IMITATION OF HUMAN BODY POSES AND HAND GESTURES USING A PARTICLE BASED FLUIDICS METHOD

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

BY

UMUT TİLKİ

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

OCTOBER 2012

Approval of the thesis:

IMITATION OF HUMAN BODY POSES AND HAND GESTURES USING A PARTICLE BASED FLUIDICS METHOD

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

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. İsmet Erkmen
Head of Department, **Electrical and Electronics Engineering** ________________

Prof. Dr. İsmet Erkmen
Supervisor, **Electrical and Electronics Eng. Dept., METU** ________________

Prof. Dr. Aydan M. Erkmen
Co-supervisor, **Electrical and Electronics Eng. Dept., METU** ________________

Examining Committee Members:

Prof. Dr. Erol Kocaoğlan
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. İsmet Erkmen
Electrical and Electronics Engineering Dept., METU ________________

Prof. Dr. Veysel Gazi
Electrical and Electronics Engineering Dept., İKBU ________________

Assoc. Prof. Dr. Duygun Erol Barkana
Electrical and Electronics Engineering Dept., Yeditepe University ________________

Assist. Prof. Dr. Yiğit Yazıcıoğlu
Mechanical Engineering Dept., METU ________________

Date: ________________

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

Name, Last Name: UMUT TİLKİ

Signature :

ABSTRACT

IMITATION OF HUMAN BODY POSES AND HAND GESTURES USING A PARTICLE BASED FLUIDICS METHOD

Tilki, Umut
Ph.D., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. İsmet Erkmen
Co-Supervisor : Prof. Dr. Aydan M. Erkmen

October 2012, 120 pages

In this thesis, a new approach is developed, avoiding the correspondence problem caused by the difference in embodiment between imitator and demonstrator in imitation learning. In our work, the imitator is a fluidic system of dynamics totally different than the imitatee, which is a human performing hand gestures and human body postures. The fluidic system is composed of fluid particles, which are used for the discretization of the problem domain. In this work, we demonstrate the fluidics formation control so as to imitate by observation initially given human body poses and hand gestures. Our fluidic formation control is based on setting suitable parameters of Smoothed Particle Hydrodynamics (SPH), which is a particle based Lagrangian method, according to imitation learning. In the controller part, we developed three approaches: In the first one, we used Artificial Neural Networks (ANN) for training of the input-output pairs on the fluidic imitation system. We extracted shape based feature vectors for human hand gestures as inputs of the system and for output we took the fluid dynamics parameters. In the second approach, we employed the Principal Component Analysis (PCA) method for human hand gesture and human body pose classification and imitation. Lastly, we developed a region based controller which assigns the fluid parameters according to the human body poses and hand gestures. In this controller, our algorithm determines the best fitting ellipses on human body regions and human hand finger positions and maps ellipse parameters to the fluid parameters.

The fluid parameters adjusted by the fluidics imitation controller are body force (f), density, stiffness coefficient and velocity of particles (V) so as to lead formations of fluidic swarms to human body poses and hand gestures.

Keywords: imitation learning, correspondance problem, learning by observation, fluidics controller, smoothed particle hydrodynamics

ÖZ

İNSAN VÜCUT DURUŞLARININ VE EL İŞARETLERİNİN PARÇACIK TABANLI AKIŞKANLAR DİNAMİĞİ METODU KULLANILARAK TAKLİT EDİLMESİ

Tilki, Umut
Doktora, Elektrik Elektronik Mühendisliğ Bölümü
Tez Yöneticisi : Prof. Dr. İsmet Erkmen
Ortak Tez Yöneticisi : Prof. Dr. Aydan M. Erkmen

Ekim 2012, 120 sayfa

Bu tezde, taklit ederek öğrenme metodundaki taklit eden ve taklit edilen arasındaki şekillenme farkından dolayı ortaya çıkan uyuşma problemine karşı yeni bir yaklaşım geliştirilmiştir. Çalışmamızda, insan vücut duruşları, insan el hareketleri ve işaretleri taklit edilen olarak alınmış ve taklit eden ise dinamiği taklit edilenden tamamen farklı olan akışkan bir sistemdir. Akışkanlar dinamiği tabanlı sistem problem alanını ayrıştırmada kullanılan akışkan parçacıklardan oluşturulmuştur. Bu çalışmada, insan vücut duruşları ve insan el hareketleri özelliklerinin gözlenmesi ile taklit edilmesini saglayacak akışkanlar dinamiği tabanlı biçimlendirme kontrolü gerçekleştirilmiştir. Bu çalışmada kullanılan biçimledirme kontrolörü Lagrangian, parçacık tabanlı bir metod olan Yumuşatılmış Parçacık Hidrodinamiği (YPH) parametrelerinin taklit ederek öğrenmeye göre uygun değerlere ayarlanmasına dayanır. Kontrolör kısmında üç farklı yaklaşım geliştirildi: Birincisinde akışkan tabanlı taklit ederek öğrenme sisteminde giriş çıkış çiftlerinin eğitilmesinde Yapay Sinir Ağları (YSA) kullanıldı. Sistemin girişi olarak insan el işaretlerine ait özellik vektörlerini çıkarıldı ve buna karşılık çıkış olarak akışkanlar dinamiğine ait parametreler akışkan parçacıklarından oluşan koloniye uygulandı.İkinci yaklaşımda, insan vücut duruşlarının ve insan el işaretlerinin sınıflandırılmasında ve taklit edilmesinde Temel Bileşenler Analizi (TBA) kullanıldı. Son olarak, insan vücut duruşlarına ve el işaretlerine göre akışkan parametrelerini ayarlayan bölgesel bir kontrolör geliştirildi. Bu kontrolör için geliştirilen algoritma insan vücut ve el parmak bölgelerine en iyi oturan elipsi saptayıp, elips parametrelerini akışkan parametrelerine dönüştürmektedir.

