algorithm

2.2.1.6 Synapse algorithm

The aim of the grammar induction system is to learn minimum length of production rules for given positive and negative samples. In this system this is done by iterative deepening of search space. That is the search is started with one nonterminal, S, and zero rules and new rules and nonterminals are added until a solution is found. The number of maximum rules, R_{max}, is incremented until a limit, R_{limit} and if no solution is found a new nonterminal is added and search is repeated. Limit on number of nonterminal symbols, K_{max}, is defined by user to limit the search so that the procedure returns with failure in finite amount of time for non-context free grammars.

Maybe the most important point of the Synapse algorithm is the **failure** operation which goes to the last choice point and retrieves system state accordingly.

The pseudo-code for Synapse algorithm is given (Fig2.7).

```
procedure synapse(S_P,S_N): list of strings, K_max, R_limit: integer);
% S_P is a sequence of positive sample strings w_1,w_2,…,w_n
% S_N is a sequence of negative sample strings v_1,v_2,…,v_m
% K_max is the maximum number of nonterminal symbols
% R_limit is the limit of the number of rules in the search.
begin
    var P: set of rules;
    var K, R, R_max,i,j: integer;
    for K ← 1 until K_max do for R_max ← 1 until R_limit do
    begin
    P ← Ø, R ← 0;
    for i ← 1 until n do
        begin
        P ← inductive_cyk(w_i,P,K,R,R_max);
        for j ← 1 until m do if cyk(v_j,P) then failure;
        end
    Output the rules P;
    return
    % For finding all solutions, replace return by failure;
    end
    Print "No grammar is found";
end
```

Figure 2.7 Nondeterministic Program for Synapse

2.2.2 Multilayer neural networks

Multilayer neural networks are frequently used as classifiers, or more specially as detectors, in many studies. A multilayer neural network is composed of at least three layers of nodes where each layer of nodes' outputs are connected to next layer's nodes' inputs as shown in Figure 2.8.

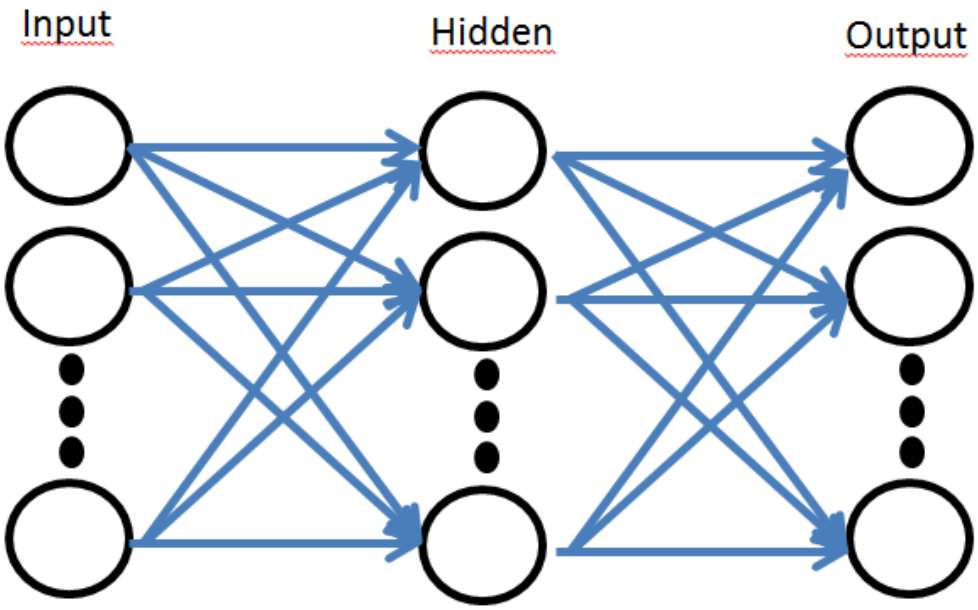


Figure 2.8 Multilayer neural network structure

Learning in neural networks can be done using the backpropagation algorithm.

2.2.3. Shape Context Descriptor

Shape context is a feature descriptor used in object recognition. Serge Belongie and Jitendra Malik proposed the term in their paper(63) **.**

Shape context descriptor [50] is a rich descriptor to represent and match shapes. Input is an edge map extracted by an edge detector. Finite number of points from the edge map, hundred in this study, is selected randomly. Shape context is described for each point on the shape as histogram of relative positions of all other points of the shape with respect to the chosen point. So, for each shape, hundred

histograms are hold to represent the shape. A log-polar histogram of 12 angular bins and 5 range bins is chosen as in the original study. In order to compare two shapes two things are considered. First, a distance metric is defined to describe how close a shape is to another shape. Second, each point on the first shape is matched with a point on the second shape. Distance between two points' shape contexts is defined using $\chi 2$ test statistic:

$$C_{ij} \equiv C(p_i, q_j) = \frac{1}{2}\sum_{k=1}^{K} \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)} \quad (1)$$

where hi(k) and hj(k) are K-bin normalized histogram at pi and qj, respectively. Distance between two shapes is calculated as the sum of distances between matched points' shape contexts:

$$D = \sum_{n} C(p_n, q_n) \quad (2)$$

This is also the metric to be minimized to match points on the first shape to the points on the second. This minimization is done using the Hungarian algorithm [13].

CHAPTER 3

FOOTBALL VIDEO GAME (FVG) DATASET

A new dataset for human behavior understanding is generated in this study by using a football video game. The details of the dataset, that we call FVG dataset, is explained in this chapter. First, general information about the FVG dataset is given. Then each action and activity is explained.

3.1. General information about the FVG dataset

Note that most of the datasets in the literature consist of simple actions like walking, running, handwaving, etc. Actually human activities range from simple actions to more complex actions. Between simplest actions and a complete TV series video, there exist actions of varying complexity. Sport games take place somewhere in the middle of this complexity list. Football is chosen for this reason. In order to build a more controlled dataset, instead of using real football games, football video game is chosen.

FIFA2011 video game is played naturally and samples are recorded at 1920x1080 resolution with 25 fps using video capture software of NCH software suite.

The whole dataset consists of two parts of different level of complexity. In the lower level, there are atomic actions which are atomic player actions longing for half second like running or hitting a ball. In the higher level, there are activities which are nothing but sequences of atomic actions accomplished one after another. A good example is dripling which consists of sequences of runs and manevours

following each other. In the following sections details are given for each level of complexity.

3.2.FVG dataset

Why football matches? Necessity for a new dataset

Several popular datasets from the literature are acquired and inspected. It is seen that they will not be the most proper setting to prove the usefulness of the proposed approach. This is because of the following: Since they mostly consist of relatively short and simple actions, it is expected that they will be caught in the lower layers which will be implemented using the existing techniques. The novelty of the proposed approach, that is learning grammar rules from data, will be seen best on complex multi person activities where the structure of the activity can be represented by rules of relations between simpler primitive actions. This is the first reason to choose football matches as the first application domain: they are long videos of about 90 minutes, composed of well structured activities of several persons.

The question of "why not Hollywood movies" has the following answer: they will also prove to be a good domain for the proposed approach but it will be more difficult to track humans and catch the primitive actions due to variety of environment. On the other hand, environment of the football matches are relatively standard and background is almost identical: a green field with marked lines conforming to a defined international standard. It is easier to detect region of interest. Furthermore, the primitive actions in a football match are relatively low numbered (running, beating, shooting as main primitives and may be a handful of more to define a full match) compared with any action of life in actual movies.

Newly generated FVG dataset is explained in this chapter in detail. First, general information about the FVG dataset is given. Then each action and activity is explained.

3.2.1 FVG atomic action dataset

Some of the raw videos recorded are used for generating primitive action dataset. Each of the atomic actions last for approximately half second and consists of 12 frames. Totally there are 215 primitive actions in the dataset. MoP algorithm is tested on this dataset. The types of atomic actions are as follows:

1) *ballAlone* (a): none of the players controls ball and ball is alone,

2) *beat* (b): two or more rival players fight for control of ball,

3) *hitBall* (h): a player hits the ball,

4) *manevour* (m): a player changes direction while moving with ball,

5) *run* (r): a player controlling ball runs straight with ball,

6) *takeBall* (t): a player takes control of ball.

In figure 3.1 samples from FVG action dataset is given for each different action

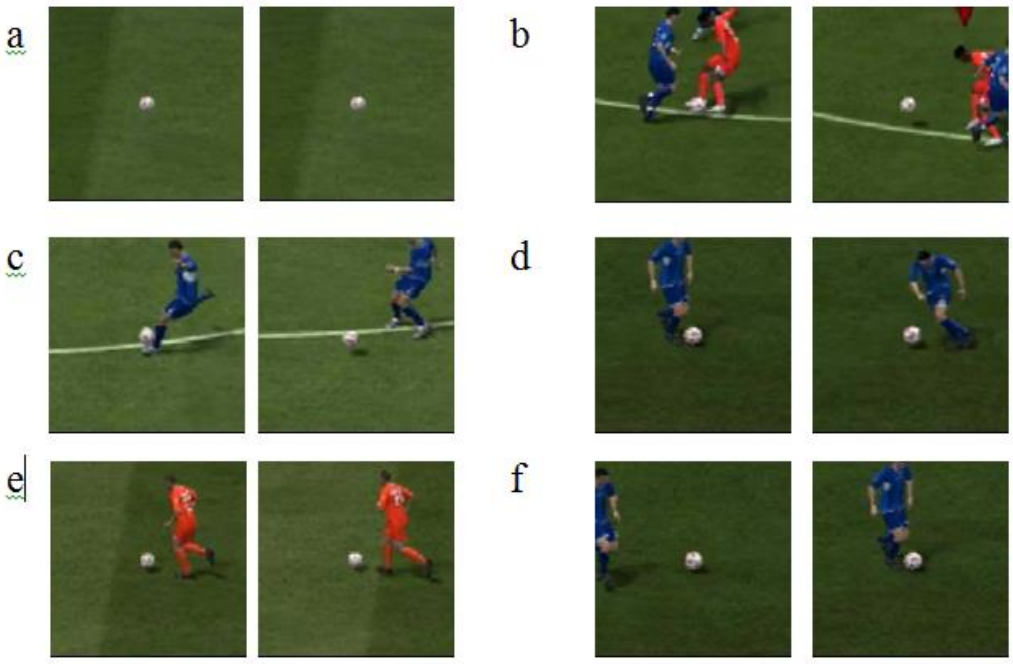


Figure 3.1 Sample figures from primitive action dataset a) ballAlone (a), b) beat (b), c) hitBall (h), d) manevour (m), e) run (r), f) takeBall (t)

Number of samples from each primitive action is given in the Table 3.1 below.

Table 3.1 Number of samples from each primitive action in the FVG dataset

ballAlone	beat	hitBall	manevour	run	takeBall	total
25	15	15	15	115	30	215

3.2.2 FVG Activity Dataset

Complex activities are videos consisting of several primitive actions cascaded sequentially. In complex activity dataset, there are totally 40 videos each consisting one of five types of complex activities:

1) *Dripling* (D) consists of runs and optionally manevours,

2) *LongPass* (lP) consists of hitBall followed by one or more ballAlone and finished by takeBall,

3) *Lose* (L) consists of run followed by beat and finished by ballAlone,

4) *ShortPass* (sP) resembles longPass with the difference that there is no ballAlone but only hitBall and takeBall,

5) *Tackle* (T) consists of run followed by beat and finished by run.

