

GAIT ANALYSIS AND ITS IMPLEMENTATION

by

Serhat GÜVENİLİR

Submitted to the Institute of Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Electrical and Electronics Engineering

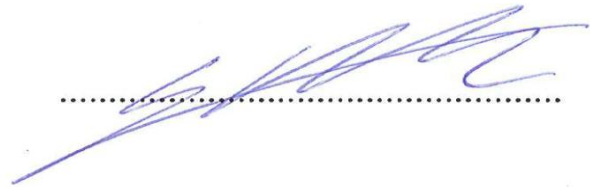
Yeditepe University

2011

GAIT ANALYSIS AND ITS IMPLEMENTATION

APPROVED BY:

Assoc. Prof. Dr. Cem Ünsalan
(Advisor)



Assist. Prof. Dr. Duygun E. Barkana



Assist. Prof. Dr. Dionysis Goularas



DATE OF APPROVAL: ... / ... / 2011

ACKNOWLEDGEMENTS

I would like to give my sincere thanks to my advisor Assoc. Prof. Dr. Cem Ünsalan for his assistance, encouragement and guidance. In addition to this, I am very thankful to my second home Yeditepe University, Deniz Pazarıcı, my instructors and research assistants the people who always encouraged me all the time.

I would like to dedicate my thesis to my family and my wife Elçin Güvenilir who were and will be unforgettable and indispensable in my heart till the end of my life.

ABSTRACT

GAIT ANALYSIS AND ITS IMPLEMENTATION

A new application domain of computer vision has emerged over the past few years dealing with the analysis of images involving humans. This application domain covers face recognition, hand-gesture recognition, and whole body or body parts recognition. Similarly, identification of a person's activity by gait extracted from video has recently become a popular research problem. Therefore, in recent years, gait analysis and recognition has received substantial attention from both research communities and the industry. It is especially useful for public, military and commercial security purposes.

In this thesis, we consider the problem in two parts. First, we locate and track moving objects in an image sequence. Then, we analyze the motion using gait analysis methods. Analyzing a person's activity by gait has a unique advantage over other biometrics as follows. It has a potential to be used at a distance when other biometrics might be at too low a resolution or might be obscured. The ability to recognize humans and their activities by vision is the key for a machine to interact intelligently and effortlessly with a human-inhabited environment. In this thesis, our aim is to implement human tracking and gait recognition.

ÖZET

GAIT ANALİZİ VE UYGULANMASI

Son yıllarda bilgisayarlı görüntüleme alanında insan hareketlerinin incelenmesi konusunda yeni bir uygulama ortaya çıktı. Bu yeni uygulama alanı yüz tanıma, el işaretlerinin tanınması, insan vücudunun ve uzuvlarının analizi gibi konuları kapsamaktadır. Hareket analizi ise insan hareketini yürüyüşleri ve davranışları ile tanıma şeklindedir. Son yıllarda hareket analizi araştırma topluluklarının ve sektörün ilgisini çekmiştir. Bu nedenle, hareket eden cisimleri videolarda tanıma bilgisayarlı görü alanında gerekli ve kritik bir alan olmuştur. Bu sistemler askeri, ortak güvenlik ve kamu için yararlıdır.

Bu çalışmada hareket analizi problemini ikiye ayırdık. İlk olarak, cisimlerin görüntü dizilerinde yerleri saptanmış ve takip edilmişlerdir. Daha sonra, davranışlarını hareket analizi yöntemleri ile inceledik. İnsan hareketlerini bu şekilde analiz etmenin diğer biyometrik özelliklere göre en büyük avantajı, potansiyel olarak uzak olan ve düşük çözünürlüklü imgelere uygulanabilir olmasıdır. Bu da, insane aktivitelerini bilgisayar sistemleri tarafından, otomatik olarak, akıllı ve efor harcamadan ayırmak için en önemli unsurdur. Bu çalışmada ana hedef, insan takibi ve hareket analizi idi.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	iii
ABSTRACT.....	iv
ÖZET	viii
TABLE OF CONTENTS.....	viii
LIST OF FIGURES	viii
LIST OF TABLES.....	xi
LIST OF SYMBOLS / ABBREVIATIONS.....	xiii
1. INTRODUCTION	1
1.1. GAIT ANALYSIS	1
1.2. APPLICATION AREAS AND SCENARIO	3
1.3. LAYOUT OF THE THESIS	4
2. LITERATURE REVIEW	6
2.1. OVERVIEW	6
2.2. TRACKING	6
2.2.1. Background Subtraction	7
2.2.1.1. Running Gaussian Average.....	7
2.2.1.2. Temporal Median Filter	8
2.2.1.3. Frame Difference	8
2.2.1.4. Mixture of Gaussians	8
2.2.1.5. Eigenbackgrounds.....	9
2.3. FEATURE BASED METHODS	9
2.3.1. Multiple Hypothesis Tracking (MHT).....	10
2.3.2. Hidden Markov Model (HMM).....	10
2.3.3. Artificial Neural Networks (ANN).....	12
2.3.4. Kalman Filtering and Mean Shift	12
2.4. MODEL BASED METHODS	14
2.5. OPTICAL FLOW BASED METHODS	16
3. IMAGE PROCESSING FUNDAMENTALS	18
3.1. COLOR MODELS ON IMAGE PROCESSING	18
3.1.1. RGB Color Model.....	18

3.1.2. CMY Color Model.....	19
3.1.3. HSI Color Model (HSV, HSL)	20
3.1.4. RGB to HSI Conversion	20
3.2. MORPHOLOGICAL FILTERS	22
3.2.1. Erosion	22
3.2.2. Dilation	23
3.2.3. Opening.....	23
3.2.4. Closing	23
4. GAIT ANALYSIS IMPLEMENTATION	26
4.1. FEATURE EXTRACTION	26
4.1.1. Human Detection and Tracking	26
4.1.1.1. Background Subtraction.....	26
4.1.1.2. Lighting Differences	29
4.1.1.3. Finding the Action Regions	30
4.1.1.4. Silhouette Representation	30
4.1.1.5. Signature Extraction.....	30
5. EXPERIMENTS	40
6. CONCLUSIONS	57
APPENDIX A : RGB to HSI MATLAB CODE IMPLEMENTATION	57
APPENDIX B : GAIT ANALYSIS MATLAB CODE IMPLEMENTATION.....	62
REFERENCES	73

LIST OF FIGURES

Figure 1.1.	Approaches to automatic gait recognition [3]	2
Figure 1.2.	Gait recognition in action [11]	4
Figure 2.1.	Boundary of a silhouette [26]	10
Figure 2.2.	Probabilistic parameters of a Hidden Markov Model X – states, Y – possible observations, a – state transition probabilities, b- output probabilities	11
Figure 2.3.	General architecture of the HMM	11
Figure 2.4.	a) Structural gait description b) Relative modeling based on angles [3]	14
Figure 2.5.	Different modeling types [3]	15
Figure 2.6.	Optic flow experienced by a rotating observer. At each location represented by the direction and length of each arrow [52]	16
Figure 3.1.	RGB color scheme [48]	19
Figure 3.2.	CMY color scheme [49]	19
Figure 3.3.	HSI color scheme [52]	20
Figure 3.4.	Example of erosion filter on binary image	22
Figure 3.5.	Erosion of an image, a. original image, b. erosion	23

Figure 3.6. Example of dilation on binary image	23
Figure 3.7. Dilation example of the same image, a. original image, b. dilation	24
Figure 3.8. Opening example of the same image, a. original image, b. opening example	24
Figure 3.9. Closing example of an image, a. original image, b. closing example	25
Figure 4.1. An example of silhouette extraction	28
Figure 4.2. Sample images of walking person	28
Figure 4.3. Running person and silhouette from our database	29
Figure 4.4. Standing person and silhouette from our database	29
Figure 4.5. Walking person and silhouette from our database	29
Figure 4.6. Noise occurred by lighting difference in image segmentation	30
Figure 4.7. Target rectangle and tracking window	31
Figure 4.8. Tracking of standing human	32
Figure 4.9. Tracking of human in different positions	32
Figure 4.10. Tracking frame by frame of walking human	33
Figure 4.11. Illustration of silhouette shape representation [26]	34
Figure 4.12. Image sequence from walking figure	34

Figure 4.13. Centroid of each image	35
Figure 4.14. An example of creating the block sizes	36
Figure 4.15. Moving points for these images, and magnitude of the flow [54]	37
Figure 5.1. Schematic of experimental apparatus	40
Figure 5.2. Frames of result man_walking1 video	43
Figure 5.3. Frames of result man_walking2 video	44
Figure 5.4. Frames of result Melih video	45
Figure 5.5. Frames of result Selcuk video	46
Figure 5.6. Frames of result Melih_2 video	47
Figure 5.7. Frames of result Selim video	48
Figure 5.8. Frames of result Selcuk_run video	49
Figure 5.9. Frames of result Selcuk_run video with new values	49
Figure 5.10. Frames of result Selcuk_run2 video with new values	50
Figure 5.11. Frames of result Selcuk_2 video with new values	50
Figure 5.12. Frames of result man_walking1 video with new values	51
Figure 5.13. Some samples in the SOTON gait database	55

LIST OF TABLES

Table 5.1.	Speed of movement table of different human motions	41
Table 5.2.	Some values from self-recorded videos	42
Table 5.3.	Walking man video results	52
Table 5.4.	Motion results for Selcuk_run2	52
Table 5.5.	Motion results for Selcuk_2	52
Table 5.6.	Motion results for Melih	53
Table 5.7.	Motion results for Selim	53
Table 5.8.	Walking_man video results with new threshold	54
Table 5.9.	Motion results for Selcuk_2 with new threshold	54
Table 5.10.	Motion results for Selcuk_run	54
Table 5.11.	Motion results for Selim with new threshold	55
Table 5.12.	Comparison of several different approaches on SOTON gait database	56
Table A.1.	RGB to HSI Matlab Code Implementation	59
Table B.1.	Low pass function	62
Table B.2.	Region of interest function	63

Table B.3.	Block_average function	64
Table B.4.	Mean filter function	65
Table B.5.	Draw roi function	66
Table B.6.	Create action movies function	66
Table B.7.	Gait centroid speed function	68
Table B.8.	Action recognizer function	70

LIST OF SYMBOLS / ABBREVIATIONS

A^c	Complement of set A
$a \parallel b$	Concatenation of string a and b
$b(x) \leftarrow 1$	1 goes to b(x)
$f(x,y)$	Image pixel value at coordinates x and y
$m(x)$	Weighted mean of density
m	Mean vector
O	Point at infinity
(x, y)	Cartesian coordinate pair x and y
$x \leftarrow X$	x randomly selected from X
$x \subseteq A$	x is subset of (or is included in) A
$\bar{x} = x_c, \bar{y} = y_c$	Centroid of image in x and y positions
\hat{x}	State space equation
xP	x times P added
y_k	Output signal of calculated label
\circ	Opening operator
\bullet	Closing operator
\emptyset	Empty set
\cup	Union of sets
\cap	Intersection of sets
α	Empirical weight
\notin	Not element of set A
\in	Element of set A
θ	Angle of threshold value
\ominus	Erosion operator
μ_t	Average of n number of pixels
\oplus	Dilation operator
$\varphi(\cdot)$	Activation function

ANN	Artificial Neural Network
CMY	Cyan, Magenta, Yellow color triple
DARPA	Defense Advanced Research Project Agency
HMM	Hidden Markov Model
HSV	Hue, Saturation, Value color triple
KF	Kalman Filtering
MHT	Multiple Hypothesis Tracking
RGB	Red, Green, Blue color triple
STK	Shi-Tomasi Kanade Tracker
YCrCb	Luma, Red chroma, Blurring color triple

1. INTRODUCTION

Biometrics is a technology that makes use of the physiological or behavioral characteristics to authenticate the identities of people [1]. It is used in a wide variety of applications, which makes a precise definition difficult to establish. The combination of human motion analysis and biometrics in surveillance systems has become a popular research direction over the past few years. There are numerous biometric measures which can be used to help derive an individual's identity. Biometric identification should be an automated process, as manual feature extraction will be time consuming, undesirable and costly. Recognizing human action from image sequences is an active research area in biometrics and computer vision. It attempts to detect, track and identify people. More generally, it aims to understand the human behavior from image sequences [2]. This is called gait analysis in the literature.

1.1. GAIT ANALYSIS

The usual meaning of “gait” is “manner of walking” or “manner of movement on foot” [3]. Recognition by gait is actually one of the newest biometrics [3]. It has been receiving growing interest within the computer vision community. Therefore, several gait metrics have been developed [4]. This interest is strongly driven by the need for automated person identification systems for visual surveillance and monitoring applications in security-sensitive environments such as banks, parking lots, and airports.

The first human ID program initialized by DARPA (Defense Advanced Research Project Agency) [5], aims to develop multimode surveillance technologies for successfully detecting, classifying and identifying humans to enhance the protection of facilities from terrorist attacks [6]. Its focus is on dynamic face recognition and recognition from body dynamics including gait. Gait based human identification aims essentially to discriminate individuals by the way they walk. It is closely related to methods that deal with whole-body human movements. In Figure 1.1, different techniques used to identify gait signatures are shown.

Approx. Year	<i>Model-free analysis</i>		<i>Model-based analysis</i>	
	Moving Shape		Modeled	
1994 to 2000	spatiotemporal pattern [30]; Principal Components Analysis (PCA) [25]	shape of motion [21]; PCA + Canonical Analysis [16]	single oscillator [8]	
2001 On-wards	Moving Shape unwrapped silhouette [42]; silhouette similarity [31]; relational statistics [39]; self similarity [1]; key frame analysis [7]; frieze patterns [22; area [10]; symmetry [15]; key poses [49]	Shape+Motion eigenspace sequences [43]; hidden Markov model [18, 35]; average silhouette [24]; moments [32]; ellipsoidal fits [20]; kinematic features [4]; gait style and content [19]	Structural stride parameters [3]; human parameters [5]; joint trajectories [44]	Modeled articulated model [41]; dual oscillator [46]; linked feature trajectories [48]
		video oscillations [6]		

Figure 1.1. Approaches to automatic gait recognition [3]

Different from gait based human identification, gait analysis classifies human motion into categories, such as walking, running, and jumping [2]. The most attractive feature of gait as a biometric is its unobtrusiveness. In comparison with other first-generation biometrics such as fingerprint, face, hand gesture, iris, and voice can be captured only by physical contact or at a close distance from recording probe [7, 8]. We can summarize the advantages of the gait analysis as a biometric identification technique as follows:

- **Unobtrusive:** The gait of a person walking can be extracted without the user knowing they are being analyzed and without any cooperation from the user in the information gathering stage unlike fingerprinting or iris.
- **Distance recognition:** The gait of an individual can be captured at a distance unlike other biometrics such as fingerprint recognition.
- **Reduced detail:** Gait recognition does not require images that have been captured to be a very high quality unlike other biometrics such as face recognition, which can be affected easily by low resolution.
- **Difficult to conceal:** Gait of an individual is difficult to disguise, by trying to do so the individual will probably appear more suspicious. With other biometrics

techniques such as face recognition, the individuals face can be easily altered or be hidden.

Despite the above advantages of the gait, an individual's gait signature can be affected by certain factors, such as:

- Stimulants: Drugs and alcohol will affect the way in which a person walks.
- Physical changes: During pregnancy, after an accident affecting the leg, or after severe weight gain / loss can all affect the movement characteristic of an individual.
- Psychological: A person's mood can also affect his or her gait.
- Clothing: Wearing different clothing may cause an automatic signature extraction method to create a widely varying signature for an individual.

Although these disadvantages are inherit in a gait biometric, the process of automatic gait extraction is more difficult due to external factors such as lighting conditions, self occlusion, and type of clothes which can all affect the data acquisition process [9, 10].

1.2. APPLICATION AREAS AND SCENARIO

Gait recognition can be used in a number of different scenarios. As gait is fundamental to human motion, it is likely that gait could find deployment in many other areas. The potential applications of human motion capture are the most significant factor of development. If an individual walks by the camera whose gait has been previously recorded and they are a known threat, then the system will recognize it and the authorities can be automatically alerted. The person can be controlled, before it becomes a threat. In Figure 1.2, a simple gait identification system architecture can be seen. According to this, the system can identify the suspicious motions in crowded areas according to the sample motion frames in its database. After this process, if the system identifies high risk, then it can start the alarm process.