Akışkan tabanlı taklit etme kontrolörü tarafından akışkan sürüye insan vücut duruşlarını ve el işaretlerini biçimi verecek şekilde akışkanlar dinamiğine ait uygun değerlere ayarlanan parametreler gövde kuvveti (f), yoğunluk, sertlik katsayıları ve parçacık hızlarıdır.

Anahtar Kelimeler: taklit ederek öğrenme, uyumluluk problemi, gözlemliyerek öğrenme, akışkan tabanlı kontrollör, yumuşatılmış parçacık dinamiği

To my family

ACKNOWLEDGMENTS

First of all I would like to express my thanks to my thesis supervisor Prof. Dr. İsmet Erkmen and my co-supervisor Prof. Dr. Aydan M. Erkmen for their support, invaluable advices, encouragement for studying highly challenging topic. Their motivation and belief on the topic were the biggest encouragement for myself.

I would also thank my thesis progress committee members, Prof. Dr. Erol Kocaoğlan and Prof. Dr. Veysel Gazi for attending all my progress presentations although Prof. Kocaoğlan was in Northern Cyprus and Prof. Gazi in İstanbul. Their invaluable suggestions and encouragements improved the quality of the thesis. Also I would like thank the other jury members Assist. Prof. Dr. Yiğit Yazıcıoğlu and Assoc. Prof. Dr. Duygun Erol Barkana for sparing time.

During my Ph.D education, I worked closely with other ÖYP assistant in the electrical and electronics engineering department, Dr. Evren Ekmekçi, Dr. Tansu Filik, Dr. A. Hayrettin Yüzer, Dr. Oğuzhan Erdem, Dr. Feza Carlak, Yılmaz Kalkan, Atilla Dönük, ,Turgay Koç. I want to thank all of them for their close friendship and motivations.

This thesis work is studied in Mechatronics, Robotics, and Control (MRC) Laboratory. I would like to thank all the members of the laboratory, Sedat Doğru, Akif Durdu, Dr. Sebahattin Topal and Barış Özyer for sharing most of the time. We had very long conservations and discussions about the thesis, and shared most of time in the laboratory and also outside of the laboratory.

My deepest gratitude goes to my parents for bringing me up and for giving support during all my life. My parents, Nesrin and Necati Tilki and my sister Cansu Tilki deserve infinite thanks for being such a peaceful family. It is really good to feel their supports, understanding and love.

Lastly, but not the least, I would like to thank my lovely wife, Serap Ünal Tilki for her love, support, comprehension, and for always being with me. This thesis cannot be completed without her unlimited support and patience. And the new member of our family, my son, Efe Tilki. This life is better now with his presence. I am deeply sorry for hardly having any time with you during the elaboration of this thesis.

TABLE OF CONTENTS

ABSTRACT iv

ÖZ vi

ACKNOWLEDGMENTS ix

TABLE OF CONTENTS xi

LIST OF TABLES xiv

LIST OF FIGURES xv

CHAPTERS

1 INTRODUCTION 1
  1.1 Imitation Learning 1
  1.2 Correspondence Problem 2
  1.3 Objectives and Motivation 3
  1.4 Methodology 4
    1.4.1 Analogy 5
    1.4.2 SPH 5
    1.4.3 A Framework for Control of Fluid Particles 6
    1.4.4 Principal Component Analysis 8
  1.5 Contribution 9
  1.6 Outline of Thesis 10
2 RELATED WORK AND MOTIVATION 11
  2.1 Ongoing Works in Imitation Learning 11
  2.2 Correspondence Problem 12
  2.3 Human/Robot Imitation 14
  2.4 Imitation in Humans 15