Some samples for each complex activity in FVG dataset are given in Table 3.2 below:

Table 3.2 Samples for each complex activity in FVG dataset

Dripling	LongPass	Lose	ShortPass	Tackle	passingAround
rr rmr mrr	hat haat haaat	rbaa rba rbm	ht	rbr rbbr	ppdppdp ppdpppdtpp

Number of samples and grammar production rules for each complex activity is given in Table 3.3 below. An average duration of 1.5 seconds (36 frames) is recorded.

Table 3.3 Number of samples and grammar production rules for each complex activity in FVG dataset.

Dripling	LongPass	Lose	ShortPass	Tackle	passingAround
10	10	5	10	5	10
S -> rr S -> mr S -> rS S -> Sr S -> Sm	S -> at S -> aS S -> hS	S -> ba S -> rS S -> bm S -> Sa	S -> ht	S -> er S -> rS S -> eS	Given below

CFG that recognizes the passing the ball around activity is:

S→pAp

A→pdA|dpA|Apd|Adp|ptA|tpA|Apt|Atp|plA|lpA|Apl|Alp|p|d|t|l|pA|Ap|null

CHAPTER 4

PROPOSED APPROACH

In this chapter, the proposed approach is explained in detail. The general structure of the proposed approach is given in section 4.1. Ball detection and tracking is explained in section 4.2. Next atomic action recognition layer is explained in section 4.3. And finally, activity recognition layer is explained in section 4.4.

4.1. The General Structure of the Proposed Approach

In this thesis, a hierarchical approach to solution of human behavior understanding problem is proposed. Block diagram of the proposed approach is shown in Figure 4.1. A video including humans performing some complex activity is input to the system. In this study we generated a new dataset, namely football activity dataset, composed of videos of football video game as explained in Chapter 3. The first thing to do in the input video is to extract the region of interest which is around ball. Note that most active region is generally around ball in a football game. The output of football ball detection and tracking block is region of interest which is nothing but subvideo of certain duration centered around ball.

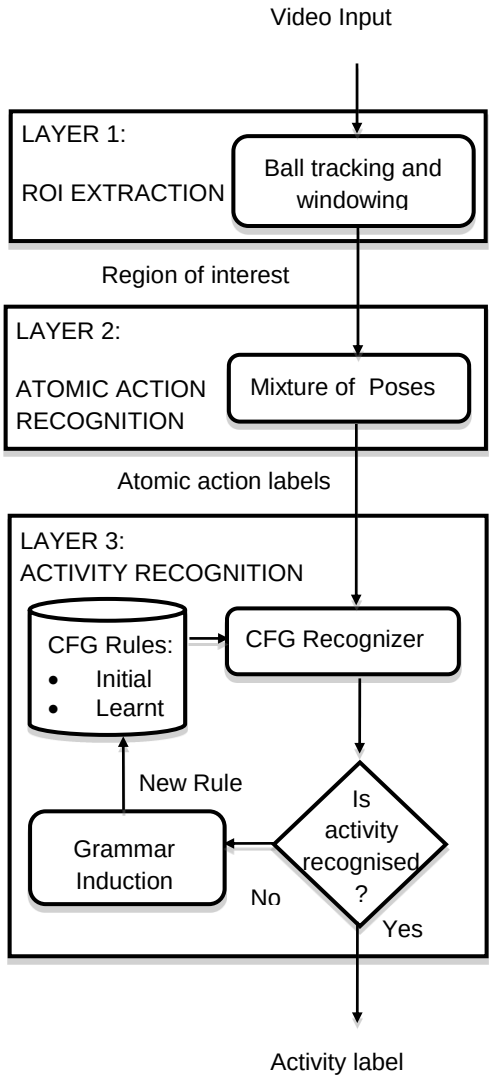


Figure 4.1: Block diagram of the proposed approach

Note that an activity as explained in Chapter 3, consists of atomic actions performed one after another. After the region of interest is extracted next thing to do is to recognize the atomic action performed around ball. There are six atomic actions in the dataset as explained in Chapter 3. In this layer three types of algorithms are used: 1) Dollar et al's cuboid features with k-nn and SVM classifier, 2) Mixture of Poses and 3) Mixture of Poses with Fisher Vector. Performance tests of each approach is given in Chapter 5 of this thesis. Output of this layer is the recognized primitive action series of which is fed to the next higher layer.

After recognition of series of atomic actions, next is to recognize activities consisting of series of atomic actions recognized. In our study there are two types of activities: those that are pre-taught to the system manually which are known activity types, like known attack tactics and those that are learnt by the system using grammar induction rules explained in section 2.2. This highest layer is composed of context-free-grammars recognizing pre-taught activities and grammar induction learning new rules.

Next in this chapter each layer is explained in detail.

4.2 Region of Interest Extraction

As a ball detection and tracking algorithm Liang et.al.'s [51] approach is implemented and tested. But results are far from being satisfactory. This is because of the following. Ball detection in video game differs from that in real videos. This is because appearance of ball in adjacent frames may differ a lot in video games while they vary little in real videos. As an example appearances of ball in two consecutive frames in video game is shown in Figure 4.2 below.

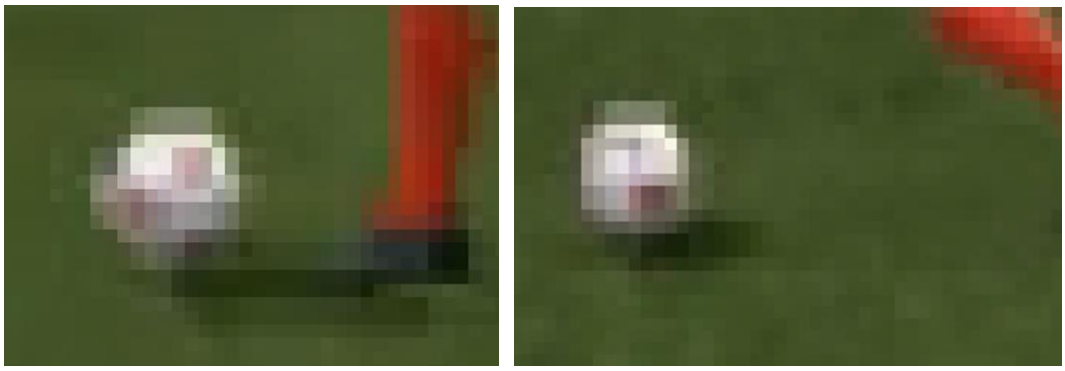
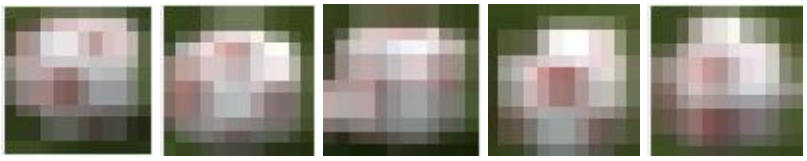


Figure 4.2 Appearance of ball in two consecutive frames in video game

Due to this fact a discriminative approach is taken for solution of ball detection problem. In this study neural networks are used as discriminative function for ball detection.

In this study a multilayer neural network with 20 nodes is used. As training algorithm gradient descent is used with 2000 iterations

A new ball appearance dataset is generated with 142 positive samples(Fig4.3a) and 273 negative samples (Fig4.3b).



(a)

(b)

Figure 4.3 (a) Positive and (b) negative samples from ball appearance dataset

With normal training false positives are more than acceptable. So, learning with queries is done to improve performance. As initial detector neural networks trained

with gradient descent as described above is used. A computer GUI shown in Figure 4.4 below is designed. Frame is scanned and whenever a ball is detected it is asked whether this is real ball or not. False positives which are detected as ball but not actually ball are added to the training dataset. Note that this may resemble to training of Ada-Boost algorithm, and in this way most instructive negative samples are collected with learning with queries method.

Figure 4.4 GUI designed to iteratively train neural network with learning with queries method

This way performance of ball detection algorithm is enhanced to more than 98 percent as shown in Figure 4.6 below.

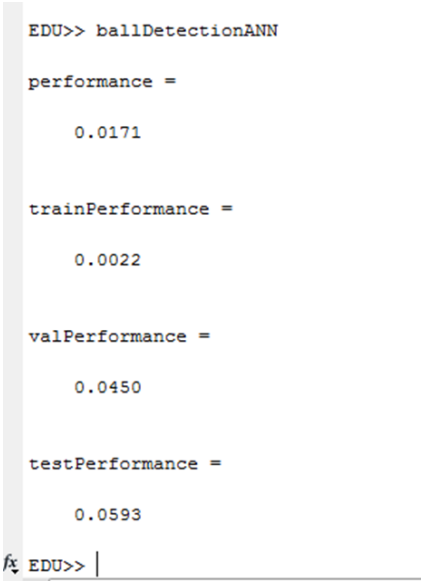


Figure 4.6 Performance of ball detection algorithm enhanced with learning with queries

Growing window tracker

In first frame ball is detected using neural networks described above by searching all frame and marking the highest score subframe as ball. Note that this is relatively computationally expensive since all of the frame is scanned and detector is used for each subframe. In order to reduce computational complexity a tracker is to be used. As tracker, particle filter is tried first. With 1000 particles performance is poor. With higher number of particles it is inhibitively slow. An alternative tracker, growing window tracker, is proposed and used.

Note that since ball has limited velocity and each consecutive frame is relatively close to each other in time, ball position between consecutive frames cannot be above certain pixel distance. Having this in mind it is more logical to begin to search for ball in next frame within neighbourhood of ball's last position. This is done using a search within a window whose size is initialized as 20x20 pixel-square for first frame. This 20 is selected by observations of balls maximum velocity. If the ball is not detected in first consecutive frame, then for next consecutive frame

window size is increased to 40x40 pixel-square and search is performed within this window. This is repeated for 5 consecutive frames. This is because uncertainty of ball's location in each undetected consecutive frame is increased by maximum velocity of ball. Note that ball's velocity is not estimated because it may change quite unexpectedly between consecutive frames. After 5 undetected consecutive frames, a new detection scheme is initialized by searching for the entire frame.

4.3 Atomic Action Recognition layer

4.3.1 Mixture of Poses

In this study, we proposed a global shape descriptor that we call Mixture of Poses (MoP) in order to be used for human behavior understanding problem. The method we proposed is first tested on Weizmann dataset and presented in [49]. Although our approach may be thought as if it is resembling that of [11], it is quite different because in this study we use mixture of Gaussians as a global shape representation while their approach uses them mainly as local shape and motion descriptors. Here the action classes are decided not considering the probability value at a specific point in the distribution representing the action class, but by calculating the distances of two distrubutions, which are the mixture of poses representing the new coming action to be classfied and mixture of poses representing each action stored in the library. Additionally, number of mixtures in our system is adaptive depending on complexity of motion.