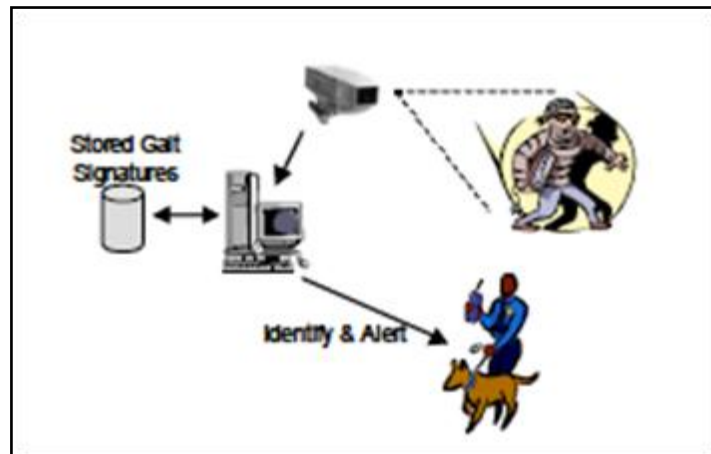


Figure 1.2. Gait recognition in action [11]

When considering crowded areas, we can separate them to three major parts as surveillance, control, and analysis. The surveillance area covers applications where one or more subjects are being tracked over time and possibly monitored for special actions [11]. Classic examples for the surveillance are parking lots, airports, banks, and shopping malls. Biometric identification systems like gait recognition has ability to identify the suspicious actions in these places.

The control area is based on applications where captured motion is used to provide controlling functionalities. It could be used as an interface to virtual environments or animation to control remotely located implementations [12].

The third and the last application area is focused on the detailed analysis of the captured motion data. This application type is very useful in clinical studies, diagnostic of orthopedic patients or athletes to improve their performance. Therefore, such systems have large amount of potential application domains.

1.3. LAYOUT OF THE THESIS

The main objective of this thesis is developing a system capable of recognizing gait of individuals derived from a video sequence. It has basic modules as follows: statistical background extraction from the video frames; tracking humans; classifying their gait like

walking and running. Input images for system are taken from a commercial web camera. To explain these modules, some image processing fundamentals are described. Also, feature extraction for classification is described. This is followed by the implementation details. Finally, the system outputs are given with the conclusion and what can be done as a future work.

2. LITERATURE REVIEW

2.1. OVERVIEW

Human gait has always been an active research topic in biometric, kinesiology, and physical medicine. The study of a gait as a discriminating trait was first attempted a few decades ago from medical and behavioral viewpoint [13, 14]. Later, several attempts were made to investigate the gait recognition problem from the perspective of capturing and analyzing gait signals [15]. Gait recognition is closely related to computer vision methods that detect, track and analyze human behaviors as discriminating different human motion types, such as running, walking, jogging, and standing. Generally, methods can be categorized into three classes according to their properties as: feature, model and optical-flow based [16]. Their individual characteristics will be summarized in the following sections. The common point for all these methods is the tracking of an object or human silhouette. Therefore, we first introduce the basic tracking methods.

2.2. TRACKING

Real time and autonomous object tracking remains a challenging task for various computer vision applications such as: visual surveillance, driver assistance, and human action understanding. The goal of object tracking is to determine the position of the object in the image sequence continuously and reliably in dynamic scenes. Object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Tracking objects can be complex due to: loss of information, noise in images, complex object motion, complex object shapes, and real-time processing requirements.

Parametric contours have been applied successfully to achieve real-time performance but they have difficulties in handling topological changes such as merging and splitting of object regions [11]. The level set method is a more powerful technique and has various models. However, the high computational cost of the level set method has limited its

popularity in real-time scenarios [11]. Therefore, background subtraction is the best choice for real time applications according to its computational cost and simplicity.

2.2.1. Background Subtraction

Tracking objects or humans using background subtraction is commonly used both indoor and outdoor environments. Background subtraction is a simple but effective method for detecting moving objects by comparing the intensity of the background and observed images in videos. General method in the background subtraction is detecting the moving objects from the difference between the current frame and a reference frame which can be called as the background image.

There are several methods explained below for background subtraction. Simple methods aim to the maximize speed and minimize the memory requirements. However, more sophisticated approaches aim to achieve more accuracy under different circumstances. These can be counted as: running Gaussian average, temporal median filter, frame difference, mixture of Gaussians, and eigenbackgrounds.

2.2.1.1. Running Gaussian Average

Wren *et al.* [11] proposed to model the background at each pixel location. The model is based on ideally fitting a Gaussian probability density function on the last n pixel values. Instead of evaluating each frame from scratch, the average is computed as

$$\mu_t = \alpha I_t + (1 - \alpha) \mu_{t-1} \quad (2.1)$$

where, I is the pixel's current value and μ_t is the previous average. α is an empirical weight often chosen as a tradeoff between stability and quick update. In addition to speed, the running average is given by the memory requirement for each pixel. Koller *et al.* [17] remarked that model as unduly updated also at the occurrence of such foreground values [18]. Intensity images, build ups can be made with color-spaces such as (R, G, B) or (Y, U, V).

If real time requirements constrain the computational load, the update rate can be set to less than the sample rate. However, when the background model has lower update rate, then this decreases the system's ability for quickly responding to the actual background dynamically.

2.2.1.2. Temporal Median Filter

Lo and Velastin [19] proposed to use the median value of the last n frames as the background model computed as

$$\mu_t = \alpha I_t + (1 - \alpha) \mu_{t-1} \quad (2.2)$$

Cucchiara *et al.* [20] argued that such a median value provides an adequate background model even if the n frames are subsampled with respect to the original frame rate by a factor of 10. In addition, they proposed to compute the median on a special set of values containing the last n , sub-sampled frames and w times the last computed median value. The main disadvantage of this approach is that its computation requires a buffer with recent pixel values [21].

2.2.1.3. Frame Difference

As we have implemented this method in our system, we will investigate it in the following chapters in detail.

2.2.1.4. Mixture of Gaussians

In the background subtraction process, some background objects are not permanent and appear at a rate faster than the background update. A typical example is that of an outdoor scene with trees partially covering a building: the same pixel location will show values from tree leaves, tree branches, and the building itself [21]. Stauffer and Grimson [22] raised the case for a multi-valued background model, which is able to cope with multiple background objects. Actually, their model can be more properly defined as an image model since it provides a description of both foreground and background values. At each t frame time, two problems must be simultaneously solved a) assigning the new observed value x to the best matching distribution and b) estimating the updated model parameters. Among many studies stemming from Stauffer and Grimson's work, Power and

Schoonees [22] elegantly describes the theoretical framework and provides useful corrections.

2.2.1.5. Eigenbackgrounds

Eigenbackgrounds approach, proposed by Oliver *et al.* [23], is also based on the same idea, but applied on the whole image. In this method, there are two phases: learning and classification. In the learning phase, samples from the image are acquired. Then, the average image is computed and subtracted from all images. In the classification phase, for every new image a new eigenspace is projected and then this is backprojected onto the image space. Finally, foreground points are evaluated for the specified locations. This procedure can be subject to variations for improving its efficiency. Stauffer and Grimson [24] built and iteratively updated a multi model with static background. Black *et al.* [25] utilized the robust statistics with eigenbackgrounds to identify the training data and improve the specificity of the models. Eigenbackgrounds is an ideal background subtraction method that detects an object with exact sensitivity. However, this method is not practical because of its computational cost.

2.3. FEATURE BASED METHODS

Feature based methods were originally developed for tracking a small number of features in a long sequence [16]. They involve extraction of regions of interest in the image sequence and then identification of the counterparts in individual images of the sequence. The advantages of a model free approach are that the methods derived are not linked to one object. Therefore, a method for detecting human gait can also be used for animal gait analysis and vice versa with little changes [12]. This type of approach, capturing the outline of a motion and trying to analyze it is based on data already collected or some previous specified information regarding the image. In Figure 2.1, a boundary of a human silhouette and also centre of the human has been specified as an example.

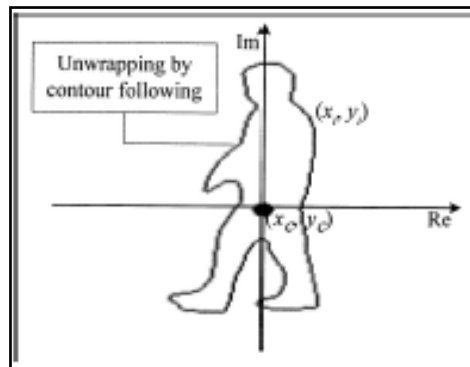


Figure 2.1. Boundary of a silhouette [26]

Typical feature based algorithms are: multiple hypothesis tracking, hidden Markov models, artificial neural networks, Kalman filtering and mean shift.

2.3.1. Multiple Hypothesis Tracking (MHT)

Reid [27] first proposed the MHT algorithm. It is generally accepted as the preferred method for solving the data association problem in modern target tracking systems. Target tracking is an essential requirement for surveillance systems employing one or more sensors, together with computer subsystems, to interpret the environment [28]. Reid's algorithm defines a systematic way in multiple data association hypotheses that can be formed and evaluated for the problem of multiple targets in a false alarm background. This algorithm considers the multiple tracking candidates and intends to find the best fit to the real image descriptors. However, the classical MHT technique by itself is computationally expensive both in time and memory. Therefore, it is not applicable for real-time applications.

2.3.2. Hidden Markov Model (HMM)

A Hidden Markov Model is normally used to extract the transformation between two images or moving 3D structures in object tracking. HMM is especially known for its application in temporal pattern recognition such as speech, handwriting, gesture recognition, partial discharges and bioinformatics. Kale and Sundaresan [29, 30] used HMM for their deployment which consider two different image features; the width of the

outer contour of binarized silhouette and the binary silhouette itself [3]. Using the Baum-Welch algorithm [2] and its modifications, researchers train HMM by adjusting the weights of the transitions to better model the relationship of the actual training samples [16, 31]. In Figure 2.2, an example of probabilistic parameters of Hidden Markov Model is provided. Also, in the same figure, the state transition probabilities and finally the output possibilities are given.

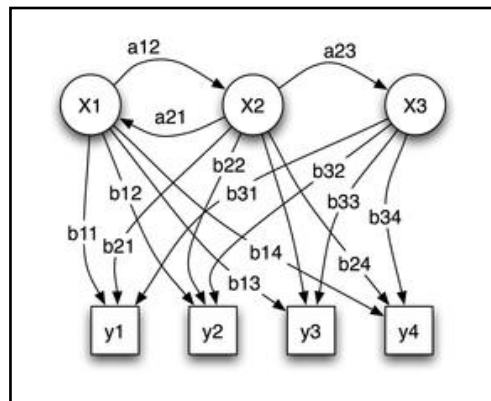


Figure 2.2. Probabilistic parameters of a Hidden Markov Model X – states, Y – possible observations, a – state transition probabilities, b - output probabilities

HMM based approaches do not require analytical solutions to certain problems. This method is also effective in handling very complicated problems. In Figure 2.3, you can see the general architecture of HMM. Here the random variable $x(t)$ is hidden state at time t where $(t) \in \{x_1, x_2, x_3\}$ and the random variable $y(t)$ is the observation at time t , where $y(t) \in \{y_1, y_2, y_3, y_4\}$. The arrows also denote the conditional dependencies.

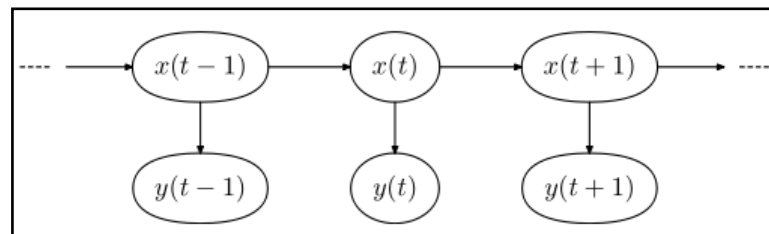


Figure 2.3. General architecture of the HMM.

2.3.3. Artificial Neural Networks (ANN)

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems. Novel structure of the information is the key element for this paradigm. ANN is composed of the high number of processing elements defined as neurons. Learning in biological systems involves the adjustments of the connections that exist between these neurons. By embedding number of simple neurons in an interactive nervous system, it is possible to provide computational power for different kind of sophisticated processes.

2.3.4. Kalman Filtering and Mean Shift

The Kalman filter is used to estimate the state of a linear system where the state is assumed to be Gaussian distributed. From a theoretical standpoint, the Kalman filter is an algorithm for efficiently calculating the interference in a linear dynamical system, which is a Bayesian model similar to HMM. However, the state space is continuous and all variables have a Gaussian distribution. Kalman filtering produces the estimates of the true values of measurements and their associated values by predicting a value. This was first described and introduced partially in technical papers by Swerling [32] and Kalman [32]. The filter is named after Kalman, because he published his results in a more prestigious journal and his work was more general. The Kalman filter model assumes that the true state at time k is evolved state from $(k-1)$. The Kalman filter consists of three equations, each manipulating the matrix representation as

$$K_k = AP_k C^T (CP_k C^T + S_Z)^{-1} \quad (2.3)$$

$$\hat{X}_{k+1} = (A \hat{x}_k + Bu_k) + K_k (y_{k+1} - C \hat{x}_k) \quad (2.4)$$

$$P_{k+1} = AP_k A^T + S_W - AP_k C^T S_Z^{-1} CP_k A^T \quad (2.5)$$

where T indicates transpose, K is called the Kalman gain and P matrix refers to error covariance. Also \hat{x} is the state space equation and $k+1$ is just A times the state estimate at time k . The Kalman filter method has been extensively used in computer vision community for tracking and hundreds of diverse areas such as, aerospace, nuclear power plant instrumentation, manufacturing, fuzzy logic and neural network training.

Mean shift algorithm is a nonparametric clustering technique that does not require prior knowledge of the number and shape of the clusters. Given n data points on a d dimensional space with kernel $K(x)$ and window radius h is;

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2.6)$$

where x means a sample point and n is the total number of points. To determine the weight of nearby points for re-estimation of the mean, one can use

$$K(x_i - x) = e^{-c\|x_i - x\|} \quad (2.7)$$

The weighted mean of the density in the window determined by K is

$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x)x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \quad (2.8)$$

where $N(K)$ is the neighborhood of K , a set of points for which $K(x) \neq 0$.

This is an iterative method starting with an initial estimate. Tracking is accomplished by iteratively finding the local minimum of the distance measure functions. This method is originally presented in 1975 by Fukunaga and Hostetler [33]. It was first introduced into the image processing community several years ago by Cheng [33]. Recently, Comaniciu and Meer [34] successfully applied this method to image segmentation and tracking. Also Yang [32] proposed a simple method to compute measures in spatial feature specifications.

This similarity measure allows the algorithm to track more general motion models in an integrated way.

2.4. MODEL BASED METHODS

Model based tracking is an example of feature-based tracking. The reason why it is independently described is due to requirement of grouping, reasoning and rendering which may defer from the feature-based tracking [16]. Model based approaches build up a model of how one would walk and then try to fit their image data to model. To succeed this prior knowledge, the investigated models are normally required. Unlike feature-based approaches, model-based approaches are in generally view and scale invariant. This is a significant advantage over feature-based methods. However, since model-based approaches rely on the identification of specific gait parameters in a sequence, they need high quality images as input to be useful. In Figure 2.4, one can see the angles between joints and the different length types of human body to be used in modeling.

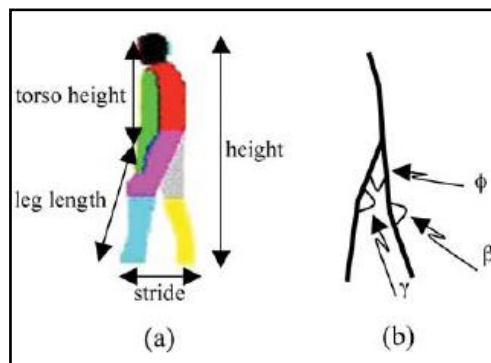


Figure 2.4 a) Structural gait description b) Relative modeling based on angles [3].

Model-based approaches concentrate on dynamics of a body shape. When modeling the body, there are various kinematical and physical constraints [18]. In Figure 2.5, different modeling types of human are given. Models used are typically stick representations as ribbons or blobs.

