Evaluation of a System for 2D Human Pose Estimation with Development of a System for Multiple Human Target Tracking Using Kalman Filter

Behnam Asadi
Master thesis at Faculty 3: Mathematics, Digital Media and Computer Science,
University of Bremen
March 2013

Supervisor: Dr. ing. Udo Frese
Second reviewer: Dr. Christoph Zetzsche

Contents

Acknowledgements **4**

1 Introduction **6**
1.1 Motivation and Aim of Work 6
1.2 Contribution of Thesis 11

2 Literature review and similar works **12**
2.1 Motion Analysis of Human Body Parts 12
2.2 Non Model Based Motion Analysis 14
2.2.1 Stick Figures 15
2.2.2 2D Contours 15
2.3 Model-Based Approaches 16
2.3.1 Stick Figures 16
2.3.2 2D Contours 17
2.3.3 3D Volumes 17
2.4 Human Detection 18
2.4.1 Classification of Human Detection 18
2.4.2 Human detection based on features extracted from image . 19
2.4.3 Summary 22

3 Detection of Human and Tracking of Multiple Targets **24**
3.1 Human Detection 24
3.1.1 Histograms of Oriented Gradients for Human Detection . 24
3.1.2 Overview of the Method 25
3.1.3 Gradient Computation 25
3.1.4 Weighted Vote into spatial and Orientation cells 26
3.1.5 Normalization of Descriptor Block 27
3.1.6 Detector Window 28

3.1.7 Classifier 28
3.2 Tracking of Target by Kalman filter 29
3.2.1 Introduction 29
3.2.2 Kalman Equation 30
3.2.3 Model and State Spaces 31
3.2.4 Data Association 32
3.2.5 Color Tracker for Evaluation of HOG and Kalman Filter . 32
3.2.6 HSV Color Model 33
3.2.7 Viewing OpenCV's HSV color space 34
3.2.8 Thresholding and Binarizing 35
3.2.9 Experiments 37

4 2D Body Pose Estimation **39**
4.1 Generating 2D Contour of Human Body 39
4.2 Principal Component Analysis 41
4.3 Dimensionality Reduction Using PCA 42
4.4 Creating the Training Set 42
4.5 Developed Software for Generating Human Contours 45
4.6 Discussion of The Model and Interpretation of Principal Components 49
4.7 CNS Contour Response 49
4.7.1 Contour Response 50

5 Experiments and Analysis **52**
5.1 CNS response from labeled contour on images with clutter noise . 53
5.2 CNS response from labeled contour on image with clear background 56
5.3 CNS response from optimal contour on image with clutter noise . 59
5.3.1 Accepting and Rejecting the Estimated Contours 59
5.4 CNS response from optimal contour on image with clear background 61

6 Results Analysis and Conclusion **64**
6.1 Analysis of Human Detector and Kalman Tracker 64
6.2 Analysis of Contour Generator and Pose Estimator 65
6.2.1 Analysis of Accepted Contours 65
6.2.2 Analysis of Rejected Contours 67
6.3 Future Work 73

7 Bibliography **78**

Acknowledgements

Primary thanks for support and priceless knowledge to my supervisor Dr. ing. Udo Frese who supported and encouraged me throughout this work, spent time to discuss problems, answered all my related questions and offered many useful suggestions, comments and feedback which were always knowledgeable and helpful. And special thanks for offering me this topic. I also would like to thank Oliver Birbach for his help and support during the thesis progress and also Dr. Christoph Zetzsche for taking time and reviewing the thesis.
In the end, I would like to appreciate my parents, in particular my mother, for her care and support.

Erklärung

Hiermit versichere ich, Behnam Asadi, diese Master Thesis ohne fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Alle Stellen, die wörtlich oder sinngemäßaus Veröffentlichungen entnommen sind, sind als solche kenntlich gemacht.

Unterschrift

Chapter 1

Introduction

1.1 Motivation and Aim of Work

Nowadays, robots are no more just machines in factory production lines and gradually entering in human everyday life taking part in their activity. Having intuitive and natural interaction is one of the key challenges for robots.
Robots that interact with humans not only need to detect humans in their environment but also need to keep tracking of their movement to avoid collision and also detect their attitude to allow them to follow human and make proper decision for interaction. Human detection, tracking and activity recognition is a highly active area of research not only in robot vision but generally in computer vision due to its necessity in applications such as video surveillance and advanced driver assistance systems. However, though significant work has been done in recent years a number of challenges have not yet been fully met.
To find out and recognize the difficulties and obstacles that a robot would face while having interaction with human in everyday life, a game that involves interaction between human player and robot has been designed and developed as a pilot application. In this game (Figure 1.1) the robot is located in the center of two imaginary concentric circles. The players in ***team A*** stand on inner circle and players of ***team B*** stand on the outer circle. The players in ***team B*** try to pass the ball to the robot. The robot is equipped with stereo cameras which enable him to track the ball and also the players. The robot has a mechanical arm on his top which enables him to kick back the coming ball. The cameras and the arm can rotate around vertical pivot during the game so robot can respond properly. The players in ***team A*** would try to intercept the ball. Therefore the robot should be

able to track the ball and predict the path of ball and also the location of players in both teams and estimate their body pose to find free players. Figures 1.2, 1.3 and 1.4 illustrate the images captured by the camera on the robot during the game. Figure 1.6 illustrates the robot during the game from an external camera.

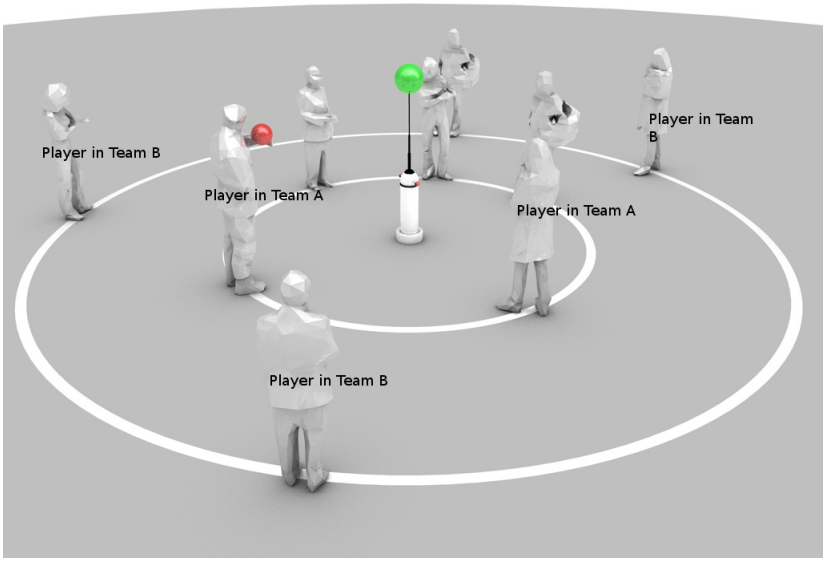


Figure 1.1: Schema of the game (graphic by Prof. Dennis Paul).

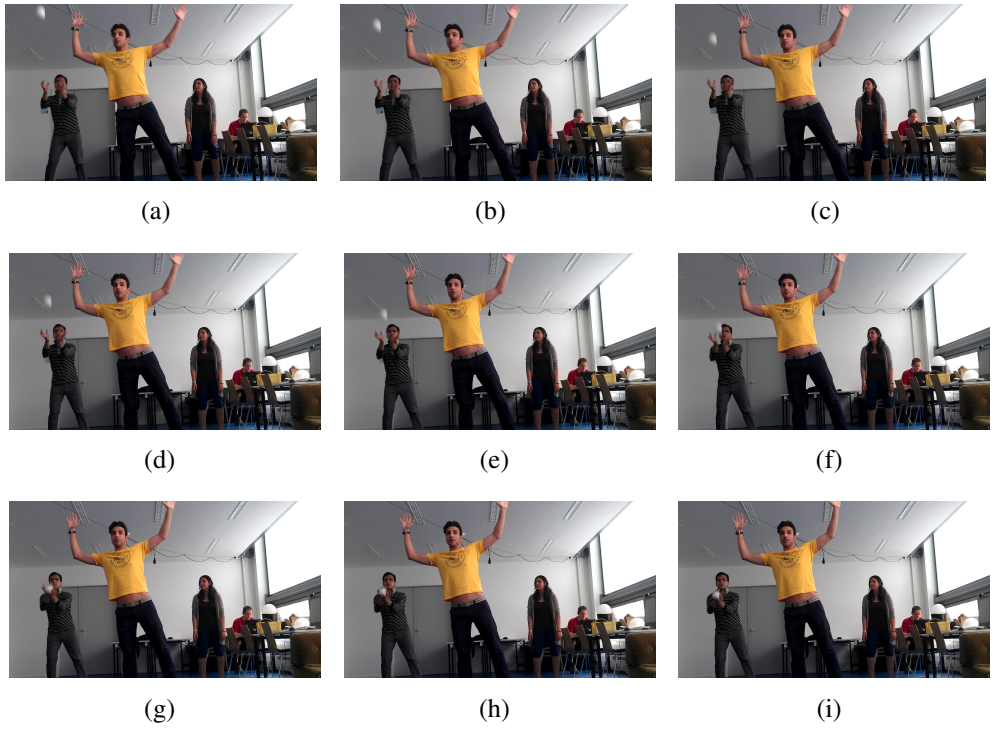
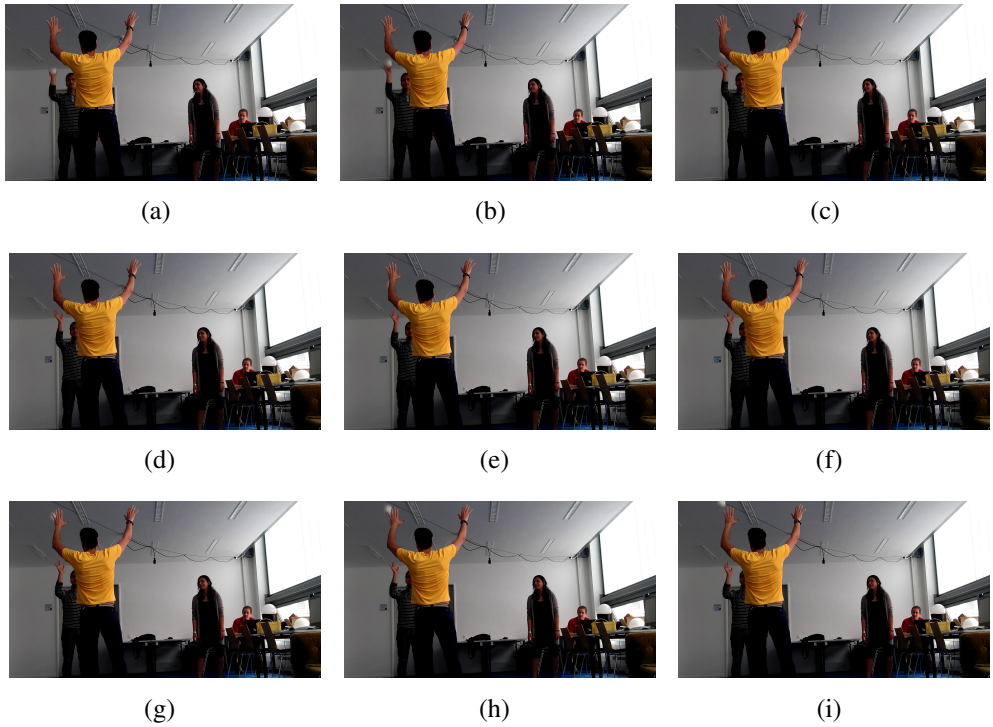


Figure 1.2: Screen shots from camera mounted on robot, the player in team A (the player in yellow shirt) tries to intercept the ball



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 1.3: Screen shots from camera mounted on robot, the player in team B throw the ball back to the robot and the player in team A tries to intercept the ball

Figure 1.4: Screen shots from camera mounted on robot, the player in team A finally intercept and catch the ball

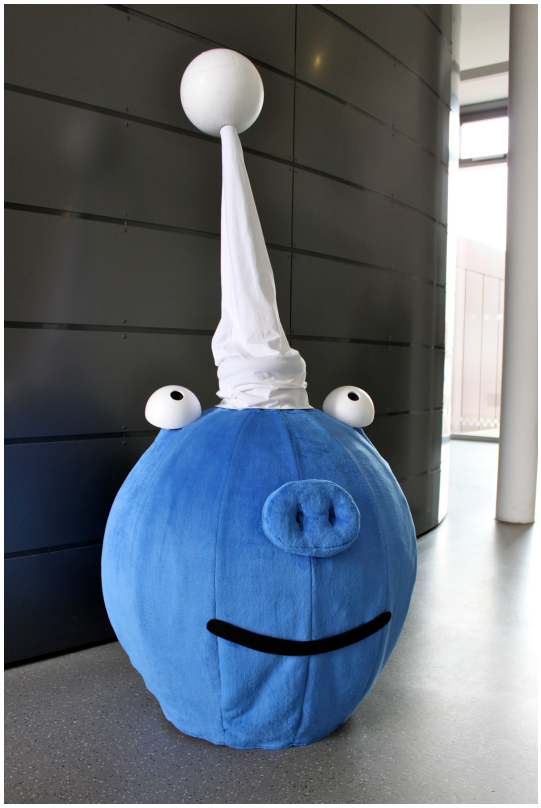


Figure 1.5: The Piggy robot (Photo by R. Wagner).

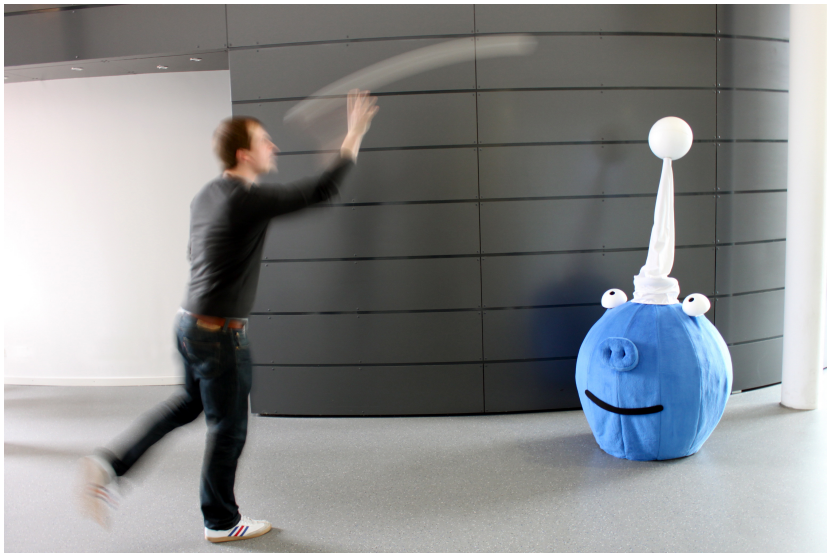


Figure 1.6: The Piggy robot during the game, kicking back the flying ball (Photo by R. Wagner)

1.2 Contribution of Thesis

The aim followed in this work is not only a building system for estimating the pose of the human in video as many approaches stated in the literature review already doing that, but the main focus is mainly equipping HOG (Histogram of Oriented Gradient) with Kalman filter to improve the performance of human detection on a mobile robot and also evaluation of CNS (Contrast Normalized Sobel) for human motion analysis in the form of estimating a 2D contour of human and testing it's performance for human pose estimation.

Chapter 2

Literature review and similar works

Understanding of human activity and motion analysis is a rising research area in computer vision which has gained a lot of attention in recent years. It is a multidisciplinary research area, involving various fields including image processing, computer vision, machine learning, pattern recognition, artificial intelligence, human kinematics, cognitive science, and even psychology.
Human motion analysis consist of following steps[1]:

- ***Body Structure Analysis***: Which could be model based or non-model based.
- ***Tracking Moving Human***: Either from a single view or multiple camera perspectives.
- ***Human Activity Recognition*** : Which could be based on state space model or template matching.

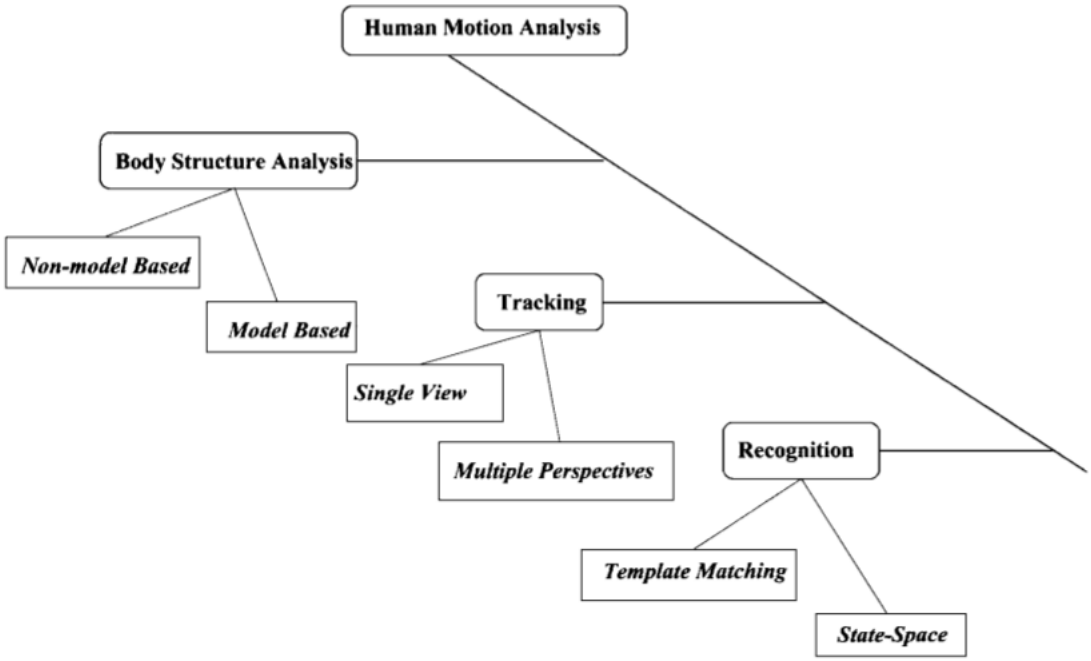
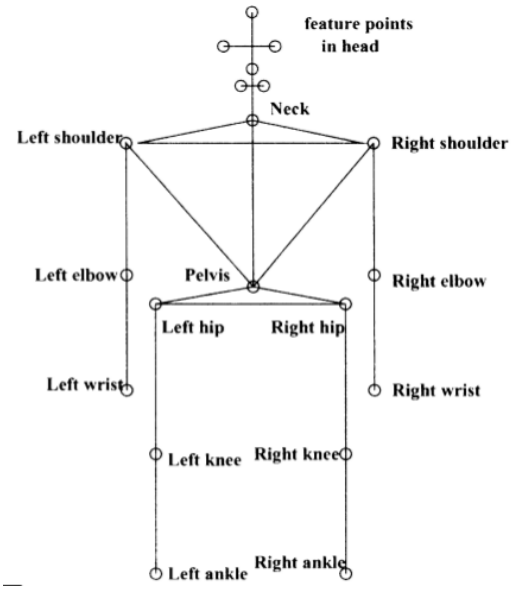


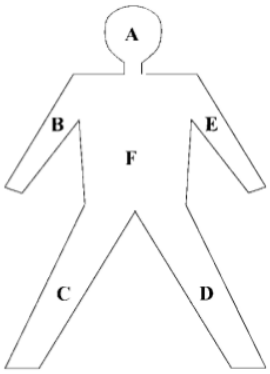
Figure 2.1: Steps in human motion analysis and different approaches for each step, graph retrieved from [1].

2.1 Motion Analysis of Human Body Parts

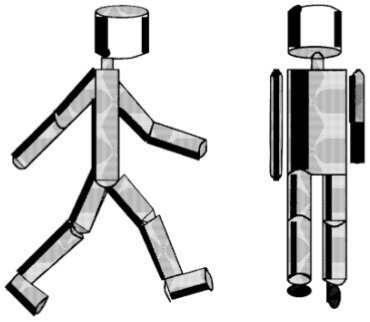
Aggarwal and Cai [1] have classified human motion analysis from different perspective of representation and also the model that has been used as well. Here we follow the same structure.
In computer vision reviews human bodies are mainly represented by **Stick Figures**, **2D Contours** or **Volumetric Models**. [2] With this representation body segments can be approximated as lines, 2D ribbons, and 3D volumes.