2.5 Imitation in Animals 16
2.6 Fluidics in Robotics 18
  2.6.1 Controlling the swarms 18
  2.6.2 Fluidics Approaches 20
3 FLUIDIC FORMATION CONTROL 23
3.1 Control Architecture 23
3.2 Feature vector extraction of human hand postures 24
3.3 Modeling the swarm imitator as a colony of fluid particles . . 27
3.4 Generating the training sets of our controller 34
  3.4.1 Example-1:Horizontal Hand Gesture 36
  3.4.2 Example-2:Pointing at angle hand preshape 37
  3.4.3 Example-3:Separation of fingers training set 37
  3.4.4 Example-4:Pinching of fingers training set 39
3.5 Imitation Results 40
4 HUMAN HAND PRESHAPE CLASSIFICATION WITH PRINCIPAL COMPONENT ANALYSIS FOR IMITATION 44
4.1 Principal Component Analysis 44
4.2 Related Works 46
4.3 Mathematical Background of PCA 48
4.4 Hand Preshape Classification 50
  4.4.1 Preprocessing Module 54
  4.4.2 PCA Module 54
  4.4.3 Test Module 56
  4.4.4 Classification 57
4.5 Statistical Analysis of Principal Components for Classification 58
4.6 Imitation of Hand Gestures with Stiffness Coefficient 60
4.7 Sensitivity Analysis 65
5 HAND GESTURE AND HUMAN BODY POSE IMITATON 71
5.1 Human Hand Gesture Imitation 71
  5.1.1 Example-1 71

5.1.2 Example-2 78

5.2 Human Body Pose Imitation 84

6 IMITATION OF HUMAN BODY POSES AND HAND GESTURES BY REGIONAL CONTROLLER 91

6.1 Introduction 91

6.2 Fluidic Control Layer 91

6.2.1 Control Architecture 91

6.2.2 Module 1: Focus Region Extraction 93

6.2.3 Module 2: Ellipse Fitting to the Regions of Focus . 96

6.2.4 Module 4 Swarm Layer: Modeling The Fluid Particles With SPH 98

6.2.5 Module 3: Mapping Between Ellipse Parameters and SPH Parameters 99

6.3 Simulation Results 101

7 CONCLUSION AND FUTURE WORK 111

7.1 Conclusion 111

7.2 Future Work 112

REFERENCES 114

CURRICULUM VITAE 119

LIST OF TABLES

TABLES

Table 4.1 PCA Recognition Rate 60
Table 4.2 Average Consumed Time for Training and Testing 60

Table 6.1 Elapsed Time According to the Step Size(sec) 109

LIST OF FIGURES

FIGURES

Figure 2.1 Examples of motor imitation of a human by a dolphin 18

Figure 3.1 Fluidic formation controller for swarms 23

Figure 3.2 Feature extraction of scissor movement-like hand behavior 24

Figure 3.3 Feature extraction for pinching and C-shaped hand gestures 26

Figure 3.4 The support domain for particle i and 1D projection of a smoothing function over it. 28

Figure 3.5 Flow chart diagramming the instance of control algorithm, performed for each particle *i*. 33