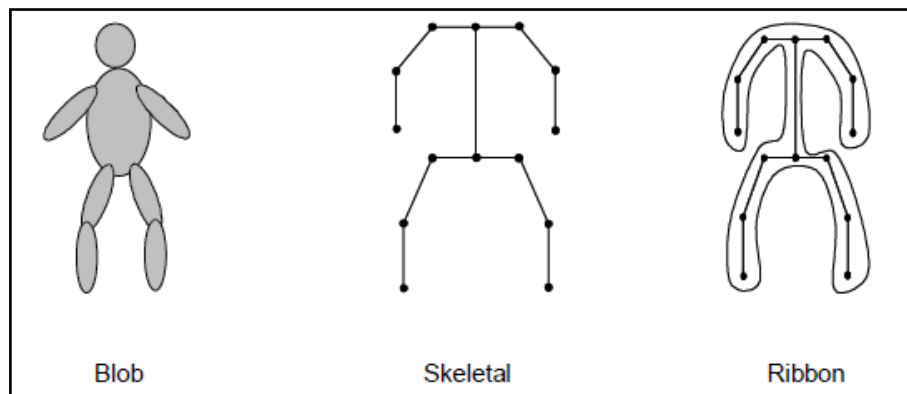


Figure 2.5. Different modeling types [3]

Wren *et al.* [11] produced a real time people finder program (PFinder) that models human body using set of blobs. BenAbdelkader *et al.*'s [36] approach using self similarity is a prime example of model-based approach. Bobick *et al.* [37] used structural human parameters as another example for structural model based approach. The method used the action of walking to derive relative body parameters which describe the subject's body and stride [3]. Wang *et al.* [37] used a model based approach derived from the dynamic information of gait by using a condensation framework to track the walker and recover joint-angle trajectories of lower limbs [3]. Lowe [38] used the Marr-Hidreth edge detector to extract edges from an image, which were chained together to form lines. Then, these lines were matched. Hough transform was utilized to achieve a similar idea [39]. Gavrilu and Davis [40] matched edges in the image with an appearance model using distance transforms. A decomposition approach and a best-first technique were used to search through the high dimensional pose parameter space.

The advantages of model base approaches are that whole image can be used before making a choice on the model fitting. This type of approach can handle the occlusion and noise better [18]. Gait signatures can be driven directly from model parameters. Also this helps to reduce the dimensionality needed to represent the data.

Besides the advantages of the model based approaches, there are also some disadvantages. Implementing a model based approach has a high computational cost due to

complex matching and searching that has to be performed. Besides that one needs more memory to meet the implementation demands.

2.5. OPTICAL FLOW BASED METHODS

Almost all work on image sequences begin with attempting to find the vector field which describes the image changing with time [41]. Optical flow is a dense field of displacement vectors defining the translation of each pixel in a region with time. To find the optical flow in the image sequence, researchers attempted to use feature based and correlation based approaches. Optical flow is commonly used as a feature in motion-based segmentation and tracking applications such as motion detection, object segmentation, time to collision. It is computed using the brightness constraint, which assumes brightness constancy of corresponding pixels in consecutive frames. In Figure 2.6, one can see an optical flow example experienced by rotating observer and magnitude of optical flow at each location is represented by the direction and length of each arrow.

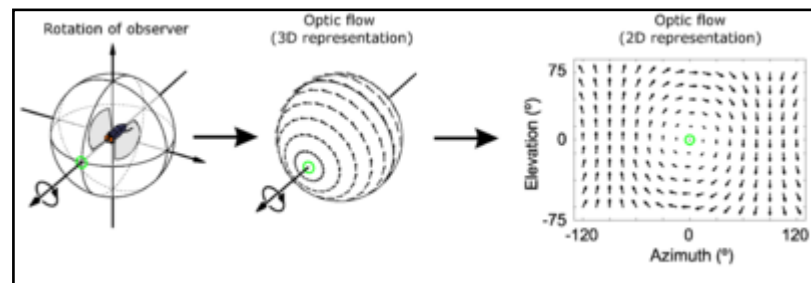


Figure 2.6. Optic flow experienced by a rotating observer. At each location represented by the direction and length of each arrow [52]

This computation is often carried out in the neighborhood of the pixel either algebraically by Lucas [42] or geometrically by Shunck [43]. Shi and Tomasi [44] proposed a well known Shi-Tomasi-Kanade (STK) tracker which iteratively computes the translation of a region centered on an interest point. This tracking version works fast and efficiently in most circumstances. Black [45] proposed a coarse optical flow method developed via correlation. Also Buxton [46] proposed a method based on a model of a motion of edges in an image sequence.

Optic flow is found by matching two dimensional features. One of the strongest advantages of this approach is the fact that it is not affected from the problems of the aperture. Secondly, the work is largely aimed at producing good results from images of real events, especially taken from outside. Thus there is a rarely a difference between the image flow and projected world motion so as a result the term optic flow will be used freely and generally.

3. IMAGE PROCESSING FUNDAMENTALS

Digital image processing is focused on data processing on a 2-D array of numbers as $f(x, y)$ [47]. The array is the numeric representation of an image. It can be represented as

$$f(x, y) = \begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & \cdots & \vdots \\ f(N,1) & f(N,2) & \cdots & f(N,N) \end{bmatrix} \quad (3.1)$$

3.1. COLOR MODELS ON IMAGE PROCESSING

3.1.1. RGB Color Model

RGB is an additive color model in which red, green, and blue light are added together in various ways to reproduce an array of colors [48]. The main purpose of the RGB color model is displaying the images in electronic systems, such as televisions and computers.

RGB is device dependent and based on the Cartesian coordinate system. Different devices detect or reproduce a given RGB value differently, since the color elements and their response to individual R, G, and B levels are changing from manufacturer to manufacturer. Thus an RGB value does not define the same color across devices. In Figure 3.1, you can see the RGB color scheme. According to the figure, the intersection of the red and blue gives magenta, cyan occurs due to intersection of the blue and green and finally yellow is the intersection area of green and red.

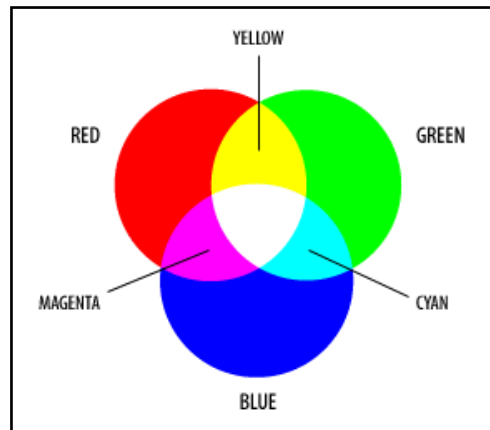


Figure 3.1. RGB color scheme [48]

3.1.2. CMY Color Model

CMY is a subtractive model and used for printing process. It is actually CMYK which refers to four different inks used in printing, Cyan, Magenta, Yellow, and Key black. As RGB (Red, Green, Blue) is the primary color model, the secondary colors derived from primary is mentioned as CMY (Cyan, Magenta, Yellow). In Figure 3.2, you can see the opposite set type of the RGB. This time Green, Red, and Blue occur due to intersection of the secondary color model.

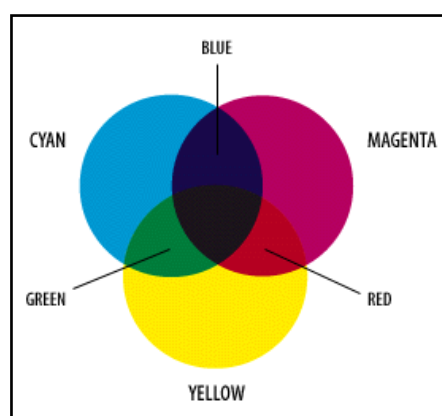


Figure 3.2. CMY color scheme [49]

It is necessary to calculate the differences in intensities to obtain the components R, G and B for a true display on a computer screen. This is the reason of the subtractive

name for CMY combination [49]. The set of equations between the three components of RGB and CMY is as follows

$$\begin{aligned} R &= Y + M - C \\ G &= Y + C - M \\ B &= M + C - Y \end{aligned} \quad (3.2)$$

3.1.3. HSI Color Model (HSV, HSL)

This color space depends on the idea of tint, saturation and tone. Hue component defined the dominant color (such as red, green and yellow) of an area [50]. HSL and HSV are the most common cylindrical-coordinate representations of points in an RGB color model. They were developed in the 1970s for computer graphics applications, and used for color-modifications tools in image analysis and computer vision [51]. Figure 3.3 shows the HSI color scheme according to the value and lightness.

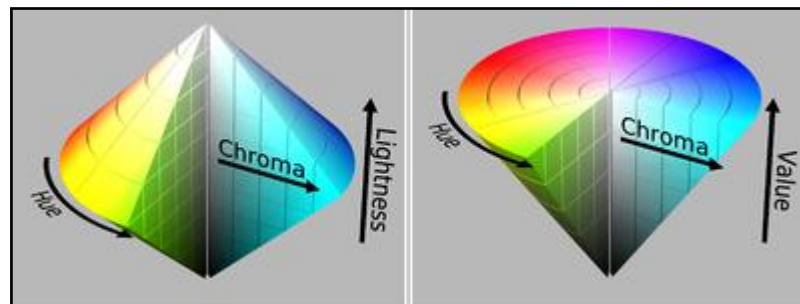


Figure 3.3. HSI color scheme [52]

This model is the best for human eye as it is kind of describing color. When humans view a color object, we describe it with its hue, saturation and brightness [52].

3.1.4. RGB to HSI Conversion

When converting the RGB values to HIS, the normalized RGB values are used as

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B} \quad (3.3)$$

Using these values, we can obtain the HSV values as

$$h = \cos^{-1} \left\{ \frac{0.5 * [(r-g) + (r-b)]}{(r-g)^2 + (r-b)(g-b)^{\frac{1}{2}}} \right\} \quad h \in [0, \pi] \text{ for } b \leq g \quad (3.4)$$

$$h = 2 * \pi - \cos^{-1} \left\{ \frac{0.5 * [(r-g) + (r-b)]}{(r-g)^2 + (r-b)(g-b)^{\frac{1}{2}}} \right\} \quad h \in [\pi, 2\pi] \text{ for } b > g \quad (3.5)$$

$$s = 1 - 3 * \min(r, g, b); \quad s \in [0, 1] \quad (3.6)$$

$$h = H * \frac{\pi}{180}; \quad s = S / 100; \quad i = I / 255 \quad (3.7)$$

Also, we can obtain the RGB representation starting from HSI as

$$x = i * (1 - s) \quad (3.8)$$

$$y = i * \left[1 + \frac{s * \cosh}{\cos(\pi/3 - h)} \right] \quad (3.9)$$

$$z = 3i - (x + y) \quad (3.10)$$

When $h < 2\pi/3$, $b = x$; $r = y$ and $g = z$

When $2\pi/3 \leq h < 4\pi/3$, $h = h - 2\pi/3$ and $r = x$; $g = y$; $b = z$ (3.11)

When $4\pi/3 \leq h < 2\pi$, $h = h - 4\pi/3$ and $r = z$; $g = x$; $b = y$

The resulting r, g, b values are normalized, which are in the range of [0, 255]

3.2. MORPHOLOGICAL FILTERS

Mathematical morphology is used for extracting image components that are useful in representation and description of a shape [53]. Mathematical morphology is a set and was initiated in the late 1960's to analyze binary images from geological and biomedical data [54, 55]. In the late 1970's, this process was extended to gray-level images and in the mid of the 1980's it was brought to the mainstream of image processing [55]. The above evaluation of ideas has formed as field of morphological image processing. All morphological filters are based on two basic operations which are erosion and dilation.

3.2.1. Erosion

The erosion operation is defined as the minimum of the difference of a local region of an image. The basic effect of this operator is to remove the boundaries of foreground pixels. Therefore, it is an excellent way of removing small noise in the image. The erosion of an input image A by a structuring element B is defined as

$$A \ominus B = \{x : B + x \subseteq A\} = \bigcap \{A + b : b \in -B\} \quad (3.12)$$

This means that to perform the erosion of A by B , we translate B by x so it lies inside A . The set of all points x satisfying this condition constitutes $A \ominus B$. In Figure 3.4 and 3.5, we can see example of erosion filter conversion and image example.

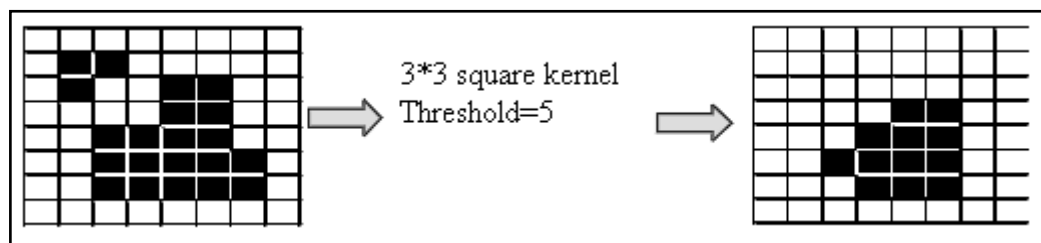


Figure 3.4. Example of erosion filter on binary image.

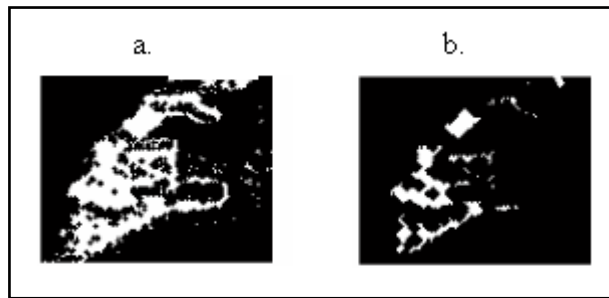


Figure 3.5. Erosion of an image, a. original image, b. erosion

3.2.2. Dilation

The dual operation to erosion is dilation. It is defined as the maximum of the sum of a local region of an image. The effect of dilation is to gradually enlarge the foreground region by a factor. Dilation of an input image A by a structuring element B is defined as

$$A \oplus B = \bigcup \{B + a : a \in A\} \quad (3.13)$$

This means that in order to perform the dilation of A by B , we translate B by all points of A . The union of the translation constitutes $A \oplus B$. In Figure 3.6 and 3.7, we provide an example of dilation filter conversion and image example.

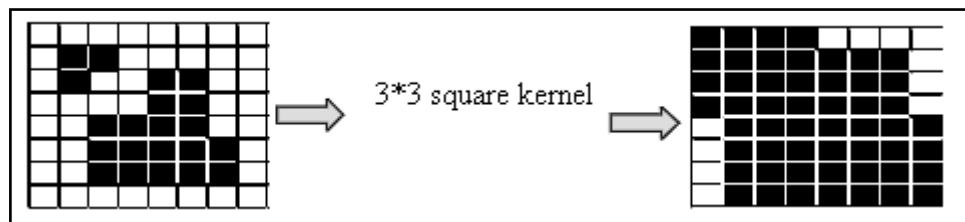


Figure 3.6. Example of dilation on binary image.

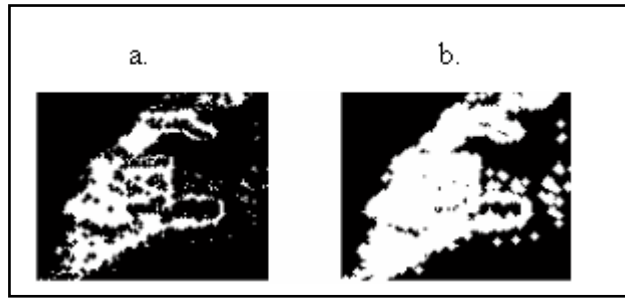


Figure 3.7. Dilation example of the same image, a. original image, b. dilation

3.2.3. Opening

A secondary operation of great importance in mathematical morphology is opening. Opening of an image is defined as the dilation of the erosion of the image given as

$$A \circ B = (A \ominus B) \oplus B = \bigcup \{B + x : B + x \subseteq A\} \quad (3.14)$$

Opening generally smooths the contour of an object and it breaks the narrow gaps [53]. This means that in order to open A by B, we translate the B by x so according to that definition x lies inside the A. In Figure 3.8, you can see an example from opening operation.