(a) Stick Figure



(b) 2D Contour



(c) Volumetric Model

Figure 2.2: Model representation for human body in motion analysis, images derived from work in [3] [4] [5] respectively.

Based on deploying a priori knowledge regarding shape models there are are two methods for motion analysis **model based** and **non model based**.
The complexity for representing the human body model changes from stick figures to 2D contours to 3D volumes In both model-based and nonmodel-based approaches.
The main idea of modeling human body with stick figures is based on the fact that

human body motion can be seen as movement of the supporting bones.
2D contours are projection of human body from 3D world into 2D image. 3D modeling of human body with volumetric models, such as elliptical cylinders, spheres and cones are able to describe the human body with more detail but require more parameters [6].
Both approaches have the similar procedure:

- feature extraction.
- feature correspondence.
- high-level processing.

The main difference between the two model based and non model based approach is in feature correspondence between consecutive frames. In model based methods there is an assumption that a priori shape models match in the images into a predefined model.
After model matched into the real images, feature correspondence would automatically be done.
In the case that there is no a priori shape models, correspondence between successive frames is based upon prediction or estimation of features related to position, velocity, shape, texture, and color. However these two approaches can also be combined at various levels to verify the matching between consecutive frames and, finally, to accomplish more complex tasks [1].

2.2 Non Model Based Motion Analysis

As Aggarwal and Cai stated in [1], "Most approaches to 2D or 3D interpretation of human body structure focus on motion estimation of the joints of body segments. When no a priori shape models are assumed, heuristic assumptions are usually used to establish the correspondence of joints between successive frames. These assumptions impose constraints on feature correspondence, decrease the search space, and eventually, result in a unique match". [1]

2.2.1 Stick Figures

Stick figures developed initially by Johansson [7], consist of line segments which are linked by joints. It is one the simplest form for representing human body. An

example of a stick figure can be seen in 2.2(a). In stick figures, motion estimation and activity recognition is based on motion of joints. In [7], Johansson marked joints as moving light displays (MLD). Afterward in a work by Rashid [8], he tried to use projected MLD to recover a connected human structure. Stick figures developed to 3D structure by [3] and [9].
"In a work by Webb and Aggarwal, they imposed the fixed axis assumption, which assumes that the motion of each rigid object (or part of an articulated object) is constrained so that its axis of rotation remains fixed in direction. Therefore, the depth of the joints can be estimated from their 2D projections. All of these approaches inevitably demand a high degree of accuracy in extracting body segments and joints. The segmentation problem is avoided by directly using MLDs that implies their restrictions to human images with natural clothing" [1], [3] and [9].

2.2.2 2D Contours

2D contours are also used for representing the human body. In 2D contours, the human body segments are comparable to 2D ribbons or blobs. An example of 2D contours can be seen in 2.2(b).
Shio and Sklansky [5] worked on 2D translational motion of human blobs. In this work they relied on motion for body decomposition. A method alike to optical flow used to compute magnitude and direction of the pixel velocity of blobs and then blobs were grouped based on that. After few frames, the velocity of each part converges to a global average.
In [10] Kurakake and Nevatia find correspondences between extracted ribbons to locate the joints of walking humans. Correspondences were conducted using various geometric constraints. Joints were labeled the center of the area of two overlapped ribbons .
Kakadiaris [11] used a physics-based framework for body segmentation and joint location from image sequences of the moving subject. In this work joints are identified when the body segments connected to it have motion.
Rowley and Rehg [12] worked on segmentation of optical flow fields of articulated objects. They add kinematic motion constraints to each pixel data to extend the work in [13]. In this work they have used EM (expectation maximization) in a way that segmentation is done in the E-step and motion analysis in the M-step. These two steps are computed iteratively. The motion addressed in the paper is restricted to 2D affine transforms.

2.3 Model-Based Approaches

In the previous section I reviewed non model based methods. In non model based methods no a priori shape models available therefore feature correspondence between consecutive frames is difficult. To overcome this problem predefined models for feature correspondence and body structure recovery is being used. In the following we review model based approaches.

2.3.1 Stick Figures

Chen and Lee [14] proposed a model 2.2(a) that represent the head, torso, hip, arms, and legs. The model consists of 14 joints and 17 lines connecting them. Physical constraints were employed in order to analysis the gait. With this model they recovered the 3D configuration of a moving human from a projected 2D image. The need to accurate extract 2D stick figures and brute forcing all possible combinations of 3D configurations from 2D projection made the method computationally expensive [1].
Bharatkumar, Daigle, Pandy, Cai, and Aggarwal [15] in their work tried to construct a general kinematic model for gait analysis for walking humans. They used stick figures to model the lower limbs of the moving human body. "Stick figures were obtained from ordinary images of persons dressed in tight fitting clothes without any markers by using the medial axis transformation. The body segment angle and joint displacement were measured and smoothed from real image sequences, and then a common kinematic pattern was detected for each walking cycle. A high correlation was found between the real images and the model" [1]. Being sensitive to the perspective angle that image has been acquired and view-based dependency is the main drawback of this work.
Iwasawa [16] in his paper introduces a new real-time method for estimating the posture of a human (in thermal images) regardless of the background and lighting conditions. Images were acquired by an infrared camera. The height of the human image and the distance between the subject and the camera were precalibrated.
The orientation of the upper half of the body was obtained as the principle axis of inertia of the human silhouette. After that, significant points such as the top of the head, the tips of the hands and foot are heuristically located. The position of elbow and knee were detected by using a genetic algorithm. This work has a similar drawback like [15] and the model depends on the view.

2.3.2 2D Contours

Leung and Yang [17] developed a vision system "First Sight" for labeling the outline of a moving human body and applied that to estimate human subjects performing gymnastic movements. Two main processes are implemented in First Sight. The first process extracts the outline of a moving human body from an image sequence by subtracting consecutive frames and finding moving blobs and then finding edges. "Next, a sophisticated 2D ribbon model 2.2(b) is applied to explore the structural and shape relationships between the body parts and their associated motion constraints.
A spatial-temporal relaxation process was proposed to determine if an extracted 2D ribbon belongs to a part of the body or that of the background. In the end, a description of the body parts and the appropriate body joints is obtained, based on the structure and motion. Extensive knowledge of the structure shape, and posture of the human body is used in the model" [1].

2.3.3 3D Volumes

Sensitivity to the angle between camera and subject (view dependency) is the main drawback for any 2D model. To overcome this problem 3D volumetric models have been proposed. 3D volumetric models are computationally more expensive and require more parameters. Pioneering works for 3D modeling was done by Badler [18] and also Yamamoto [19].
In [18] Badler and O'Rourke applied a volumetric model for 3D model-based human motion analysis that consists of 25 joints and 24 segments while taking into account motion constraints for human body parts. This work involves four major steps: **prediction, simulation, image analysis, and parsing**. In the prediction step the image analysis phase based on the previous prediction tries to find the body segments. After the search space for 3D location of body segment is narrow down, the parser tries to fit the location-time relationships into certain linear functions. In the following in the prediction phase, the predictor deploys the determined linear functions to estimate the position of the parts in the next frame. In the end a simulator with extensive knowledge of the human body translates the prediction data into corresponding 3D regions, which in the next iteration will be used by the image analysis phase.
Another way to represent human body in 3D volumes is elliptical cylinders.
For an elliptical cylinder the length of the axis and the major and minor axes of the ellipse cross section is needed so it needs fewer parameters in comparison with

[18],
Hogg [20] used the cylinder model and developed a computer program (WALKER) based on that which recovers the 3D structure of a walking person.
Rohr [21] used the 14 elliptical cylinders to represent the human body with the origin of the coordinate system at the center of the torso. Rohr used eigenvector line fitting to outline the human image and then fitted the 2D projections into the 3D human model using a similar distance measure. Both works [20] and [21] tried to generate 3D descriptions of a human walking by modeling.

2.4 Human Detection

In all 2D or 3D approaches and model based or non model based methods, the very first thing that any of them needs is an automatic methods for detecting the human from a visual input which could be an image or a video. Once this step done, depending on the application, the system can go further into processing details of understanding the human activity. In the following we discuss the techniques for human detection in visual input.

2.4.1 Classification of Human Detection

There are mainly two approaches for the problem of detecting human in an image or video (sequence of images).
The first one is based on background subtraction techniques and requires a fixed background. These method subtract frames from previous frame or from a fixed frame and finds the foreground objects from that. Afterward based on some characteristics such as shape, color, or motion the algorithm would classify these objects into like human, animal, vehicle, etc. Ogale [22] compiled most recent works based on background subtraction and the feature that is being used to detect humans in foreground objects and summarized in Table 2.1.

Authors	Subtraction technique	Feature for finding human
Wren et al.[23]	Color/Ref. image	Color,contour
Beleznai et al.[24]	Color/Ref. image	Region model
Haga et al.[25]	Color/Ref. image	F1-F2-F3
Eng et al.[26]	Color/Ref. image	Color
Elzein et al.[27]	Motion/Frame diff.	Wavelets
Toth and Aach[28]	Motion/Frame diff.	Fourier shape
Zhou and Hoang[29]	Motion/Frame diff.	Shape
Yoon and Kim[30]	Motion + Color diff.	Geom. Pix. Val.
Xu and Fujimura[31]	Depth	Motion
Han and Bhanu[32]	Infrared	IR+color
Jiang et al.[33]	Infrared	IR+color

Table 2.1: Methods for human detection based on background subtraction with feature for finding human [22].

2.4.2 Human detection based on features extracted from image

The second group of approaches for human detection doesn't require a fixed background but tries to detect humans based on features such as shape (in the form of contours or other descriptors), color (skin color detection), motion, or combinations of these. Ogale [22] compiled some widely cited works for human detection based on finding and matching features (without need for background subtraction). Table 2.2 gives an overview of different methods for human detection based on extracted features that will be discussed in the following section.
Due to the nature of the game and because the camera moves, the background is not fixed so human detection based on background subtraction could not be applied. Furthermore the background subtraction methods are designed for indoors application and not outdoors where situation such as light condition might changes. Therefore our attention is on the second group specially on the HOG (Histogram of Oriented Gradient).

Authors	Model for Human	Classifier
Cutler and Davis[34]	Periodic Motion	Motion similarity
Utsumi and Tetsutani[35]	Geom. Pix. Val.	Distance
Gavrila and Giebel[36]	Shape template	Chamfer dist.
Viola et al.[37]	shape+motion	Adaboost cascade
Sidenbladh[38]	Optical flow	SVM (RBF)
Dalal and Triggs[39]	Hist. of gradients	SVM (Linear)

Table 2.2: Methods for human detection based on extracted features [22].

Robust Real-Time Periodic Motion Detection [34]

Cutler and Davis [34] have developed a new technique to detect and analyze periodic motion such as walking.
The video from a camera is first stabilized and then frame differencing and thresholding is performed to detect independently moving regions. Next morphological operations are applied to obtain a set of tracked objects. By tracking objects of interest, an object's self-similarity is computed as it evolves during the time. Each segmented object is aligned along the time axis. By using similarity measures such as correlation for each object, a self-similarity matrix is computed which is periodic for periodic motions.
To track and classify objects a real-time system has been implemented which uses periodicity to classify objects. Cutler and Davis showed that their method can distinguish the motion of a human from a dog. The system is not only able to detect human periodic motion, but it also able to provide other information regarding gait analysis such as stride length [34],[22].

Detecting human motion with support vector machines [38]

Sidenbladh [38] in his work presented a method for detection of humans in video sequences. The main application of the method is outdoor surveillance.
The appearance of humans varies hugely due to shading, different light condition (amount and direction of light), clothing, weather, etc.
The idea in this work is to detect patterns of human motion which are relatively independent of appearance and environmental factors. The authors also observed that it is harder for a person to camouflage motion but easier to change appearance [38], [22].

In this work, a training set of examples of human and non-human motion optic flow is used. The subjects are moving in different angles to the camera plane, on different image scales. Then a support vector machine (SVM) with a radial basis function (RBF) kernel is trained with the training set to create a human classifier. The trained SVM searches for human-like motion patterns on optical flow of images. In images which humans are partially occluded this method may not be suitable for detection [22].

Detecting pedestrians using patterns of motion and appearance [37]

This work is about a pedestrian detection system that combines image intensity information with motion information while previous works have built detectors based on motion information or detectors based on appearance information.
This method detect pedestrian in the input video which mostly have upright poses. Viola et al. used a detection that take advantage of both motion (scanning over two consecutive frames) and appearance (using an AdaBoost classifier) information to detect a walking person. The training data set consists of images and videos of human and non-human examples. The implemented code can analyze 4 frames per second, capable of detecting pedestrians with very small scale (20×15 pixels). This work builds on the detection work of Viola and Jones.
The Main contributions of this method is:

- Development of a representation of image motion which is extremely efficient
- Implementation of a state of the art pedestrian detection system which operates on low resolution images under difficult conditions (such as rain and snow).

The static detector uses images as inputs and efficiently extracts simple rectangular features using integral images. A cascade of classifiers is created to achieve superior detection and low false positives. Each stage of the classifier is trained on true and false positives from the previous stage using Adaboost to select weak classifiers (which are the simple rectangular features mentioned earlier).
Similarly the dynamic detector is trained with combination of static and motion rectangular features. Both detectors are fast on a large pedestrian data set with good result on detection. [22]

Human detection based on histogram of oriented gradient [39]

Dalal and Triggs in this work used a histogram of gradients as the feature space for making a classifier. The work is based on the fact that shape of a object can be well represented by a distribution of local intensity gradients or edge directions. For that purpose the image is divided into small spatial parts (cells) and then the histograms of edge orientations computed over all the pixels of the cell.
Several parameter for orientation, spatial binning resolutions and normalization schemes tested for finding the maximum performance. A linear SVM classifier trained on the gradient histogram features from a dataset of human and non-human examples is used. New input images can be classified by the trained classifier for detecting humans. In section 3.1.1 we discuss this approach more in detail.

2.4.3 Summary

Several methods in the recent works for human detection from visual input have been reviewed in this section. The reviewed literatures classified in two main category. Methods in first category work based on the background subtraction technique. These methods either subtract two consecutive frames or they subtract each frame from a fixed frame (a background frame). They also use various approaches for detecting humans in the regions remained from subtracting like color of region, shape of region, infrared scan, etc. These method are widely used for static cameras. Methods in the second category work based on extracting features from image and training a classifier for human/ non human images. These method could be used for both static cameras and mobile robots. Overall, there seems to be an increasing trend towards methods which extract features from image and directly operates on that rather than background subtracting. Due to the nature of the game which requires a moving camera the method in the first category are not applicable in this work. Among the methods in the second category, "Histograms of Oriented Gradients" has been used for human detection due to following criteria:

1. Very high rank of referencing in recent publication and literature (4782 times).
2. Support in computer vision groups.
3. Very good rate of success in human detection.
4. Robustness against cluttered backgrounds different illumination condition.

5. Widely support in open source community.

6. Various implementations and support in OpenCV.

Chapter 3

Detection of Human and Tracking of Multiple Targets

The very first step to estimate the pose of the player is detecting the human player in the video. Detecting humans in images is a challenging task due to the wide range of poses and appearance that human can have, in cluttered backgrounds under difficult illumination. In this work HOG has been used for human detection due to its characteristic such as robustness to light condition, shading and cluttered backgrounds, high performance speed in run time. In OpenCV 2.3 and later HOG has been implemented as part of "Object Detection" namespace. In this work codes from OpenCV has been used and details of algorithm and parameters for the codes has been discussed in the following sections.

3.1 Human Detection

3.1.1 Histograms of Oriented Gradients for Human Detection

Dalal and Triggs [39] in their study showed that using locally normalized Histograms of Oriented Gradient (HOG) as descriptors provides excellent performance in comparison to other existing feature sets including wavelets [41], [42], edge orientation histograms [43], SIFT descriptors [44] and shape contexts [45].

3.1.2 Overview of the Method

The main idea in HOG is that distribution of local intensity gradients or edge directions can be used as good descriptor for shape and appearance of object, even without precise knowledge of the corresponding gradient or edge positions. In HOG this idea has been done by dividing the image window into small spatial regions which is called "cells" and afterward calculating a local 1-D histograms of edge orientations over the pixels of each cell. In order to make it robust to different conditions of illumination, shadowing, etc, local responses have been contrast-normalized. This has been achieved by making a block of cells and then accumulating a measure of local histograms over each blocks. This value has been used for normalizing all of the cells in the block. Dalal and Triggs [39] referred to this normalized descriptor blocks as Histogram of Oriented Gradient (HOG) descriptors. The use of orientation histograms is not a novel concept and has been applied for hand gesture recognition and for automatic face and gesture recognition [43] or for pattern recognition. It reached the full state of development in in Lowes Scale Invariant Feature Transformation (SIFT)[44] when combined with local spatial histogramming and normalization.

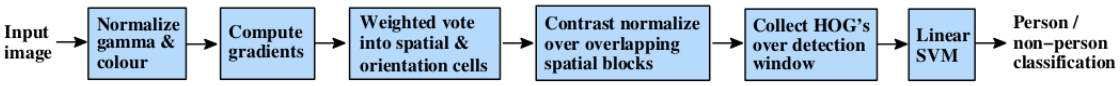


Figure 3.1: Feature extraction and object detection in HOG, Tiling the detection window in an overlapping grid of HOG descriptors and then using a SVM based window classifier gives the human detection chain. Image acquired from [39].

3.1.3 Gradient Computation

The performance of the detector is very sensitive and depended on the way in which gradients are computed.
Dalal and Triggs tested gradients using different masks with different parameters for smoothing functions.
Various mask were tested including 1-D point derivatives, cubic-corrected, 3×3 Sobel masks and 2×2 diagonal ones $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$.
Gaussian smoothing function were used for smoothing. Several smoothing scales were tested including σ =0 (none). Dalal and Triggs showed that the simple 1-D

$[-1, 0, 1]$ masks at σ =0 work best. Larger masks decrease the performance, and smoothing declines it significantly. In colorful images, at first a gradient for each channel is calculated and then the one with the largest norm used as the pixel's gradient.

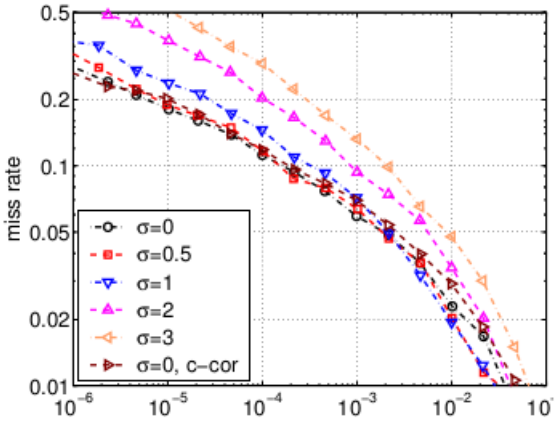


Figure 3.2: Effect of gradient scale sigma, false positives per window. Image acquired from [39]

Details of gradient computation for HOG

The details of the HOG computation are illustrated as follows.
If H is our gray scale image and and H(x, y) is the gray value of H in pixel (x,y):
Horizontal gradient $px(x, y)$:

$$px(x, y) = H(x + 1, y) - H(x - 1, y) \tag{3.1}$$

Vertical gradient $py(x, y)$:

$$py(x, y) = H(x, y + 1) - H(x, y - 1) \tag{3.2}$$

Magnitude of gradient for each pixel:

$$G(x, y) = \sqrt{px^2(x, y) + py^2(x, y)} \tag{3.3}$$

Orientation of gradient for each pixel:

$$Orientation(x, y) = arctan(px(x, y)/px(x, y)) \tag{3.4}$$

3.1.4 Weighted Vote into spatial and Orientation cells

In the next step a histogram of the angles of the gradient of each pixel within a cell should be formed. The orientation bins are evenly divided over 0-180 degree (9 bins). Each pixel adds a vote to the respective bin that it falls in. The vote is a function of the gradient magnitude at the pixel. It might be the followings:

- The gradient magnitude itself.
- The gradient magnitude square.
- The gradient magnitude square root
- A clipped form of the magnitude representing soft presence/absence of an edge at the pixel.