Figure 3.6 Body force vectors for an attraction point. 35

Figure 3.7 Horizontal Pointing Training Set 36

Figure 3.8 Pointing at angle training set 37

Figure 3.9 Separation of fingers 38

Figure 3.10 Body force distribution for separation 38

Figure 3.11 Separation of fingers with particles 39

Figure 3.12 Pinching hand gesture 40

Figure 3.13 Body force distribution for pinching hand preshape 40

Figure 3.14 Pinching hand preshape with particles 41

Figure 3.15 Imitation: Separation of fingers 41

Figure 3.16 Imitation of separation of fingers 42

Figure 3.17 Pinching hand posture 43

Figure 3.18 Imitation of pinching hand preshape 43

Figure 4.1 Examples of hand preshape classes 52

Figure 4.2 Flowchart of the PCA for classification 53

Figure 4.3 Example of training set 55

Figure 4.4 Cylindrical preshape 56

Figure 4.5 Test image: closed hook 57

Figure 4.6 Euclidean distances for the test image 58

Figure 4.7 Most and Least Similar Images 58

Figure 4.8 Recognition rate for classification 59

Figure 4.9 Initial particle distribution 62

Figure 4.10 Separation of particles with stiffness coefficient 63

Figure 4.11 Curving of two fingers with stiffness coefficient 64

Figure 4.12 Average particle velocity for separation hand gesture 66

Figure 4.13 Average particle velocity for curving hand gesture 67

Figure 4.14 Initial particle position 68

Figure 4.15 Velocity field of the particles 68

Figure 4.16 Physical viscosity field on the particles 69

Figure 4.17 Velocity field for $\beta = 2$ 70

Figure 4.18 Velocity field for $\beta = 50$ 70

Figure 5.1 Hand preshapes combining scissor + hook basic preshapes. 72

Figure 5.2 Separation of particles 73

Figure 5.3 Closing of the finger tips on the object 74

Figure 5.4 Aggregation of particles 75

Figure 5.5 Moving forward of the object 76

Figure 5.6 Moving forward of particles 77

Figure 5.7 First group frames of hand gesture 78

Figure 5.8 Snapshots for separation of particles 80

Figure 5.9 Second phase of the hand gesture 81

Figure 5.10 Snapshots for curvature with horizontal edge with particles 82

Figure 5.11 Body force distribution for line edge hand gesture 83

Figure 5.12 Body force distribution for curvature hand gesture 84

Figure 5.13 Human body pose classes 85

Figure 5.14 Human body tracking system on Simulink 86

Figure 5.15 Examples of detected human body motions 87

Figure 5.16 Training data set for human body poses 88

Figure 5.17 Training data set for human body poses 89

Figure 5.18 Training data set for human body poses 90

Figure 6.1 Flow chart of the proposed approach 92

Figure 6.2 Human body movements for imitation 94

Figure 6.3 Hand gesture movements for imitation 94

Figure 6.4 Left and right arm segmentation for different human poses 95

Figure 6.5 Human body torso and head regions 96

Figure 6.6 Focus region extraction for hand gesture 96

Figure 6.7 Fitted ellipses to the segmented human body regions 98

Figure 6.8 Focus region extraction for hand gesture 99

Figure 6.9 Separation of arms two sides 102

Figure 6.10 Moving left arm down side 103

Figure 6.11 Left arm close to the right arm 103

Figure 6.12 Scissor like hand gesture 105

Figure 6.13 Cylindrical to hook preshape 105

Figure 6.14 Pinching like hand gesture 105

Figure 6.15 Sensitivity analysis of fluid parameters 106

Figure 6.16 Interparticle distances for different hand gestures 107

Figure 6.17 Pressure distributions for different hand gestures 108

Figure 6.18 Particle number sensitivity for hand gestures 110

CHAPTER 1

INTRODUCTION

1.1 Imitation Learning

Imitation learning is one of the ways of social learning mechanism that enables the human or robot agents to learn new skills. With basic words, the definition of imitation is a type of social behavior in which one of the agents (human, animal or robot) copies the form of behavior or action of another agent. Imitation is a learning mechanism for transferring knowledge from the skilled agent (demonstrator) to the unskilled agent (imitator) using direct demonstration methods.

In imitation learning or in other words, programming by demonstration (PbD) there are two primary goals [1]. The first one is studying human beings and exploring how social learning process works and the second one is to see how social learning can work in general and how to build better artifacts using that knowledge. Four fundamental questions are important in imitation: who, what, when, and how to imitate. The important problem during imitation learning, is to answer the "how to imitate" question. The answer basically is obtained by creating an appropriate mapping between the actions afforded by particular embodiments and the matching by the model and imitator (solving the correspondence problem). In this work we focus on the question of how to imitate when the demonstrator and imitator do not the share same embodiment.

The traditional imitation approaches concentrate on finding the correct features of the model and developing a controller that generates the appropriate movements of the model, and the imitator maps those action sequences appropriately. Since the demonstrator and the imitator do not share and perceive the same context, this approach limits itself to answer the question of how to imitate the demonstrator. On the contrary, agent-based approach for imitation concentrates not only on the imitator's behavior but also other agents' behaviors in the environment [2]. The agent based approaches have a broader point of view and include important questions in the implementation of imitation: who to imitate, when to imitate, what to imitate, how to imitate and how to evaluate the imitation.

1.2 Correspondence Problem

Imitation learning, where teacher demonstrates the behaviors for the imitator to follow, has been used to train agents to control complicated systems like human hand imitation by a robot hand. Particularly an action or task is performed by the demonstrator, and the details of this action, usually in the form of observation and action pairs, are passed on to the learner [3]. However, most of the work puts a burden to the teacher, since the learning process depends on the actual demonstration of the teacher [4].

Imitation learning is a powerful and practical way of learning to develop robot behaviors. Even so, development remains a challenge and possible demonstration limitations, for example correspondence issues between the robot (imitator) and demonstrator, can degrade performance. The goal of imitation learning is to have the agent (robot or human) which learns the task, by observing the demonstrator agent.

Learning by observation is an essential and noninvasive part of imitation without interfering with the imitatee's task. Many problems have to be handled for learning to imitate without affecting the imitatee. The primary difficulty is the correspondence problem, which is the mapping of actions or action sequences between a demonstrator (imitatee) and an imitator. This difficulty can be overcome if imitatee and imitator have similar kinematic structures. However learning to imitate by observing others that are different kinematically, is the recent focus of machines imitating human or animals, or machines imitating other machines [5]. For systems having kinematically different structures, the correspondence problem becomes an initial important issue that prevents imitation by observation, if not solved within the imitation methodology. Even when two human beings of the same dynamical structure imitate each other, it has been found in the medical literature that there exists a chronic disease called ideomotor apraxia that hampers the correspondence problem, leading to faulty organ matching, e.g., the demonstrator lifting a hand, the imitator may lift a leg [6]. During the imitation between two kinematically different systems, this disease is inherent. For example, imitation of human hand motions by a 3-fingered robot hand has an inherent correspondence problem where no one to one organ matching exists, the imitator being under actuated with less limbs than that of the demonstrator. The correspondence problem aside, many researchers speculated that imitative learning has valuable characteristics such as being noninvasive in the learning process which may in many cases speed up learning by not requiring communication between teacher and student thus, not interrupting the imitatee's task through interference [7].

In this work, we tackle the problem of imitating human hand postures by a system that possesses completely different dynamics, thus unable to initiate an imitational organ matching. Such an imitation is performed in nature by animals such as dolphins imitating their trainer's postures and gestures. We focus on developing an approach for imitation through observation of an imitatee's pose or gestures without the need of any organ matching. Towards this end, we focus on imitation of human hand gestures or body poses by a swarm having totally different dynamics than a human. More specifically this imitation is handled as the colony formation control of the swarm so as to resemble basic human hand preshapes. We consider in this work, a swarm that has a strikingly different dynamics than a human, which is more specifically a colony of fluid particles.