The block diagram of the proposed approach is shown in Figure 4.7.

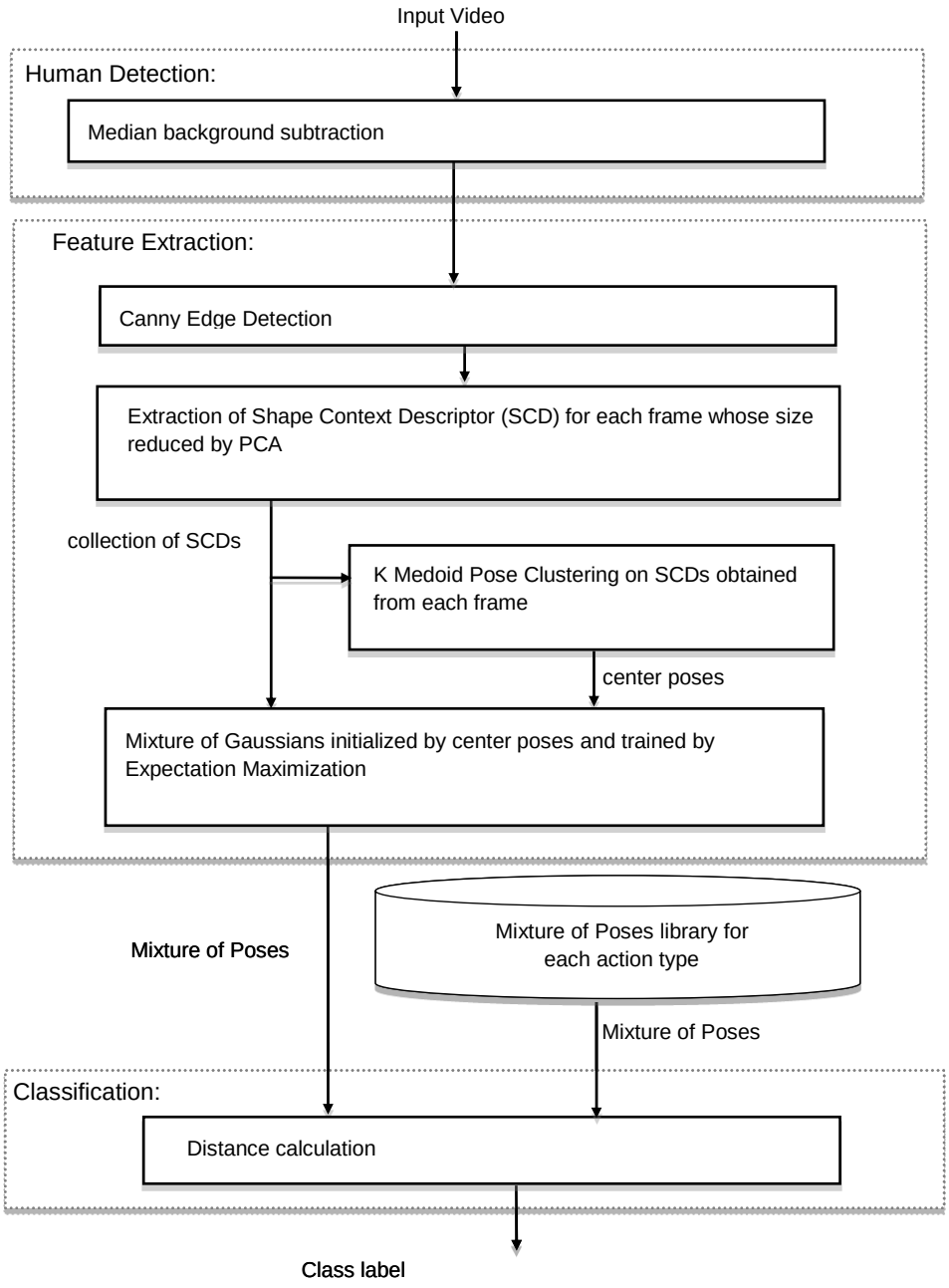


Figure 4.7 Block diagram of Mixture of Poses

The input video is processed using color based background subtraction technique and a foreground video is found (see Fig4.8). Using Canny edge detection, contours of the foreground are extracted and fed to the feature extraction module. Notice that, the border of the whole body could be extracted for the jump and run actions since the whole body is moving in these cases. Not the whole body, but only the borders of arms and hands, sometimes also head, are extracted the for the wave1 (one hand) and wave2 (two hands) actions, since body is not moving in these cases.

Figure 4.8 Output of foreground extractor. ballAlone, beat, hitBall, manevour, run, takeBall

Shape context descriptor of each frame is computed using the method described in chaper 2. The frames of a video are clustered using a variant of k-medoids clustering. The centroids of the clusters represent each action video and are called center poses. The procedure up to here is common in both training and test phases.

Although it is assumed that the center poses are relatively informative representations of action videos, in order to have more successful classification, more informative representation is required. This is done by representing the distribution of center poses belonging to the same action by a mixture of Gaussians trained by expectation maximization algorithm, that we call mixture of poses of that

action. A library is prepared by finding the mixture of poses for each different action type of each person.

In recognition phase, a test video is also represented with a mixture of poses and compared to each mixture of poses in the library prepared in training phase. The class label of mixture of pose in the library having the smallest distance is selected as the one recognized.

Mixture of Gaussians

An unknown distribution can be approximated by mixture of simple parametric distributions where Gaussians are widely used as base functions. k-medoid clustering is used to initialize the means of Gaussians. Expectation maximization is used to find the parameters of the Gaussians constituting the mixture.

After training phase a library of Mixture of Poses is prepared where each entity is corresponding to Mixture of Gaussian extracted considering a single action performed by a human recorded in videos. In recognition phase, test video is compared with each mixture in the library and the one with the closest distance is assigned as label.

In this study the L2 norm of the difference of two mixtures of Gaussians is used as the distance between them:

$$D(MoG_i(x) - MoG_j(x)) = \left\| MoG_i(x) - MoG_j(x) \right\|_2 = \left(\int_{x \in R^n} (MoG_i(x) - MoG_j(x)^2 dx \right)^{1/2} \quad (3)$$

Here Rn is the n dimensional input space on which mixture of Gaussians are defined. For practical purposes, instead of this integral calculation, summation of the difference of their values within hypercubes placed around the Gaussian centers and extending for two standard deviations in each dimension on the n dimensional discretized input space is calculated.

4.3.2 Fisher Vector

In MoP, distance calculation between different MoGs is computationally most expensive part. In order to decrease computation time we have used Fisher vector representation[52]. By using Fisher vector representation, MoG is represented as a single vector. Each mode in MoG is mapped to a part of the vector. Note that for different number of modes, different lengths of vectors are obtained. All vectors are truncated to the shortest vector length. Note that Fisher Vector Representation is used in two recent studies on human behavior understanding and successful results are reported [53], [54]. Notice that these two studies are later than our MoP; however recognition performance of MoP is still better than Fisher vectors as shown in results section.

4.3.3 Dollar et al.'s cuboid features

For atomic action recognition we have used Dollar et.al.'s [9] cuboid features as an alternative. As interest point detector 1D Gabor filter is used to extract local maxima. Around interest points found gradient features are extracted in cuboid volumes. As classifier nearest neighbor classifier and SVM are used. Note that this cuboid features have taken considerable amount of citation in literature. And they are suitable for classifying 3D data of volume of atomic actions. These are the reasons that they are chosen as a first alternative in atomic action recognition layer. The performance results are given in results section.

4.4 Activity Recognition Layer

In this layer complex activities given in Table 3.2 are recognized. Note that these are videos of several seconds of duration. Two types of approaches are used for comparison. As a first alternative we have used HMM which is nothing but probabilistic extension of FSM. HMMs are used widely in literature especially for speech recognition where there is a time series of data to be classified. Here in our study we also have a time series of discrete data to be classified as one of complex activities. So, HMM is one suitable solution for our problem. As a second

alternative, we have used CFG which is theoretically capable of recognizing more complex data than FSM can. Although there are studies in literature using CFG for activity recognition, we have not seen a study that uses learning in CFG for activity recognition. This is one way that our study is novel.

Each approach is explained below.

4.4.1 HMM for activity recognition

HMM is a type of stochastic signal model [41] which is used as a classifier for time series signals like speech, gesture, etc… In this study we have used HMM as one alternative for activity recognition layer. Input to HMM is series of atomic actions recognized by lower layer and output is the activity recognized. Performance results are given in results section.

4.4.2 CFG Learning for activity recognition

Application to activity recognition:

Within CFG framework, activities correspond to languages where each instance of an activity is a string in the language. Each terminal symbol corresponds to an action that is recognized by a low level module, like HMM.

As a domain, football matches are chosen for the reasons explained in section 3.1. Proposed solution is not limited only to football matches but any long video composed of several sequential activities, which may have some overlap in temporal and spatial dimensions. For example for a full Hollywood movie input, the output will be simple (possibly coinciding) subtitles each defining what is happening within certain amount of time.

Proof of concept demo

As a proof of concept scenario following setting is used:

DRIPLING: .These are complex activities involving several runs and possibly several manevours.

Positive samples: {'rr', 'rr', 'rmr', 'mrr', 'mr', 'rmr', 'rmr', 'rmr', 'rr', 'rr', 'rrm', 'rr', 'rmr', 'rr', 'rmr', 'rmr', 'rmr', 'rmr', 'mrr', 'rr'}

Negative samples: { 'hat', 'haat', 'hat', 'haat', 'haaat', 'haat', 'haat', 'hat', 'haat', 'hat', 'haat', 'hat', 'hat', 'hat', 'hat', 'hat', 'hat', 'hat', 'haat', 'hat', 'hat', 'hat', 'hat', 'haat', 'rbaa', 'rba', 'rba', 'rbm', 'rbm', 'rbaa', 'rba', 'rba', 'rba', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'ht', 'rbr', 'rbr', 'rbr', 'rbbr', 'rbbr', 'rbr', 'rbr', 'rbr', 'rbr', 'rbr', 'rbr', 'rbr', 'rbbr', 'rbr'}

Test sample: {‘rmrm’}

So, the recognition system is trained with positive and negative samples. Following is the output grammar of learning stage. Note that grammar is learnt correctly.

S -> rr

S -> mr

S -> rS

S -> Sr

S -> Sm

Next, test sample is given. Following is the output of recognizer. Note that test sample is classified correctly.

```
EDU>> test_cyk
isElementOfLanguage =
    1
```

CHAPTER 5

EXPERIMENTAL RESULTS

Results of each layer's success is given in this chapter. Region of interest extraction layer's success is given in section 5.1. Atomic action recognition results are given in section 5.2. Activity recognition results are given in section 5.3.

5.1 Ball detection and tracking performance results

As ball detection multilayer neural networks is used as explained in chapter 4. When MLP was trained with gradient descent the success rate was below acceptable level. Especially false alarm rate was unacceptably high to inhibit use in higher layers. For this reason ball detection layer is enhanced using learning with queries approach explained previously. Below are results for both native multilayer neural networks trained with gradient descent and those for enhanced detector with learning with queries.