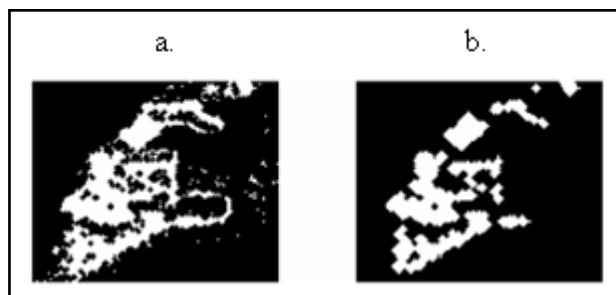


Figure 3.8. Opening example of the same image, a. original image, b. opening example

3.2.4. Closing

Closing is the dual of opening. Closing of an image is defined as the erosion of the dilation of an image as

$$A \bullet B = (A \oplus B) \ominus B \quad (3.15)$$

Closing operation eliminates small holes and fills the gaps in the contour [53]. This means that closing of A by B is the dilation of A by B, followed by erosion of the result by B. In Figure 3.9, you can see an example from closing operation.

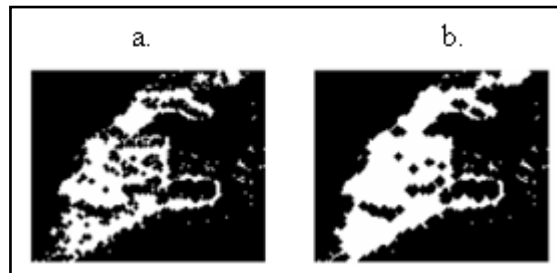


Figure 3.9. Closing example of an image, a. original image, b. closing example

4. GAIT ANALYSIS IMPLEMENTATION

Recognizing gait depends on how the silhouette changes for individual sequences. Our system for gait analysis works on the principle of frequency component of direction of movement. Before training and recognition, our system converts the movie into individual frames. In each image frame, the human object is detected.

4.1. FEATURE EXTRACTION

In the feature extraction procedure, first we extract the foreground and detect the action region. Image sequences are gathered while the subject is walking, running etc. As our focus is on motion, we can restrict the experimental situation to a single subject moving in the field of view. After this point, we calculate the centre of person in each frame. Since the centre of person contains the x and y coordinates and all the movements to be detected are due to displacement in horizontal direction, this is the reason why that component is only considered to find the frequency of movement. After that, we use certain threshold values to decide on the type of movement.

4.1.1. Human Detection and Tracking

Human detection and tracking is the first step to gait analysis. To track moving silhouettes of a walking or running human from the background image in each frame, change detection and tracking algorithm is applied which is based on background subtraction. The action region has been defined as the rectangular area where any specific action is occurred in the image frame. The action region depends on the human body shape. Usually the action region is smaller than the image area. Therefore, we select the action region inside the image frame which includes the average human body shape of specific image sequence. The main constraint here, as explained earlier, is the static camera and dynamic subject. However, this constraint performs well on our data set. Please note that, robust motion detection in an unpredictable environment is a challenging

problem for vision techniques as it contains number of problems such as shadows or motion clutters.

4.1.1.1. Background Subtraction

Background subtraction has been widely used in foreground object detection when the fixed camera is used while the scene is dynamic. The known background image is subtracted from the current frame, comparing the intensities of pixels, then thresholding is applied as

$$f(x, y) = \begin{cases} \text{background} & , \text{ if } \text{abs}(I_{\text{current}}(x, y) - I_{\text{known}}(x, y)) \leq \phi \\ \text{foreground} & , \text{ otherwise} \end{cases} \quad (4.1)$$

The process is complicated by the fact that if the background is not static, i.e. a changing television screen in the background should not be considered as a foreground object although the scene is continuously changing. By the order of the Figures 4.1 to 4.5 you can see the examples of background subtraction from different motions and also a good example of the consecutive images from a gait sequence.

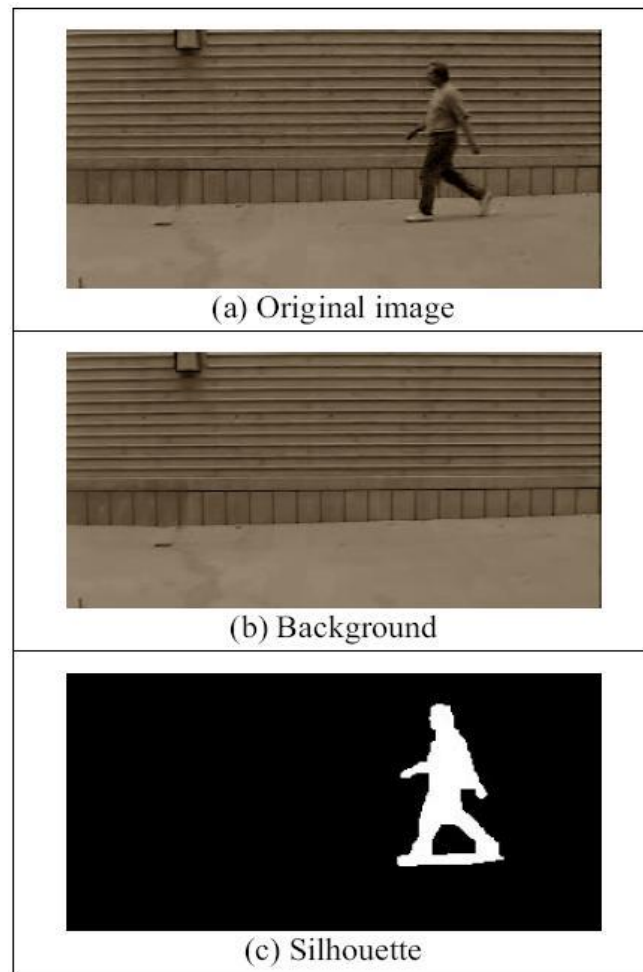


Figure 4.1. An example of silhouette extraction



Figure 4.2. Sample images of walking person



Figure 4.3. Running person and silhouette from our database



Figure 4.4. Standing person and silhouette from our database



Figure 4.5. Walking person and silhouette from our database

4.1.1.2. Lighting Differences

Lighting differences are always obtained between the background and the current image as brightness in the environment changes. We can call these illumination changes such as reflections, shadows, moving clouds or inter-reflections between sequences of the

moving figure frames. In Figure 4.6, you can see examples of problematic images containing noise caused by lighting difference.

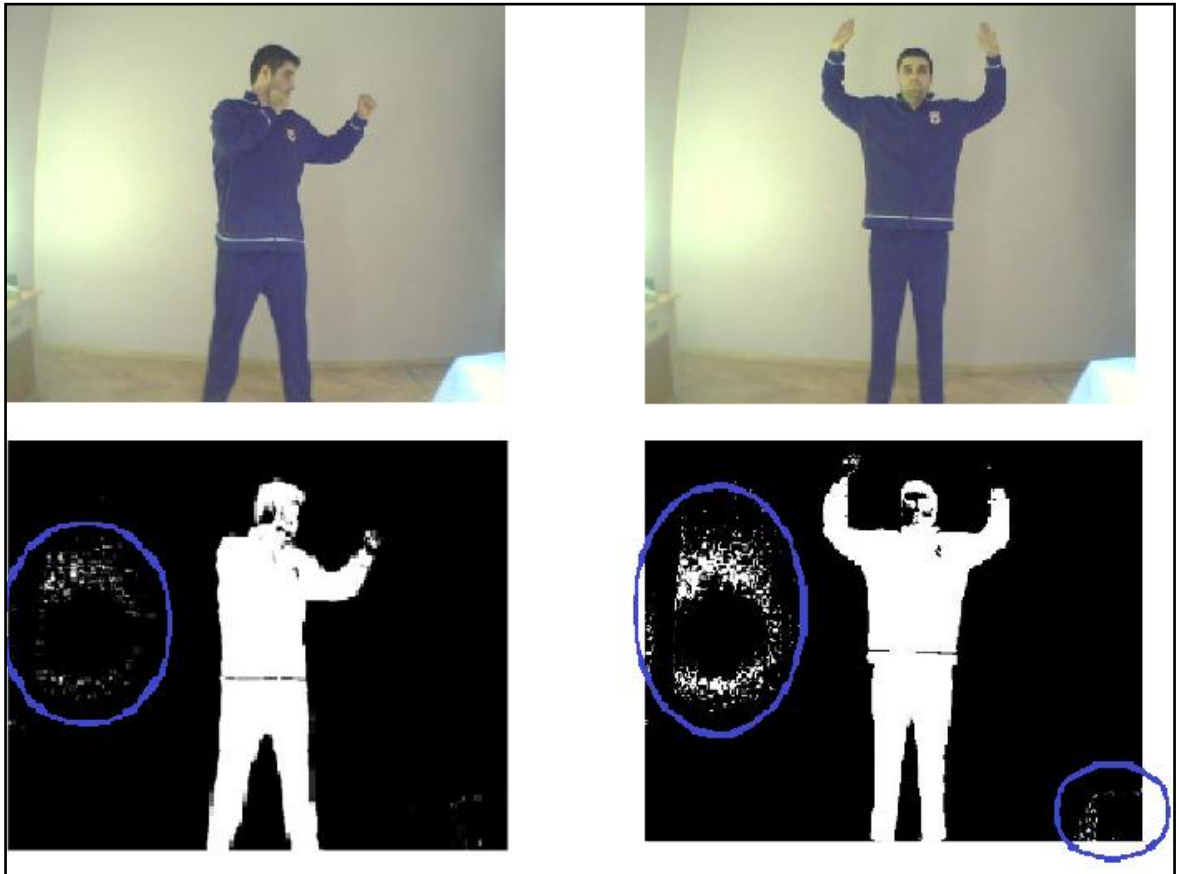


Figure 4.6. Noise occurred by lighting difference in image segmentation

Background subtraction between the current image and background image provides the foreground. However, it has different amounts of noise over the image and is not suitable for use. It is clearly seen from the above images the sharper light changes causes this noise in the foreground image. To isolate the moving figure, we first determine these noisy regions. Therefore, we benefit from the properties of the area of connected components in the thresholded image. Assuming that the human occupies most of the scene, we pick the largest two regions in the image to eliminate noise terms.

4.1.1.3. Finding the Action Regions

After eliminating the noise in the frames, each foreground region is tracked from frame to frame by a simple method based on bounding boxes. To do so, we define the

action regions as rectangular areas where any specific action has been occurred. The rectangle used to entirely include the tracked object is called the tracking window. This tracking window depends on the human body shape, distance between the video sensor and the person which is performing the action. In Figure 4.7, you can see an example of rectangle that we use as tracking window. The horizontal and vertical size of tracking window can be denoted as h_x and h_y respectively.

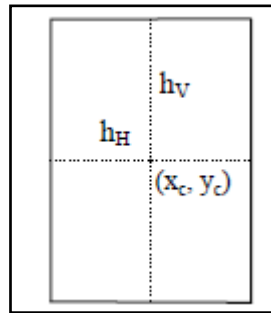


Figure 4.7. Target rectangle and tracking window

If we set parameters h_x and h_y as the half of the tracking window then we can set $h_H = 2h_x$ and $h_v = 2h_y$. Also the center of the target is (x_c, y_c) . Usually the action region is smaller than the image area. Therefore, the action region inside the image frame is selected which includes the human body shape. By the order of Figures 4.8, 4.9 and 4.10 we can see a tracking example of a standing human in a video sequence, tracking of human in different positions and lastly example of tracking frames from a walking human video sequence.



Figure 4.8. Tracking of standing human

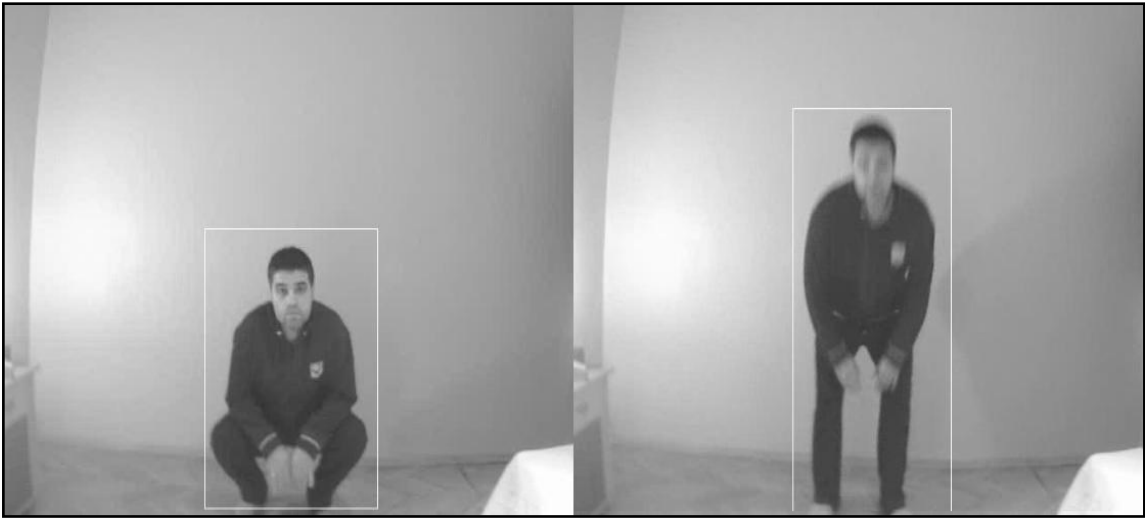


Figure 4.9. Tracking of human in different positions



Figure 4.10. Tracking frame by frame of walking human

4.1.1.4. Silhouette Representation

An important cue in determining the underlying motion of a walking figure is the temporal changes in the walker's silhouette. It makes the proposed method insensitive to changes of color and texture of clothing and the color of the foreground objects [25]. Since these silhouettes are the basis for identification, it is important that they should be clear and noise free. From the foreground image, when all noise effects have been removed, moving silhouette of a walking figure has been tracked. We compute the centroid of all points in x and y coordinates. By choosing the centroid as a reference origin, we unwrap the outer contour and turn it into the distance signal as $S = \{d_1, d_2, \dots, d_i, \dots, d_N, \}$ composed of all distances d_i between each pixel (x_i, y_i) .

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (4.2)$$

In Figure 4.11, the illustration of silhouette shape representation is given. In Figure 4.12, some walking frames from a video sequence are given. Finally, in Figure 4.13, we can see the same frames which the centroids of the frames have been calculated and shown on the frames.

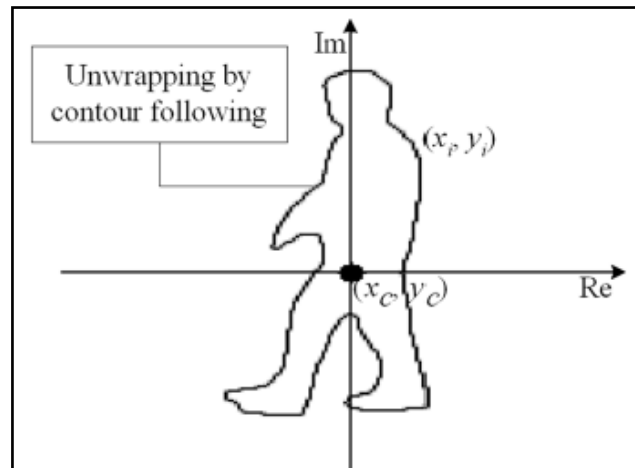


Figure 4.11. Illustration of silhouette shape representation [26]



Figure 4.12. Image sequence from walking figure

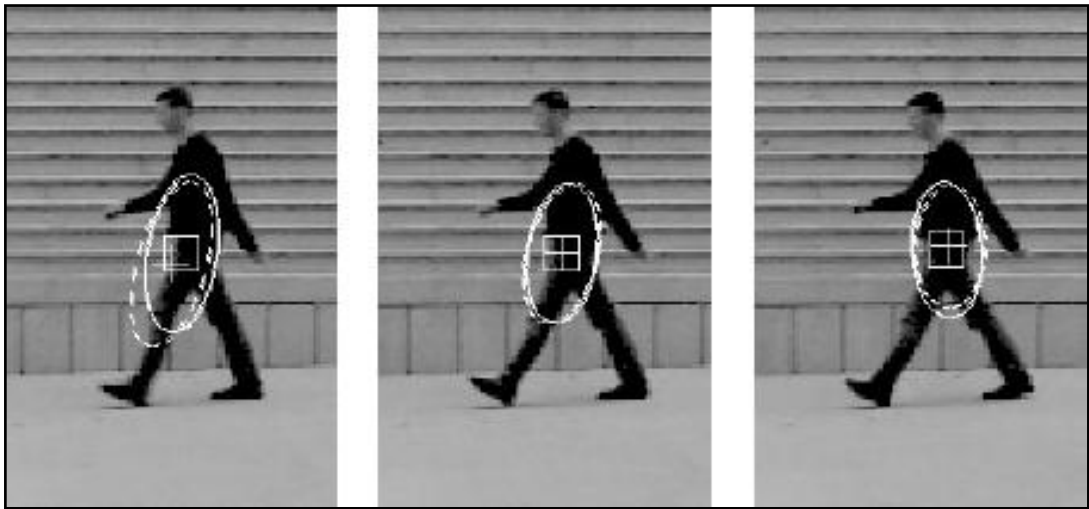


Figure 4.13. Centroid of each image

After we calculate all information from the centroid of the frames, we decided to add another process to stabilize results. Therefore, we formed the videos to a fixed block size. We have used 16 for block sizes. If we have a video sequence with size 484x644 or another different size than 480x640, we need to resize the video sequence to 480x640 for our program. We suppose to take the average of the pixel values to reach our purpose. We used 16 block size and take the average of the pixels horizontally and vertically. This means that each pixel can be represented using a block which contains its neighbors value. With this implementation, almost in all images, the value of a particular pixel depends on its neighbor.