It has been shown by [39] that magnitude itself gives the best results.
The vote of each pixel in the bin is: $V_x(x, y)$

$$V_x(x, y) = \begin{cases} G(x, y) & \text{if } Orientation(x, y) \in bin_k \\ 0 & \text{Otherwise} \end{cases} \tag{3.5}$$

3.1.5 Normalization of Descriptor Block

Variations in illumination in different parts of image change the gradient strengths over a wide range. In order to suppress these variations and also make it less sensitive to illumination condition a local contrast normalization is necessary for good performance. Dalal and Triggs evaluated different schemes for normalization. Most of these schemes group cells into larger blocks and contrast normalize each block. The final descriptor is then a vector formed from the normalized cell responses of all the blocks in the detection window.
During normalization, blocks typically overlapped the other blocks so that each scalar cell response contributes several components to the final descriptor vector. This overlapping dramatically increases the performance. [39]
Dalal and Triggs evaluated four different block normalization schemes. Let v be the unnormalized descriptor vector and $\|v\|_1$ and $\|v\|_1$ 1-norm and 2-norm respectively and ϵ be a small constant. The schemes are:

1. L2-norm: $v \longrightarrow v/\sqrt{\|v\|_2^2 + \epsilon^2}$
2. L2-Hys: which is L2-norm while limiting the maximum values of v to 0.2.
3. L1-norm: $v \longrightarrow v/\sqrt{\|v\|_1 + \epsilon}$
4. L1-sqrt: $v \longrightarrow \sqrt{v/(\|v\|_1 + \epsilon)}$

Dalal and Triggs in their work showed that L2-Hys, L2-norm and L1-sqrt all perform equally well but L1-norm reduces performance by 5% and omitting normalization reduces it by 27% [39].

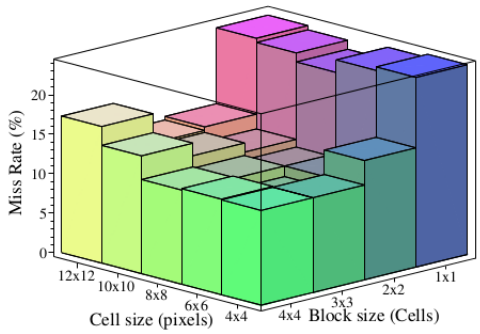


Figure 3.3: 3×3 blocks of 6×6 pixel cells perform best. Image acquired from [39]

3.1.6 Detector Window

Dalal and Triggs showed that detection window with size of 64×128 and about 16 pixels of margin around the person on all four sides, would provide the best result. Decreasing it from 16 to 8 pixels (48×112 detection window) decreases performance. Same detection (64×128) has been used for detecting player in this work.

3.1.7 Classifier

Dalal and Triggs trained a linear SVM with a soft margin of 0.01 (C=0.01). Same configuration (linear SVM with soft margin of 0.01) has been used in this work for detection players.

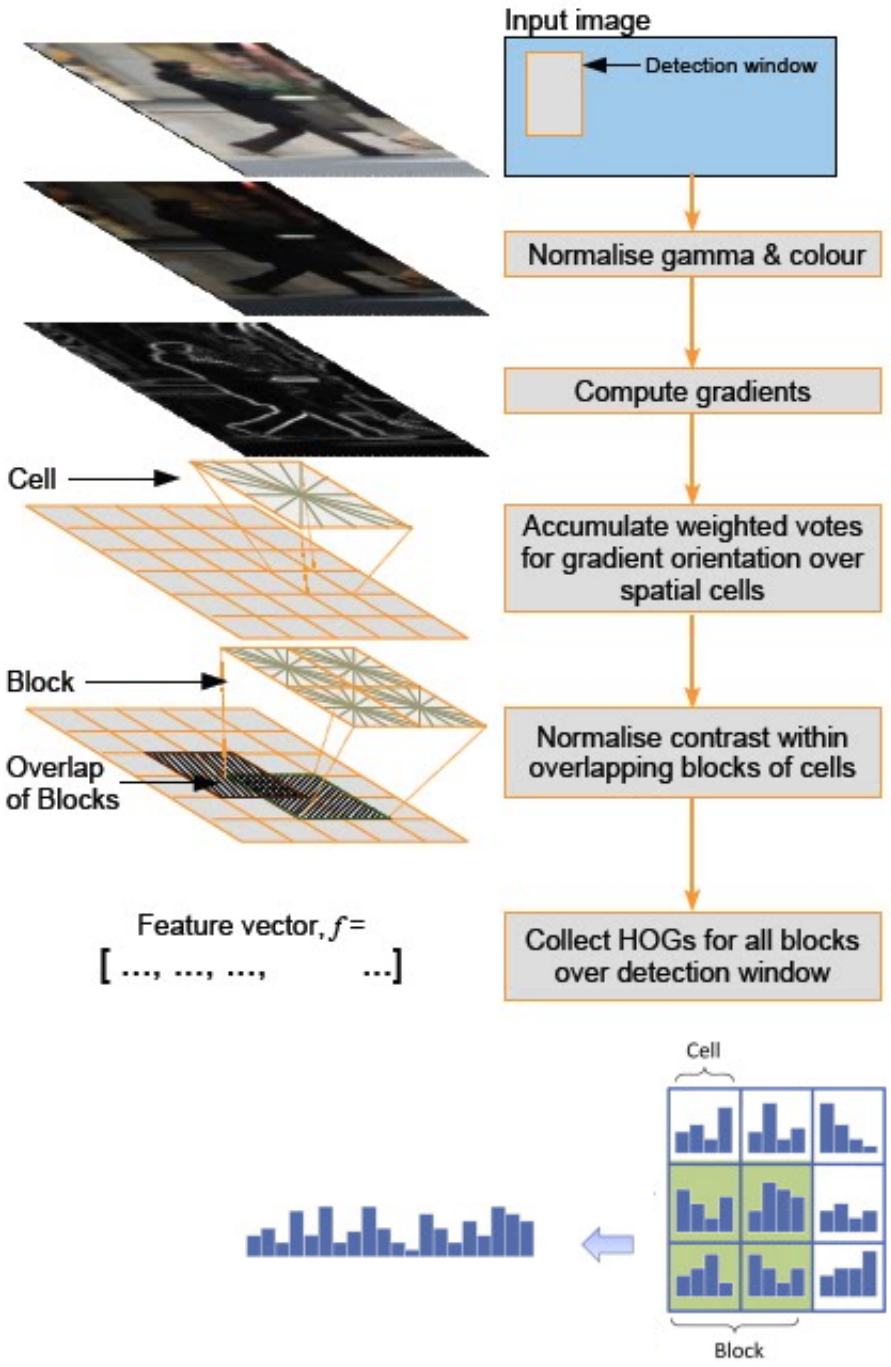


Figure 3.4: Overview of HOG, The detector window is tiled with a grid of overlapping blocks, Each block contains a grid of spatial cells. For each cell, the weighted vote of image gradients in orientation histogram is accumulated. These are locally normalized and collected into one big feature vector. Images acquired from [46].

3.2 Tracking of Target by Kalman filter

Despite the robustness of HOG for detecting humans, it still fails in some images. Furthermore in cases that a player is behind of an other player and his/her body is occluded, HOG is unable to find that particular player. In the following, we introduce a Kalman filter for tracking humans to sole this problem.

3.2.1 Introduction

For the first time, In 1960, R.E. Kalman published his paper describing a recursive solution to the discrete-data linear filtering problem[47]. The Kalman filter is a set of mathematical equations that provides a recursive mean to estimate the state of a process, in a way that error is minimized statistically. Kalman filter combines all information including:

- knowledge of the system and measurement.
- Statistical description of the noise in system and measurement errors and uncertainty in the dynamics models.
- initial conditions of states.

The word recursive means that, the Kalman filter does not require all previous data to be kept in storage and reprocessed every time a new measurement is taken. The filter is robust in estimating states of system in past, present and future. [48] There are two main assumption in the Kalman filter:

- Underlying system has linear dynamics.
- All error terms and measurements have a Gaussian distribution.

When the system is non-linear, typically it is linearized by approximating the system between small intervals [49]. Kalman filter has two main groups of equations:

1. Time update equations
2. Measurement update equations.

The time update equations project forward the current state and error covariance estimates to obtain the a priori estimates for the next time step. They can also be thought of as predictor equations.

The measurement update equations incorporate the new measurements into the a priori estimate to obtain an improved a posteriori estimate. Measurement update equations can be thought of as corrector equations [50].

3.2.2 Kalman Equation

Equations 3.6 and 3.7 express the Time update and equations 3.8, 3.9 and 3.10 express Measurement update respectively.

Time Update "Prediction"

1) Projecting State:

$$x_k^- = Fx_{k-1} + Bu_{k-1} + w_k \tag{3.6}$$

2) Projecting Error Covariance:

$$P_k^- = FP_{k-1}F^T + Q_{k-1} \tag{3.7}$$

Measurment Update "Correction"

1) Computing Kalman Gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k T + R_k)^{-1} \tag{3.8}$$

2) Updating Estimation with Measurment:

$$x_k = x_k^- + K_k(z_k^- - H_k x_k^-) \tag{3.9}$$

3) Updating Error Covariance:

$$P_k = (I - K_k H_k)P_k^- \tag{3.10}$$

Here x_k is an $n \times 1$ dimensional matrix of system state. F is an $n - by - n$ matrix and which sometimes called the transfer matrix.
u_k is a control inputs vector which allows external controls on the system.
B is an n-by-c matrix that relates these control inputs to the state change.

w_k is the process noise, associated with random events or forces that directly affect the actual state of the system. The assumption is components of w_k are unknown and have Gaussian distribution $N(0, Q_k)$ for some $n - by - n$ covariance matrix Q_k. Q can change during the time, but often it does not.
z_k m-dimensional vector and is the measurements that may or may not be direct measurements of the state variable x_k.

$$z_k = H_k x_k + v_k \tag{3.11}$$

H_k is an $m - by - n$ matrix which connects measurement and state space. v_k is the measurement error, which is also assumed to have Gaussian distributions $N(0, R_k)$ for some $m - by - m$ covariance matrix R_k

3.2.3 Model and State Spaces

OpenCV 2.2 and later comes with classes for implementing Kalman filter. In this work I have used the OpenCV classes and codes for creating the Kalman tracker and the contribution is designing the model and the state space. Our space $\mathbf{X}_k$ is:

$$\mathbf{X}_k = \begin{bmatrix} x_k \\ y_k \\ x'_k \\ y'_k \end{bmatrix}$$

Where x_k and y_k are the center of gravity of player in the x, y axis and x'_k , y'_k are derivative with respect to time (velocity in x, y axis) and transfer matrix F is:

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We measure only position of player, therefore the observation vector $\mathbf{Z}_k$ is:

$$\mathbf{Z}_k = \begin{bmatrix} z_x \\ z_y \end{bmatrix}$$

z_x and z_y are the center of gravity of player in the x, y axis.
This implies that H is:

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

In our case we supposed that the velocity of the players is constant. We would choose R_k based on our estimate of how accurately we have measured the player's position.

3.2.4 Data Association

For our experiment we limit the number of player to two (but it could be any arbitrary number bigger than two) depending the position and location players HOG might detect zero, one or two player. Furthermore HOG is memory-less which means the algorithm doesn't provide any information regarding the detected player is related to which player in the previous frame. In the other words, Player A might be the first detected human in video and player B the second person detected by HOG, but in the next frame player B detected first and player A be the second human detected by HOG. Thus there should be a mechanism for data association. To overcome this problem, we calculate the Euclidean distance between found players in current frame and detected players in previous frames. Then player A in current frame is one with minimum distance to player A in previous frame, the same is true for player B.
In the case that the number of detected players by HOG is less than the expected number of players (which is two in our experiment) then we use Kalman filter to estimate the location of the player.

3.2.5 Color Tracker for Evaluation of HOG and Kalman Filter

In order to evaluate the accuracy of prediction of the developed Kalman filter, there is a need for ground truth. There are two possibilities for labeling and verifying the result of Kalman filter :

- Manually labellings players in images for each frame and check if COG of person falls in the bounding box predicted by Kalman filter.
- Automatically detect players accurately and then check if COG of person falls in the bounding box predicted by Kalman filter. In this approach in all the frames human should be detected without missing any frame.

Due to the large number of frames, the second approach has been favored. For that purpose a color tracker has been developed and the result of color tracker has been used as ground truth. Players are being distinguished by two different colors, in this case one player has a red shirt and the other one yellow shirt. RGB color format can represent any standard color or brightness. In RGB color format colors produced by mixing red, green and blue. These are called RGB additive primary colors. Usually this is stored as a 24-bit number, using 8-bits for each color component (0 to 255). For example, White is: 255 of Red + 255 of Green + 255 of Blue.
The resulting mixtures in RGB color space can reproduce a wide variety of colors But the relationship between the amount of each color and the resulting color is not intuitive. furthermore RGB values will vary a lot depending on strong or dim lighting conditions and shadows, etc. To track our player based on color, an other color model space which has better interface between input parameters and generated color was needed. in following we introduce better alternate for RGB color model in determining players based on color.

3.2.6 HSV Color Model

The HSV color model (Smith [51]) was developed in the 1970s and is a representation of points in an RGB color model which provides better understanding of output color. HSV stands for Hue, Saturation and Value. HSV color model has cylindrical coordinate model.
Figure 3.5 illustrate the Hue, Saturation, and Value in cylindrical coordinate.

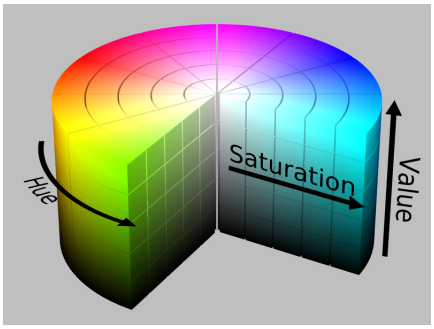


Figure 3.5: HSV color cylinder.

1. In HSV cylinder, ”Hue” is the angle around the central vertical axis. In the

other words, one could say Hue is the color. Hue is often represented as a circular angle and has angular dimension, Which mean value at 0° is equal to value at 360°. In HSV, the red color starts at 0°, green color starts at 120° and by going forward the blue would be visited at 240°, and then wrapping back to red at 360°.

2. "Saturation" is the distance from the axis. In the other words saturation means the amount of greyness, therefore a saturation value around 1.0 is a very strong color and saturation value of 0 means it is grey looking.

3. "Value" corresponds to the distance along the axis and in means the brightness of the pixel, therefore value near 0.1 is black and near 0.9 is white. [52]. [1]

[1] OpenCV's HSV color model is different to the more common HSV color models in graphics software. In OpenCV, Hue varies between 0 to 179 , and a Saturation and Values vary between 0 and 255. In other software Hue varies between 0 to 360 and Saturation and Values vary between 0 to 100.

3.2.7 Viewing OpenCV's HSV color space

ColorWheelHSV

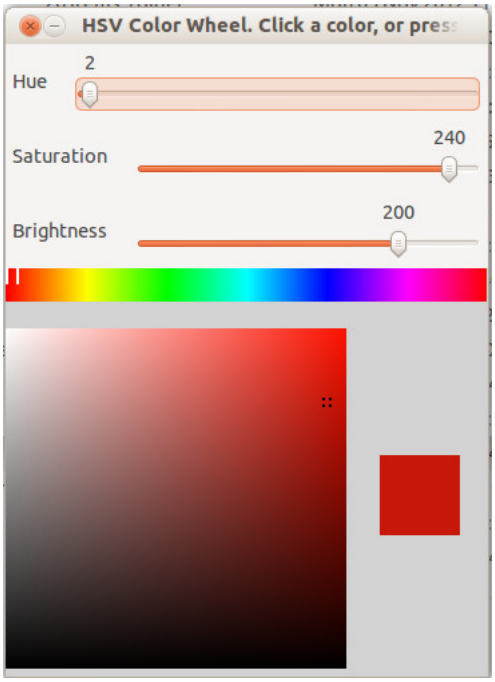


Figure 3.6: A HSV color selector.

The following range for detecting yellow color has been:

- Hue highest value =24
- Hue lowest value=20
- Saturation highest value= 255
- Saturation lowest value =140
- Value highest value=255
- Value lowest value=102

Because red color is located on 0 degree it cloud have two range values, one starting from 0

- Hue highest value =10
- Hue lowest value=0
- Saturation highest value= 255
- Saturation lowest value =40
- Value highest value=255
- Value lowest value=40

and the other one ending to 0 degree.

- Hue highest value =179
- Hue lowest value=160
- Saturation highest value= 255
- Saturation lowest value =150
- Value highest value=255
- Value lowest value=200

3.2.8 Thresholding and Binarizing

After finding proper parameters for yellow and red, raw input image is being blurred by a Gaussian filter. Then we convert image into HSV color model. Afterward we threshold and binarized image by color value range for yellow and red respectively.

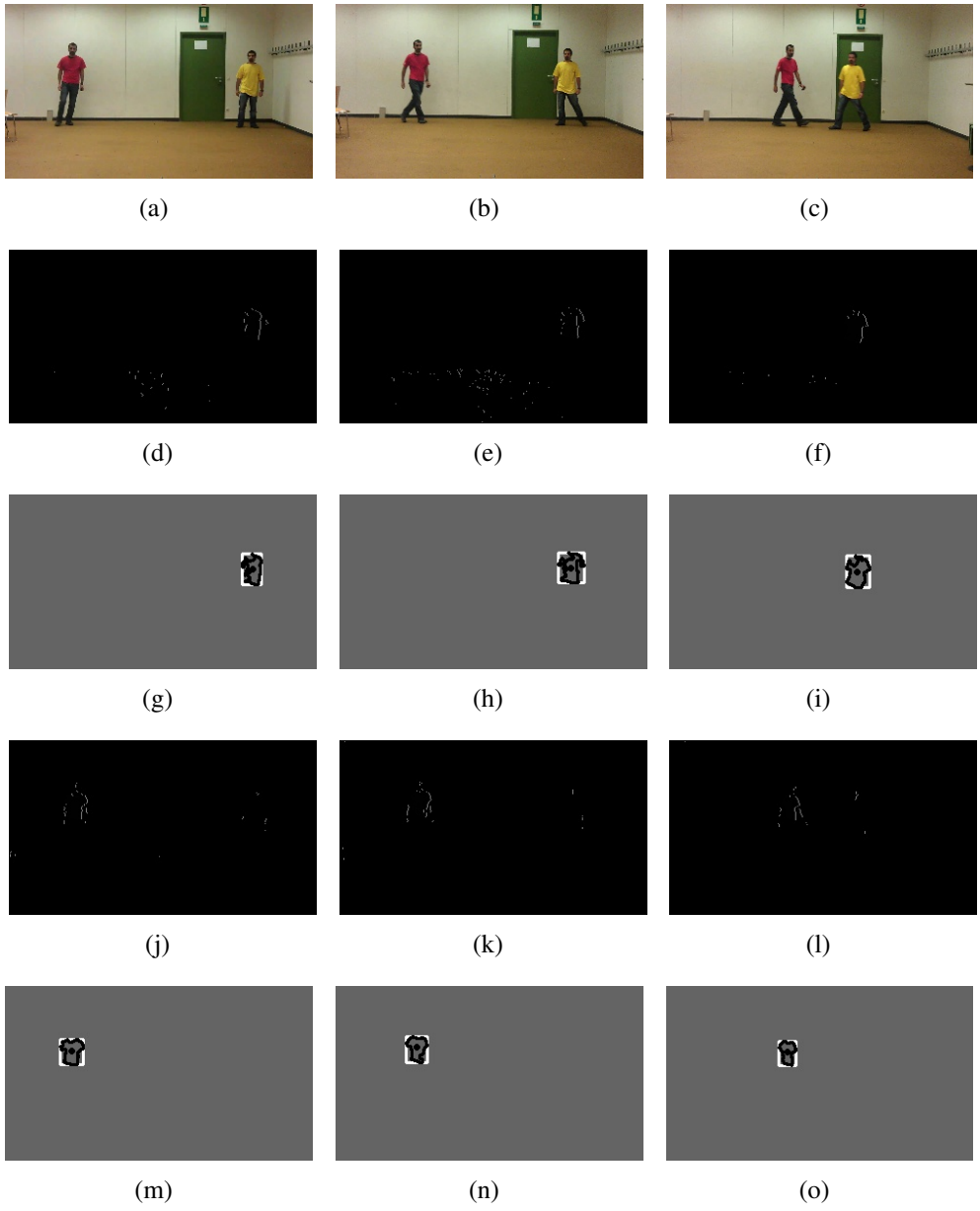


Figure 3.7: a,b and c: Raw input image. d, e and f are thresholded images for yellow color and g, h and i are tracked person.