Below in Fig 5.1, two screenshots from MATLAB program written for building ball dataset is given. In Fig 5.1a, a positive sample is selected and added to the ball dataset with label +1. In Fig 5.1b, a negative sample is selected and added to the ball dataset with label -1.

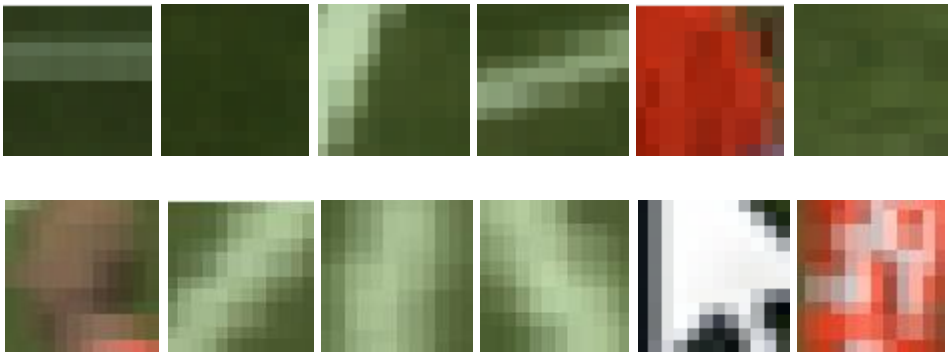
(a)

(b)

Figure 5.1 MATLAB program used for training ball detector (a) positive sample (b) negative sample

Some positive and negative samples from ball dataset are given in Fig 5.2. In Fig 5.2a, some positive samples with various appearance are shown. Note that appearance of ball differs in quite amount for different samples. In Fig 5.2b, some negative samples are shown.

(a)



(b)

Figure 5.2 Some positive (a) (first two rows) and negative (b) (last two rows) samples used for ball detection training

A screenshot from football video game with ball detected and marked in blue circle is given in Fig 5.3. Some screenshots around ball detected at various moments are given in Fig 5.4. Note that ball is detected correctly and marked with blue circle.

Figure 5.3 Ball detected is marked with blue circle in full screen

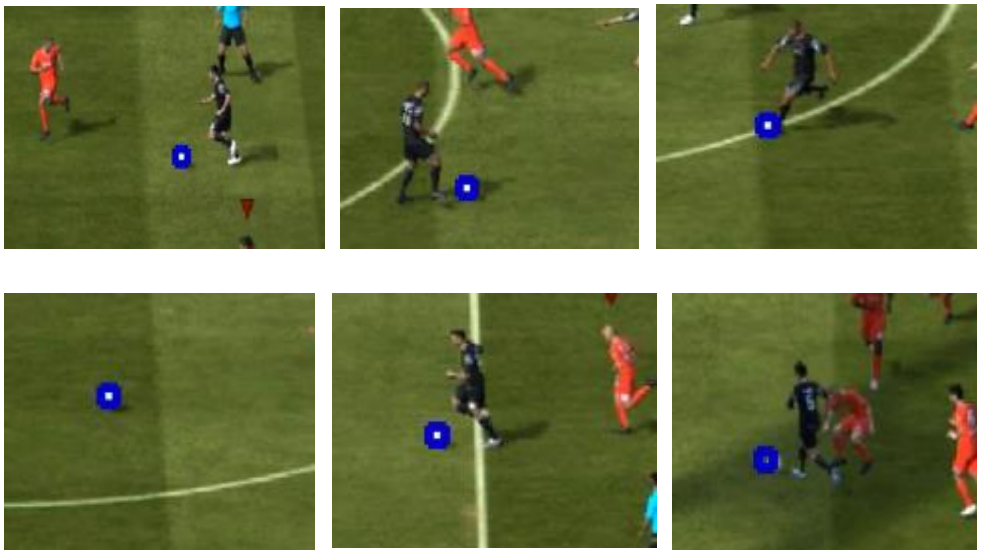


Figure 5.4 Ball detected is marked with blue circle in various frames

5.1.1 Multilayer neural network trained with gradient descent performance results

When trained with gradient descent, performance of ball detection is below acceptable level. Especially, false alarm rate is unacceptably high at 12.8%.

Table 5.1 Performance matrix for ball detection with multilayer neural network trained with gradient descent

	Ball	not ball
ball	99	1
not ball	128	872

5.1.2 Ball detection enhanced with learning with queries performance results

In order to reduce false alarm rate, learning with queries method is employed. Negative samples detected as ball are added to the dataset recursively. This way false alarm rate is reduced well below 4%.

Table 5.2 Enhance performance in ball detection with learning with queries

	Ball	not ball
ball	99	1
not ball	37	963

5.2 Atomic action recognition performance results

In middle layer three approaches are used in this study for comparison. One is the well known Dollar et.al.'s cuboid features. The other is a novel approach proposed in this study named as Mixture of Poses. Third one is Mixture of Poses with Fisher Vector. The newly proposed approach gave clearly better results as shown in Tables below.

5.2.1 Performance of MoP on Weizmann dataset

Note that we have also tested Mixture of Poses approach with a widely used dataset in the literature, namely Weizmann dataset and achieved 95.1% performance for k=400 in PCA. Refer to Table 5.3 for the success rates of various number of components for PCA:

Table 5.3 Performance results of MoP on Weizmann dataset

Number of Principle Components	Coverage Percentage	Success rate
50	60%	55.4%
100	70%	85.5%
200	79%	92.8%
400	87%	95.1%

5.2.2 Confusion matrix with MoP in the atomic action performance results

Table 5.4: Confusion matrix with MoP in the atomic action recognition layer

	ballAlone	beat	hitBall	manevour	run	takeBall
ballAlone	21	0	1	0	0	3
beat	0	3	2	2	7	1
hitBall	0	2	2	1	2	8
manevour	0	2	2	1	9	1
run	0	1	1	2	110	1
takeBall	2	4	6	1	5	12

Note that performance is clearly higher than that of cuboid features (69.3% vs 54%). For MoP success rate of 69.3% is observed. Using Fisher Vector representation performance is slightly degraded to 67% while speed is enhanced

considerably. The confusion matrix for MoP is given in Table 5.4. Note that ballAlone and run primitives are mostly classified correctly while others are often confused with another primitive action.

5.2.3 Dollar et al. features performance results

Confusion matrix for system with Dollar et.al.'s cuboid features in the atomic action recognition layer is given below. Note that since it is less successful than Mixture of Poses in the atomic action recognition layer, it is as expectedly of lower performance.

Dollar et.al.'s cuboid features algorithm is also tested with the primitive action dataset and a success rate of 53.95% is reported. Below is the confusion matrix for cuboid features.

Table 5.5: Confusion matrix with cuboid features in the atomic action recognition layer

	ballAlone	beat	hitBall	Manevour	run	takeBall
ballAlone	17	0	1	0	5	2
beat	0	5	0	1	6	3
hitBall	1	1	2	6	1	4
manevour	0	1	2	1	6	5
run	5	7	0	5	87	11
takeBall	1	4	5	5	11	4

5.3 Activity recognition performance results

The performance results for activity recognition layer with MoP in atomic action recognition layer are given in Table 5.6 below. In theory CFG can recognized more complex activities that HMM can not. An example for such a case is finding complex activities with passes more than total of the other activities. This type of complex activity can be recognized by CFG but not by HMM. An example

sequence with passing the ball around complex activity is shown below (p is used to represent ps and pl activities together). The underlined parts in the sequence are corresponding to passing around activity.:

ptddptddd**ppdppdp**tdlpdtlpddt**ppdpppdtpp**d

CFG that recognizes these sequences is:

S→pAp

A→pdA|dpA|Apd|Adp|ptA|tpA|Apt|Atp|plA|lpA|Apl|Alp|p|d|t|l|pA|Ap|null

The generating sequences for two of sequences above are:

For ppdppdp: S→pAp, A→pdA, A→Apd, A→p

For ppdpppdtpp: S→pAp, A→pdA, A→pA, A→pA, A→pdA, A→tpA, A→null

Theoretically higher capability of CFG comes with the price of computation time.

Table 5.6 Confusion matrix for whole system with CFG in activity recognition layer and MoP in atomic action recognition layer

	dripling	longPass	lose	shortPass	tackle	passing Around	None
dripling	7	0	0	0	0	0	3
longPass	0	6	0	0	0	0	4
lose	0	0	3	0	0	0	2
shortPass	0	0	0	6	0	0	4
tackle	0	0	0	0	3	0	2
passing Around	0	0	0	0	0	2	8

Table 5.7 Confusion matrix for whole system with with CFG in activity recognition layer and cuboid features in atomic action recognition layer

	dripling	longPass	lose	shortPass	tackle	passing Around	None
dripling	4	0	0	0	0	0	6
longPass	0	5	0	0	0	0	5
lose	0	0	2	0	0	0	3
shortPass	0	0	0	4	0	0	6
tackle	0	0	0	0	2	0	3
passing Around	0	0	0	0	0	1	9

Note that the performance of the activity layer is higher when MoP is used in action recognition layer compared to cuboid features. This comes with high computation time.

Table 5.8 Confusion matrix for whole system with with HMM in activity recognition layer and MoP in atomic action recognition layer

	dripling	longPass	Lose	shortPass	tackle	passing Around	None
dripling	7	0	0	0	0	0	3
longPass	0	6	0	0	0	0	4
lose	0	0	3	0	0	0	2
shortPass	0	0	0	6	0	0	4
tackle	0	0	0	0	3	0	2
passing Around	0	0	0	0	0	1	9

When the activity classes are fed manually, the confusion matrix for CFG and HMM are given in Tables 5.9 and 5.10 respectively. Note that for passing the ball around activity, CFG performs better than HMM, as expected theoretically. In this case success rate for CFG is 100% and for HMM it is 96%.

Table 5.9 Confusion matrix for whole system with CFG in activity recognition layer and manual feed in atomic action recognition layer

	dripling	longPass	lose	shortPass	tackle	passing Around	None
dripling	10	0	0	0	0	0	0
longPass	0	10	0	0	0	0	0
lose	0	0	5	0	0	0	0
shortPass	0	0	0	10	0	0	0
tackle	0	0	0	0	5	0	0
passing Around	0	0	0	0	0	10	0

Table 5.10 Confusion matrix for whole system with HMM in activity recognition layer and manual feed in atomic action recognition layer

	dripling	longPass	lose	shortPass	tackle	passing Around	None
dripling	10	0	0	0	0	0	0
longPass	0	10	0	0	0	0	0
lose	0	0	5	0	0	0	0
shortPass	0	0	0	10	0	0	0
tackle	0	0	0	0	5	0	0
passing Around	0	0	0	0	0	8	2

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Human activities range from simple breathing to whole human history. Middle in the complexity hierarchy is sports games where multiple persons commit complex activities consisting of sequential simpler primitive actions. In this study, a novel dataset using football video game (FVG) is generated. FVG dataset has two parts which are FVG-AtomicAction dataset and FVG-Activiy dataset. The samples in the FVG-AtomicAction dataset are videos of twelve frame length captured around the ball. In this dataset there are six different atomic action types, *namely ballAlone*, *beat, hitBall, maneuver*, *run* and *takeBall*. Each atomic activity video consists of 12 frames. The FVG-Activity dataset consists of videos corresponding to six different activity types namely *drippling, longPass, lose, shortPass, tackle* and pass Around. The length of videos in the FVG-activity data set lasts for a few seconds and their length changes depending on activity type and how they are performed.