1.3 Objectives and Motivation

The focus of this work is the development of decentralized strategies to control very large groups of fluid particles. We can view this work in two parts: In the first part the human hand gesture and human body pose classifications are performed by using different classification and control methods like artificial neural networks, principle component analysis, and ellipse fitting technique. In the second part, according to the output of the controller and classifier, computational fluid parameters are set to appropriate values that guide suitably aggregation of the fluid particles, this generating a formation resembling the human hand gesture or body pose. Thus the imitation learning procedure is successfully realized between dynamically different structures, namely that of human hand gestures, and body poses, and that of fluid particles.

Our motivation comes from the two different sub areas. One is solving the correspondence problem when the imitator and the demonstrator do not share same dynamical structure. In our case the demonstrator is the hand gestures and body poses performed by the human, while the imitator is the fluid body which is composed of fluid particles. The other motivation stems from the fact that a great variety of characteristics desirable for a group of robotic - particles can be observed in fluids.

Some of the examples of those characteristics:

- fluid particles can be deformed easily
- fluid particles can contour objects easily
- the flow field variables and also the fluid phase can be easily manipulated in order to construct the desired behaviors and formations.

1.4 Methodology

In this thesis, we present a new approach avoiding the correspondence problem caused by the difference in embodiment between imitator and demonstrator in imitation learning. In our case the demonstrator is a human performing hand gestures and having body poses, and the imitator is the swarm of fluid particles. Although we do not mention a lower bound for the number of fluid particles in the swarm, the system in our approach should contain approximately at least tens of fluid particles in the simulations. For the control framework, we developed two frameworks: one is about the motion of individual fluid particles and the other one is adjusting the fluid parameter values to give the desired motion to the particles. That is, the formalism that we propose is capable of governing both the local interactions of individual particles and the global behavior of the whole system. The mathematical foundation of our formalism is based on the physical principles governing the flow of fluids. Hence, we have thoroughly exploited such branches of science as Fluid Mechanics, Fluid Dynamics, and Computational Fluid Dynamics (CFD).

1.4.1 Analogy

Fluids have diffusive and selfspreading nature such that they flow in the direction of decreasing density and spread out to fill in or pour into the space of their container. In nature, fluids exist in two structures namely, compressible fluids (like gases), incompressible fluids (like liquids). Gases diffuse into the space and achieve homogeneous density distribution over the environment regardless of its complexity. This property is very favorable for mobile sensor networks systems since the maximization of covering or deployment of the environment is a very crucial factor. On the other hand, liquids have much more directional motions than gases. For our problem, the imitation human hand gestures and body poses is chosen to be done by a swarm of incompressible fluid particles, since we need much more directional movements than gases, and the equation of motion of the particles are modeled as liquid like behavior. Another important characteristic of fluids is that any flow variation or disturbance in one part of the fluid affects the rest of fluid by propagation.

1.4.2 SPH

Fluid dynamics is related with the flow of the fluid and its mathematical model is based on the three fundamental physical principles:

- Conservation of mass
- Conservation of momentum
- Conservation of energy

The governing equations of the fluid dynamics are obtained by solving these principles. Due to the fact that analytical solution of these equations is not possible, computational methods are used. Among these computational methods, Smoothed Particle Hydrodynamics (SPH) is a meshfree particle method that models a fluid body as a collection of moving particles and numerically analyzes the flow equations in these particle locations. This method was originally proposed by Lucy et al. and Monaghan [8] - [9] for solving astrophysical problems. It recently became commonly used in fluid simulations and is very suitable for distributed and parallel computations.

SPH equations are derived from the continuum governing equations by interpolating from a set of disordered particles. In fluid dynamics problems, each particle represents a small volume of the fluid and the interpolation is performed by using differentiable interpolation kernels which approximate a delta function. The continuum equations are converted to a set of ordinary differential equations, where each one controls the evolution of an attribute of a specific particle.

1.4.3 A Framework for Control of Fluid Particles

The overall shape of the fluid flow highly depends on the environment. The obstacles and the medium of the environment determine the shape of the flow. For example, gas flows and liquid flows have different characteristics. Gases spread on the air and try to cover homogenously. On the other hand the flow of the liquids depends on the initial conditions, such as initial force and has directional motions in the environment. In computational fluid dynamics and SPH there are a lot of parameters that distinguish a particular flow from one another and result in quite different flow patterns.

Our aim in this work is to utilize these parameters to generate desired motions of the fluid particles in ways that are similar to the human hand gestures and body poses given as inputs to the system. These parameters are set by our controller based on either artificial neural networks or on classification of human hand gesture and body poses done by principal component analysis. According to the hand gestures and body poses performed by the human trainer, the fluid particle parameter values are set, and by controlled aggregation of fluid particles, desired flow motions are generated. There are no limitations about the values taken by parameters even if parameter terms are set to unphysical values.