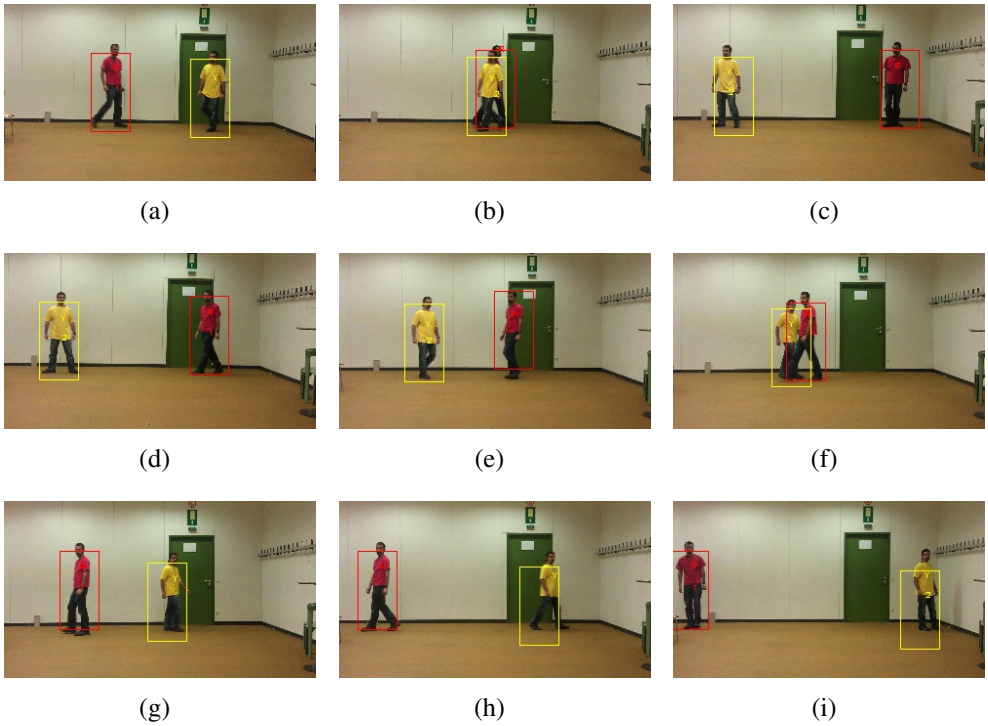
As it can be seen in the figure 3.7 d,e and a lot of sporadic noise formed after thresholding.This happens because a lot of points in the image have values close

to red or yellow. Many solutions could be used to get rid of these points such as clustering. One fast and efficient solution used here is finding the biggest component in the binary image. The result can be seen in the figure 3.7 g, h and i.

3.2.9 Experiments

To evaluate the performance and accuracy of developed Kalman filter, we try to predict the location of yellow player and red player by using HOG and and track them by our Kalman filter. Meanwhile we know the exact location of them from our color tracker.
The video recorded by a camera of 1920×1088 and later of resized to one sixth (new Width: 320 pixels, Height: 181 pixels) and 30 frames per second.
In the experiment we have checked if the predicted location by Kalman filter falls in the bounding box detected by color tracker.

Total Number of Frames	1590	In Percent $\%$
Number of Frames HOG Detected Red player	947	0.5955
Number of Frames HOG Detected Yellow player	483	0.30377
Number of Frames Kalman correctly predicted Red player	1545	0.97169
Number of Frames Kalman correctly predicted Yellow player	1327	0.83459



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.8: The bounding box shows the Kalman filter prediction while the letter 1 or 2 indicate the human detection by HOG and letter R and Y are location of player detected by color tracker.

Chapter 4

2D Body Pose Estimation

In previous chapters we developed a method for detecting and tracking the players. In the next, we are interested to estimate the pose of the players in the image. In order to estimate the pose, we need a system that can generate 2D contours of the human body in different poses and then evaluate the generated contours on the image to see which one of the generated contours can describe the pose of the player in the image. Figure 4.1 illustrates the steps for pose estimation.

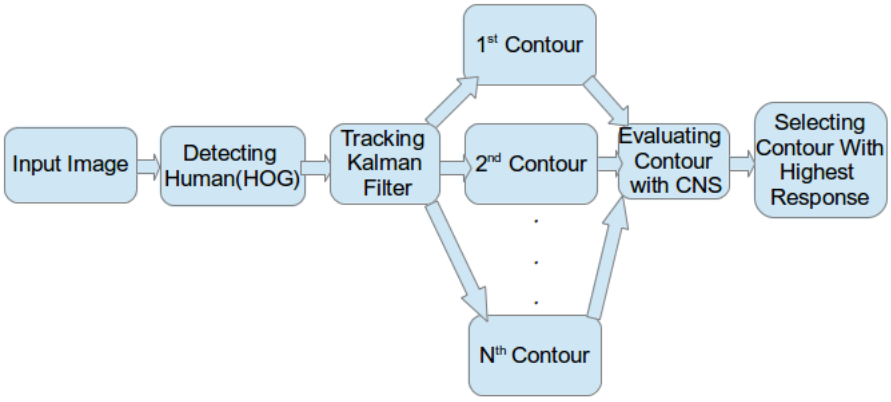


Figure 4.1: Step in Estimating Pose of Player

4.1 Generating 2D Contour of Human Body

A 2D contour is a set of points connected to each other. By changing the relative position of these point we can generate contours that can describe a human, car, tree or any arbitrary shape in 2D space. In this work first we need a system that can generate 2D contour of a human in different pose. Our approach for such a system will be discussed in Sections 4.2, 4.3 and 4.4. After this step we need a system for evaluating the generated contour on an image to see how close is the contour to the pose of player in image. The solution for this problem will be discussed in Section 4.7.1.

We considered 44 points on the human body to create a contour. These points are pointing prominent part on human body like top of the head, ears, neck, shoulder, elbow, wrist, knee, ankle and feet. There are several ways to connect these points to each other i.e. straight lines, pronominal, different kind of curve fitting. In order to make the contour smooth in a way it could describe human body curvatures meanwhile keep it computationally inexpensive we used ***cubic splines*** for connecting these point. A ***cubic splines*** is a spline constructed of piecewise third-order polynomials. In this method a third-order polynomials is fitted to every three consecutive points. In this work, GSL (GNU Scientific Library) has been used for generating interpolated points. GSL is a free numerical library for C and C++ under the GNU General Public License [53].

Figure 4.2 illustrates the difference between connecting points with straight line vs connecting them with cubic spline method.

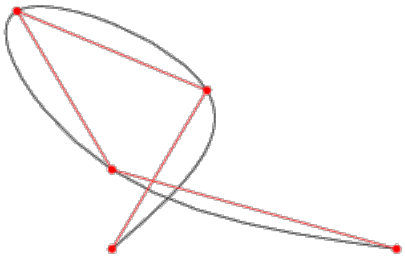


Figure 4.2: Four point connected with straight line in red and with third order pronominal in black, image acquired from http://mathworld.wolfram.com/CubicSpline.html

Now by changing the location of any these 44 points, the interpolated points would also change and a new contour would be formed. But creating the contour by this

method is pretty complicated and computationally expensive. We don't know how should we move these points to generate a contour of a human, furthermore searching through 44 dimensions is computationally expensive.
To make an automatic way for generating human contours we created a training set of several images of a player in different poses. Then we manually registered these 44 key points on the body of the player in each individual image. Based on the items in the training set, we can generate contours of human in different pose in detection phase.
We also tried to reduced the dimensions of our problem from 44 to some smaller meaningful number. For that purpose we used dimensionality reduction by PCA (Principal component analysis).
We put the data regarding the x and y position of the point in these contours plus the interpolated point as a raw entry in a matrix of data. In the next step we tried to find principal components of these data by finding the eigenvalue and eigenvector of these data. After calculating the eigenvalues of these data in the matrix, we sort them from largest value to smallest one and we pick those who contribute 90%. In the other words we sum up all the eigenvalues and then from sorted eigenvalues in descending order (largest first) we contribute eigenvalues until the sum of them is less than 90% of total eigenvalues. In the following sections we discuss more about the software which has been developed for selecting key points on the image and also interpolation process, dimensionality reduction, automatic contour generator and evaluating the contour on the image to find human body pose.

4.2 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical procedure that allows us to identify the principal directions in which the data varies.
By calculating eigenvectors and eigenvalues of covariance matrix of our data principal components could be found.
This process is equivalent to finding the axis system in which the co-variance matrix is diagonal.
By ordering the eigenvectors descending based on their eigenvalues (largest first), we can establish an ordered orthogonal basis in which the first eigenvector (the eigenvector with the largest eigenvalue) is the direction of greatest variation and the second eigenvector is the direction of next highest variation and so on.

Let's assume that we have N observations in our data set and our data has n dimensions. Let A be covariance matrix of our data which has the dimension of $n \times n$ and $x_1, x_2, ..., x_n$ are eigenvectors and $\lambda_1, \lambda_2, ..., \lambda_n$ are respective eigenvalues, therefore we have:

$\mu_x = \frac{1}{N} \sum_{i=1}^{N} P_x^{(i)}$
$A = \sum (P_x^{(i)} - \mu_x)(P_x^{(i)} - \mu_x)^T$
$A \times x_1 = \lambda_1 \times x_1$
$A \times x_2 = \lambda_2 \times x_2$
....
$A \times x_n = \lambda_n \times x_n$

writing:

$$\Phi = \begin{pmatrix} x_1, & x_2, ..., x_n \end{pmatrix} \tag{4.1}$$

and

$$\wedge = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} \tag{4.2}$$

gives us the the equation:
$A\Phi = \Phi\wedge$

4.3 Dimensionality Reduction Using PCA

By multiplying our data by a subset of the eigenvector matrix (the Φ matrix) we transform our data into a new space which has less dimension. Obviously we would loose some information about our data but by selecting the principal directions in which the data varies the most, this loss in the is minimized.
If our original data has n dimensions and we want to reduce it to p dimension we select only first p top eigenvectors and put them in a new matrix (in column order), obviously the dimension of this matrix is $n \times p$. To project our data into new space we subtract our data from mean vector and then multiply it by the first p column of the eigenvector matrix (Φ_{pca}):

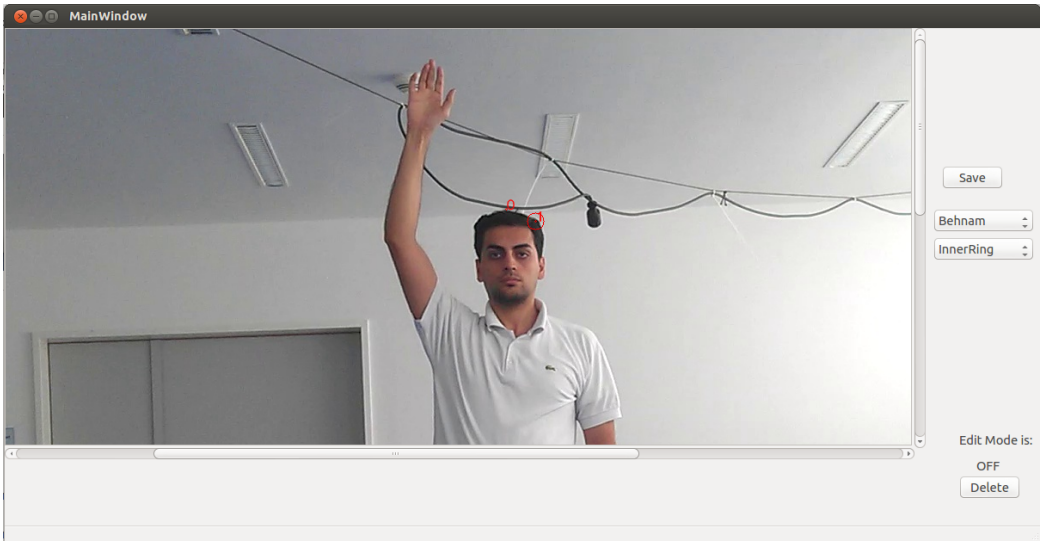
$$p_\Phi = (p_x - \mu_x) \times \Phi_{pca} \tag{4.3}$$

To transform back our data from this new space to the old one, we use:

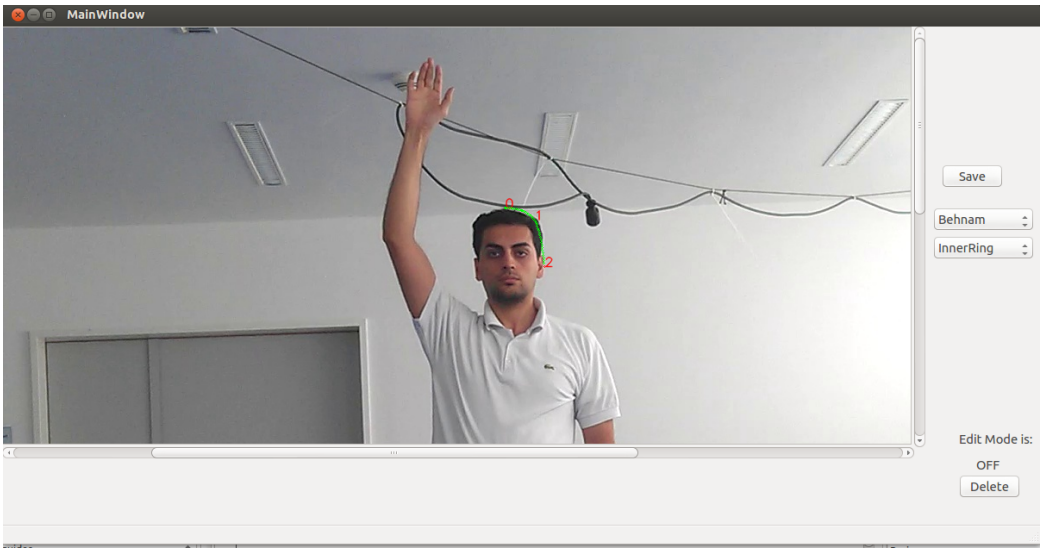
$$p_x = p_\Phi \times \Phi_{pca}^T + \mu_x \tag{4.4}$$

4.4 Creating the Training Set

To create a training set for the PCA (principal component analysis) we recorded the activity of a player in different poses and we manually labeled these 44 key points on head, ears, neck, shoulder, elbow, wrist, knee, ankle and feet. For that purpose a software has been developed. The software works in a way that after selecting three points it will immediately generate interpolated points and allow the user to modify the points so that selected points could fit the curves of human better.

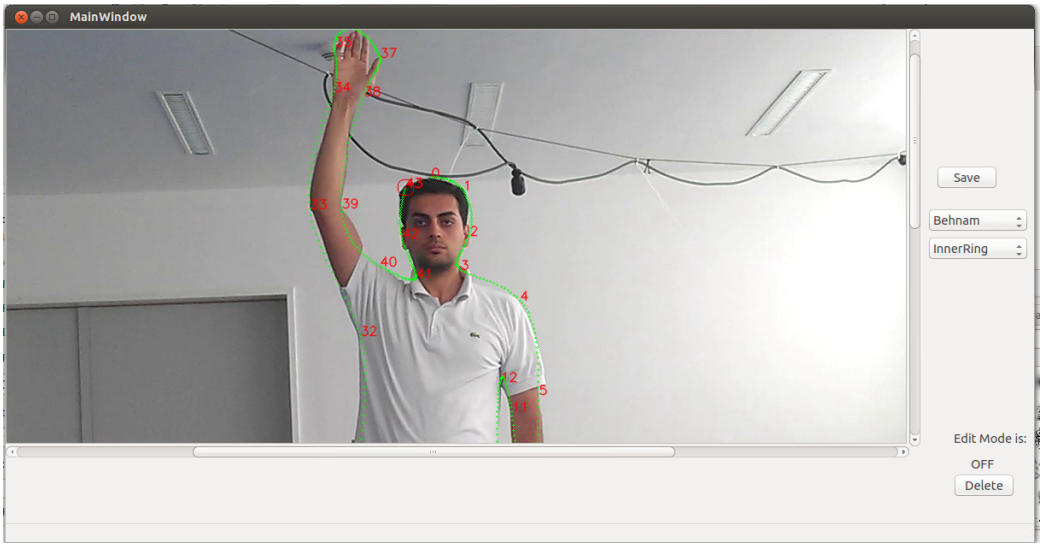


(a)

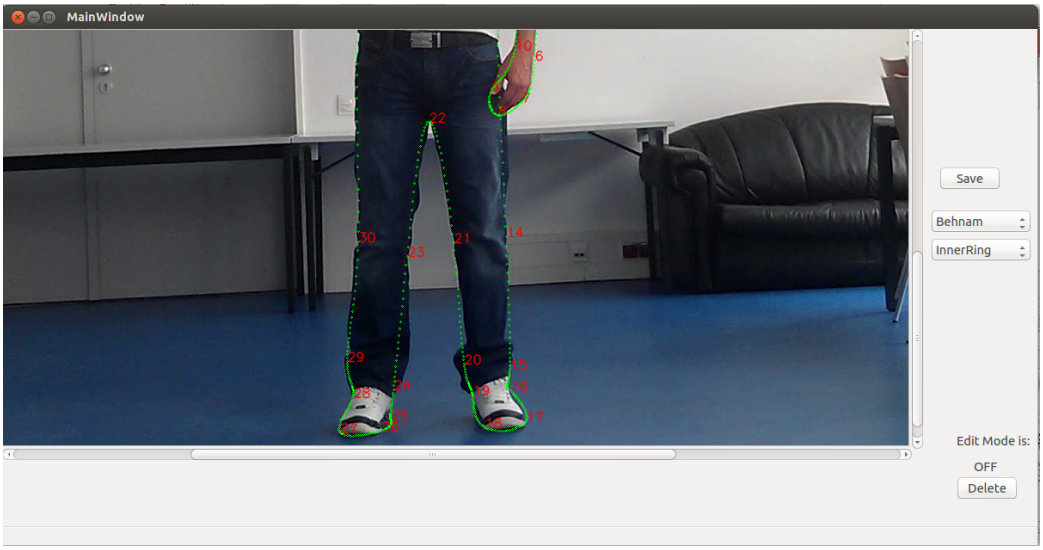


(b)

Figure 4.3: In figure **(a)** two points are selected, after selecting the third point in figure **(b)**, intermediate interpolated point are generated and drawn



(a)



(b)

Figure 4.4: By selecting 44 points over the player body a contour is created

By selecting 44 points on each image and generating 3 interpolated points for each pair of points and considering the (x,y) coordinates of each point our data would have $44 \times 2 \times 4 = 352$ dimension. To calculate PCA we have used OpenCV. OpenCV (Open Source Computer Vision Library) is a library of for real-time

computer vision, developed by Intel, and now supported by Willow Garage [54]. By taking into account the 90% rule that was mentioned in 4.1 ,we found that optimal number of dimensions is 4 and we reduce our problem into 4 DOF. By assigning a value for these four parameters in new space and applying the equation (4.4) and back projecting the data we would get data with 352 dimension which is our contour. In order to visualize the generated contour with these 4 principal component a software has been developed which is discussed in next section.

4.5 Developed Software for Generating Human Contours

To visualize the generated contour with these 4 parameters a software has been designed with four sliders for each principal component respectively. By changing the position of each slider, a new value would be set for respective principal component and by doing a back projecting with PCA, a new contour would be generated and displayed on the image. Figure 4.5 illustrates the designed GUI.

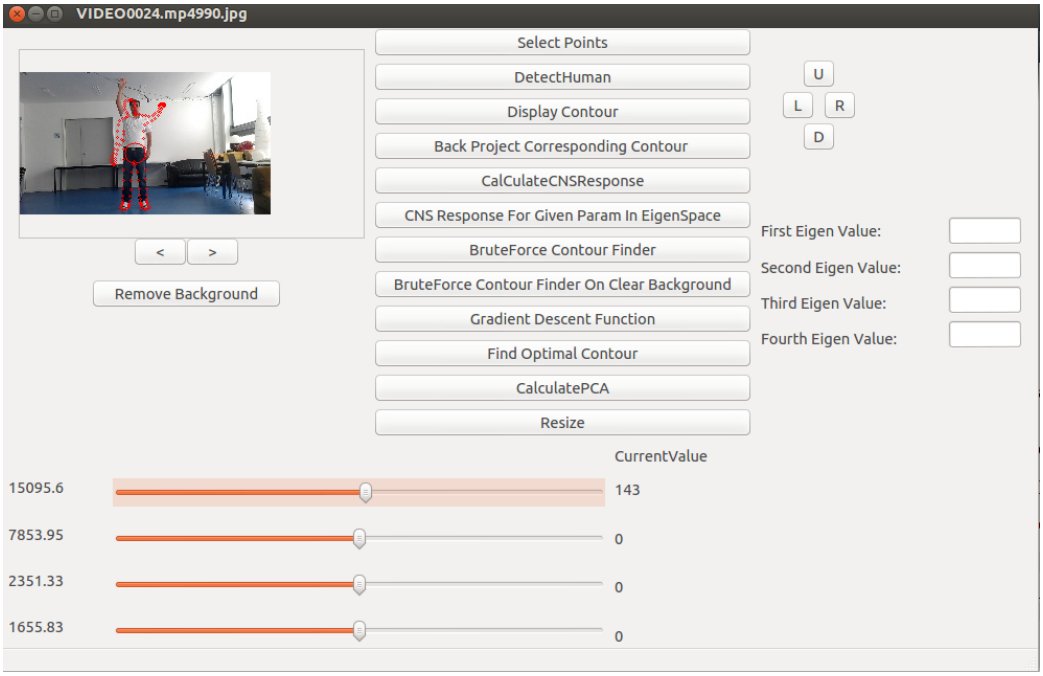


Figure 4.5: Developed GUI for generating human contour.