In order to recognize both atomic actions and activities, a novel hierarchical architecture is proposed and tested. In the lowest layer of the architecture, an MLP based detector and tracker proposed for ball detection and tracking is used. Next in the middle layer, a novel approach, namely Mixture of Poses (MoP), proposed for atomic action recognition is used. In fact MoP is Gaussian Mixture Model (GMM) of Shape Context Descriptors (SCD) extracted from each frame of atomic action videos in the dataset and the number of mixtures are determined automatically. MoP are used as feature vectors and the classification is done by calculating the L2

distance between the MoP obtained for the sample to be classified and MoPs stored in the library.

In the highest layer a novel sequential recognizer, namely CFG, is used. Note that CFG is used in other studies for human action recognition in the literature [34], [37], but to our best knowledge it is the first time that grammar induction (CYK) is used for this purpose.

Results obtained by MoP atomic action recognition are compared with a widely referenced feature, namely Dollar et. al's cuboid features. Additionally a version of MoP, where Gaussian Mixtures are represented by Fisher Vectors (FV) and classfied by SVM is considered for comparison. Results obtained by CFG for activity recognition are compared with the results obtained by well known HMM.

Quite encouraging results are achieved on FVG dataset for the proposed architecture. For atomic actions 69.3% recognition rate is observed for proposed MoP approach, which clearly exceeds 54.0% of Dollar et. al's success rate. Employing Fisher vector with MoP slightly decreases performance to 67.0 % while achieving high speed. Newly proposed MoP approach is also tested on Weizmann dataset, one of the mostly used datasets in the literature, and 95.1% success rate is reported.

For activities in the FVG dataset, 62.5% success rate is observed with MoP-CFG pair while 42.5% is the success of cuboid-CFG pair.

Note that for the activities in the dataset, success rates of CFG and HMM are exactly the same. But it is known that CFG can recognize more complex activities that HMM cannot. We have given an example for this case. In football matches, when one of the teams achieves the score they aim, they may do some activities to just pass time. The best known one is *passing the ball around* activity. In passing the ball around activity, players of winning team just pass the ball to each other in order to pass time. This type of complex activity can be described as an activity where number of passes is more than number of other activities. An example sequence involving two of such complex activities is given as an example together

with the CFG that can recognize it. Note that the performance of CFG is higher than HMM in recognizing such a complex activity when labels of the simple actions constituting the complex activity are fed manually to CFG and HMM, although their performances are the same for the other activities in the dataset. For manual feeding, the success rate on the activity dataset is 100% for CFG and it is 96% for HMM. This confirms the theoretical fact that CFG is higher in complexity hierarchy than HMM and can recognize languages that HMM cannot.

Note that some activities that are meaningful in football game are not included in our FVG dataset. Some examples are goal, corner kick, touch. These activities are not included in the dataset since the ball is not seen in most of the frames so it is problematic to extract videos automatically around the ball. As a future study we are planning to include these activities and related actions in our dataset. Additionally, as a future work, real football games are planned to be used for a new dataset.

We are also planning to determine number of MoPs representing an action automatically depending on the variety of the action performed. As another improvement more advanced background subtraction techniques can be used. Distance between MoPs can be evaluated using symmetric KL divergence to improve speed and accuracy. As another future work, we are planning to build a full football game annotation system where a long football game video is input to the system and annotations are outputs. In order to achieve this continuous annotation scheme, a strong post-processing after primitive sequence generation is required.

REFERENCES

[1] Turaga et.al., Machine Recognition of Human Activities, IEEE transactions on circuits and systems for video technology, Vol. 18, No. 11, November 2008

[2] N. İkizler, P. Duygulu, Human Action Recognition Using Distribution of Oriented Rectangular Patches, ICCV, 2007.

[3] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[4] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 984–989.

[5] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007

[6] Veerzxaraghavan, A. K. Roy-Chowdhury, R. Chellappa, "Matching Shape Sequences in Video with Applications in Human Movement Analysis", IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 12, pp. 1896–1909, Dec. 2005

[7] Kepenekçi, Human Activity Recognition by Gait Analysis, PhD Thesis, 2011

[8] Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in Proc. Int. Conf. Pattern Recognit., 2004, pp. 32–36.

[9] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in Proc. IEEE Int. Workshop Vis. Surveillance Performance Eval. Tracking Surveillance, 2005, pp. 65–72.

[10] J. C. Niebles, H.Wang, and L. F. Fei, “Unsupervised learning of human action categories using spatial-temporal words,” in Proc. British Mach. Vis. Conf., 2006, pp. 1249–1258.

[11] J. C. Niebles and L. Fei-Fei, “A hierarchical model of shape and appearance for human action classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2007, pp. 1–8.

[12] R. Polana and R. C. Nelson, “Detection and recognition of periodic, nonrigid motion,” *Int. J. Comput. Vis.*, vol. 23, no. 3, pp. 261–282, 1997.

[13] C. Thurau, V. Hlavac, “Pose primitive based human action recognition in videos or still images”, in Proc. IEEE Conference on CVPR, 2008.

[14] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In CVPR’05, 2005.

[15] J. H. J.Ward. Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc., 58:236–244, 1963

[16] T. F. Syeda-Mahmood, M.Vasilescu, and S. Sethi, “Recognizing action events from multiple viewpoints,” in *Proc. IEEE Workshop Detection Recognit. Events Video*, 2001, pp. 64–72.

[17] Y. Yacoob and M. J. Black, “Parameterized modeling and recognition of activities,” *Comput. Vis. Image Understand.*, vol. 73, no. 2, pp. 232–247, 1999.

[18] O. Chomat and J. L. Crowley, “Probabilistic recognition of activity using local appearance,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1999, vol. 02, pp. 104–109.

[19] S. Savarese, A. Del Pozo, J. C. Niebles, and L. Fei-Fei, “Spatial-temporal correlations for unsupervised action classification,” presented at the IEEE Workshop Motion Video Comput., Copper Mountain, CO, Jan. 8–9, 2008.

[20] S. Nowozin, G. Bakir, and K. Tsuda, “Discriminative subsequence mining for action classification,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[21] S. F. Wong, T. K. Kim, and R. Cipolla, “Learning motion categories using both semantic and structural information,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–6.

[22] Y. Song, L. Goncalves, and P. Perona, “Unsupervised learning of human motion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 814–827, Jul. 2003.

[23] Chen, Ming-Yu and Hauptmann, Alexander, MoSIFT: Recognizing Human Actions in Surveillance Videos, Computer Science Department, CMU, Paper 929, 2009.

[24] A. Fathi, G. Mori, Action Recognition by Learning Mid-level motion features, CVPR, 2008.

[25] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in IEEE International Conference on Computer Vision, 2003, pp. 726–733.

[26] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proceedings of the DARPA Image Understanding Workshop, April 1981, pp. 121–130.

[27] Y. Ke, R. Sukthankar, and M. Hebert, “Efficient visual event detection using volumetric features,” in Proc. IEEE Int. Conf. Comput. Vis., 2005, pp. 166–173.

[28] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In CVPR, 2001.

[29] J. Wu, J. M. Rehg, and M. D. Mullin. Learning a rare event detection cascade by direct feature selection. In Proc. NIPS, 2002.

[30] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 405–412.

[31] Y. Wang, G. Mori, Human action recognition by semilatent topic models, IEEE TPAMI, vol. 31, pp. 1762-74, 2009.

[32] L. Suter, D. Wang, Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model, CVPR, pp. 1-8, 2007.

[33] Y. Mori, W. Greg, Max-Margin Hidden Conditional Random Fields for Human Action Recognition, CVPR, 2009.

[34] M. S. Ryoo and J. K. Aggarwal, "Recognition of composite human activities through context-free grammar based representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1709–1718.

[35] S. Park and J. K. Aggarwal, Simultaneous tracking of multiple body parts of interacting persons, CVIU 102(1), pp. 1-21, April 2006.

[36] J. F. Allen and G. Ferguson, Actions and Events in Interval Temporal Logic, Journal of Logic and Computation, 4(5):531-579, 1994.

[37] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 852–872, Aug. 2000.

[38] T.J. Darrell and A.P. Pentland, ªSpace-Time Gestures,º Proc. Conf. Computer Vision and Pattern Recognition, pp. 335±340, 1993.

[39] A. Stolcke, ªAn Efficient Probabilistic Context-Free Parsing Algorithm That Computes Prefix Probabilities,º Computational Linguistics, vol. 21, no. 2, pp. 165±201, 1995.

[40] J.C. Earley, ªAn Efficient Context-Free Parsing Algorithm,º PhD thesis, Carnegie-Mellon Univ., 1968.

[41] LR Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE, 1989

[42] J. Martin, Introduction to Languages and the Theory of Computation, 3rd edition, McGraw-Hill, 2003.

[43] J.E. Hopcroft, R. Motwani, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, 3rd edition, 2007.

[44] K. Nakamura, T. Ishiwata, Synthesizing Context Free Grammars from Sample Strings Based on Inductive CYK Algorithm, Lecture Notes in Computer Science, 2000, Volume 1891/2000, 207-208.

[45] R. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches, John Wiley & Sons, Inc, 1992.

[46] E. M. Gold, Language identification in the limit. Information and Control, 10:447-474, 1967.

[47] Marcus, G.F. (1993). Negative evidence in language acquisition. Cognition, 46, 53-85.

[48] Floyd, R. W., Nondeterministic Algorithms, Jour. of ACM 14, No. 4 (1967) 636-644.

[49] Halıcı, U., Gökçe O., Mixture of Poses for human behavior understanding, CSIP 2013, China.

[50] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Trans. Pattern Anal. Mach. Intell., 24(4):509–522, 2002.

[51] Liang et.al., A scheme for ball detection and tracking in broadcast soccer video,…

[52] A. Vedaldi, B. Fulkerson, {VLFeat}: An Open and Portable Library of Computer Vision Algorithms, 2008, http://www.vlfeat.org/

[53] Castro F.M., Marin-Jimenez M.J., Medina-Carnicer R., 2014, Pyramidal Fisher Motion for Multiview Gait Recognition, Proc. ICPR.

[54] Peng X., Zou C., Qiao Y., Peng Q., 2014, Action recognition with stacked fisher vectors - Computer Vision–ECCV 2014, – Springer, pp 581-595.

[55] Ofli F., Chaudhry R., Kurillo G., Vidal R., Bajcsy R., 2014, Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition, Journal of Visual Communication and Image Representation Volume 25, Issue 1, Pages 24–38.

[56] Yang X., Tian Y., 2014, Effective 3D action recognition using EigenJoints, Journal of Visual Communication and Image Representation Volume 25, Issue 1, p 2–11.