We can usually expect the value of a pixel to be more closely related to the values of pixels close to it than to those further away. This is because most points in an image are spatially coherent with their neighbors. In Figure 4.14 you can see an example shown of the block sizes in horizontal and vertical directions.

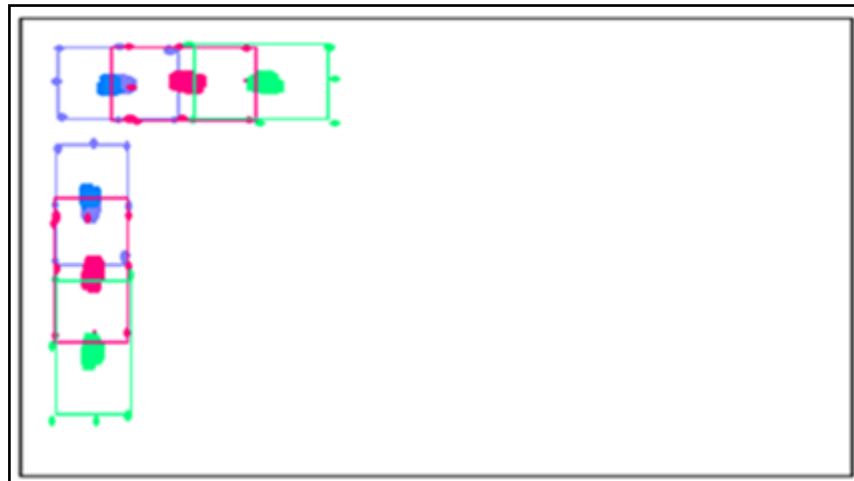


Figure 4.14. An example of creating the block sizes

4.1.1.5. Signature Extraction

After calculating the moving object properties, we continue the operation with matching the appropriate labels as: running, walking, no movement and raise hands. We used a simple type of optical flow method for this purpose. The optical flow algorithm, for each pixel, searches among a limited set of discrete displacements for the displacement $(u(x, y), v(x, y))$ that minimizes the sum of absolute differences between a frame in one image and the corresponding displaced frame in other image. The x and y coordinates of a centroid are two scalar measures of the distribution of motion.

In Figure 4.15, the first line shows some frames from a video sequence. Also in the second line you can see the moving points of the frames showed in the first line and finally in the third line you can see the magnitude of the frames showed.

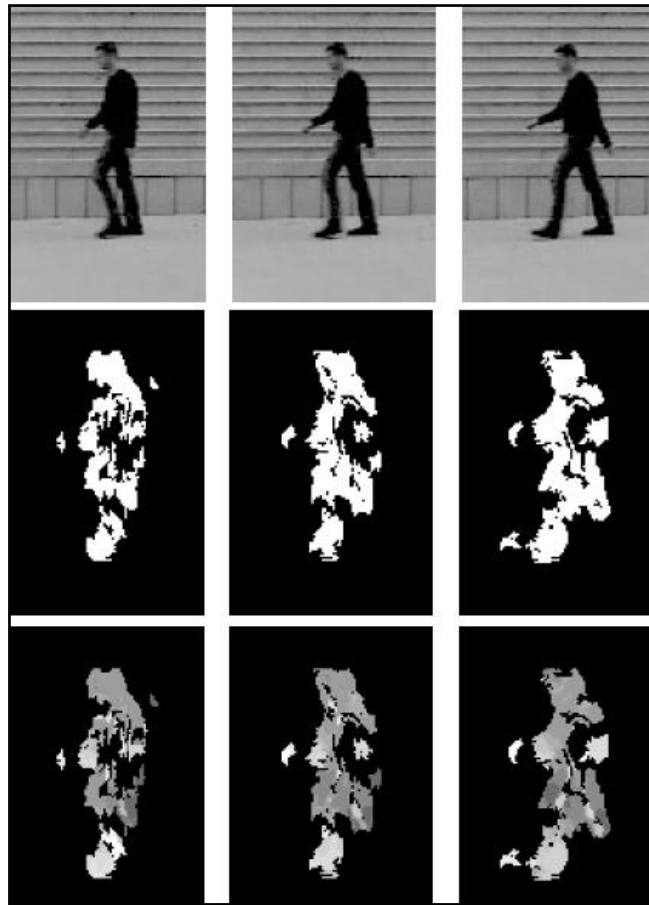


Figure 4.15. Moving points for these images, and magnitude of the flow [54]

The shape of motion varies periodically during the motion sequence. The relative positions of the centroid of T (moving points) vary systematically over the sequence. The ratio of the lengths of the major and minor axes is scale-invariant measure of the distribution of motion which reflects both position and velocity of the moving points. Human gait has single, fundamental driving frequency as a result of the fact that every part of the body must move in cooperation. For example, for every step forward taken with the left foot, the right arm swings forward exactly once.

Below you can see the all scalar measure types which describe the shape of a single gait. Since all the components of the gait, such as movement of individual; body parts, have the same fundamental frequency, all signals derived by summing the movements of these parts.

$$\text{X Coordinate of centroid is } \sum xT / \sum T \quad (4.3)$$

$$\text{Y Coordinate of centroid is } \sum yT / \sum T \quad (4.4)$$

$$\text{X coordinate of difference of centroids is } X_{wc} - X_c \quad (4.5)$$

$$\text{Y coordinate of difference of centroids is } Y_{wc} - Y_c \quad (4.6)$$

As we know the frame rate of the video sequences and also how many frame consists in the video, it is easy to determine the speed of the movement along the time and according to the x and y direction. If there is a fixed motion way ahead in the video sequence, it can be threshold as the speed of that movement. The program for gait analysis works on principle of frequency component of direction of movement. The flow component of the system provides dense measurements of optical flow for a set of points in the image. We use all points and analyze their distribution. To describe the spatial distribution, we compute the centroid of all moving points. The shape of motion is the distribution of flow which characterized by measures of flow. The shape of motion varies systematically during a motion sequence. The features of the centroids characterize the flow at a particular instant in time.

The program converts a movie into individual frames and extracts the person object from each frame. Size, length and frame rate informations have been taken from the video sequence. For every individual frame the program calculates the difference with itself and the previous frame. It then calculates difference between itself and previous as frame skip value. It then calculates the centre of person and centroid labels in each frame, since the centre of person contain both x and y coordinates, and all the movements which are to be detected are due to displacement in horizontal direction, so that's why that component is only considered to find the frequency of movement. The frequency of each movement is calculated by the optical features of the movement. After calculation, all labels have been passed from mean filter. After this operation with weighting operation real label has been identified.

After the labels have been matched with the motion frames of both procedures which are centroid and frame by frame investigation, we give this information as an input to a spatial filter to make results smoother. We used a low pass filter to remove the noise from the results. Here our indication as noise is, the false label matches with the motion frames in the video sequence. For example, there is a walking movement for 10 frames in the video sequence and results give us fourth frame as walking, fifth frame as running and the sixth frame as walking again. Here we are looking for both fourth and sixth frames. According to their results, we decide that the movement is walking and we never show the running label on the results.

Lastly for a result label we use very simple artificial neural network implementation and make a voting between the results of the centroid and the low pass procedures. The input value is calculated by summing the weighted input values from its links.

In mathematical terms

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (4.7)$$

and

$$y_k = \varphi(u_k + b_k) \quad (4.8)$$

where x_1, x_2, \dots, x_m are the input signals; w_1, w_2, \dots, w_m are the weights of labels. $\varphi(.)$ is the function and y_k is the output signal for the calculated label. Label b shows us the bias and the output for a simple neuron k is u. The use of bias has an affine transformation effect on output u. We put a rectangular area on the tracked human in each frame and set the image which correspondence the correct label. The type of movement is decided using certain threshold values. Program put the label image to the right bottom for every individual frame and for a result we combined all the individual result frames and result video sequence occur. We will give more detailed information about the threshold of the movement in next chapter with the experimental information.

5. EXPERIMENTS

To verify the usefulness of the proposed system, we analyzed it on a set of image sequences containing walking subjects viewed from a static camera, facing towards a static background. In the second stage, we also setup some test sequences ourselves. These video sequences contain different type of movements together and we try to indentify the human motions consequently. In our experiments, we sampled the gait of people using the apparatus seen below. In Figure 5.1, we provide the schematic apparatus of the recording scheme in experimental video sequences.

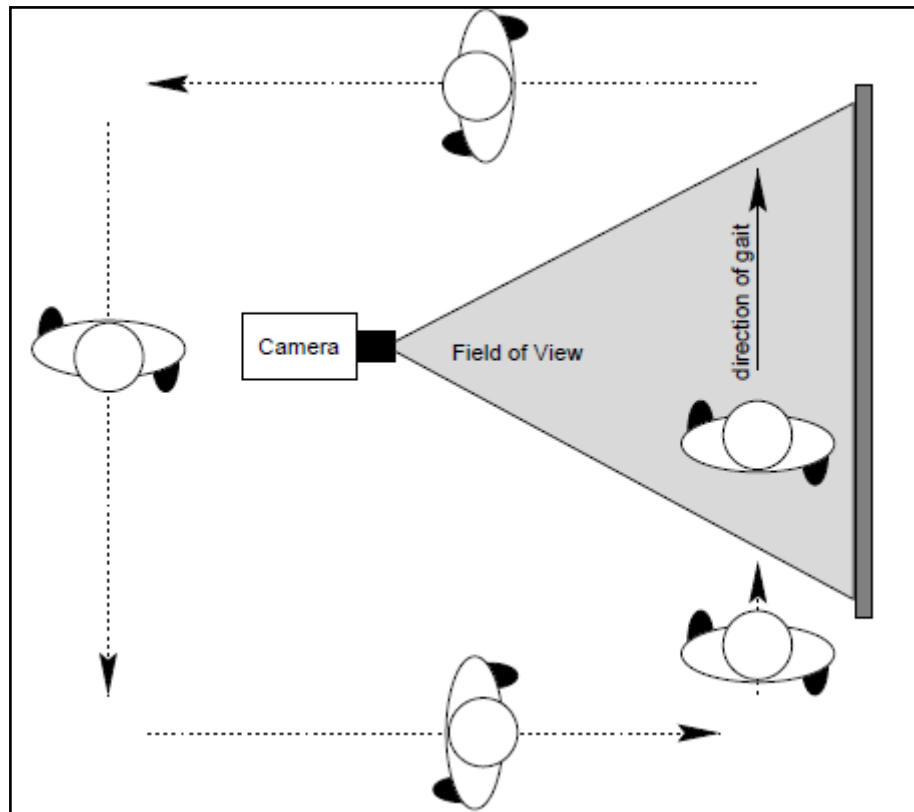


Figure 5.1. Schematic of experimental apparatus.

There are different subjects and each sequence was recorded in 24-bit color, at a resolution of 640x480 pixels. It must be noted that this video was of very good quality and this helped in the segmentation. The type of movement is decided using certain threshold values. Since the thresholds of frequency for each person also varies with the age, e.g. a

child walks faster compared to an old man, so these thresholds can always be changed, depending upon the target person whose gait is to be determined. This difference can be seen especially in videos, if there is a single movement on the video, the frequency of the movement is more stable than compound movements in one video. When we apply the temporal difference between the two adjacent frames, the difference between these frames represents the moving object. If there is a slow variation in few seconds, this shows us the motion of raising hands. If there is a high variation, this represents the running action. If the variation is average, than this shows us the walking action. Finally, no variation in the signal represents the no-movement action.

The experiment results can be broken into two main parts. The video that contains only one singular movement and the video contain different movements consequently. According to the video type, results show little differences. Especially, the difference occurs between the threshold values of the walking and running movement.

Below are some results generated using following videos: (the code runs for movies with initial and last frames must contain a person object, also plain background is preferred, and only one type of movement is detected at a time).

Table 5.1. Speed of movement table of different human motions.

File name	Speed of movement	Gait type
Seco.avi	6.8247	Standing
Man_walking1.mpg	87.7546	Walking
Man_walking2.mpg	87.3928	Walking
Man_walking3.mpg	94.4076	Walking
Man_walking4.mpg	73.9558	Walking
Man_walking5.mpg	72.7261	Walking
Seco2_new.avi	325.4964	Running
1_new.avi	65.9779	Walking
2_stand.avi	6.991	Standing

Besides, as can be seen in the above table, there is a significant change in the speed of movement in the running and standing positions. This difference is because of the exact change in the nature of the video sequence. In this kind of video sequence, all movements need to be determined frame by frame. This causes relative changes in the video sequence. Table 5.2 shows some example values from video and the movement type in the video.

Table 5.2. Some values taken from self-recorded videos.

File name	Speed of Stand	Walk	Run
1	57.82	78.39	134.57
Selcuk_2	53.18	73.64	126.68
Selcuk_run	X	X	132.49
2_stand	62.40	X	X
Melih	X	81.72	X

The basic difference between the video types is the elements of a figure such as moving arms, or legs have much high velocities, if we are interested in only one movement at a time rather than the different movements in one video. Lower variations between motion changes caused low noise differences between features showed in the video sequences. The threshold values arranged according to the motion which has the most significant variation value. Changing these threshold values typically caused some changes in the set of features. The key factor is that, whatever combination of threshold values or parameter is used, that combination must be held constant for recognition to work.

According to the above explanation, it can be detected as parameters and features used in the system can affect the threshold values. It is entirely possible that there are correlations between the motions seen consequently in one video sequence. Below you can see the examples from our experiments for both video type which consists of only one motion in one video sequence and different motion types in one video sequence. Firstly we are interested with the video types only has one motion at a time and the threshold values are adjusted according to this assumption. After this experiment, you can also see some results from the video sequences which have different motion types consequently in

one video sequence. We also show some comparison regarding the results of both video types and try to explain our method more effectively. In Figure 5.2, we can see some frames from result video of a walking man video sequence. You can see that if there is only one motion in the video sequence. Results are stable and correct.