For the proposed system, human hand gestures and human body poses are the inputs of the developed imitation system. These hand gestures and body poses are imitated by fluid particles in which there is an inherent correspondence problem since no one-to-one organ matching can be feasible. The motion of the fluid particles are modeled by smoothed particle hydrodynamics which is a mesh free particle based approximation method. The governing equations of the fluid dynamics are adapted to our system and each particle solves the equations to get its own velocities.

Since the proposed particle based SPH implementation is scalable with the number of particles, the algorithm runs independently for each fluid particle. Moreover, the decentralized approach is beneficial for the robustness of the system because centralized approaches suffer from partial failures that may lead to be failure of the overall system.

The important property of this particle based approach is that the behavior of the fluid body is macroscopically modeled and predicted.

In this work, we propose three different types of controller. In the first one, we use artificial neural networks to predict the fluid parameter values. For this purpose, first we extract the feature vector of the human hand gesture and according to this feature vector, the appropriate fluid parameter values are assigned. In this case the fluid parameters we use in formation control is the body force term which directly enters to the governing equation and has a guidance effect on the particles. These body force values are obtained after some experiments assigning the potential fields on the environment. The second one is based on classification of human hand gestures which is used as the input of the system generated by principal component analysis. According to this classification, the fluid parameter values are determined so as to guide the aggregation of the particles in a way similar to human hand gestures. The third one is based on the extraction of the human body and hand gesture regions of focus. While for human body pose imitation, these focus regions are torso, head, right and left arm locations, for imitation of hand gestures the focus regions are finger knuckles. The developed algorithm fits ellipses to these regions of focus, and maps ellipse parameters to the fluid parameters, which are for this third case body forces, stiffness and viscosity coefficients.

1.4.4 Principal Component Analysis

In the decision mechanism, for classification of the human hand gestures and body poses, principal component analysis (PCA) method is used. PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has a variance as high as possible (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that component is orthogonal to (uncorrelated with) the preceding ones.

In pattern recognition, problems may occur for the classification and recognition of data sets with high dimensional spaces. Significant improvements can be achieved by first mapping the data into lower dimensional space. The goal of PCA is to reduce the dimensionality of the data while retaining as much as possible the variation present in the original dataset. PCA allows us to compute a linear transformation that maps data from a high dimensional space to a lower dimensional space. With the dimension reduction in PCA, information loss is inevitable. Therefore with the minimization of the error function, much of the information is preserved. The best low-dimensional space can be determined by the "best" eigenvectors of the covariance matrix of the data set (i.e., the eigenvectors corresponding to the "largest" eigenvalues are called "principal components"). Basically the objectives of the PCA are to reduce the dimensionality of the data set and to identify new meaningful underlying variables. It means it allows us to reduce a set of observed variables into a smaller set of artificial variables called principal components. The resulting principal components may then be used in subsequent analysis which in our case is the use of these components to set parameters of the formation control of a fluidic swarm.

1.5 Contribution

In this thesis, we propose a fluid based controller which guides fluid particles in the imitation of human hand gestures and body poses. We base our formalism on the physics of fluids through some analogies that we establish between particle-based system and fluid bodies. In the formulation of our approach we use SPH as a distributed computational method for each particle.

The contribution of this thesis is the fluidic formation control for the imitation of human hand gestures and also body poses. This formation control is achieved by suitably adjusting parameters of fluid dynamics approximated by Smoothed Particle Hydrodynamics (SPH) which is a mesh free computational particle based Lagrangian method, whose resolution can easily be adjusted by fluid variables such as density. The most important advantage is the adaptive mesh free nature of the SPH method: Since it is not a grid based formulation, SPH is not affected by the arbitrariness of the particle distribution, and grid size. Therefore, it can handle the problem of large deformations which resides at the very foundational nature of a colony of particles imitating large deformations (curving, branching etc.) in hand gestures. We have found that SPH is highly suitable for the fluidic formation control of a swarm imitator, that imitates human hand posture with variable degree of resolution and can reliably duplicate observed movements represented by features extracted from human poses.

The contributions of this thesis can be summarized as follows:

- Our approach realizes that even when there is a correspondence problem between two embodiments, a kinematically different embodiment than a human, in our case fluid particles, can imitate human hand gestures and human body motions.
- Development of a decentralized formation controller for fluid particles for the imitation of human hand gestures and human body poses.
- To the best of our knowledge it is the first time such extreme embodiments (human and swarm of fluid particles) used for an imitation problem. For example it is not like humanoid robot imitating a human (it is somehow possible to es-

tablish one to one organ matching since humanoid robots have similar organs with human beings, like legs, arms, heads, hands etc.)

1.6 Outline of Thesis

This thesis is organized in six chapters. In this chapter we briefly introduced the main theme of the thesis. Our motivation and problem definition are given in this chapter. Also we briefly explained the methodology that we use in our proposed system.

Chapter 2 reviews the related work found in literature.

In chapter 3 we give the proposed fluid based approach in details. In this chapter we demonstrate our first approach for formation control which is a the Neural Network based controller and imitation results are given and discussed.