[57] Wang J., Liu Z., Wu Y., Yuan J., 2014, Learning Actionlet Ensemble for 3D Human Action Recognition, IEEE Trans. on PAMI, 36(5), p 914-927.

APPENDIX A

MATLAB CODES

CYK algorithm

```
function elementOfG = cyk ( w, R, K, P )
elementOfG = 0;
tempsize = size(w);
n = tempsize(2);
T = zeros(n,n,K+1);
tempT1 = zeros(1, K+1);
tempT2 = zeros(1, K+1);
tempT = zeros(1, K+1);
for i=1:n
    T(i,1,1)= w(i);
end
for j=2:n
    %j
    for i=1:(n-j+1)
        %i
        T(i,j,1)=0;%T(i,j)=empty set
        for k=1:(j-1)
            %k
            tempT1(:) = T(i,k,:);
            tempT2(:) = T(i+k,j-k,:);
            temp1 = 1;
            while ( tempT1(temp1) ~= 0 )
                B = tempT1(temp1);
                temp2 = 1;
                while ( tempT2(temp2) ~= 0 )
                    C = tempT2(temp2);
                    A = apply(P,K,R,B,C);
                    temp3 = 1;
                    while ( A(temp3) ~= 0 )
                        tempT(:) = T(i,j,:);
                        if ( isNotElement ( A(temp3), tempT ) )
                            tempSize = findSize(tempT);
                            T(i,j,tempSize+1)=A(temp3);
                        end
                        temp3=temp3+1;
                    end
                    temp2 = temp2 + 1;
                end
                temp1 = temp1 + 1;
            end
        end
    end
end
tempT(:) = T(1,n,:);
notElement = isNotElement(1, tempT);
if notElement == 0
    elementOfG = 1;
else
    elementOfG = 0;
end
```

Inductive CYK algorithm

```
function [returnMode, failMode, R, P] = inductive_cyk(failMode, w, R, K, P, Rmax)

%globals
global sysP sysT sysSp sysReg sysK sysR sysRmax
global sysStkPtr sysStkPtrLimit Rlimit Sp TUN
global maxn Kmax

tempsize = size(w);
n = tempsize(2);
T = zeros(maxn,maxn,Kmax+1);
tempT1 = zeros(1, Kmax+1);
tempT2 = zeros(1, Kmax+1);
tempT = zeros(1, Kmax+1);
for i=1:n
    T(i,1,1)= w(i);
end
canBeDerived = 0;
while (canBeDerived == 0)
    if (failMode == 0)
        j=2;
    end
    while ( (failMode == 1) || (j<=n) )
        if failMode == 0
            i=1;
        end
        while ( (failMode == 1) || (i<=(n-j+1)) )
            if failMode == 0
                T(i,j,1)=0;%T(i,j)=empty set
                k=1;
            end
            while ( (failMode == 1) || (k<=(j-1)) )
                if failMode == 0
                    tempT1(:) = T(i,k,:);
                    tempT2(:) = T(i+k,j-k,:);
                    ib = 1;
                end
                while ( (failMode == 1) || (tempT1(ib) ~= 0) )
                    if failMode == 0
                        B = tempT1(ib);
                        ic = 1;
                    end
                    while ( (failMode == 1) || (tempT2(ic) ~= 0) )
                        if failMode == 0
                            C = tempT2(ic);
                            ia = 1;
                        end
                        while ( (failMode == 1) || (ia<=K) )
                            if failMode == 0
                                A = TUN(ia);%A = Nk(ia);
                            end
                            if ( (failMode == 1) || (isNotInTheRules(R,P,A,B,C)) )
                                if failMode == 0
                                    Reg=[j i k ib ic ia B C A];
                                    pushSystemState( P, T, Reg, K, R, Rmax );
                                    if sysStkPtr >= sysStkPtrLimit
                                        fprintf('system stack pointer in the limit!
\n');

                                        returnMode = 2;%failure due to system stack
limit

                                        return;
                                    end
                                end
                                if (failMode == 1)
                                    if sysStkPtr <= 1
                                        fprintf('system stack pointer 1. Empty
stack! \n');

                                        returnMode = 3;
```

```
                                return;
                            end
                            [P T Reg K R Rmax] = pullSystemState ();
                            j = Reg(1);
                            i = Reg(2);
                            k = Reg(3);
                            ib = Reg(4);
                            ic = Reg(5);
                            ia = Reg(6);
                            B = Reg(7);
                            C = Reg(8);
                            A = Reg(9);
                            if ( R<Rmax )
                                R = R+1;
                                P(R,1)=A;
                                P(R,2)=B;
                                P(R,3)=C;
                                failMode = 0;
                            else
                                %FAILURE!
                                failMode = 1;
                                break;
                            end

                        end
                    end
                    if failMode == 0
                        ia = ia+1;
                    end
                end

                if failMode == 0
                    %-------------------------------------------------------
                    % T[i,j]=T[i,j]Uapply(P,{B},{C})
                    setofNs = apply(P,K,R,B,C);
                    temp3 = 1;
                    while ( setofNs(temp3) ~= 0 )
                        tempT(:) = T(i,j,:);
                        if ( isNotElement ( setofNs(temp3), tempT ) )
                            tempSize = findSize(tempT);
                            T(i,j,tempSize+1)=setofNs(temp3);
                        end
                        temp3=temp3+1;
                    end
                    % END T[i,j]=T[i,j]Uapply(P,{B},{C}) END
                    %-------------------------------------------------------
                    ic=ic+1;
                else
                    break;
                end
            end
            if failMode == 0
                ib=ib+1;
            else
                break;
            end
        end
        if failMode == 0
            k=k+1;
        else
            break;
        end
    end
    if failMode == 0
        i=i+1;
    else
        break;
    end;
end
```

```
        if failMode == 0
            j=j+1;
        end
    end

    % if S element T(1,n) return(P) else FAILURE! pullSystemState() goto('N3')
    canBeDerived = 0;
    T1n = T(1,n,:);
    if ( isNotElement ( 1, T1n ) )
        canBeDerived = 0;
    else
        canBeDerived = 1;
    end
    if ( canBeDerived == 1 )
        returnMode = 0;%return with success
        failMode = 0;
        return;
    else
        %FAILURE!
        failMode = 1;
    end
end
```

Synapse algorithm

```
%synapse
function P = synapse(SPos, SNeg)

%globals
global sysP sysT sysSp sysReg sysK sysR sysRmax
global sysStkPtr sysStkPtrLimit Rlimit Sp TUN
global TUNcharacters maxn Kmax

TUN = [1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20];
sysStkPtrLimit = 1000;

%find max size of SPos
maxn=0;
for i=1:length(SPos)
    wi = char(SPos(i));
    if maxn < length(wi)
        maxn = length(wi);
    end
end

solutionFound = 0;
for K=1:Kmax
    for Rmax=1:Rlimit
        K
        Rmax
        fprintf(' \n ------------------------ \n');
        P = zeros(Rlimit,3);
        R=0;
        resetSystemState();
        returnMode = 0;
        while ( (returnMode == 0) && (Sp <= length(SPos)) )
            wi = char(SPos(Sp));
            w=convert2TUN(wi, TUNcharacters);
            failMode = 0;
            returnMode = 0;
            while ( returnMode == 0)
                [returnMode, failMode, R, P] = inductive_cyk(failMode, w, R, K, P,
Rmax);
                if ( (Rmax==6) && (K==4) && (R==6) )
                    fprintf('\nSp=%d\n',Sp);
                    wi
                    printRules(P);
```

```
                    pause
                end
                if returnMode == 0
                    isElementOfLanguage=0;
                    for j=1:length(SNeg)
                        wi = char(SNeg(j));
                        w=convert2TUN(wi, TUNcharacters);
                        isElementOfLanguage =  cyk( w, R, K, P);
                        if isElementOfLanguage == 1
                            break;
                        end
                    end
                    if isElementOfLanguage == 1
                        failMode = 1;
                        wi = char(SPos(Sp));
                        w=convert2TUN(wi, TUNcharacters);
                    else
                        Sp=Sp+1;
                        break;
                    end
                end
            end
        end
        if returnMode == 0
            solutionFound = 1;
        end
        if solutionFound == 1
            fprintf(' \n !! SOLUTION FOUND !! \n');
            %P
            return;
        end
    end
end
```

APPENDIX B

FRAMES OF ATOMIC ACTIONS

The frames constituting atomic actions are provided in the Figures B1-B6. Notice that some of the frames within the action sequence is the same. The reason for this repetion is frame rate for the game recording is less than 25 fps.