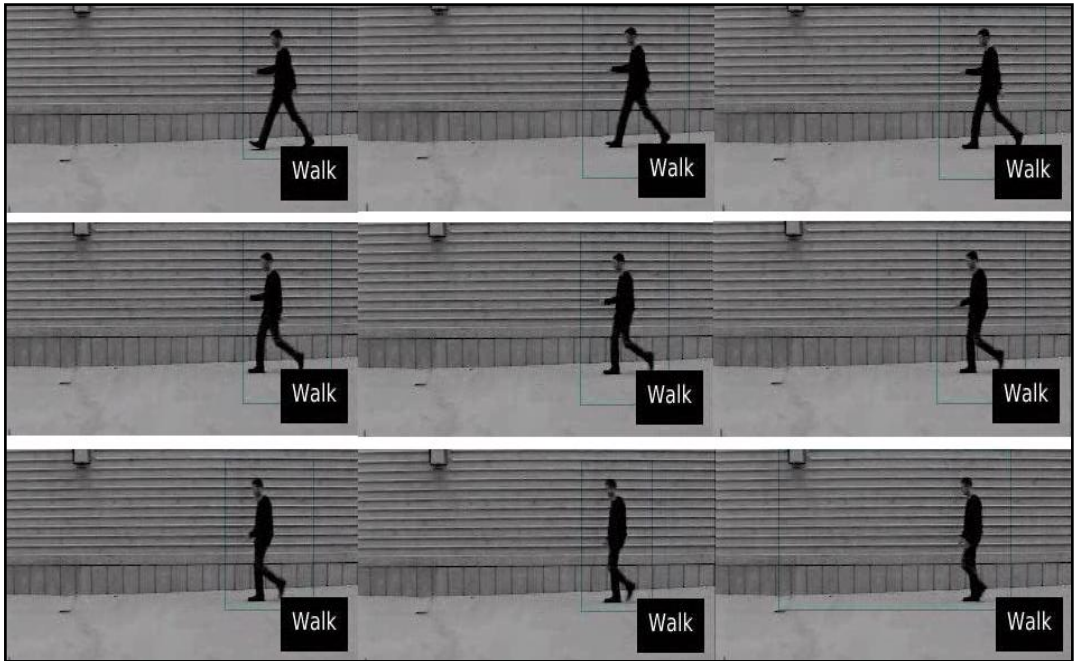


Figure 5.2. Frames of result walking_man1 video

In Figure 5.3, we can see other example frames from result of another walking video sequence.

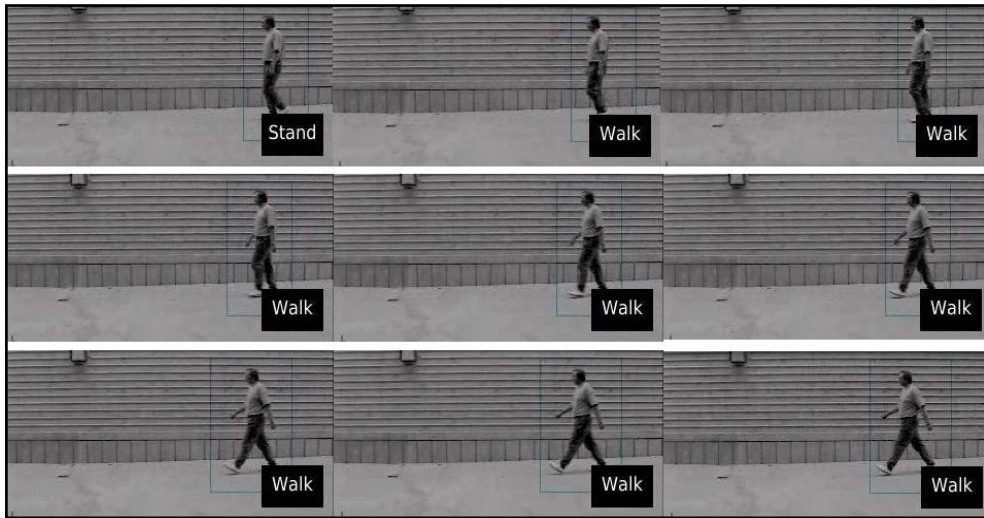


Figure 5.3. Frames of result man_walking2 video

Again for the above results you can see that motion in all frames detected as true instead of the first frame which is detected as STAND instead of WALK. The reason for this is in the first frame the human motion seems as STAND instead of walk and from the frame perspective because the speed of movement here is so slow and close to the STAND, motion detected as STAND. In Figure 5.4, we can see some result frames from Melih video. In this video frames you can see STAND and WALK results.



Figure 5.4. Frames of result Melih video

You can see the same problem occurred in this video on second and third frames as explained above. Instead of figure starts to walk, on the frame it seems as figure is standing and the low speed of movement program system thinks it is STAND position instead of walking. In Figure 5.5 you can see some result frames from Selcuk video. In this figure results are related to STAND motion.

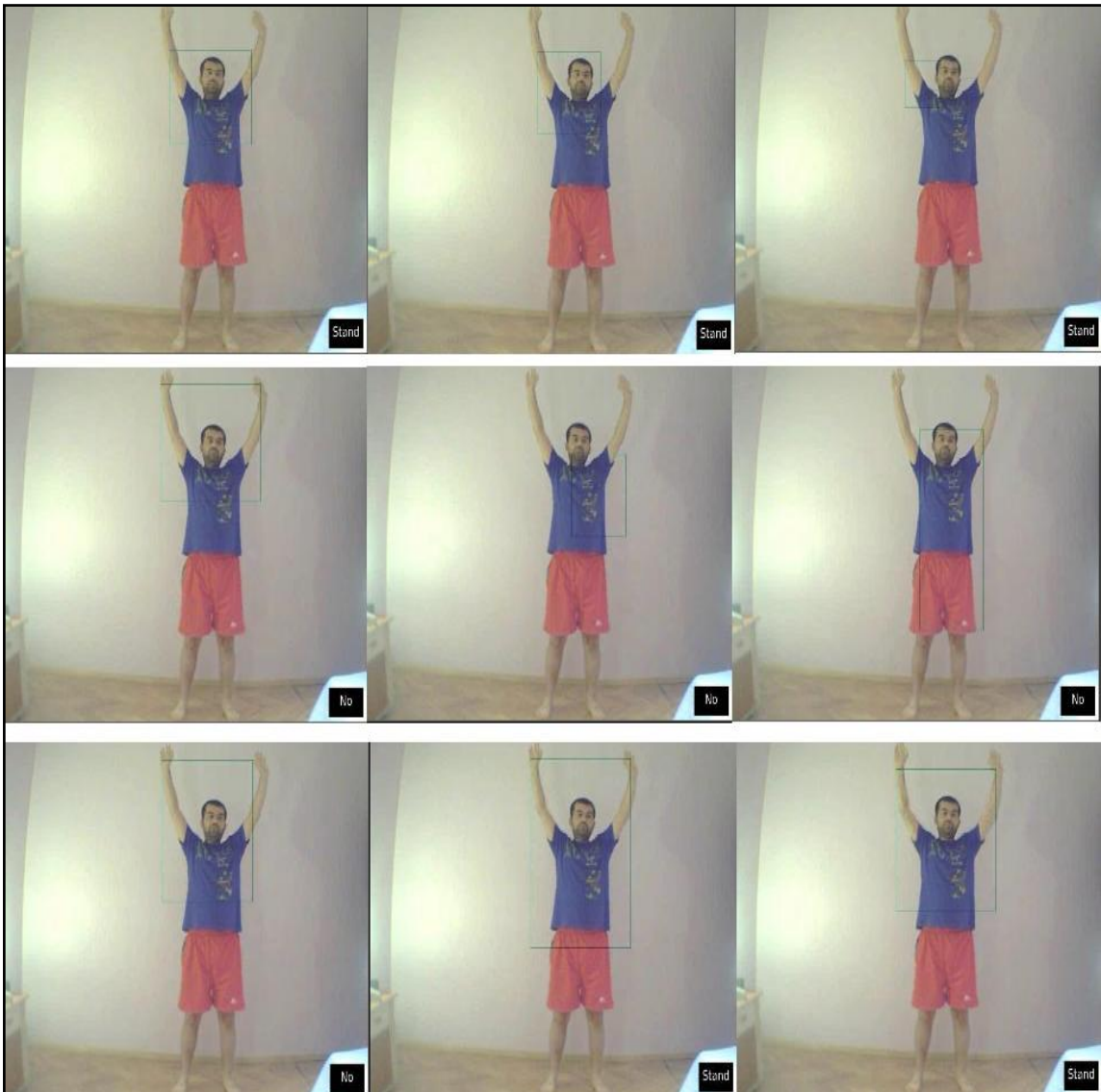


Figure 5.5. Frames of result Selcuk video

In the above frames, you can see three different results. For the first three frames, our system recognizes the human motion as the **STAND** action. However, for the next three frames, it shows **No-movement**. This happens because Selcuk did not make any movement. So, the system thought there is not any movement on the frame. Also there is not any speed info coming from the frame and system printed **NO** sign at left bottom. At this point we can say the last three frames are like evidence to the previous three frames. At the final three frames, the first frame is still shows **NO** movement however in the second frame, there is a little movement which can be identified from the rectangular on the frame. With this little movement, the program again shows us the **STAND** action. In

Figure 5.6, we can see different result frames of different motion types from Melih_2 video.

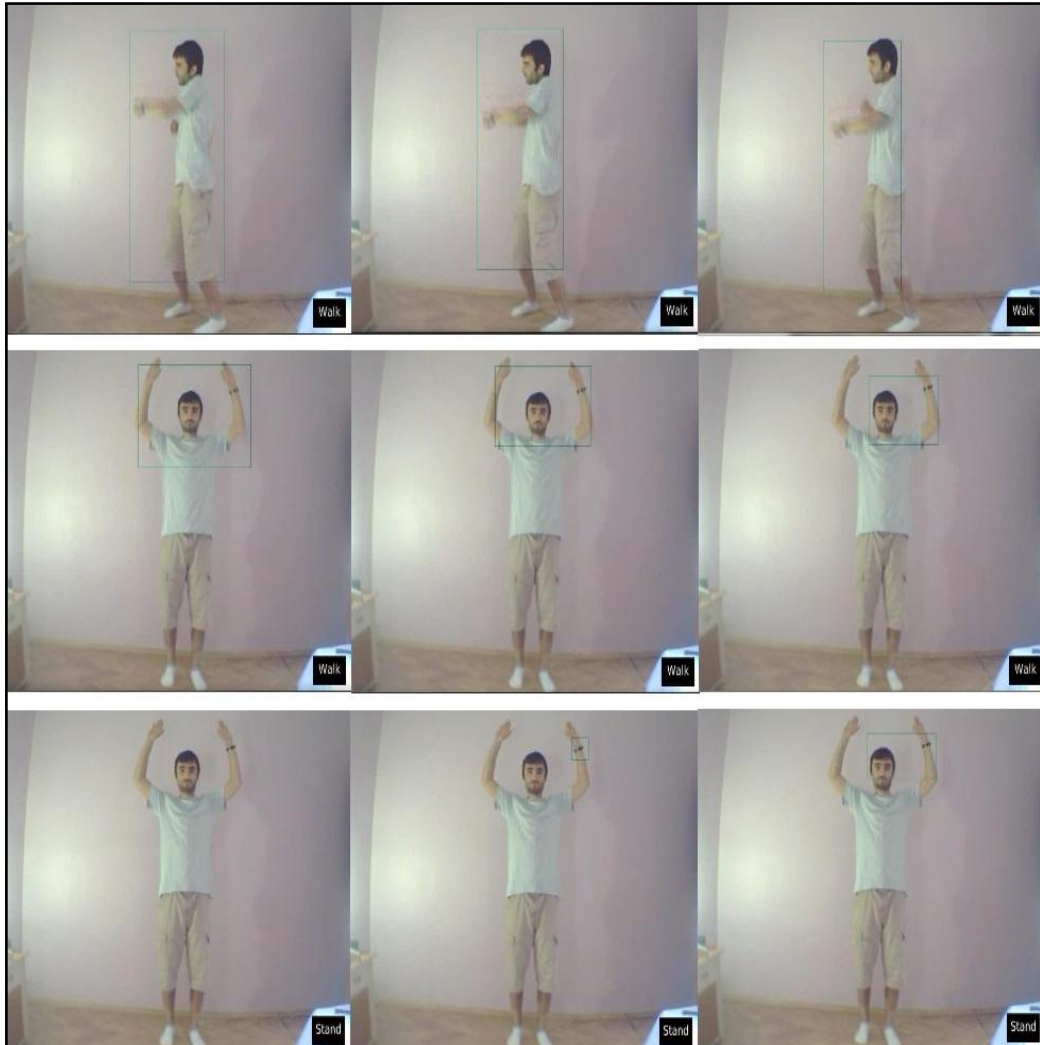


Figure 5.6. Frames of result Melih_2 video

In the above frames, different results can be seen. For the first three frames, our system recognizes the human motion as the WALK action. For the next three frames, it shows the WALK movement. This might happen because of the first frames after the motion changed. However, again this can be considered as error. For the last three frames, our system recognizes the motion is different and prints the WALK action correctly. In Figure 5.7, you can see result frames of three different motions from the Selim video.



Figure 5.7. Frames of result Selim video

For the above video results according to the frames all the actions have been performed correctly. WALK action for first three frames, STAND action for the remaining frames are indicated. In Figure 5.8, we can see some example frames from Selcuk_run video.



Figure 5.8. Frames of result Selcuk_run video

For the above video results it can be seen that all the frames have been indicated wrongly. System prints WALK action instead of RUN. This has occurred due to mismatched speed information because of the exact change in the nature of the video sequence. If there is a single movement in the video and the program takes the first and last frames instead of calculating the motions frame by frame, then the speed values used for thresholding the motion changes.

In Figure 5.9, you can see the same frames results with different threshold values for running motion. In this new threshold value, we only consider the frame by frame changes. On this new setup, the speed of run motion decreases dramatically. This is as expected for the video sequences recorded by ourselves. Also all motions in a video sequence have been identified perfectly. However this time we have some problematic results with the video sequences which have been recorded for general usage databases.



Figure 5.9. Frames of result Selcuk_run video with new values

In Figure 5.10 and 5.11, we can see another result frames with new threshold values for different motion types.



Figure 5.10. Frames of result Selcuk_run2 video with new values

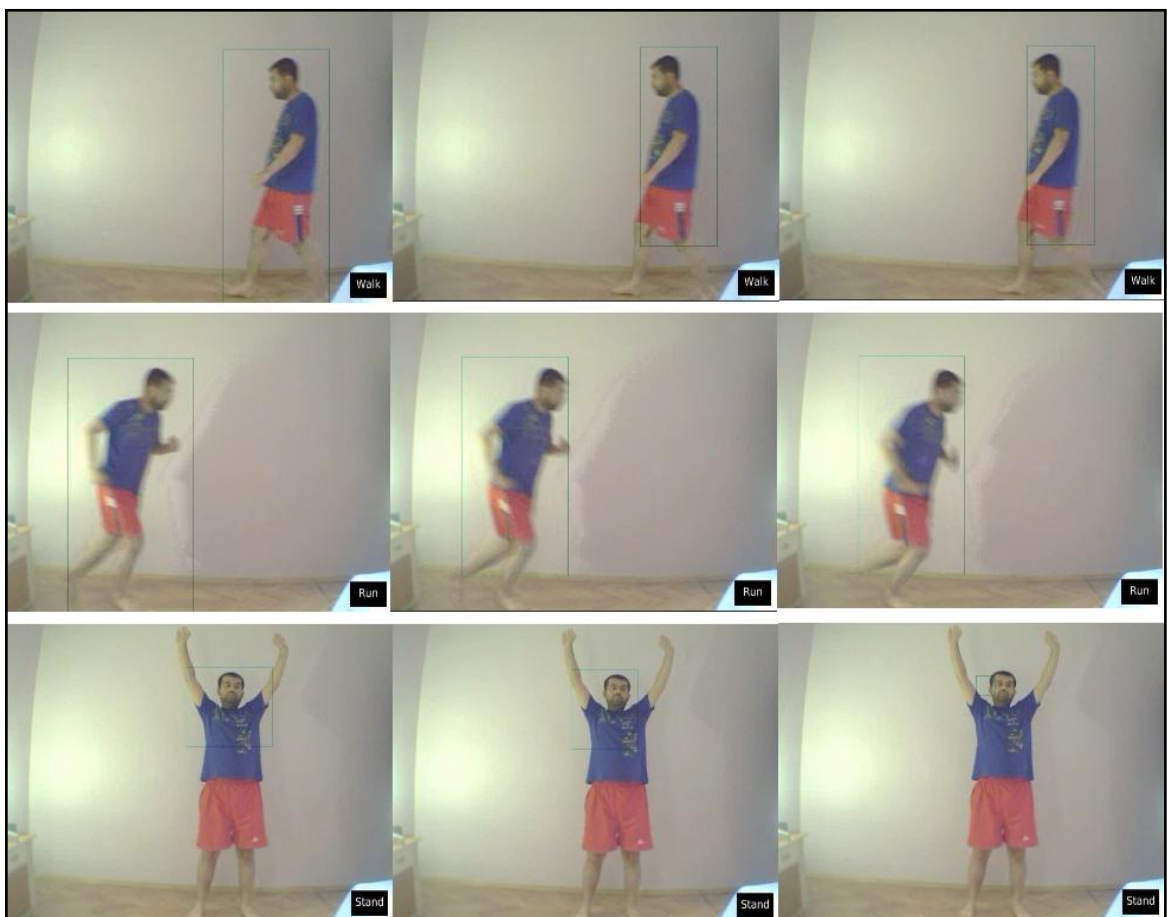


Figure 5.11. Frames of result Selcuk_2 video with new values

For the above video results according to the frames all the actions have been performed rightly. WALK action for the first three frames, and right after that RUN action for the second and third row is indicated as STAND. Of course there are faulty frames occurred between these and this will be indicated in the overall results. Also in Figure 5.12, we can see some result frames from walking video sequence with new threshold values.