If we set zero for all these four variables (by leaving the slider position in the center) according to equation 4.4 we would have:

$$p_x = p_\Phi \times \Phi_{pca}^T + \mu_x$$

and

$$p_\Phi = [0, 0, 0, 0]$$

result in:

$$p_x = \mu_x$$

which is average contour of our data. Figure 4.6 illustrates the average contour while all the four principal components are set zero.

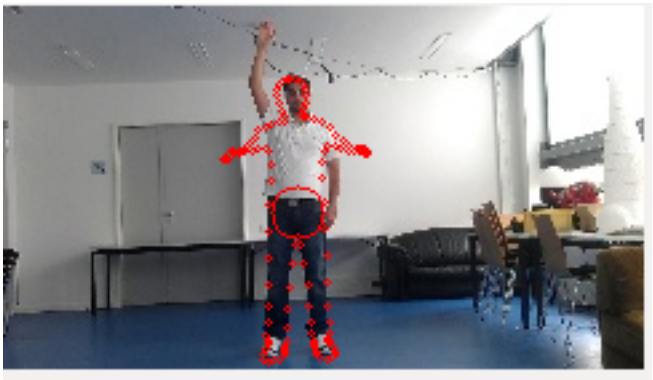


Figure 4.6: Average contour of our data, respective values for principal component are $p_\Phi = [0, 0, 0, 0]$

By changing the value of the first principal component, arms would move in opposite directions in the generated contour. Figure 4.7 demonstrates the generated contour by assigning positive and negative value for the first principal component.

(a)

(b)

Figure 4.7: In figure **a** principal component values are set to $p_\Phi = [143, 0, 0, 0]$ and In figure **b** principal component values are set to $p_\Phi = [-149, 0, 0, 0]$.

As a result, By choosing positive values for first principal component right hand goes up and left hand goes down and by choosing negative value left hand goes up and right hand goes down.
By changing the value of the second principal component and keeping other principal component at zero, hands would move in same direction. Figure 4.8 demonstrates the generated contour by assigning positive and negative value for the second principal component.

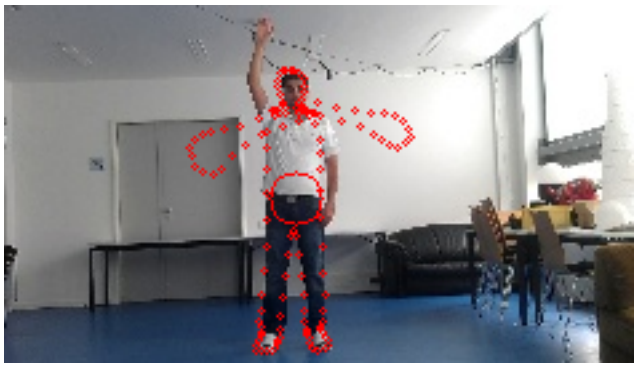
(a)

(b)

Figure 4.8: In figure **a** principal component values are set to $p_\Phi = [0, 143, 0, 0]$ and In figure **b** principal component values are set to $p_\Phi = [0, -109, 0, 0]$.

Changing in the third principal component would open or fold both hands in opposite direction. Figure 4.9 shows how positive and negative values for the third slider would change the position of hand.

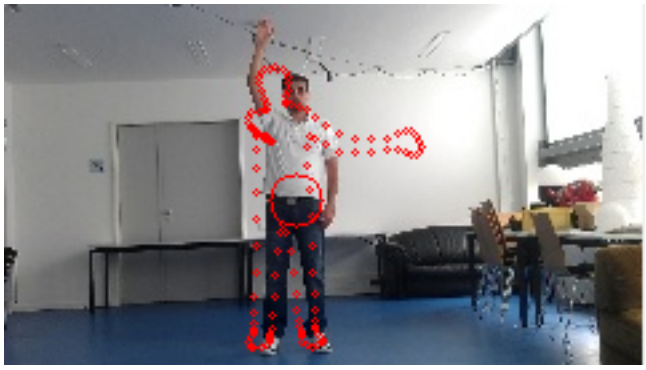
(a)



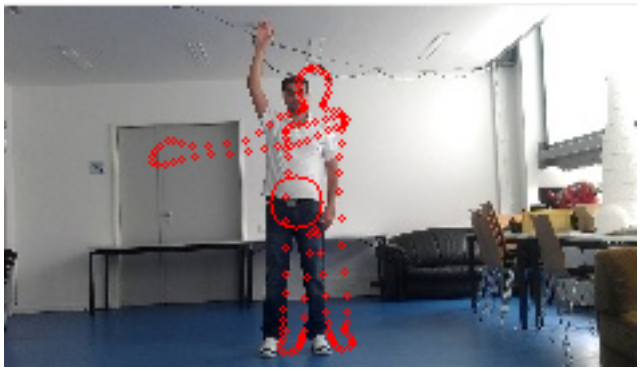
(b)

Figure 4.9: In figure **a** principal component values are set to $p_\Phi = [0, 0, -140, 0]$ and In figure **b** principal component values are set to $p_\Phi = [0, 0, 140, 0]$.

The last principal component would extend (stretch) both hands to the left or to the right (depending on positive or negative value). Figure 4.10 illustrate changes in the fourth slider and respective contour.



(a)



(b)

Figure 4.10: In figure **a** principal component values are set to $p_\Phi = [0, 0, 0, 98]$ and In figure **b** principal component values are set to $p_\Phi = [0, 0, 0, -140]$.

4.6 Discussion of The Model and Interpretation of Principal Components

In previous section we discussed the proposed method for creating the training set and the method for dimensionality reduction. In this section we discuss the observed results from the model.
Due to the fact that in the training set only up right poses has been visited, so the proposed method generates only human contours in up right poses.
In the majority of poses, the hands of the players moves against each others in opposite direction so the first principal component successfully captured these

changes. Thus in back projection of PCA, changing the value of the first principal component (while keeping other principal component fixed) generates contours in which if the right hand goes up the left hand would goes down and if the left hand goes up, the right hand goes down.
The second largest group of poses in the training set are the poses where hands are going up and down together simultaneously (for blocking and intercepting the ball). The second principal component successfully captured these changes and by changing the value of the second principal component (while keeping other principal components fixed) the hands are going up and down together in the generated contours.
The last group of poses in the training set are poses in which hands are folding or stretchering (either simultaneously or against each other). The third and fourth principal component captured these changes and generate respective poses in back projection of the PCA. All in all the model is able to generate the contours in the training set with 4 DOF. In the following we discuss our approach for evaluation the generated contours.

4.7 CNS Contour Response

So far we have introduced a method for generating a 2D contour in different poses by restricting the problem using PCA into 4DOF. In the next we have to evaluate the generated contour on an image to see how well it matches. For that purpose we have used a library called CNS (Contrast Normalized Sobel)[55].

4.7.1 Contour Response

In the CNS filter, the contour response $\in [0, 1]$ is defined as an integral over all the contour points:

$$ContourResponse(p) = \int_0^1 response(p(\lambda), \angle p^{'}(\lambda) + \pi/2) d\lambda \tag{4.5}$$

The $response(p(\lambda), \angle p^{'}(\lambda) + \pi/2)$ on an arbitrary contour point $p(\lambda)$ indicates, how much the local image at $p(\lambda)$ looks like the a contour in direction of $\angle p^{'}(\lambda)$. The $response$ function is the function of two parameters, the size of the gradient on the image and the angle between the normal of the contour and the gradient vector. Figure 4.11 illustrates the relation between parameters of $response$ function and its output value. As it can be seen the $response$ output is maximized

when the size of the gradient vector is maximized (maximum size of the gradient vector is one due to normalization) and the angle between gradient vector and normal of contour is zero ($\delta = 0$).

$$\delta = \theta - \angle cns(x, y) \tag{4.6}$$

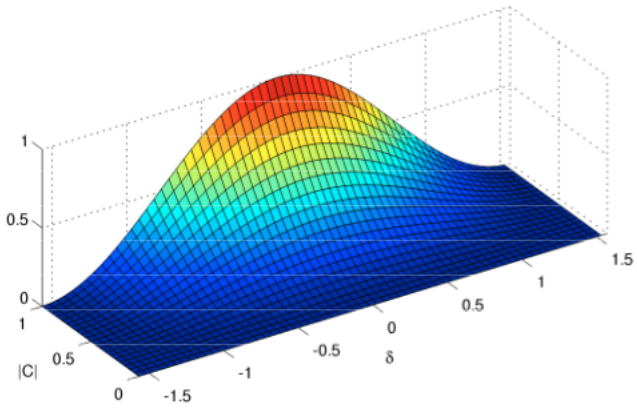


Figure 4.11: Plot of the response as a function of CNS norm $|cns(x, y)|$ and angular mismatch between contour normal and gradient vector, Image acquired from [56].

Equation 4.7 shows how the value of the response function on an arbitrary (x, y) point on the image is calculated.

$$response(x, y, \theta) = \left(\left(\begin{pmatrix} cos\theta \\ sin\theta \end{pmatrix}\right)^T CNS(x, y)\right)^2 \tag{4.7}$$

And Equation 4.8 shows how CNS is calculated.

$$CNS = \frac{\sqrt{2}\left[X * \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, X * \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}\right]^T}{\sqrt{X^2 * 16\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} - \left(X * \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}\right)^2}} \tag{4.8}$$

Equation 4.8 can be seen as:

$$CNS = \frac{SobelX, SobelY}{NormalizationFactor} \tag{4.9}$$

The denominator (Normalization Factor) is a weighted image variance that follows from illumination invariance. Equation 4.10 is formed by rewriting equation 4.7 and 4.8

$$response(x, y, \theta) = \frac{2(X * [cos\theta(\begin{smallmatrix}1&0&-1\\2&0&-2\\1&0&-1\end{smallmatrix}) + sin\theta(\begin{smallmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{smallmatrix})])^2}{X^2 * 16(\begin{smallmatrix}1&2&1\\2&4&2\\1&2&1\end{smallmatrix}) - (X * (\begin{smallmatrix}1&2&1\\2&4&2\\1&2&1\end{smallmatrix}))^2} \begin{pmatrix} x \\ y \end{pmatrix} \quad (4.10)$$

More about the CNS could be found at [56] and [55]. It's more efficient to compute the response of contour on an array of 16×16 rather than a single (x,y) point on image.

Chapter 5

Experiments and Analysis

To evaluate the robustness and efficiency of the proposed method, a series of experiments has been done. In the first type of experiments, we calculate the CNS response on our image set by calling the respective contour (labeled contour) of that image from our data set. This experiment has been done on images contaminated with clutter noise and also on images with clear background. In the second type of experiments we tried to find the contour that can describe the pose of the player in image by searching all possible values for all four principal components (brute forcing).
The purpose of the experiments is to investigate the answer for the following questions:

- What should be the range of values for principal components?
- Whats should be the step size in increasing values for the principal components?
- How robust is the CNS against clutter noise?
- On what bases should the estimated contour be labeled as a successful one?
- What is the success rate of the proposed method?
- In the case of failure in estimating the pose, CNS did the mistake or did the contour generator based on PCA, or both?
- What is the computational time for estimating a pose?

- What can be done to improve the efficiency and reduce the computation time?

In the following, the experiments will be discussed and in the next chapter the results of the experiments are analyzed.

5.1 CNS response from labeled contour on images with clutter noise

In this experiment we first used HOG to locate the person in the image. Then by knowing the id number of the image we loaded the labeled contour and afterward we calculated the CNS response on every image.

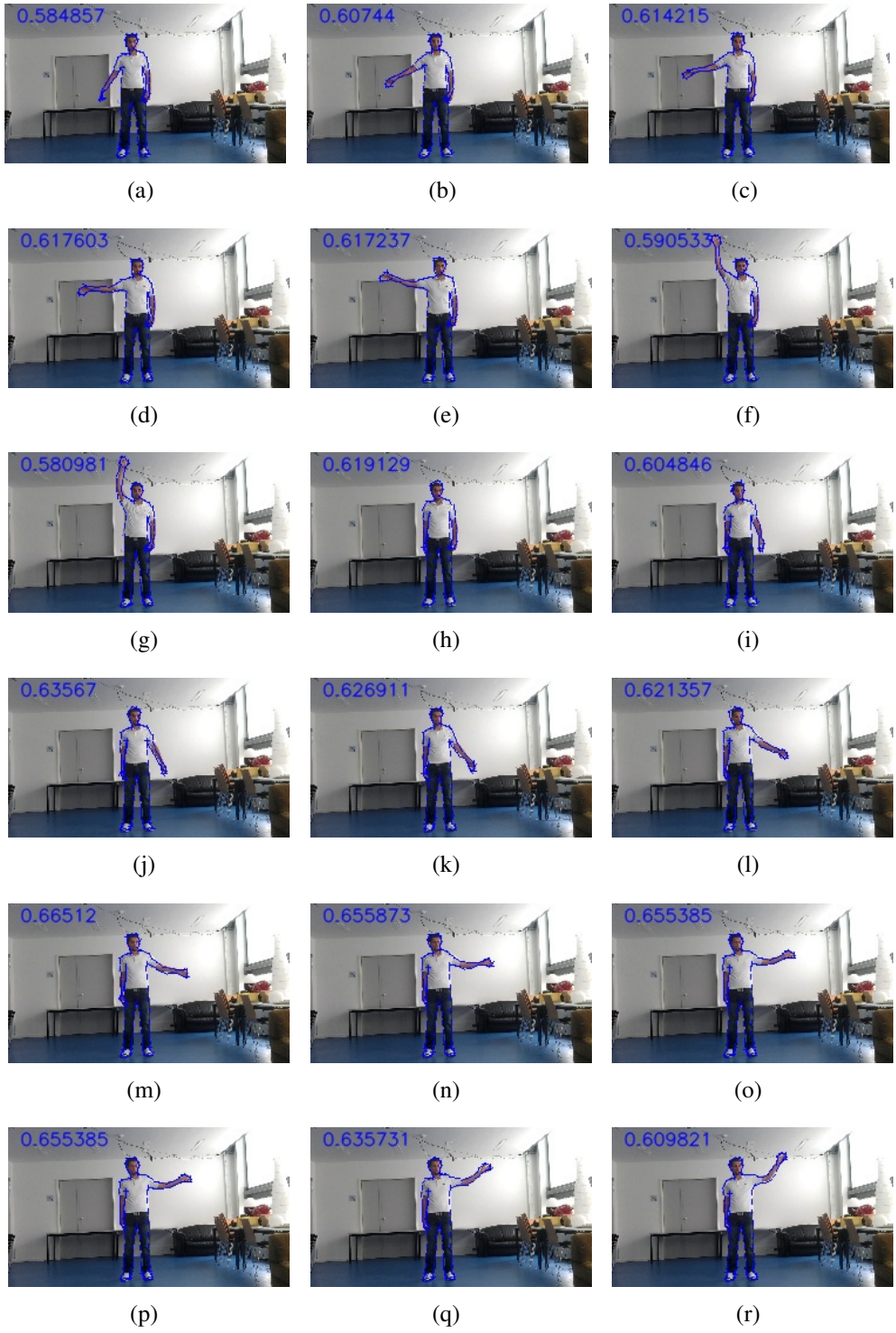


Figure 5.1: Labeled contours loaded and CNS response for the contour on the image calculated and written on the top left of the image.

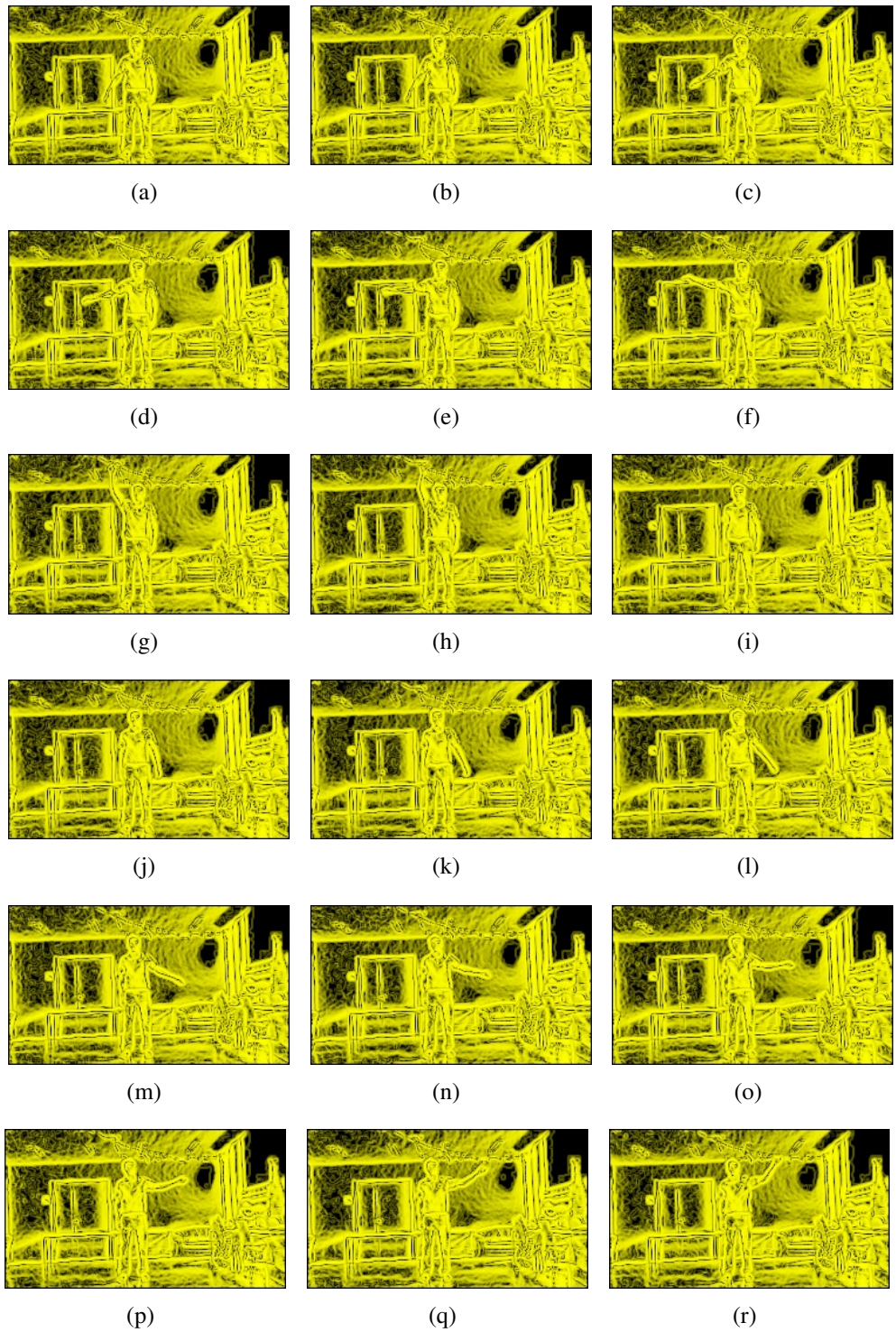


Figure 5.2: The CNS images of the experiment with clutter noise in background . Highlighted part on the image indicate the magnitude of gradient on the image (the more highlighted, the greater gradient magnitude)

5.2 CNS response from labeled contour on image with clear background

In this experiment a mask generated from the labeled contour has been fitted to the image and the image has been clipped based on that. Similar to the previous experiment CNS response has been calculated and written on the top left of the image.