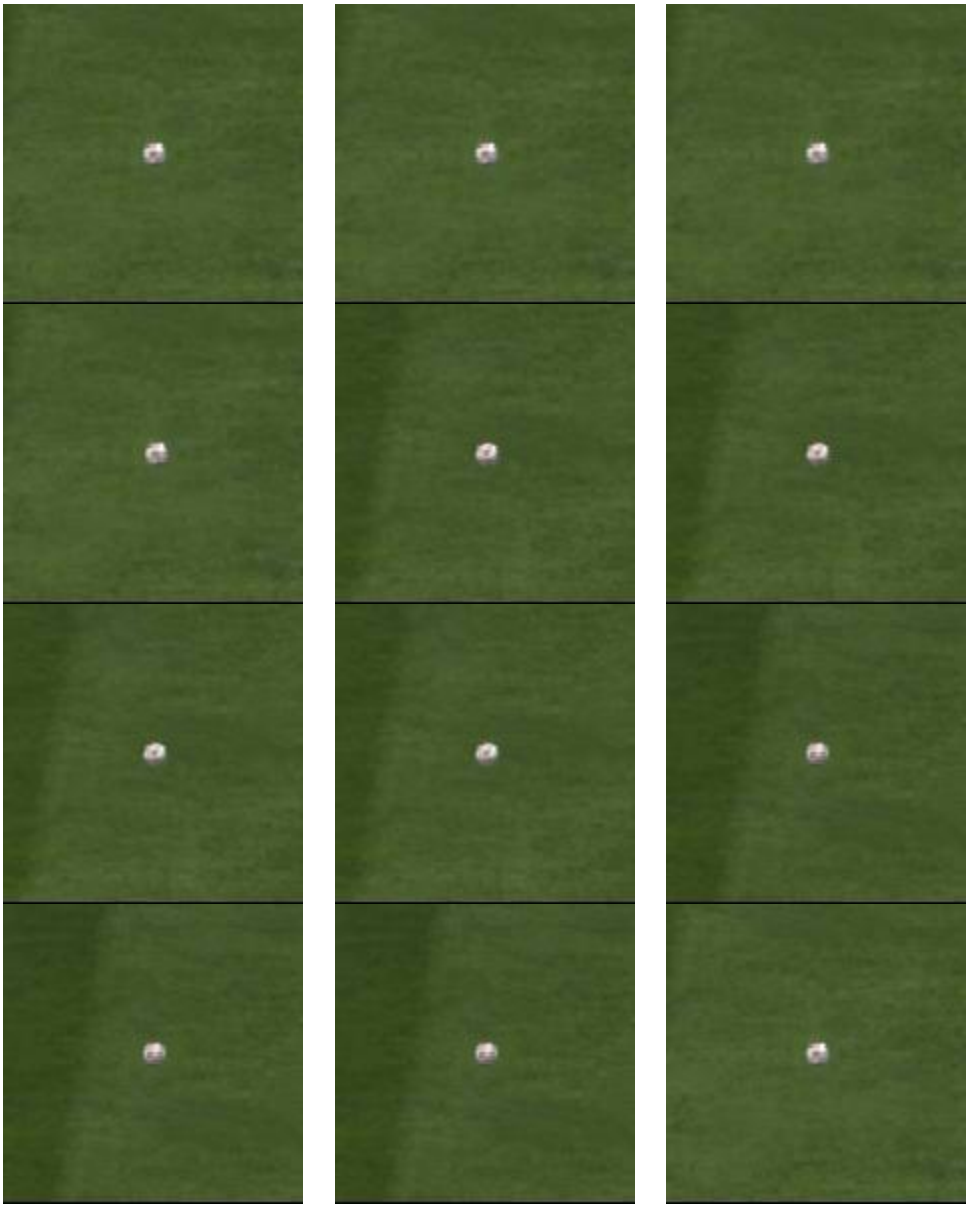


Figure B.1. Frames of a sample ballAlone atomic action sequence

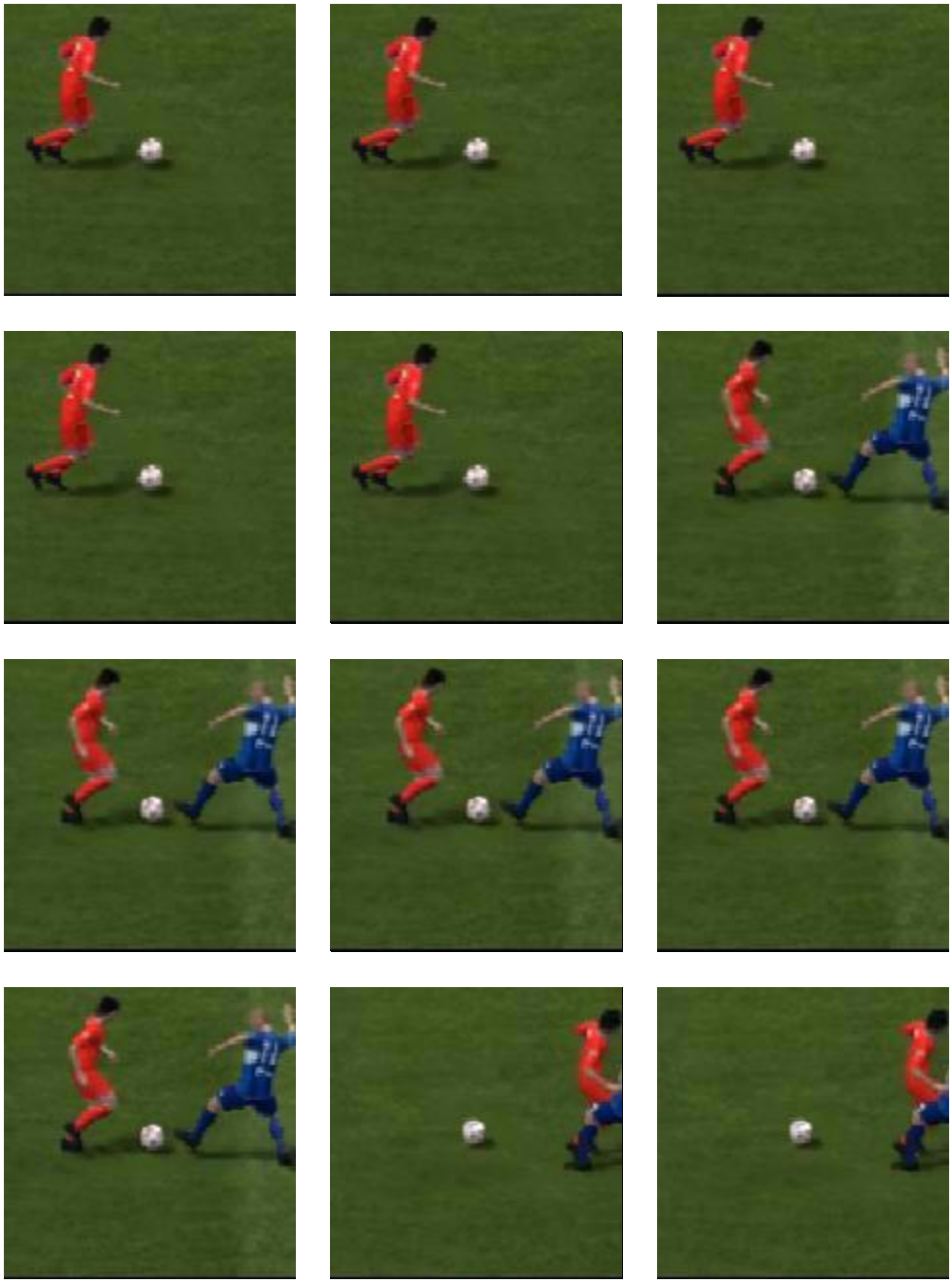


Figure B.2. Frames of a sample beat atomic action sequence

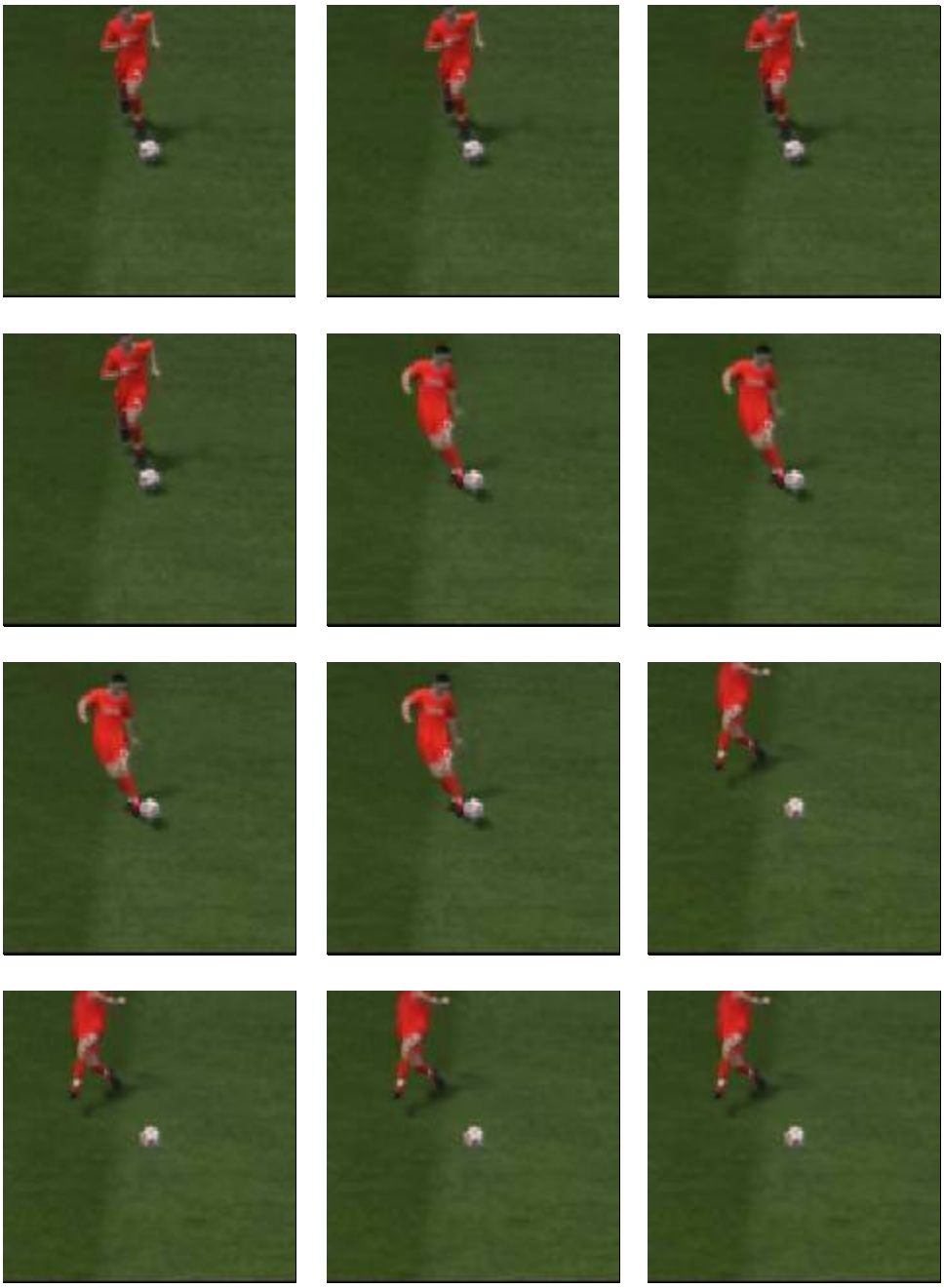


Figure B.3. Frames of a sample hitBall atomic action sequence

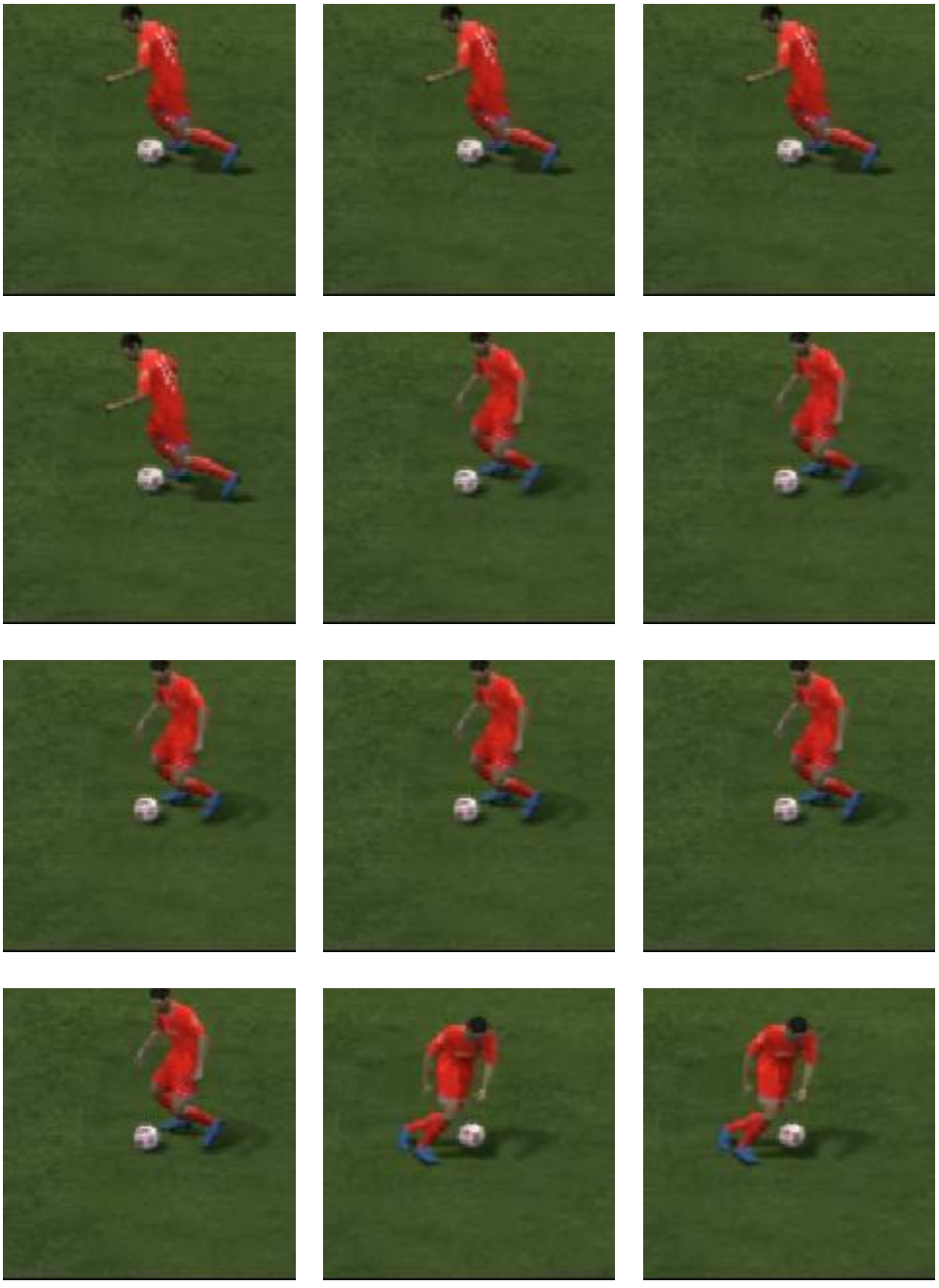


Figure B.4 Frames of a sample manevour atomic action sequence

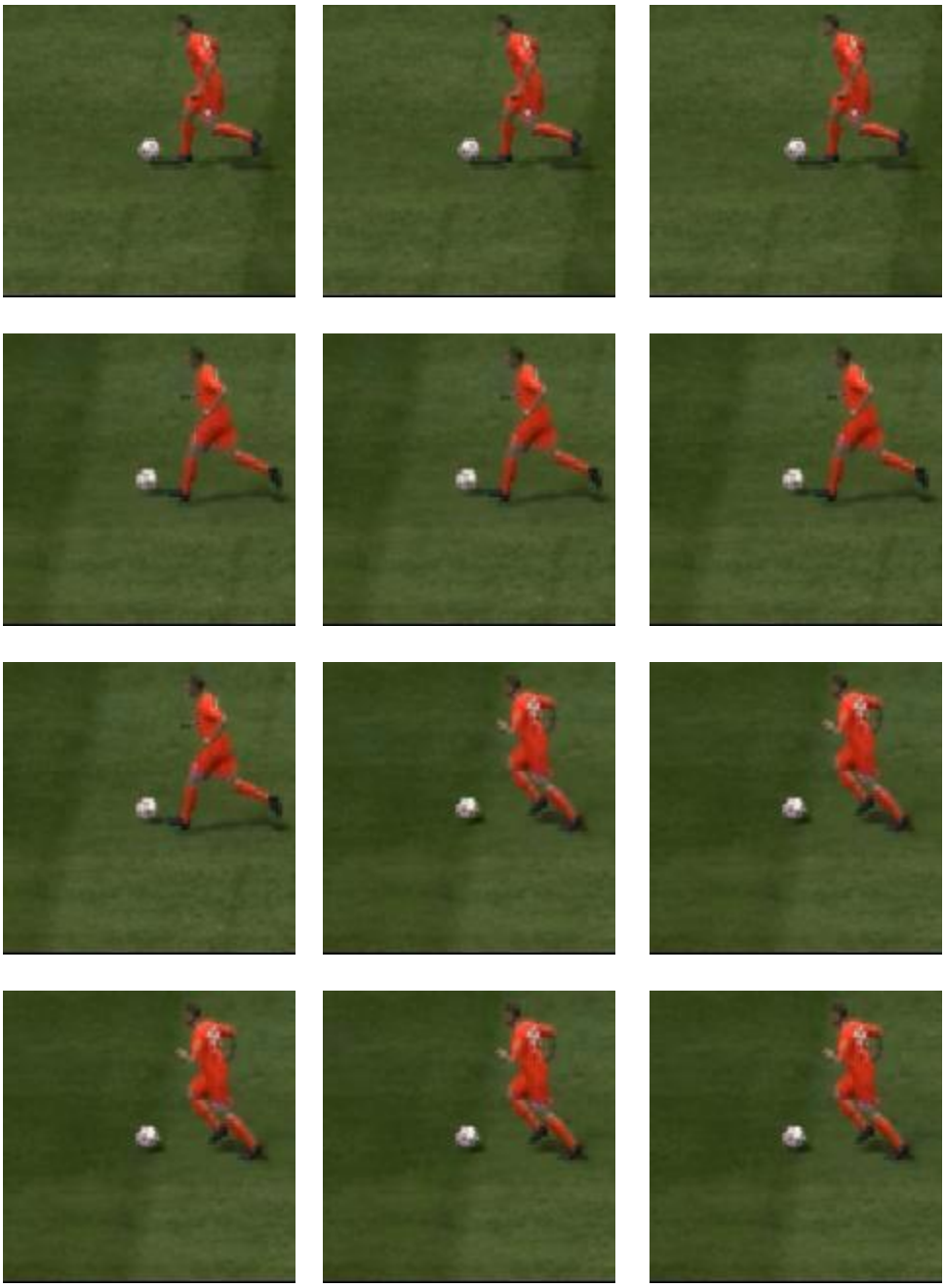


Figure B.5. Frames of a sample run atomic action sequence

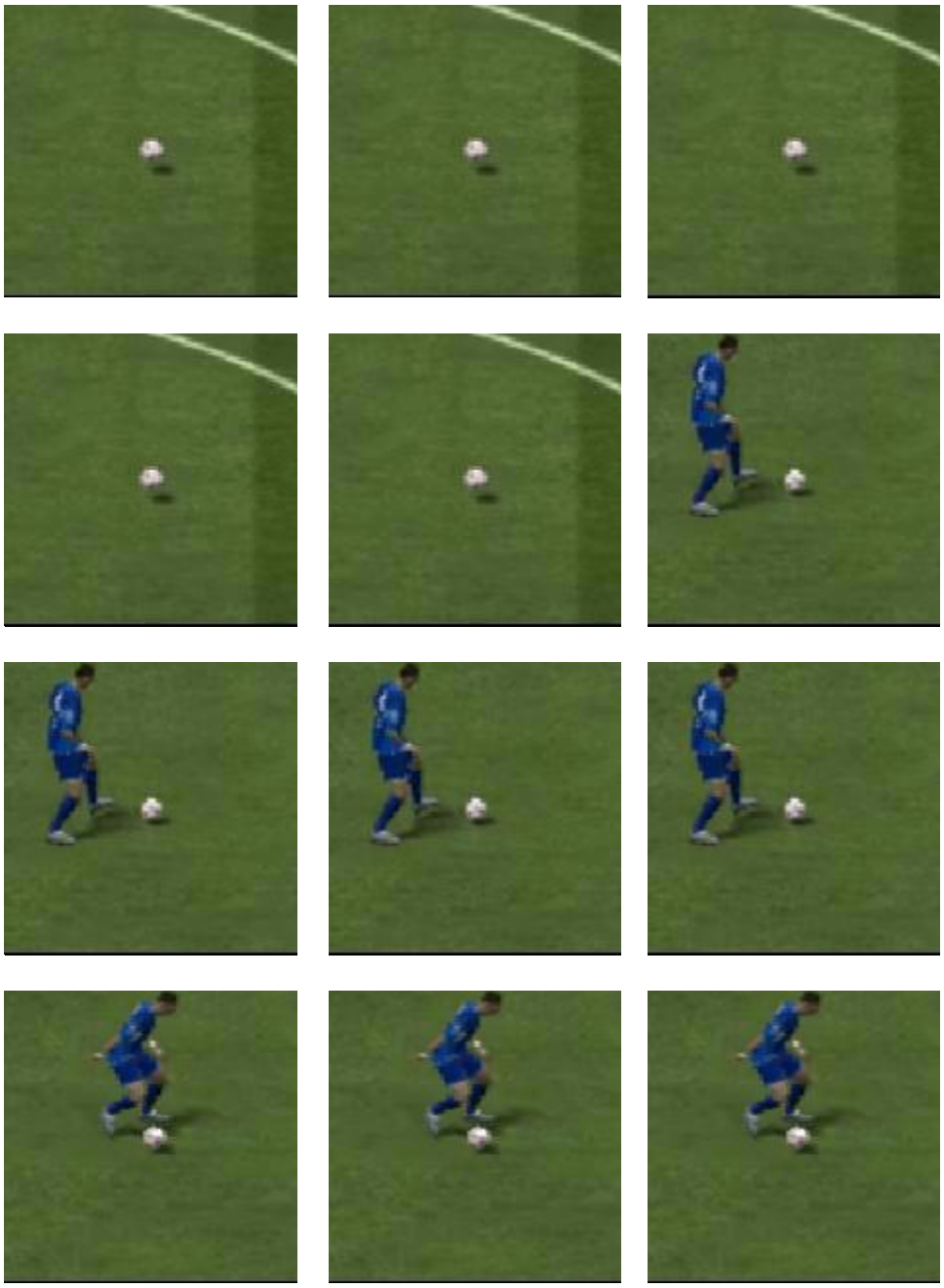


Figure B.6. Frames of a sample takeBall atomic action sequence

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Gökçe, Celal Onur

Nationality: Turkish (TC)

Date and Place of Birth: 15 January 1979, Kuyucak/Aydın

Marital Status: Married

Phone: +90 539 517 35 34

email: gokce@eng.ankara.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	Sabancı University Mechatronics Eng.	2002
BS	Boğaziçi University Electrical-Electronics Eng.	2000
High School	Afyon Süleyman Demirel Scientific High School, Afyon	1996

WORK EXPERIENCE

Year	Place	Enrollment
2009- Present	Ankara University Electrical-Electronics Eng.	Research Assistant
2005-2008	MİKES A.Ş.	Software engineer
2004-2005	MİLSOFT A.Ş.	Systems engineer
2003-2004	Turkish Armed Forces (compulsory)	Software engineer
2000-2003	Sabancı University	Research Assistant

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. Halıcı U., Gökçe O., A grammar based hierarchical architecture for human behavior understanding to recognize complex actions in football video game (submitted for journal publication)

2. Halıcı U., Gökçe O., "Mixture of poses for human behavior understanding", Proc. IEEE CSIP, (2013)

HOBBIES

Chess, Sci-fi novels, Robotics