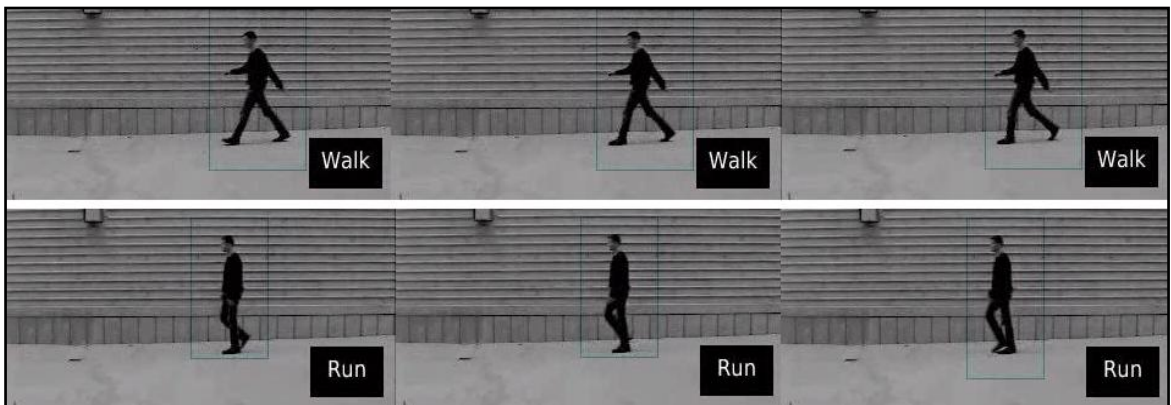


Figure 5.12. Frames of result man_walking1 video with new values

As we have explained above, with new threshold value we have some problematic values on video frames which have been recorded for general usage databases for only RUN action. In this kind of video sequences, this has been occurring because there is only one motion in the video sequence and the speed of movement has been varied with the video type.

Instead of running threshold, we have same stable results for other motion types. Because of this reason, we concluded to put the running threshold for the video sequences which has been recorded for general usage databases. Also we show both result types for this two different threshold values.

Below are the results tables for the first threshold value. Video sequences which have been recorded for general usage databases. At this threshold value, running is thought as high variation and has high speed value.

Table 5.3. Walking_man video results

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Walking_man1	83	6	1	3	73	87.5
Walking_man2	83	3	X	3	76	91.5
Walking_man3	78	6	X	3	67	86.0
Walking_man4	95	2	X	3	90	94.5

Table 5.4. Motion Results for Selcuk_run2 video

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	15	10	X	X	5	67
Run	35	5	5	X	25	14
No	100	10	3	87	X	87
Walk	X	X	X	X	X	X

Table 5.5. Motion Results for Selcuk_2 video

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	65	35	X	10	20	54
Run	60	5	5	X	50	7
No	50	2	X	48	X	96
Walk	90	3	X	X	87	96

Table 5.6. Motion Results for Melih video

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	X	X	X	X	X	X
Run	X	X	X	X	X	X
No	80	3	X	77	X	96
Walk	140	5	X	X	135	96

Table 5.7. Motion Results for Selim video

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	50	30	X	X	20	60
Run	X	X	X	X	X	X
No	90	5	X	85	X	94
Walk	110	5	X	5	100	91

As you can see, instead of running motion we have high success rates in other motion types especially for the WALK. For the STAND motion, it is clear that the performance decreases. The real cause of this result is the speed of WALK and STAND actions are very close to each other.

Below are the results in tabular form for the second threshold value. We have recorded the video sequences. At this threshold value, running is thought as slower variation and has less speed value. From the tables, it is clearly seen that the results on our recorded video sequences are encouraging for the running action. At this point, the results for walking of man videos success decrease as we consider the running action. The variation type difference causes this different result for this video type. Also for Selcuk_run2 and Selim files, the only difference is the success rate of the running motion. Other motion types are not affected in the same kind of video sequences.

Table 5.8. Walking_man video results with new threshold

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Walking_man1	83	3	53	2	25	30
Walking_man2	83	4	50	2	27	33
Walking_man3	78	4	47	2	24	31
Walking_man4	95	2	65	2	29	31

Table 5.9. Motion Results for Selcuk_run2 video with new threshold

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	15	10	X	X	5	67
Run	35	3	28	X	4	80
No	100	10	X	85	5	85
Walk	X	X	X	X	X	X

Table 5.10. Motion Results for Selcuk_run video

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	X	X	X	X	X	X
Run	35	X	25	X	10	72
No	X	X	X	X	X	X
Walk	X	X	X	X	X	X

Table 5.11. Motion Results for Selim video with new threshold

	Total Frame	Stand	Run	No	Walk	Success Rate (%)
Stand	50	30	X	X	20	60
Run	X	X	X	X	X	X
No	90	5	X	85	X	94
Walk	110	5	X	5	100	91

We can show you a comparison of several different approaches on the SOTON database. We give this information because the video sequences in the SOTON database are close to our man_walking video sequences. In Figure 5.13, there are the some sample examples from SOTON gait database.



Figure 5.13. Some samples in the SOTON gait database

Our system results are really encouraging as we worked on a similar database. In Table 5.12, you can see different approaches on the SOTON gait database.

Table 5.12. Comparison of several different approaches on SOTON gait database

Methods	Data Set	Correct Classification Rate (%)
Shutler 2000	4 subjects, 4 seq. per subject	87.50
Hayfron-Acquah 2001	4 subjects, 4 seq. per subject	100.00
Foster 2001	6 subjects, 4 seq. per subject	83.00

In our experiment which is so close to above database worst result is 86% and best result is 96%.

6. CONCLUSIONS

With the increasing demand on visual surveillance systems, human identification at a distance gained more interest. Gait, is a potential behavioral feature for this application. Therefore, many allied studies have demonstrated that it can be used as a useful biometric feature for recognition [4]. The unique advantage of gait is that it can be used at a distance when other biometrics could be obscured or at too low resolution to be perceived. Automatic recognition by gait is now in a position where its properties and potency are comparable with other biometrics.

The development of computer vision techniques have also assured that vision-based automatic gait analysis can be gradually achieved. However, the lack of a common database and evaluation methodology has been an apparent limitation in the development of gait recognition algorithms. As we know, a large number of papers in the literature reported good recognition results on a limited size databases.

The aim of this thesis was to develop a system that is capable of recognizing actions using gait analysis. The combination of a background subtraction procedure and a simple correspondence method is used to segment and track silhouettes of a walking figure. Simple feature selection reduces the computational cost significantly during the training and recognition processes. Our system achieves high recognition rates. Our system is also general and can also be applied on the motion of animals. Although the analysis of our results is valid, we are limited in our ability to extrapolate them. Our sample size is small, so we cannot conclude much about gaits in the entire human population. In an effort to control the conditions of the experiment, we considered only pedestrians walking across the camera field of view. There is no reason to expect that shape of motion features are invariant to viewing angle, so we expect that the shape of motion features we have acquired to fail if the view is not approximately perpendicular to the direction of gait.

As a conclusion, recent developments in gait research indicate that the gait technologies still need to mature and limited practical applications should be expected in immediate future. At present there is a potential for initial deployment of gait for

recognition in conjunction with other biometrics. However gait analysis and recognition is an open and challenging research area for future.

APPENDIX A: RGB to HSI MATLAB CODE IMPLEMENTATION

Algorithm A.1. RGB to HSI Matlab Code Implementation

```

function rgbtohsi(x)
F=imread(x);
F=im2double(F);
r=F(:,:,1);
g=F(:,:,2);
b=F(:,:,3);
th=acos((0.5*((r-g)+(r-b)))/((sqrt((r-g).^2+(r-b).*(g-b)))+eps));
H=th;
H(b>g)=2*pi-H(b>g);
H=H/(2*pi);
S=1-3.*(min(min(r,g),b))./(r+g+b+eps);
I=(r+g+b)/3;
hsi=cat(3,H,S,I);
HE=H*2*pi;
HE=histeq(HE);
HE=HE/(2*pi);
SE=histeq(S);
IE=histeq(I);
choice=input('1:RGB to HSI\n2:Display Hue, Saturation and Intensity
Images\n3:HSI to RGB\n4:Hue-Equalization\n5:Saturation-
Equalization\n6:Intensity-Equalization\n7:HSI-Equalization\n Enter your
choice :');
switch choice
case 1
figure,imshow(F),title('RGB Image');
figure, imshow(hsi),title('HSI Image');

```

case 2

```
figure,imshow(F),title('RGB Image');  
figure, imshow(H),title('Hue Image');  
figure, imshow(S),title('Saturation Image');  
figure, imshow(I),title('Intensity Image');
```

case 3

```
C=hsitorgb(hsi);  
figure,imshow(hsi),title('HSI Image');  
figure, imshow(C),title('RGB Image');
```

case 4

```
RV=cat(3,HE,S,I);  
C=hsitorgb(RV);  
figure,imshow(hsi),title('HSI Image');  
figure,imshow(F),title('RGB Image');  
figure, imshow(C),title('RGB Image-Hue Equalized');
```

case 5

```
RV=cat(3,H,SE,I);  
C=hsitorgb(RV);  
figure,imshow(hsi),title('HSI Image');  
figure,imshow(F),title('RGB Image');  
figure, imshow(C),title('RGB Image-Saturation Equalized');
```

case 6

```
RV=cat(3,H,S,IE);  
C=hsitorgb(RV);  
figure,imshow(hsi),title('HSI Image');  
figure,imshow(F),title('RGB Image');  
figure, imshow(C),title('RGB Image-Intensity Equalized');
```

case 7

```
RV=cat(3,HE,SE,IE);  
C=hsitorgb(RV);  
figure,imshow(hsi),title('HSI Image');
```

```

    figure,imshow(F),title('RGB Image');
    figure, imshow(C),title('RGB Image-HSI Equalized');
    otherwise
        display('Wrong choice');
    end
end
function C=hsitorgb(hsi)
HV=hsi(:,:,1)*2*pi;
SV=hsi(:,:,2);
IV=hsi(:,:,3);
R=zeros(size(HV));
G=zeros(size(HV));
B=zeros(size(HV));
%RG Sector
id=find((0<=HV)& (HV<2*pi/3));
B(id)=IV(id).*(1-SV(id));
R(id)=IV(id).*(1+SV(id).*cos(HV(id))./cos(pi/3-HV(id)));
G(id)=3*IV(id)-(R(id)+B(id));
%BG Sector
id=find((2*pi/3<=HV)& (HV<4*pi/3));
R(id)=IV(id).*(1-SV(id));
G(id)=IV(id).*(1+SV(id).*cos(HV(id)-2*pi/3)./cos(pi-HV(id)));
B(id)=3*IV(id)-(R(id)+G(id));
%BR Sector
id=find((4*pi/3<=HV)& (HV<2*pi));
G(id)=IV(id).*(1-SV(id));
B(id)=IV(id).*(1+SV(id).*cos(HV(id)-4*pi/3)./cos(5*pi/3-HV(id)));
R(id)=3*IV(id)-(G(id)+B(id));
C=cat(3,R,G,B);
C=max(min(C,1),0);
end

```

APPENDIX B : GAIT ANALYSIS MATLAB CODE IMPLEMENTATION

Algorithm B.1. Gait Analysis Matlab Code Implementation

```
function [ label , roi ] = low_pass( diff, thresholds, avg_n )

% Averaging Blocks
block_avg = block_average( diff, avg_n );

% Find Area of Interest
roi = region_of_interest( block_avg );

% Maximum Difference
out_max = max( max( diff ) );

% Determining Label with Thresholds
label = 0;
for t = 1 : length( thresholds )
    if( out_max > thresholds(t) )
        label = t;
    else
        break;
    end
end

end
```

Table B.2. Region of interest function.

```
function roi = region_of_interest( mat )

smat = sparse( mat );
roi = zeros(1,4);

if sum( find(smat) ) > 0

    % indices of non-zero elements
    [ i, j ] = find( smat );

    roi(1) = min( i ); % min of x
    if roi(1) > 1
        roi(1) = roi(1) - 0.5;
    end

    roi(2) = max( i ); % max of x
    if roi(2) < size( mat, 1 )
        roi(2) = roi(2) + 0.5;
    end

    roi(3) = min( j ); % min of y
    if roi(3) > 1
        roi(3) = roi(3) - 0.5;
    end

    roi(4) = max( j ); % max of y
    if roi(4) < size( mat, 2 )
        roi(4) = roi(4) + 0.5;
    end
end
```

```

end
end
end

```

Table B.3. Block_average function

```

function [ block_avg ] = block_average( mat, block_size )

% Removing Unnecessary Pixels
mat = mat(1:size(mat,1) - mod(size(mat,1),block_size),...
1:size(mat,2) - mod(size(mat,2),block_size) );

% Averaging Blocks
[ n1, n2 ] = size( mat );
block_avg = zeros( n1 / block_size, n2 / block_size );
cnti = 0;
for i = 1 : block_size : n1
    cnti = cnti + 1;
    cntj = 0;
    for j = 1 : block_size : n2
        cntj = cntj + 1;
        avgx = sum( sum( mat( i : i + block_size-1, j : j + block_size-1 ) ) ) / (
block_size * block_size );
        block_avg( cnti, cntj ) = avgx;
    end
end
end
end

```

Table B.4. Mean filter function

```
function [ filtered ] = mean_filter( list, K )

L = length( list );
filtered = zeros( 1, L );

for i = 1 : L
    for j = i-K : i+K
        k = j;

        if( k < 1 )
            k = 1;
        else
            if k > L
                k = L;
            end
        end

        filtered(i) = filtered(i) + list(k);

    end
end

filtered = round( filtered / ( 2*K + 1 ) );

end
```

Table B.5. Draw roi function

```

function img_roi = draw_roi( img, roi, block_size )

COLOR = 0;

img_roi = img;
img_roi( block_size * (roi(1)-1) + 1 : block_size * roi(2),...
         block_size * (roi(3)-1) + 1 ) = COLOR;
img_roi( block_size * (roi(1)-1) + 1 : block_size * roi(2),...
         block_size * roi(4)) = COLOR;
img_roi( block_size * (roi(1)-1) + 1,...
         block_size * (roi(3)-1) + 1 : block_size * roi(4)) = COLOR;
img_roi( block_size * roi(2),...
         block_size * (roi(3)-1) + 1 : block_size * roi(4)) = COLOR;

end

```

Table B.6. Create action movies function

```

function [] = create_action_movies( video_file, source, frameRate,
BLOCK_SIZE, lowpass_roi, label )

labels_path = 'labels/';
margin = 10;

frameCount = length( source );

avi_actions = avifile([ video_file '_actions.avi' ] );
avi_actions.fps = frameRate;

```



```
for frame = 1 : frameCount

    frameImage = source( frame ).cdata;

    if( frame > 1)

        % Draw ROI
        if( sum( lowpass_roi( :, frame-1 ) ) > 0 )
            frameImage = draw_roi( frameImage, lowpass_roi( :, frame-1 ),
BLOCK_SIZE );
        end

        % Add Label
        labellImage = imread([ labels_path int2str(label(frame-1)) '.jpg']);
        frameImage( end-size(labellImage,1)+1-margin : end-margin,...
            end-size(labellImage,2)+1-margin : end -margin,...
            :) = labellImage;

    end

    avi_actions = addframe( avi_actions, frameImage );

end

avi_actions = close( avi_actions );

end
```