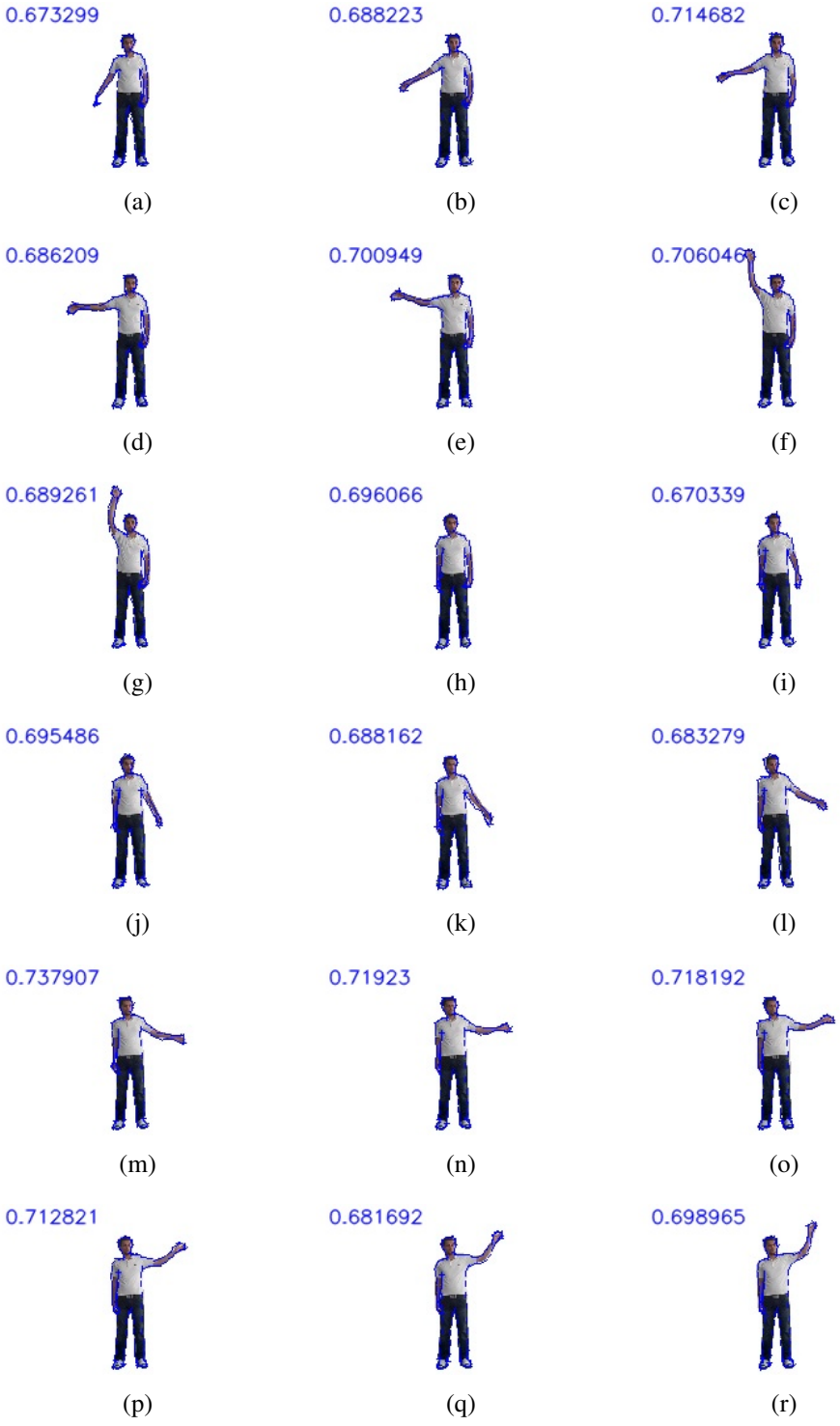


Figure 5.3: cns response for labeled contour on images with clear background.

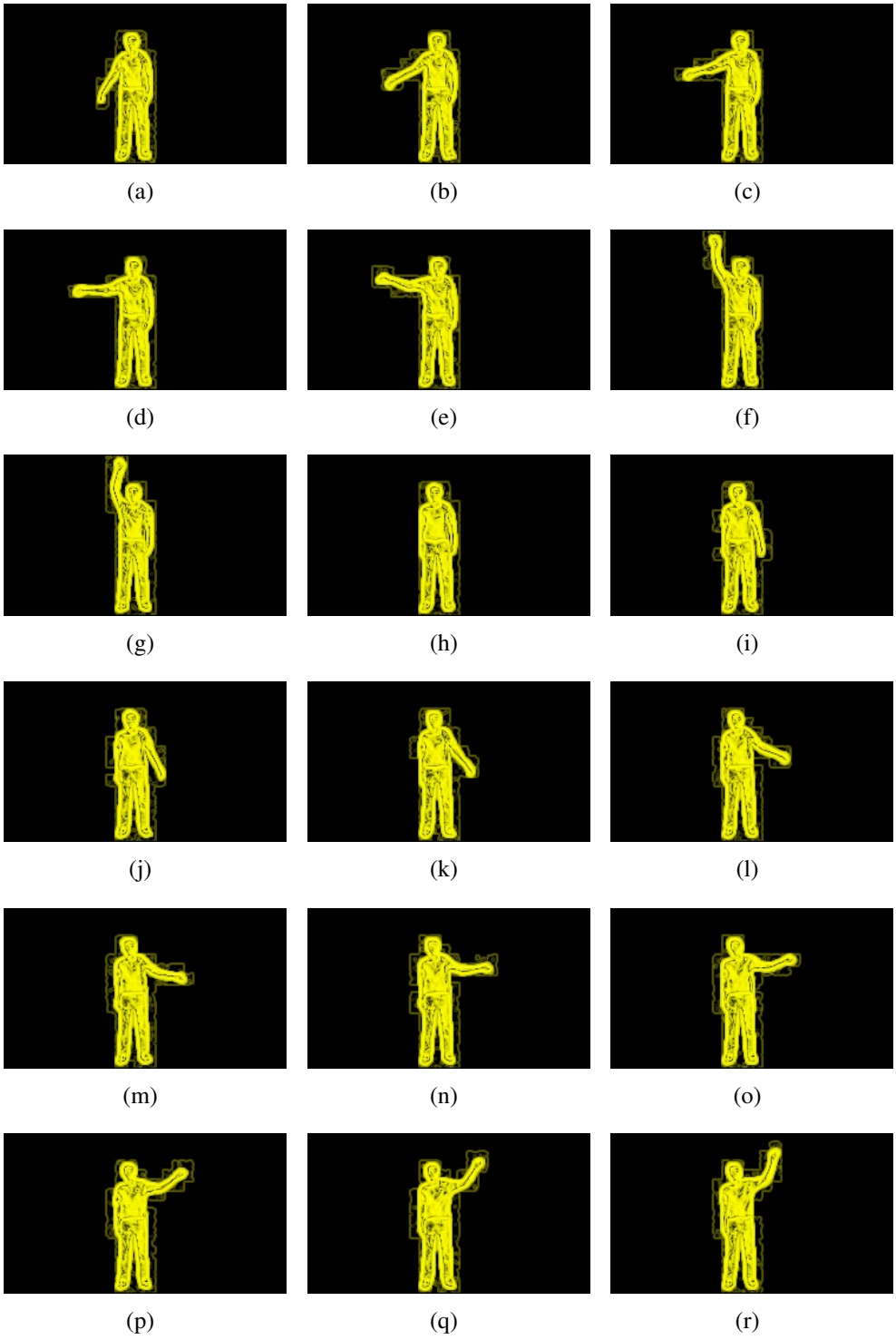


Figure 5.4: The CNS images of the experiment with clear background. Highlighted part on the image indicate the magnitude of gradient on the image (the more highlighted, the greater gradient magnitude)

5.3 CNS response from optimal contour on image with clutter noise

In this experiment for each image, several contours have been generated by brute forcing all the four principal components in PCA space and the CNS response for them has been calculated. The contour with the highest CNS response has been selected. It should be remembered that the variance of each component is equal to the corresponding eigenvalue. In other words, the standard deviation of the component is equal to the square root of the eigenvalue. To determine the range of values for each principal component, first we calculate the square root of each eigenvalue. Then the range of the search space for each principal component is set to two times of the standard deviation. The maximum and minimum value for each principal component observed in the labeled contour also endorses this range and falls within this interval range.

Order of Principal Component	Eigenvalue	Square Root	Search Range
First principal component	15095	122	$[-250, 250]$
Second principal component	7853	88	$[-150, 150]$
Third principal component	2351	48	$[-100, 100]$
Fourth principal component	1655	40	$[-100, 100]$

The step size for increasing value of each principal component is set to 10. This value for step size has been derived from observing the changes in generated contour by changing the position of sliders in the developed GUI (changing value of principal component). Values larger than 10 would cause big changes in the generated contour and any smaller number would increase the size of search space. For example if we set the step size of the first principal component on 5 instead of 10, the search space would be increased to double and the algorithm would need two times more for estimating the pose.

5.3.1 Accepting and Rejecting the Estimated Contours

Estimated contour from brute force has been labeled rejected if at least one of the limbs of player (arms, hands, legs or feet) has been wrongly estimated otherwise it has been labeled as accepted. Figure 5.5 and 5.6 demonstrates some examples of the estimated contours which has been labeled as accepted and Figure 5.7 illustrates some examples of estimated contours which has been labeled as rejected.

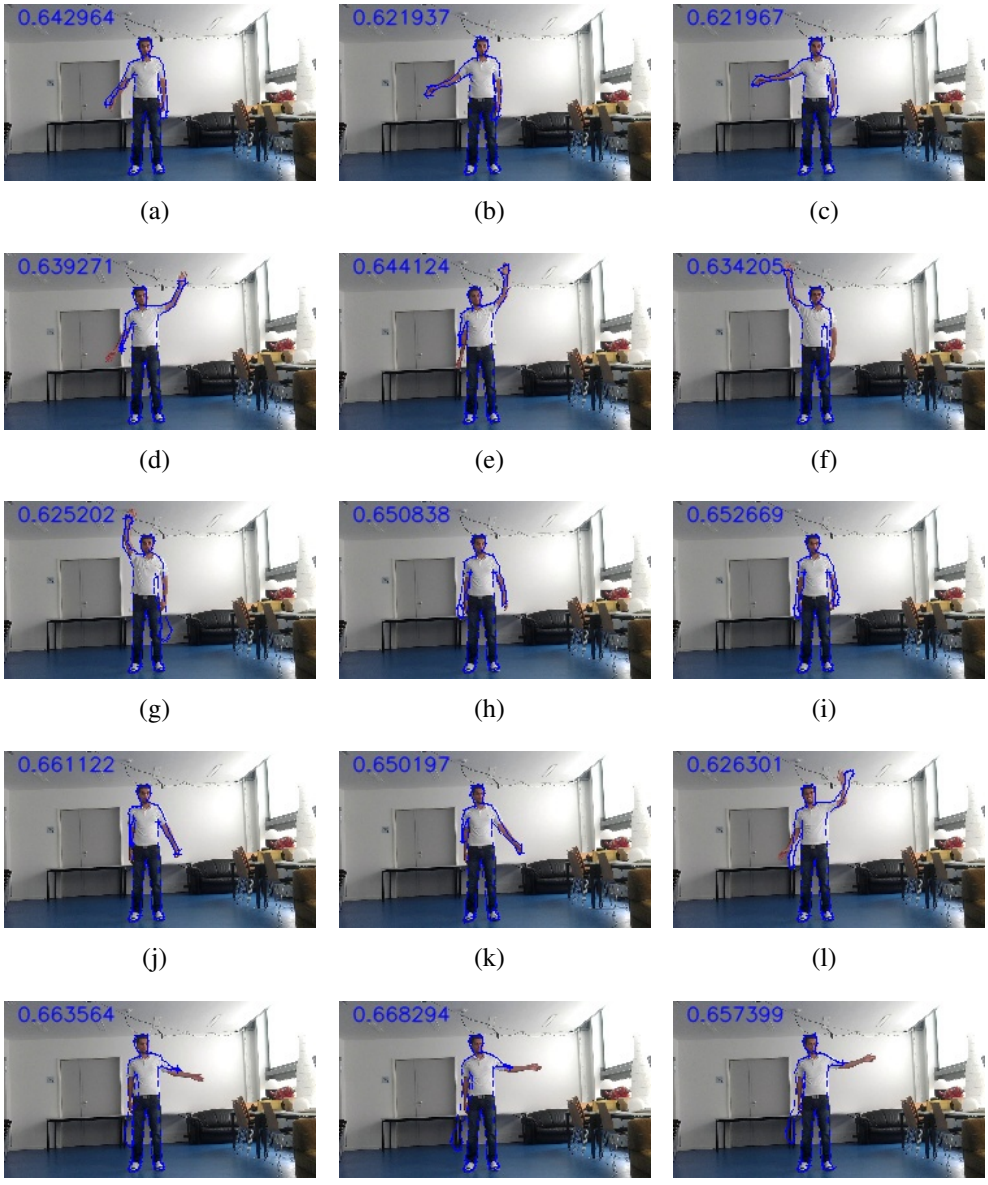


Figure 5.5: CNS response for optimal contour on images with clutter background.

5.4 CNS response from optimal contour on image with clear background

In this experiment first a mask generated from labeled contour has been fitted to the image and image has been clipped based on that. Then several contour has been generated by brute forcing all the four variable in PCA space and CNS response for them has been calculated. The contour with the highest CNS response has been selected.

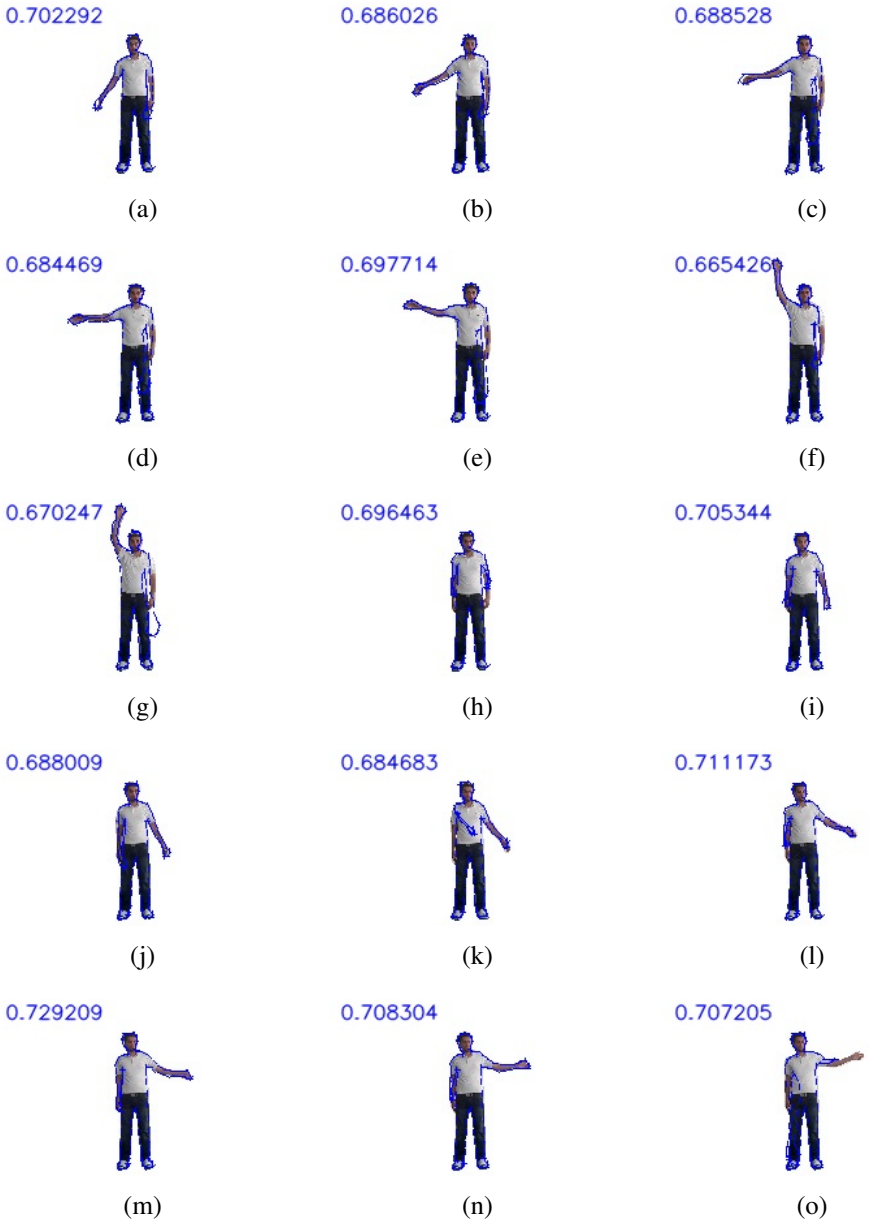


Figure 5.6: CNS response from optimal contour on image with clear background.

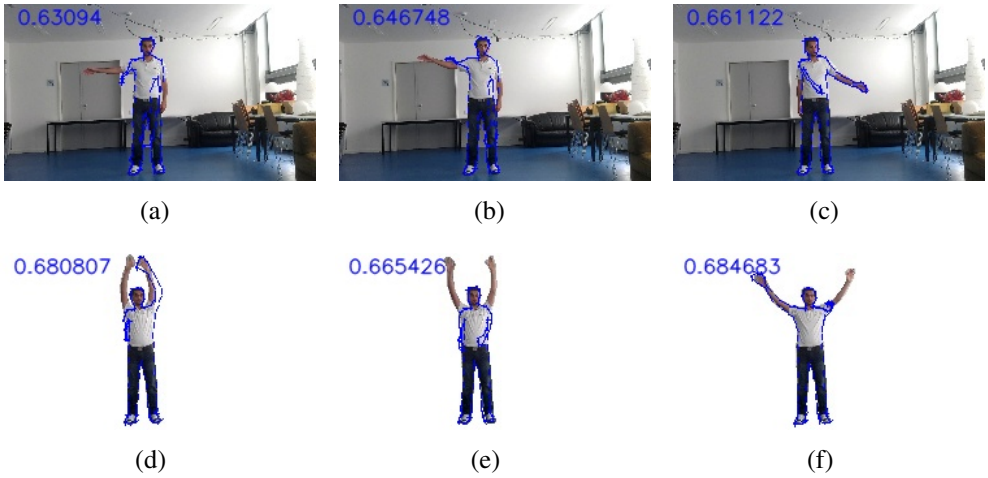


Figure 5.7: Example of contours labeled as rejected contours.

Chapter 6

Results Analysis and Conclusion

6.1 Analysis of Human Detector and Kalman Tracker

The first part of this thesis focused on detecting human players and tracking them with Kalman filter. HOG human detector algorithm correctly detected players with accuracy of 0.59% and 0.30% for first and second player respectively. This varies due to many factors including pose of players, relative angle of player and camera (camera point of view), clutter noise, etc. By utilizing a Kalman filter the accuracy of detection increased to 0.97% and 0.83% for first and second player respectively which is roughly 100% increase (numbers retrieved from Table 3.2.9). The algorithm for human detection and tracking can process nearly 30 frames per second so it can be used for real time application.
HOG is a memoryless algorithm, which means the algorithm doesnt provide any information regarding the relation between detected player in current frame and previous frame. In other words, **Player A** might be the first detected human in video and **Player B** the second person detected by HOG, but in the next frame **Player B** might be detected first and **Player A** be the second human detected by HOG. To deal with that, calculating the Euclidean distance between players found in the current frame and detected players in the previous frames and connecting players with minimum distance from previous frame successfully overcame the problem.

6.2 Analysis of Contour Generator and Pose Estimator

The second part of the thesis focused on estimating a 2D contour for the player in the video. As it has been mentioned earlier in Section 5.3.1, estimated contour from brute force has been labeled rejected if at least one of the limbs of player (arms, hands, legs or feet) has been wrongly estimated otherwise it has been labeled as accepted. Overall 0.75% of estimated contours labeled as accepted contours.
The software platform for conducting the experiments is a 32 bit Linux (kernel version 3.5.0-25) and the CPU is Intel Core i3 @ 2.27GHz with 64 bit architecture. The computation time for estimating the pose on such a platform is around 30 minute which is not suitable for real time applications.