Chapter 4 explains another controller strategy based on the Principal Component Analysis (PCA) that generates a classification of hand gestures and human body motions. Also sensitivity analysis of the fluid particles are given in this chapter.

In chapter 5 the simulation results of human hand gestures and human body poses are given and discussed.

Chapter 6 presents a controller strategy which is based on the extraction of imitational focus regions. According to the shape and position of these regions, the best fitted ellipses are determined, and the parameters of these ellipses are mapped to the fluid parameters.

Concluding remarks, future works are discussed in Chapter 7. Also the summary of the contributions of this dissertation is given in this chapter.

CHAPTER 2

RELATED WORK AND MOTIVATION

2.1 Ongoing Works in Imitation Learning

Learning from demonstration also known as “programming by demonstration”, “imitation learning”, and “teaching by showing” is a topic that has received significant attention in the field of automatic robot programming over the last 30 years.

Learning by imitation in robotics covers methods by which a robot or the agent learns new skills through human guidance or more skilled agent [10]. Imitation learning method explores the ways of teaching a robot, new skills by user friendly means of interactions with other agents.

Developmental and neurophysiological studies showed that the imitative learning takes place once gaze-following abilities are fully established and joint attention can support the shared behavior required by imitation. Belardinelli et al. present a methodology for learning gaze shifts based on imitation of the gaze shifts of a human demonstrator, wearing a specially designed gaze-machine [11]. This device allows the robot to measure gaze shifts and fixations through mutual vergence, and subsequently, generalizes the acquired data by learning the scene’s salient features and the way cascaded programming is attained.

Hidden Markov Model (HMM) and Gaussian Mixture Regression (GMR) based approaches are developed by Calinon et. all. [12] in order to learn human motions. With the proposed approach, from the multiple demonstrations redundancies are extracted and build time-independent models to reproduce the dynamics of the demonstrated movements are possible. Validation of the proposed method is demonstrated on a 7 DOFs robotic arm learning and reproducing the motion of hitting a ball with a table tennis racket.

Wood and Bryson [13] proposed in their work that skill learning in the domain of a real-time computer game environment. In the game environment the imitator agent learns certain tasks of an experienced agent by doing some observations. The agent learns to classify of the actions which are observed in order to imitate. Besides, they assumed in this work that the imitator agent partially solves the correspondence problem which means that imitators are able to map the demonstrator's actions into their own action space.

Calinon et al. [14] present a programming-by-demonstration framework that extracts the relevant features of a given task and is able to generalize the acquired knowledge to different environments. The proposed framework is validated after some experiments in which the human demonstrator teaches a humanoid robot simple manipulatory tasks.

Lopes and Santos-Victor present a developmental strategy for acquiring the capability to learn by imitation. Their system follows a developmental pathway that comprises three levels: the first one concentrates on sensory-motor coordination and, subsequently, the second one on world interaction and the last one on imitation. The validation of the proposed method is demonstrated on a humanoid robot presenting results on the acquisition of perceptual and motor skills [15].

2.2 Correspondence Problem

In most robotic imitation-based learning schemes the student attempts to find a correspondence between the actions of the teacher and its own behavior repertoire; the teacher's body structure could range from being identical to similar and to very dissimilar. However, in all these research attempts, the student assumes that it knows the model of its body structure a priori. That is, it knows beforehand how its body parts are attached to each other; the forward kinematics, mapping from the joint space to the end effector space is also known a priori in most cases [17].

In the imitation learning the demonstrator and the imitator do not have to belong to the same species (a child imitating his/her parents, a human training a dolphin) or even be biological and artificial agent (a humanoid robot imitation a human). The second example is very interesting not only for computer science and the robotic community but also for researcher from biology, psychology etc. In last decade there have been lots of attempts for solving such a problem by designing controllers that allow robots to be programmed and learn more easily and efficiently [18] - [22]

Learning to imitate by observing another system of totally different dynamics has the inherent problems of understanding what to imitate, when and how to imitate. Imitation by observation provides the advantage of speeding up the learning process, since the focus is on ”learning on the flight of observation” process which is learning online where the demonstrator (imitatee) is not required to spend time with the imitator, to specifically train the imitator by leaving his/her own tasks. The demonstrator can continue doing his/her job and the imitator usually observes and learns in parallel to imitate without interrupting the demonstrator tasks.

In their paper, Alissandrakis et al. [23] focused on the problem of body mapping in imitation learning when the imitator and the demonstrator does not share the same embodiments (degrees of freedom, body morphology, action abilities etc.). They proposed a mathematical perspective formalizing body mappings which is called correspondence matrices between imitator and demonstrator. The correspondence matrices are produced by capturing different types of associations between degrees of freedom across dissimilarly embodied agents. The proposed approach is validated in a number of simulated 3-D robotic examples, using agents described by simple kinematic models and different types of correspondence mappings.