Table B.7. Gait centroid speed function

```

function [ label, fg_smooth ] = gait_centroid_speed( diff, thresholds, timestep
)

% Centroid Image
fg_smooth = zeros( size(diff) );

% Finding Regions
L = bwlabel( diff > 0 );
stats = regionprops( L, 'all' );
areas = cat( 1, stats.Area );

if( size(areas,1) < 2 )

    % No Significant Movement
    speed_x = 0;

else

    % Finding 2 Largest Regions
    area_index = zeros(1,2);
    [ max_area area_index(1) ] = max( areas );
    areas( area_index(1) ) = 0;
    [ max_area area_index(2) ] = max( areas );

    % Considering only 2 Largest Regions
    for r = 1 : 2;
        pixels = stats( area_index(r) ).PixelList;
        for p = 1 : size( pixels, 1 )
            fg_smooth( pixels(p,2), pixels(p,1) ) = 255;
        end
    end
end

```

```
    end
end

% Determine Region Centroids
region_centroids = zeros( 2, 2 );
for r = 1 : 2
    region_centroids( r , : ) = stats( area_index(r) ).Centroid;
end

% Calculating Distace
distance_x = abs( region_centroids(1,1) - region_centroids(2,1));

% Calculating Speed
speed_x = distance_x / timestep;

end

% Determining Label with Thresholds
label = 0;
for t = 1 : length( thresholds )
    if( speed_x > thresholds(t) )
        label = t;
    else
        break;
    end
end

end
```

Table B.8. Action recognizer function

```
clear;

% Threshold for Denoising Difference Image
DIFFERENCE_THRESHOLD = 30;

% Frame Skipping for Centroid
FRAME_SKIP = 2;

% Block Size for Low Pass
BLOCK_SIZE = 16;

% Mean Filter Neighbourhood
K = 2;

% Voting Weights
WEIGHT_LOWPASS = 0.6;
WEIGHT_CENTROID = 1 - WEIGHT_LOWPASS;

% Thresholds
% Nothing | Standing | Walking | Running
thresholds_centroid = [ 55, 95, 130 ];
thresholds_lowpass = [ 55, 90, 125 ];

% Read Video File
video_path = 'videos/';
video_file = 'man_walking5';
source = aviread( [video_path video_file '.avi'] );

% Video Properties
```

```

imageSize = size( source( 1 ).cdata(:, :, 1) );
frameCount = length( source );
frameRate = get( mmreader( [ video_path video_file '.avi' ] ), 'FrameRate' );

% Labels
label_centroid = zeros( 1, frameCount - 1 );
label_lowpass = zeros( 1, frameCount - 1 );

% centroid_image = zeros( [ imageSize, frameCount - 1 ] );
lowpass_roi = zeros( 4, frameCount - 1 );

for frame = 1 : frameCount - 1

    % Current Image
    current = rgb2gray( source( frame+1 ).cdata );

    % Previous Image
    previous = rgb2gray( source( frame ).cdata );

    % Skipped Image for Centroid
    if( frame - FRAME_SKIP >= 1 )
        skipped = rgb2gray( source( frame+1 - FRAME_SKIP ).cdata );
    else
        skipped = rgb2gray( source( 1 ).cdata );
    end

    % Temporal Difference Between Two Frames ( Low Pass )
    diff_lowpass = ( double ( abs( current - previous ) ) );
    diff_lowpass = ( diff_lowpass > DIFFERENCE_THRESHOLD ) .* diff_lowpass;

    % Temporal Difference Between Two Frames ( Centroid )

```

```
diff_centroid = ( double ( abs( current - skipped ) ) ) >
DIFFERENCE_THRESHOLD;

% Gait Centroid Speed
[ label_centroid( frame ), centroid_image ]= ...
    gait_centroid_speed( diff_centroid, thresholds_centroid, FRAME_SKIP /
frameRate );

% Low Pass Filter
[ label_lowpass( frame ), lowpass_roi( :, frame ) ]= ...
    low_pass( diff_lowpass, thresholds_lowpass, BLOCK_SIZE );

end

% Mean Filtering Recognitions
label_lowpass_filtered = mean_filter( label_lowpass, K );
label_centroid_filtered = mean_filter( label_centroid, K );

% Voting Between Two Methods
label = round( WEIGHT_LOWPASS * label_lowpass_filtered +
WEIGHT_CENTROID * label_centroid_filtered );

% Creating Movie File
result_path = 'results/';
create_action_movies( [ result_path video_file ], source, frameRate,
BLOCK_SIZE, lowpass_roi, label );
```

REFERENCES

1. Jain, A., R. Bolle and S.Pankanti, *Biometrics : Personal Identification in Networked Society*, 1999.
2. Zhou, H., Y. Yuan and C. Shi, “Computer Vision and Image Understanding”, 2008.
3. Nixon, M. S. and J. Carter, “Automatic Recognition by Gait”, *IEEE Proceedings*, 2006.
4. Wang, L., T. Tan, H. Ning and W. Hu, “Silhouette Analysis-Based Gait Recognition for Human Identification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.
5. Defense Advance Research Agency Project, <http://www.darpa.mil/iao/HID.htm> , 2000.
6. Boulgouris, N. V., D. Hatzinakos and K. N. Platanoitis, “Gait Recognition : A Challenging Signal Processing Technology for Biometric Identification”, *IEEE Signal Processing Magazine*, 2005.
7. Jain, A.K., L. Hong, S. Pankanti, and R. Bolle, “An identity verification system using fingerprints”, *IEEE Proceedings*, Vol. 85, 1999.
8. Jain, A.K. and N. Duta, “Deformable matching of hand shapes for verification”, *IEEE Proceedings Conf. Image Processing, Japan*, 1999.
9. Cox, I. and S. Hingorani, “An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking”, *IEEE Trans. Pattern Anal. Mach. Intelligence*, 1996.
10. Comaniciu, D., V. Ramesh, P. Meer, *Real-time tracking of non-rigid objects using mean shift*, 2000.

11. Wren, C., A. Azarbayejani, T. Darrell and A. Pentland, "Pfinder: Real-time tracking of the human body", *IEEE Transactions on Pattern Analysis and machine Intelligence*, 1997.
12. Gavrilu, D. M., "The Visual Analysis of Human Movement: A Survey." , *Proceeding of Computer Vision and Image Understanding*, 1999.
13. Johansson, G., "Visual perception of biological motion and a model for its analysis", *Percept. Pscyphysics*, 1973.
14. Cutting, J.E. and L.T. Kozlowski, *Recognizing friends by their walk: Gait perception without familiarity cues*, 1977.
15. Cunado, D., M. S. Nixon and J. N. Carter, "Using gait as a biometric, via phase-weighted magnitude spectra", *Audio and Video based biometric person Authentication*, 1997.
16. Fashing, M., C. Tomasi, "Mean shift is a bound optimization", *IEEE Trans. Pattern Anal. Mach. Intelligence*, 2005.
17. Koller, D., J. Weber, T. Huang, G. Ogasawara, B. Rao, and S. Russell, *Towards Robust Automatic Traffic Scene Analysis in Real-time*, 1994.
18. Picardi, M., "Background subtraction techniques : a review", *IEEE International Conference on Systems*, 2004.
19. Lo, B. P. L. and S. A. Velastin, *Automatic congestion detection system for underground platforms*, 2001.
20. Cucchiara, R., C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts, and shadows in video streams", *IEEE Trans on Pattern Anal. and Machine*, 2003.

21. Huston, S.J., HG Krapp, *Visuomotor Transformation in the Fly Gaze Stabilization System*, 2008.
22. Stauffer, C. and W.E.L. Grimson, *Adaptive background mixture models for real-time tracking*, 1999.
23. Oliver, N.M., B. Rosario, and A.P. Pentland, "A Bayesian computer vision system for modeling human interactions", *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 2000.
24. Stauffer, C. and W.E.L. Grimson, "Learning Patterns of Activity using Real-Time Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.
25. Rymel, J., J. Renno, D. Greenhill, G. A. Jones, "Adaptive eigen-backgrounds for object detection" , *Image processing International Conference*, 2004.
26. Bobick, A. F. and A. Y. Johnson, "Gait recognition using static, activity-specific parameters", *IEEE Proceeding Computer Vision and Pattern Recognition*, 2001.
27. Reid, D., "An algorithm for tracking multiple targets", *IEEE Trans. Auto. Control*, 1979.
28. Blackman, S. S., "Multiple Hypothesis Tracking For Multiple Target Tracking", *IEEE A&E Systems Magazine*, 2004.
29. Sundaresan, A., A. Roy-Chowdhury, and R. Chellappa, "Hidden Markov model based framework for recognition of humans from gait sequences", *IEEE Int. Conf. Image Processing*, 2003.
30. Kale, A., A. N. Rajagopalan, A. Sundaresan, N. Cuntoor, A. Roy-Chowdhury, V. Kruger, and R. Chellappa, "Identification of humans using gait", *IEEE Trans. Image Process.*, 2004.

31. Rabiner, L., "A tutorial on Hidden Markov Models and selected applications in speech recognition", *IEEE Proceeding*, 1989.
32. Yang, C., R. Duraiswami, L. Davis, Efficient "Mean-shift tracking via a new similarity measure", *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, 2005.
33. Fukunaga, K. and L. D. Hostetler. "The estimation of the gradient of a density function, with applications in pattern recognition", *IEEE Trans. Inform. Theory*, 1975.
34. Comaniciu, D., V. Ramesh and P. Meer, "Kernel-based object tracking", *IEEE Trans. Pattern Anal. Mach. Intelligence*, 2003.
35. Little, J. J. and J. E. Boyd, "Recognizing people by their gait: the shape of motion", *MIT Press Journal Videre*, 1996.
36. BenAbdelkader, C., R. Culter, H. Nanda, and L. Davis, "EigenGait: motion-based recognition of people using image self-similarity", *Proc. Int. Conf. Audio- and Video-Based Person Authentication*, 2001.
37. Wang, L., T. Tan, H. Ning, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition", *IEEE Trans. Circuits Syst. Video Technol.*, 2004.
38. Lowe, D., *Robust model-based motion tracking through the integration of search and estimation*, 1992.
39. Wunsch, P., G. Hirzinger, "Real-time visual tracking of 3D objects with dynamic handling of occlusion", *Proceedings of 97 International Conference on Robotics and Automation*, 1997.
40. Gavrilu, D., L. Davis, "3D model-based tracking of humans in action: a multiview approach", *Proceedings of the Computer Vision and Pattern Recognition*, 1996.

41. Smith, M. S., *Finding optic flow* , Department of Clinical Neurology, Oxford University, <http://users.fmrib.ox.ac.uk/~steve/review/review/node1.html>
42. Lucas, B., T. Kanade, “An iterative image registration technique with an application to stereo vision”, *International Joint Conference on Artificial Intelligence*, 1981.
43. Schunck, B., *The image flow constraint equation*, *Comput. Vis. Graph. Image Process.*, 1986.
44. Shi, J., C. Tomasi, “Good features to track”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.
45. Beauchemin, S. S. , J. L. Barron, *The computation of optical flow*, 1995.
46. Catalano, G., S. Pedro, *Optical Flow*, 2009.
<http://www.cvmt.dk/education/teaching/f09/VGIS8/AIP/opticalFlow.pdf>
47. Gonzales, R. C., R. E. Woods, *Digital Image Processing*, Prentice-Hall , 2002.
48. RGB versus CMY Color Imagery, <http://astrosurf.com/buil/us/cmy/cmy.htm>
49. Cootes, T., T. Taylor, C. Cooper and D. Graham, “Active Shape Models – Their Training and Application”, *Computer Vision and Image Understanding*, 1995.
50. Bergstedt ,G. A., "Apparatus and method for modifying displayed color images," Assigned to Tektronix, Inc, 1983.
51. Jones, M. J., *Statistical color models with application to skin detection*, 1999.
52. Lambert, P. and T. Carron, “Symbolic fusion of luminance-hue-chroma features for region segmentation.”, *Pattern Recognition*, 1999.

53. Cheng, Y., "Mean shift, mode seeking, and clustering", *IEEE Trans. Pattern Anal. Mach. Intelligence*, 1995.
54. Matheron, G. and R. W. Schafer, *Random sets and Integral Geometry*, Wiley, New York, 1975.
55. Serra, J., *Image Analysis and Mathematical Morphology*, Academy Press, 1982.
56. Darrell, T. and A. Pentland, "Space-time gestures," *IEEE Conference on Computer Vision and Pattern Recognition*, 1993.
57. Downton, A. and H. Drouet, "Model-based image analysis for unconstrained human upper-body motion", *IEEE International Conference on Image Processing and Its Applications*, 1992.
58. Ross, D., J. Lim and M. Yang, "Adaptive probabilistic visual tracking with incremental subspace update", *Proceeding of the Eighth European Conference on Computer Vision*, 2004.
59. Jepson, A., D. Fleet and T. El Maraghi, "Robust online appearance models for visual tracking", *IEEE Trans. Pattern Anal. Mach. Intelligence*, 2003.
60. Humphreys, G. W. and V. Bruce, *Visual Cognition*. Psychology Press., 1989.
61. Lucas, B., T. Kanade, "An iterative image registration technique with an application to stereo vision", *International Joint Conference on Artificial Intelligence*, 1981.
62. Bovik, A. C., *Morphological Filtering For Image Enhancement and Feature Detection*, Elsevier Academic Press, 2005.
63. Wren, C., A. Azarhayejani, T. Darrell, and A.P. Pentland, "Pfinder: real-time tracking of the human body", *IEEE Trans. on Pattern Anal. and Machine Intell.*, 1997.

64. Turk, M. and A. Pentland, "Face recognition using eigenfaces", *IEEE Proceedings Conf. Computer Vision and Pattern Recognition*, 1991.
65. Petrou, M., C. Petrou, *Image Processing Fundamentals*, Wiley, 2010.
66. Rosenfeld, A. and A. C. Kak, *Digital Picture Processing*, Academy Press., 1982.
67. Yambor, W.S., *Analysis of PCA-based and Fisher Discriminant-based Image Recognition Algorithms*, Colorado State University, 2000.
68. Power, P. W. and J. A. Schoonees, *Understanding background mixture models for foreground segmentation*, 2002.
69. Menser, B., "Segmentation and tracking of facial regions in color image sequences", *SPOE Visual Communications and Image Processing*, 2000.
70. Rabiner, L. R., "A tutorial on Hidden Markov Models and Selected Applications in speech Recognition" , *Proceeding of the IEEE*, 1989.
<http://www.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf>
71. Hall, P.M., D. Marshall, and R.R. Martin, "Incremental Eigenanalysis for Classification", *Proceedings of the British Machine Vision Conference*, 1998.
72. Bobick, A.F., "Real time recognition of activity using temporal templates", *In workshop on application of Computer vision*, 1996.
73. Phillips, J., H. Moon, S. Rizvi, and P. Raue, "The FERET evaluation methodology for face recognition algorithms", *IEEE Trans. Pattern Anal. Machine Intelligence*, 2000.
74. Yam, C., M. Nixon, and J. Carter, "On the relationship of human walking and running: automatic person identification by gait", *Proc. Int. Conf. Pattern Recognition*, 2002.

75. Phillips, P., S. Sarkar, I. Robledo, P. Grother, and K. Bowyer, "The gait identification challenge problem: Data sets and baseline algorithm", *In Proc. Int. Conf. Pattern Recognition*, 2002.
76. Lee, L. and W. Grimson, "Gait analysis for recognition and classification", *Proc. Int. Conf. Automatic Face and Gesture Recognition*, 2002.
77. Brunelli, R. and D. Falavigna, "Person identification using multiple cues", *IEEE Trans. Pattern Anal. Machine Intelligence*, 1995.
78. Hong, L. and A. Jain, "Integrating faces and fingerprints for personal identification", *IEEE Trans. Pattern Anal. Machine Intelligence*, 1998.
79. Kittler, J., M. Hatef, R. Duin, and J. Matas, "On combining classifiers", *IEEE Trans. Pattern Anal. Machine Intelligence*, 1998.
80. Isard, M. and A. Blake, "A mixed-state condensation tracker with automatic modal switching", *Proc. Int. Conf. Computer Vision*, 1998.
81. Baumberg, A. and D. Hogg, "Generating spatiotemporal models from examples", *Proc. Brit. Machine Vision Conf.*, 1995.
82. Niyogi, S. A. and E. H. Adelson, "Analyzing and recognizing walking figures in XYT", *IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 1994.
83. Little, J. and J. Boyd, *Recognizing people by their gait: the shape of motion*, 1998.