6.2.1 Analysis of Accepted Contours

To verify the effect of clutter noise on CNS response four different experiments have been done and the CNS response of contours from these experiment illustrated in Figure 6.1. As it can be seen there is not much significant difference between CNS response from contour on images with clear background and contour on images with clutter noise and one can conclude CNS is robust to clutter noise.

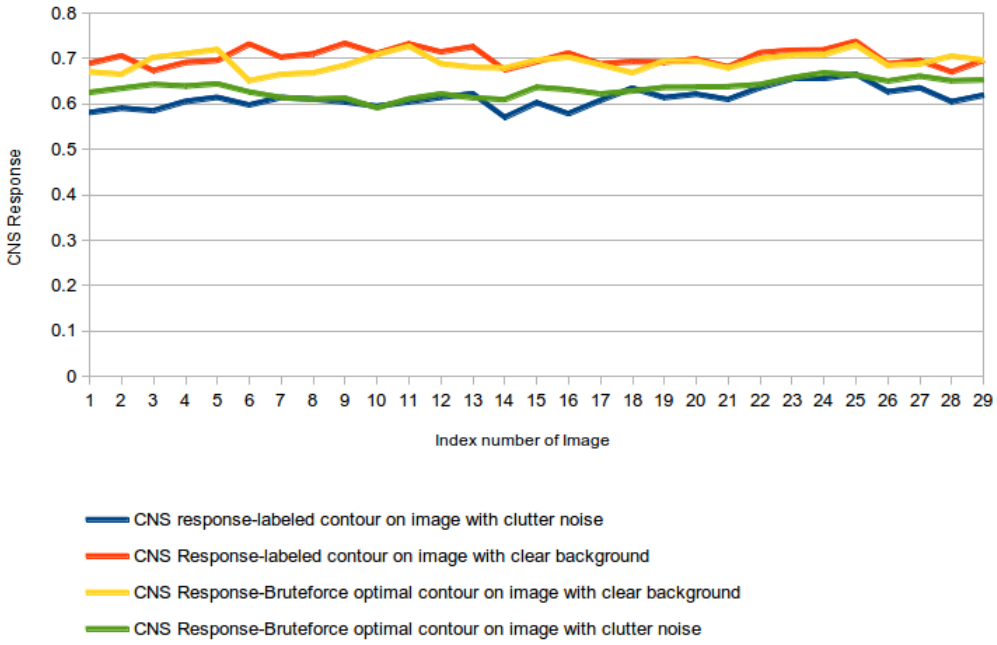


Figure 6.1: CNS response of labeled and estimated contours on images with clear and cluttered background.

Comparing number of pixels that contours must be shifted from center of gravity in x and y axes is relatively similar in brute force contour estimation and also on the labeled contour with clear or noisy background. Figure 6.2 and 6.3 illustrate the number of pixels that the contours must be shifted.

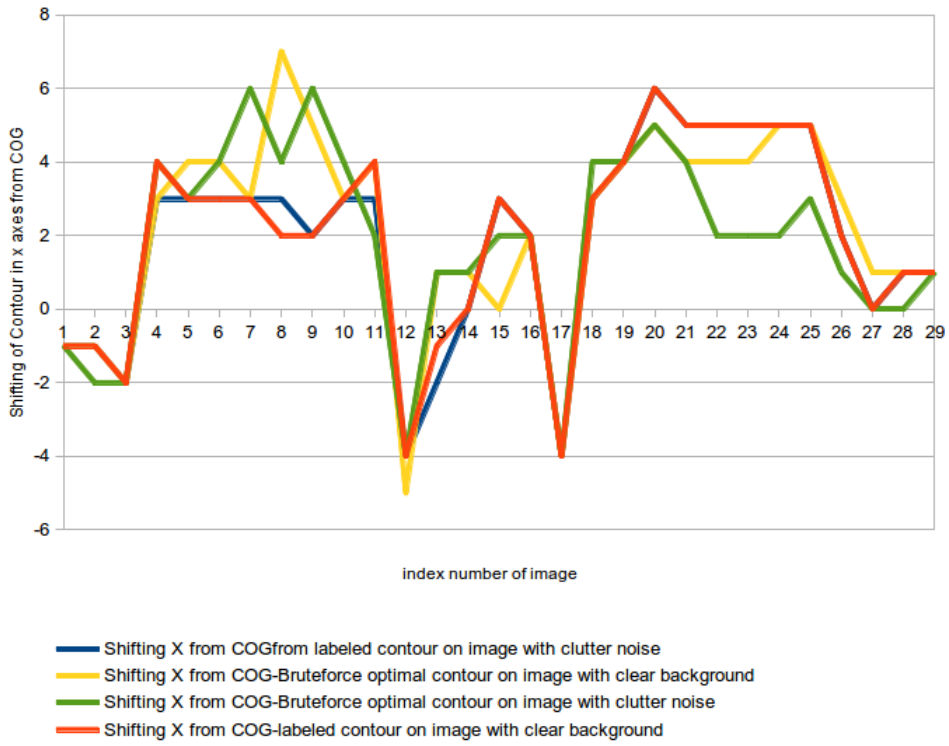


Figure 6.2: Number of pixels to shift the contour in x axes from COG.

Figure 6.3: Number of pixels to shift the contour in y axes from COG.

Comparing values of first principal component from four different experiments described in Chapter 5 shows correspondences between real values acquired from labeled contours and estimated values from brute force.

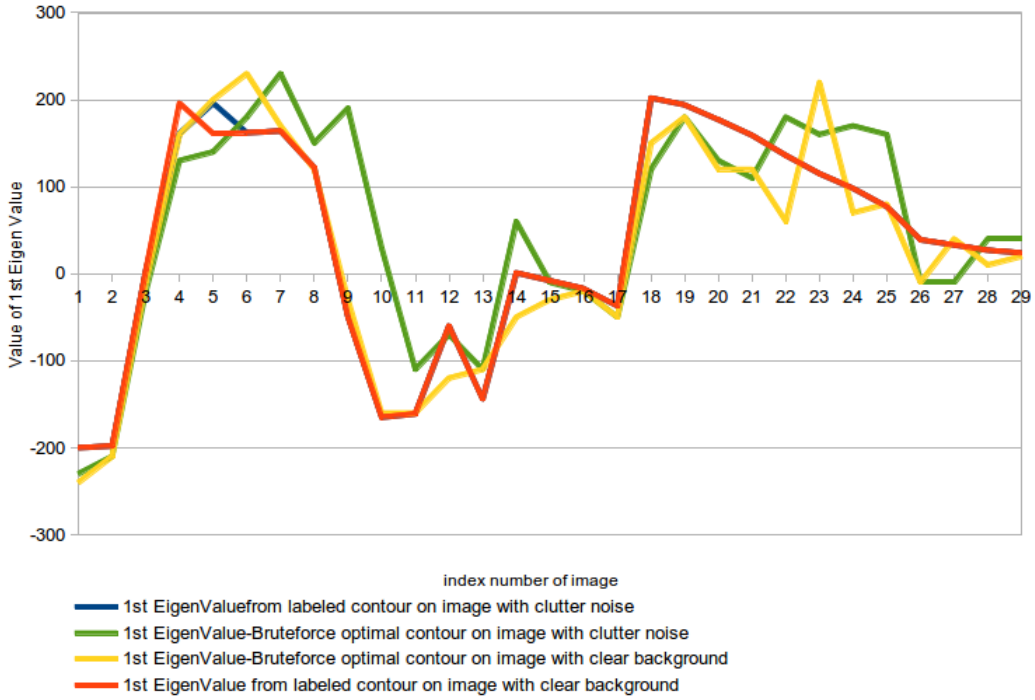


Figure 6.4: Values corresponding to first principal component in four experiments.

Similar pattern can be seen in the second, third and fourth principal component, although differences between estimated values and real values are relatively bigger in comparison with values of first principal component.

Figure 6.5: Values corresponding to second principal component in four experiments.

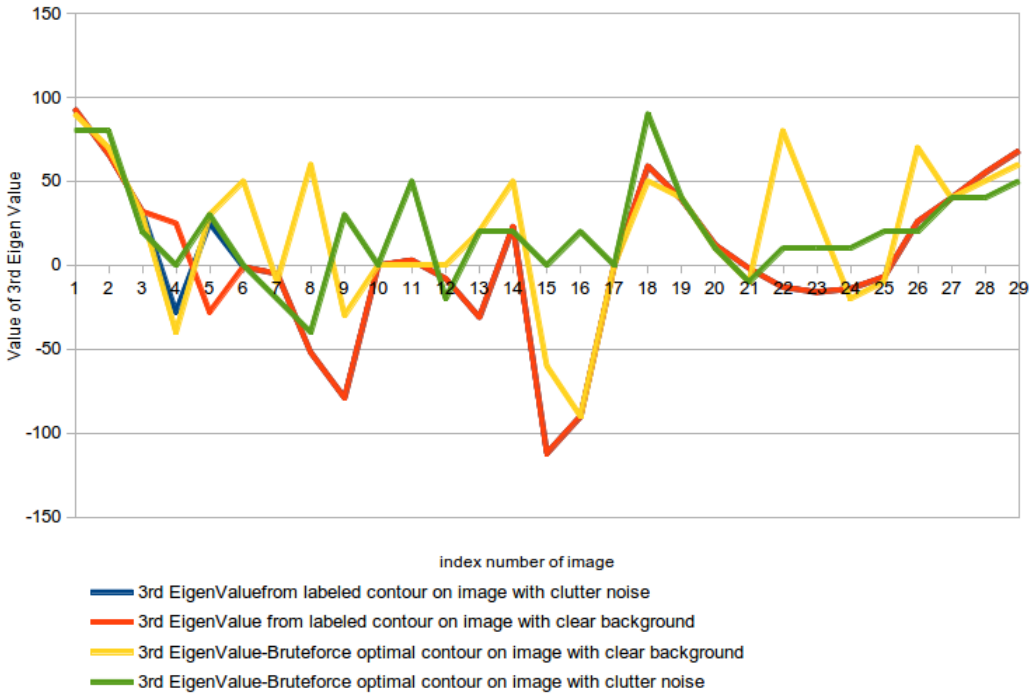


Figure 6.6: Values corresponding to third principal component in four experiments.

Figure 6.7: Values corresponding to fourth principal component in four experiments.

All in all, the proposed method for generating 2D contours by reducing the dimensionality to 4 variables and evaluating the contour with CNS can estimate the 2D pose of the player with accuracy of 75% but it is relatively slow and may not be used for real time applications. Effort for optimization the brute force for finding the optimal contour by using gradient descent function didn't succeed due to the local maximum problem in the search space. CNS is more suitable for detection of rigid objects as it has been used in [55] for finding and grabbing a mug, although it could be used for non rigid object like human body but not for real time

applications.

6.2.2 Analysis of Rejected Contours

As it has been discussed earlier, an estimated contour from brute force has been labeled as rejected one if at least one of the limbs of the player (head, arms, hands, legs or feet) has been wrongly estimated. The question that tried to be investigated in this section is: what has caused the failure? Is that CNS which can't properly evaluate the contour or it is because of the method for generating the contour with PCA, or both?
As it can be see in Figure 6.8, the CNS response of estimated contours on images with clutter noise in background (green line) is either very close or higher than the CNS response of the labeled contours (blue line).
The CNS response of the estimated contour on images with clear background (yellow line) is either very close or higher than the CNS response of the labeled contours (red line). So in both cases, CNS has given a higher response to the wrong contour than the correct contour.
By paying attention to the differences between values of first principal component in labeled contours and estimated contours in Figure 6.9 a big difference can be seen (the difference between values of red line and green line or red line and yellow line). This big difference can be seen in second and third and fourth principal component as well in Figure 6.10, 6.11 and 6.12 respectively.
As it has been discussed in 5.3, the search range of values for principal components in brute forcing is large enough to generate the poses in the training. The step size also has been selected small enough to distinguish between different pose. In conclusion, the contour generator is able to generate all possible contours from training set but CNS in 25% of image wrongly estimated the pose.

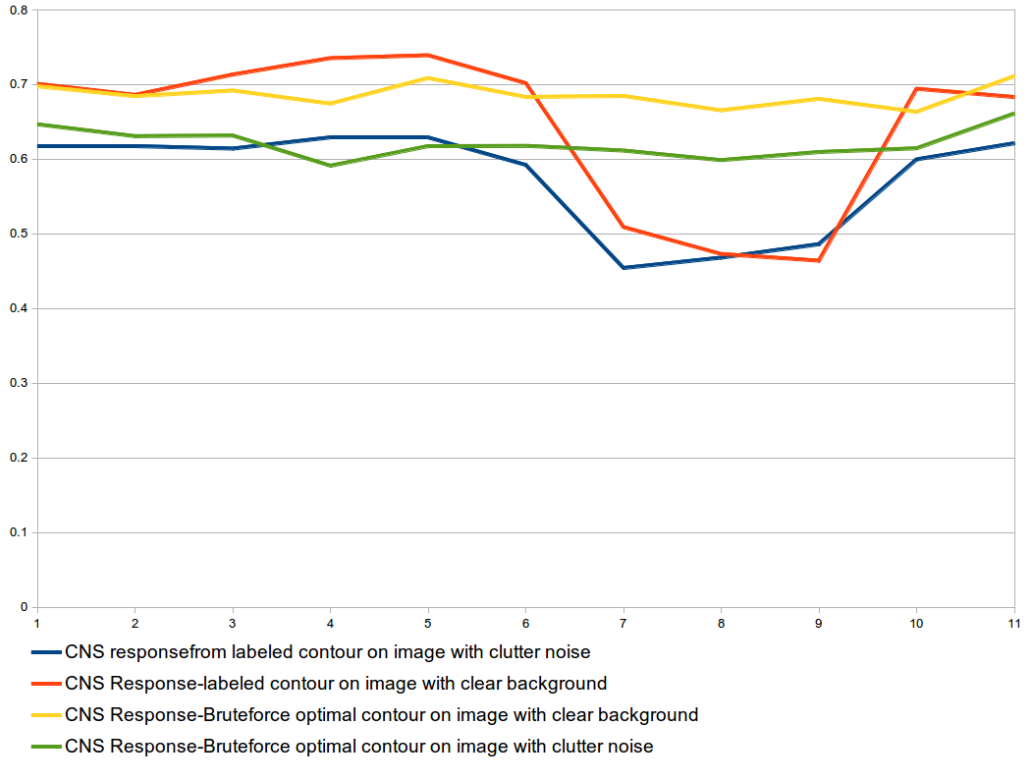


Figure 6.8: CNS response of labeled and estimated contours on images with clear and cluttered background for rejected contour.

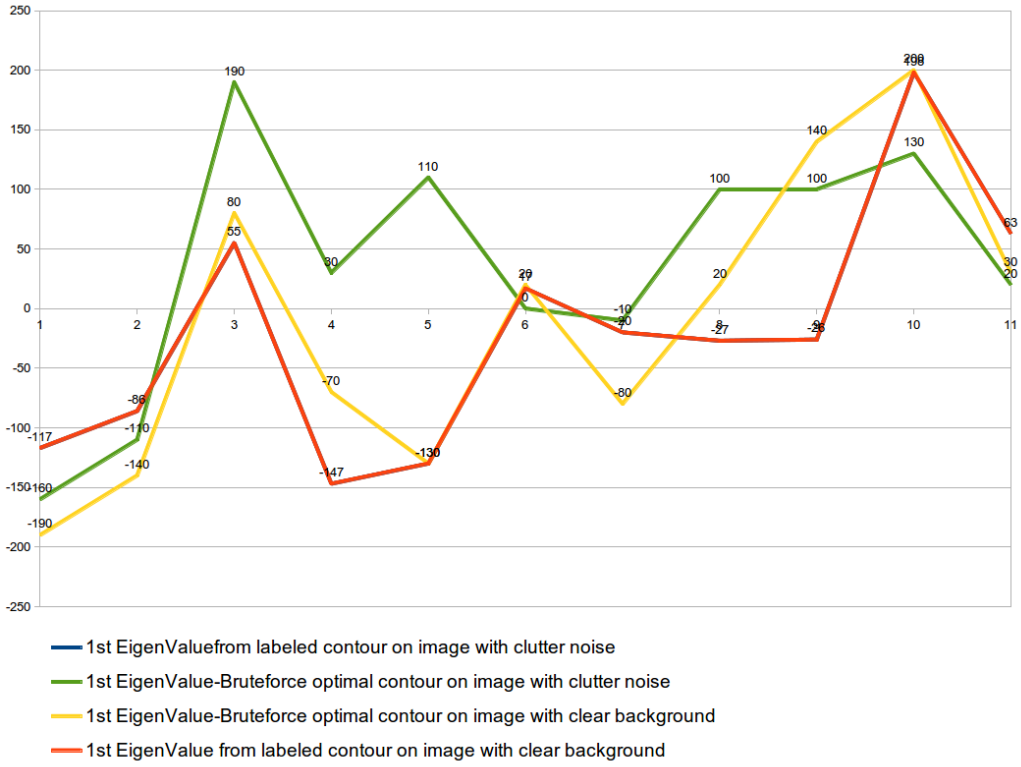


Figure 6.9: Values corresponding to first principal component in four experiments for rejected contours.

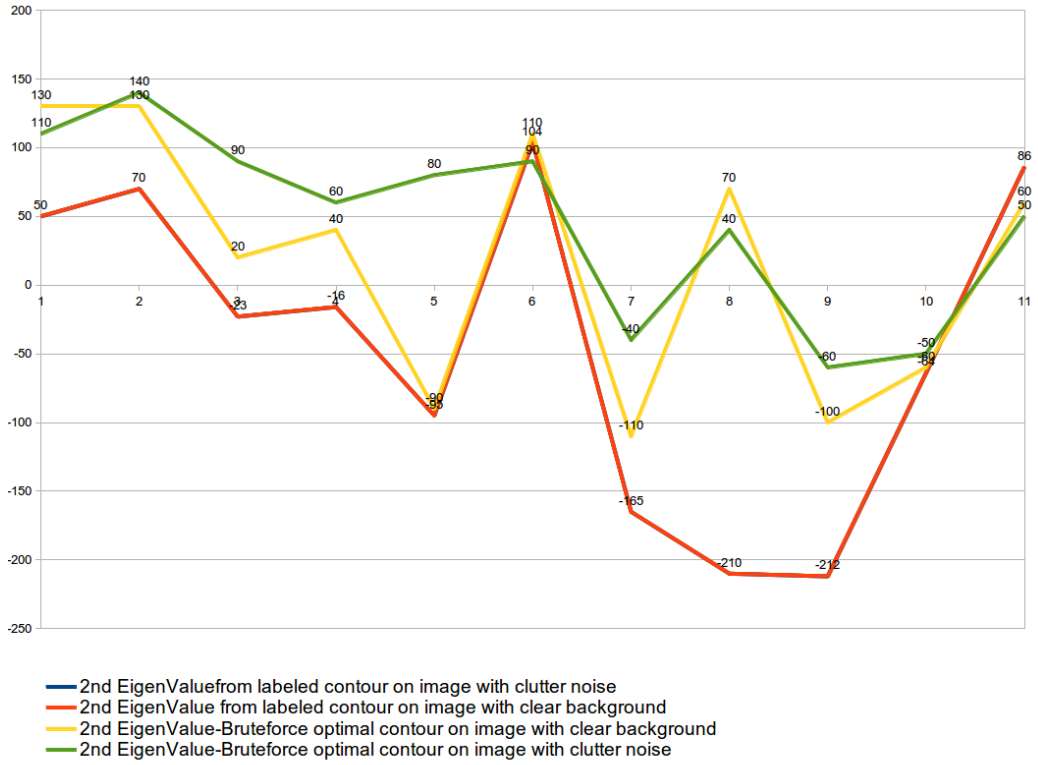


Figure 6.10: Values corresponding to second principal component in four experiments for rejected contours.

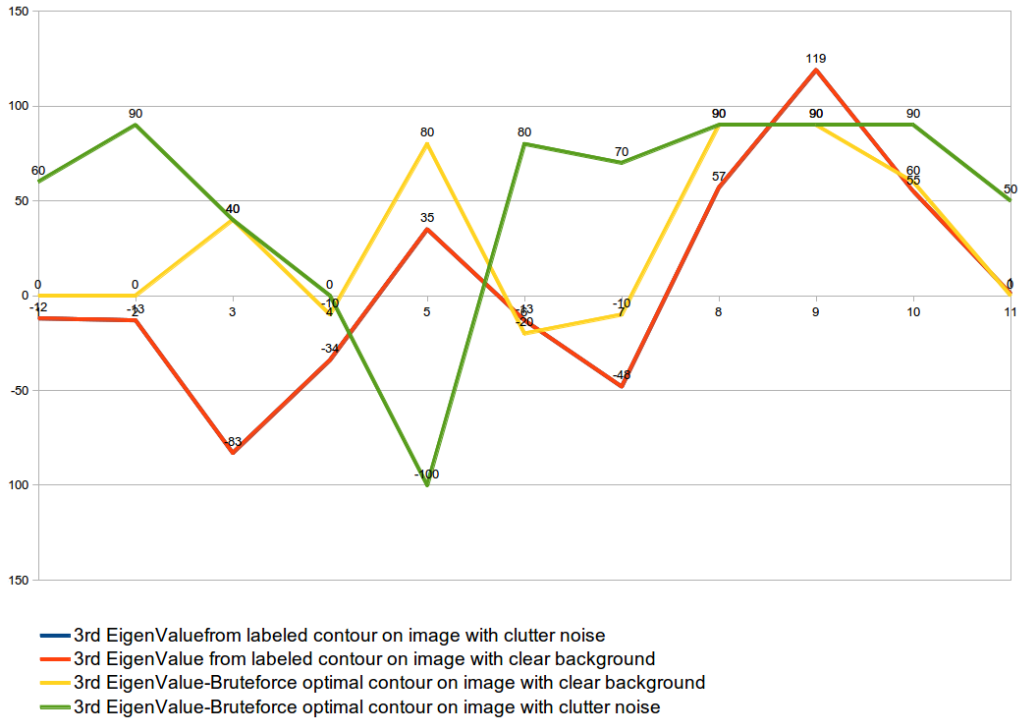


Figure 6.11: Values corresponding to third principal component in four experiments for rejected contours.

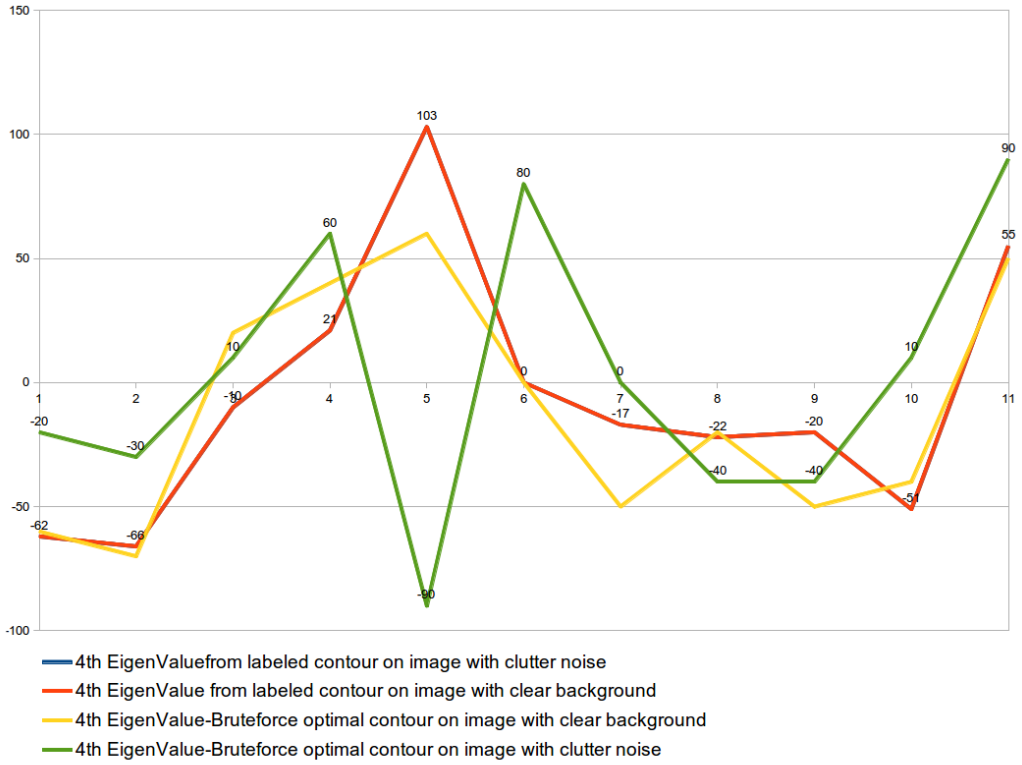


Figure 6.12: Values corresponding to third principal component in four experiments for rejected contours.

6.3 Future Work

In this work a training set has been created manually by clicking on 44 conspicuous points on human body on the training set images. This process is slow, time consuming and may not be accurate. One can make this process automatically done by using software like SCAPE (Shape Completion and Animation for PEople) [57] to generate a human shape model in different shapes and poses and automatically label the joints. By using software like SCAPE a very big training set can be created easily and very rapid.
In the proposed method, 2D contour for the whole body is being estimated at once. Estimating the contour can be done for each limb (head, torso, hands, arms, legs,

feet) separately and independent of each other. This may decrease the range of the search space for principal components and therefore shorter execution time.
Due to the fact the pose of the player smoothly changes from one frame into the next frame, after estimating the pose of the player in one frame by brute force, in the next frame range of the search space could be limited around the values of the previous one and make the pose estimation faster.
Using GPU specially CUDA (Compute Unified Device Architecture) is very popular in computer vision and image processing nowadays. This could decrease the execution time and make the algorithm runs faster.
In this work a method has been proposed and implemented for detecting of players and tracking and estimating their body pose. During the game, the robot still needs to be able to make decision which player should receive the ball. An other development for this work might be setting proper policies and making the robot able to make such decisions during the game.

List of Figures

1.1 Schema of the game (graphic by Prof. Dennis Paul). 7
1.2 Screen shots from camera mounted on robot, the player in team A (the player in yellow shirt) tries to intercept the ball 8
1.3 Screen shots from camera mounted on robot, the player in team B throw the ball back to the robot and the player in team A tries to intercept the ball 9
1.4 Screen shots from camera mounted on robot, the player in team A finally intercept and catch the ball 10
1.5 The Piggy robot (Photo by R. Wagner). 11
1.6 The Piggy robot during the game, kicking back the flying ball (Photo by R. Wagner) 11

2.1 Steps in human motion analysis and different approaches for each step, graph retrieved from [1]. 12
2.2 Model representation for human body in motion analysis, images derived from work in [3] [4] [5] respectively. 13

3.1 Feature extraction and object detection in HOG, Tiling the detection window in an overlapping grid of HOG descriptors and then using a SVM based window classifier gives the human detection chain. Image acquired from [39]. 25
3.2 Effect of gradient scale sigma, false positives per window. Image acquired from [39] 26
3.3 3×3 blocks of 6×6 pixel cells perform best. Image acquired from [39] 28

3.4 Overview of HOG, The detector window is tiled with a grid of overlapping blocks, Each block contains a grid of spatial cells. For each cell, the weighted vote of image gradients in orientation histogram is accumulated. These are locally normalized and collected into one big feature vector. Images acquired from [46]. . . 28
3.5 HSV color cylinder. 33
3.6 A HSV color selector. 34
3.7 a,b and c: Raw input image. d, e and f are thresholded images for yellow color and g, h and i are tracked person. 36
3.8 The bounding box shows the Kalman filter prediction while the letter 1 or 2 indicate the human detection by HOG and letter R and Y are location of player detected by color tracker. 38

4.1 Step in Estimating Pose of Player 39
4.2 Four point connected with straight line in red and with third order pronominal in black, image acquired from http://mathworld.wolfram.com/CubicSpline.html 40
4.3 In figure **(a)** two points are selected, after selecting the third point in figure **(b)**, intermediate interpolated point are generated and drawn 43
4.4 By selecting 44 points over the player body a contour is created . . 44
4.5 Developed GUI for generating human contour. 46
4.6 Average contour of our data, respective values for principal component are $p_\Phi = [0, 0, 0, 0]$ 47
4.7 In figure **a** principal component values are set to $p_\Phi = [143, 0, 0, 0]$ and In figure **b** principal component values are set to $p_\Phi = [-149, 0, 0, 0]$. 47
4.8 In figure **a** principal component values are set to $p_\Phi = [0, 143, 0, 0]$ and In figure **b** principal component values are set to $p_\Phi = [0, -109, 0, 0]$. 48
4.9 In figure **a** principal component values are set to $p_\Phi = [0, 0, -140, 0]$ and In figure **b** principal component values are set to $p_\Phi = [0, 0, 140, 0]$. 48
4.10 In figure **a** principal component values are set to $p_\Phi = [0, 0, 0, 98]$ and In figure **b** principal component values are set to $p_\Phi = [0, 0, 0, -140]$. 48
4.11 Plot of the response as a function of CNS norm $|cns(x, y)|$ and angular mismatch between contour normal and gradient vector, Image acquired from [56]. 50

5.1 Labeled contours loaded and CNS response for the contour on the image calculated and written on the top left of the image. 54
5.2 The CNS images of the experiment with clutter noise in background . Highlighted part on the image indicate the magnitude of gradient on the image (the more highlighted, the greater gradient magnitude) 55
5.3 cns response for labeled contour on images with clear background. 57
5.4 The CNS images of the experiment with clear background. Highlighted part on the image indicate the magnitude of gradient on the image (the more highlighted, the greater gradient magnitude) . 58
5.5 CNS response for optimal contour on images with clutter background. 60
5.6 CNS response from optimal contour on image with clear background. 62
5.7 Example of contours labeled as rejected contours. 63

6.1 CNS response of labeled and estimated contours on images with clear and cluttered background. 65
6.2 Number of pixels to shift the contour in x axes from COG. 66
6.3 Number of pixels to shift the contour in y axes from COG. 66
6.4 Values corresponding to first principal component in four experiments. 66
6.5 Values corresponding to second principal component in four experiments. 66
6.6 Values corresponding to third principal component in four experiments. 67
6.7 Values corresponding to fourth principal component in four experiments. 67
6.8 CNS response of labeled and estimated contours on images with clear and cluttered background for rejected contour. 69
6.9 Values corresponding to first principal component in four experiments for rejected contours. 70
6.10 Values corresponding to second principal component in four experiments for rejected contours. 71
6.11 Values corresponding to third principal component in four experiments for rejected contours. 72

6.12 Values corresponding to third principal component in four experiments for rejected contours. 73

List of Tables

2.1 Methods for human detection based on background subtraction with feature for finding human [22]. 19
2.2 Methods for human detection based on extracted features [22]. . . 20

Chapter 7

Bibliography

Bibliography

[1] J.K. Aggarwal and Q. Cai. Human motion analysis: a review. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90 – 102, jun 1997. doi: 10.1109/NAMW.1997.609859.

[2] Jake K. Aggarwal, Quin Cai, Wen-Hung Liao, and Bikash Sabata. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, 1998.

[3] J.A. Webb and J.K. Aggarwal. Visually interpreting the motion of objects in space. *Computer*, 14(8):40 –46, aug. 1981. ISSN 0018-9162. doi: 10.1109/C-M.1981.220561.

[4] F.J. Perales and J. Torres. A system for human motion matching between synthetic and real images based on a biomechanic graphical model. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 83 –88, nov 1994. doi: 10.1109/MNRAO.1994.346263.

[5] A. Shio and J. Sklansky. Segmentation of people in motion. In *Visual Motion, 1991., Proceedings of the IEEE Workshop on*, pages 325 –332, oct 1991. doi: 10.1109/WVM.1991.212768.

[6] S. Bandi and D. Thalmann. A configuration space approach for efficient animation of human figures. In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 38 –45, jun 1997. doi: 10.1109/NAMW.1997.609850.

[7] G. Johansson. *Visual Motion Perception*. Scientific American offprints. W.H. Freeman, 1975. URL `http://books.google.de/books?id=csn2PwAACAAJ`.

[8] R.F. Rashid and University of Rochester. Dept. of Computer Science. *Towards a System for the Interpretation of Moving Light Displays*. Reports // ROCHESTER UNIV NY. Department of Computer Science, University of Rochester, 1979. URL `http://books.google.de/books?id=n4szPwAACAAJ`.

[9] Jon A Webb and J K Aggarwal. Structure from motion of rigid and jointed objects. *Artificial Intelligence*, 19:107–130, 1982.

[10] S. Kurakake and R. Nevatia. Description and tracking of moving articulated objects. In *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pages 491 –495, aug-3 sep 1992. doi: 10.1109/ICPR.1992.201607.

[11] I.A. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: a physics-based approach. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, pages 980 –984, jun 1994. doi: 10.1109/CVPR.1994.323938.

[12] H.A. Rowley and J.M. Rehg. Analyzing articulated motion using expectation-maximization. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 935 –941, jun 1997. doi: 10.1109/CVPR.1997.609440.

[13] A. Jepson and M.J. Black. Mixture models for optical flow computation. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 760 –761, jun 1993. doi: 10.1109/CVPR.1993.341161.

[14] Z. Chen and H.-J. Lee. Knowledge-guided visual perception of 3-d human gait from a single image sequence. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(2):336 –342, mar/apr 1992. ISSN 0018-9472. doi: 10.1109/21.148408.

[15] A.G. Bharatkumar, K.E. Daigle, M.G. Pandy, Qin Cai, and J.K. Aggarwal. Lower limb kinematics of human walking with the medial axis transformation. In *Motion of Non-Rigid and Articulated Objects, 1994., Proceedings of the 1994 IEEE Workshop on*, pages 70 –76, nov 1994. doi: 10.1109/MNRAO.1994.346252.

[16] S. Iwasawa, K. Ebihara, J. Ohya, and S. Morishima. Real-time estimation of human body posture from monocular thermal images. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 15 –20, jun 1997. doi: 10.1109/CVPR.1997.609290.

[17] M.K. Leung and Yee-Hong Yang. First sight: A human body outline labeling system. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(4):359 –377, apr 1995. ISSN 0162-8828. doi: 10.1109/34.385981.

[18] J. O'Rourke and N. Badler. Model-based image analysis of human motion using constraint propagation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(6):522–536, 1980.

[19] M. Yamamoto and K. Koshikawa. Human motion analysis based on a robot arm model. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 664 –665, jun 1991. doi: 10.1109/CVPR.1991.139772.

[20] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5 – 20, 1983. ISSN 0262-8856. doi: 10.1016/0262-8856(83)90003-3. URL `http://www.sciencedirect.com/science/article/pii/0262885683900033`.

[21] K. Rohr. Towards model-based recognition of human movements in image sequences. *CVGIP: Image Understanding*, 59(1):94 – 115, 1994. ISSN 1049-9660. doi: 10.1006/ciun.1994.1006. URL `http://www.sciencedirect.com/science/article/pii/S1049966084710060`.

[22] N. A. Ogale. A survey of techniques for human detection from video. *Survey, University of Maryland*, 2006.

[23] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: real-time tracking of the human body. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 51 – 56, oct 1996. doi: 10.1109/AFGR.1996.557243.

[24] C. Beleznai, B. Fruhstuck, and H. Bischof. Human detection in groups using a fast mean shift procedure. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 1, pages 349 – 352 Vol. 1, oct. 2004. doi: 10.1109/ICIP.2004.1418762.

[25] T. Haga, K. Sumi, and Y. Yagi. Human detection in outdoor scene using spatio-temporal motion analysis. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 331 – 334 Vol.4, aug. 2004. doi: 10.1109/ICPR.2004.1333770.

[26] How-Lung Eng, Junxian Wang, A.H. Kam, and Wei-Yun Yau. A bayesian framework for robust human detection and occlusion handling human shape model. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 257 – 260 Vol.2, aug. 2004. doi: 10.1109/ICPR.2004.1334150.

[27] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 500 – 504, june 2003. doi: 10.1109/IVS.2003.1212962.

[28] D. Toth and T. Aach. Detection and recognition of moving objects using statistical motion detection and fourier descriptors. In *Image Analysis and Processing, 2003.Proceedings. 12th International Conference on*, pages 430 – 435, sept. 2003. doi: 10.1109/ICIAP.2003.1234088.

[29] Jianpeng Zhou and Jack Hoang. Real time robust human detection and tracking system. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, page 149, june 2005. doi: 10.1109/CVPR.2005.517.

[30] Sang Min Yoon and Hyunwoo Kim. Real-time multiple people detection using skin color, motion and appearance information. In *Robot and Human Interactive Communication, 2004. ROMAN 2004. 13th IEEE International Workshop on*, pages 331 – 334, sept. 2004. doi: 10.1109/ROMAN.2004.1374782.

[31] Fengliang Xu and Kikuo Fujimura. Human detection using depth and gray images. In *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*, pages 115 – 121, july 2003. doi: 10.1109/AVSS.2003.1217910.

[32] Ju Han and B. Bhanu. Detecting moving humans using color and infrared video. In *Multisensor Fusion and Integration for Intelligent Systems, MFI2003. Proceedings of IEEE International Conference on*, pages 228 – 233, july-1 aug. 2003. doi: 10.1109/MFI-2003.2003.1232662.

[33] Lijun Jiang, Feng Tian, Lim Ee Shen, Shiqian Wu, Susu Yao, Zhongkang Lu, and Lijun Xu. Perceptual-based fusion of ir and visual images for human detection. In *Intelligent Multimedia, Video and Speech Processing, 2004. Proceedings of 2004 International Symposium on*, pages 514 – 517, oct. 2004. doi: 10.1109/ISIMP.2004.1434114.

[34] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):781 –796, aug 2000. ISSN 0162-8828. doi: 10.1109/34.868681.

[35] A. Utsumi and N. Tetsutani. Human detection using geometrical pixel value structures. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 34 –39, may 2002. doi: 10.1109/AFGR.2002.1004128.

[36] D.M. Gavrila and J. Giebel. Shape-based pedestrian detection and tracking. In *Intelligent Vehicle Symposium, 2002. IEEE*, volume 1, pages 8 – 14 vol.1, june 2002. doi: 10.1109/IVS.2002.1187920.

[37] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734 –741 vol.2, oct. 2003. doi: 10.1109/ICCV.2003.1238422.

[38] H. Sidenbladh. Detecting human motion with support vector machines. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 188 – 191 Vol.2, aug. 2004. doi: 10.1109/ICPR.2004.1334092.

[39] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886 –893 vol. 1, june 2005. doi: 10.1109/CVPR.2005.177.

[40] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511 – I–518 vol.1, 2001. doi: 10.1109/CVPR.2001.990517.

[41] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(4):349 –361, apr 2001. ISSN 0162-8828. doi: 10.1109/34.917571.

[42] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 734 –741 vol.2, oct. 2003. doi: 10.1109/ICCV.2003.1238422.

[43] W.T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*, pages 100 –105, oct 1996.

[44] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94. URL `http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94`.

[45] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 454 –461 vol.1, 2001. doi: 10.1109/ICCV.2001.937552.

[46] N. Dalal and B . Triggs. Histograms of oriented gradients for human detection., 2005.

[47] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[48] Gary Bradski. *Learning OpenCV : computer vision with the OpenCV library*. O'Reilly, Farnham, 2012. ISBN 9781449314651.

[49] Peter S. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, May 28, 1979.

[50] Changyan Li, Lijun Guo, and Yichen Hu. A new method combining hog and kalman filter for video-based human detection and tracking. In *Image and Signal Processing (CISP), 2010 3rd International Congress on*, volume 1, pages 290 –293, oct. 2010. doi: 10.1109/CISP.2010.5648239.

[51] Alvy Ray Smith. Color gamut transform pairs. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '78, pages 12–19, New York, NY, USA, 1978. ACM. doi: 10.1145/800248.807361. URL http://doi.acm.org/10.1145/800248.807361.

[52] Shervin Emami. Converting between rgb and hsv color formats in opencv, October 2010. URL http://www.shervinemami.info/colorConversion.html.

[53] Brian Gough. *GNU scientific library : reference manual*. Network Theory, Bristol, 2009. ISBN 0954612078.

[54] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[55] Thomas Röfer Judith Müller, Udo Frese. Grab a mug - object detection and grasp motion planning with the nao robot. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS 2012), Osaka, Japan*, 2012. URL http://www.informatik.uni-bremen.de/agebv2/downloads/published/muellerhumanoids12.pdf.

[56] O. Birbach, U. Frese, and B. Bauml. Realtime perception for catching a flying ball with a mobile humanoid. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5955 –5962, may 2011. doi: 10.1109/ICRA.2011.5980138.

[57] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. *ACM Trans. Graph.*, 24(3):408–416, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073207. URL http://doi.acm.org/10.1145/1073204.1073207.