Alissandrakis et al. [25] introduced ALICE, the “Action Learning for Imitation via Correspondence between Embodiments,” a generic imitation framework that can be used by an imitator agent to find corresponding actions that produce similar states and effects as a model agent. In this agent based approach they have studied the results of applying different metrics and different granularities to generate different corresponding behaviors in the imitation of sequences of moves by differently embodied agents. These agents are embodied as chess pieces, whose movement constraints provide a well-known example of dissimilar “bodies” in a simple, discrete shared world, namely, a chess-board. A knight or bishop, for example, may imitate a zig-zag of three moves by a demonstrator queen, but each such agent is subject to the constraints on how it is allowed to move (its embodiment) and so can only roughly approximate the queen’s precise behavior. Nevertheless, the knight and bishop or other pieces can successfully imitate the behavior to varying degrees with respect to particular granularities (e.g., end result or trajectory matching) and various metrics. Varying the granularity and metrics for assessing attempts at imitation in this experimental setting illustrates the profound influence of these factors on the qualitative features of the resulting imitative behavior. Similarly, differently embodied animals, robots, or other autonomous agents whose behavior matches those of others can be modeled as using correspondences between their own states and actions and those of a demonstrator, using various granularities and measures of dissimilarity.

2.3 Human/Robot Imitation

Such imitation learning requires a complex set of mechanisms that detect what to learn from an imitatee by observing his/her movements and map them onto its own movements by transforming the imitatee behavior features into its own dynamical features. Such a process includes movement recognition, pose estimation and tracking, body correspondence, coordinate transformation, matching of observed movements, etc. [24].

For determining rough human body positions, body orientation and pose information Glas et.all. [26] proposed a method in which they used laser scanner data. For tracking and velocity control, particle filter approach is used in this research. Minato et al. [27] use a grid-based approach on an android by mapping human posture in three-dimensional position space. They attempt to naturally animate a robot to maintain social interaction. For posture transformation from human to an android they use a motion capture system which can measure the posture of the human subject and the android by attaching markers on the android so that all joint motions can be discriminated. Then the same markers are attached to the human’s body.

In another approach [28], for imitation of task dependent grasping, human subjects were asked to reach and grasp different objects with different orientations. These human movements were recorded using a motion capture system and analyzed offline. Analyzed data are mapped into a robotic hand attached to a robotic arm.

There are many ongoing research about imitation of human with humanoid robots nowadays. Mohan et al. [29] used imitation learning mechanism to teach a humanoid robot, iCub, to draw primitive shapes. iCub robot gradually learns to draw shapes on a drawing board after observing its human demonstrator. In another work, Calinon et. all. [30] proposed a probabilistic approach for learning human motions through imitation. In their hybrid approach they used Hidden Markov Model (HMM), Gaussian Mixture Regression (GMR) and dynamical systems which allows extraction of redundancies of multiple demonstrations.They built a time independent model that reproduces the observed movements. They validated their algorithm on an iCub humanoid robot.

In this kind of experimental setups, the major constraint is that there must be a large set of sensory data, collected from data gloves, magnetic trackers, stereo vision systems etc. to track and understand the movements of the demonstrator.

2.4 Imitation in Humans

Our main objective in the present work is to deal with imitation by observation between two dynamically different systems that carries a total mismatch of organs (inherent ideomotor apraxia disease in the imitator system [6]) so that imitation requires the understanding of what to imitate, when and how to imitate without the case of pattern matching for organ correspondence.

Humanlike intelligence requires an enormous amount of knowledge-solutions to the hard problems of survival and reproduction, which for our species have come to involve complex social and technological manipulations. Some of these solutions are passed to us genetically, and some are learned by an individual through trial and error [31].

In our social life, we learn many new ways of using our limbs such as dancing, cooking, driving etc. These abilities are mostly gained only by observing the other performers (demonstrators). This procedure is called social learning in psychology. Learning by observing is not an easy task when the demonstrator and imitator do not the share same embodiment because of the different motion capabilities. Sometimes even when two human beings of the same dynamical structure imitate each other, it has been found in medical literature that there exists a chronic disease called ideomotor apraxia that hampers the correspondence problem, leading to faulty organ matching, e.g., the demonstrator lifting a hand, the imitator may lift a leg [6].

Cognitive science research showed that the deficits in imitation is related with the fronto-parietal or callosal brain lesions. In [32], they have developed a neurocomputational model of imitation that reproduces the statistics of callosal apraxia errors. By that model they reproduced the statistics, and the exact nature of apraxic errors, and validated the model against a new set of experimental data.

Apraxia is generally defined as "a disorder of skilled movement not caused by weakness, akinesia, deafferentation, abnormal tone or posture, movement disorders such as tremor or chorea, intellectual deterioration, poor comprehension, or uncooperativeness". Apraxia is generally defined as the inability to perform voluntary movements meaning that it represents a deficit at the level of the sensory-motor transformations. On one hand, patients with apraxia may perform some spontaneous gestures that they cannot perform on command. This voluntary-automatic dissociation can be illustrated by an apraxic patient that could use his left hand to shave and comb himself, but could not execute a specific motor action such as opening the hand, so as to let go an object [33].

2.5 Imitation in Animals

In social life we also see imitation examples in animals with less similar embodiments, such as dolphins, chimpanzees imitating their trainer's or model's postures and gestures [34]. In these cases the correspondence map between the animal and human body is much more abstract than the similarly embodied demonstrator and imitator. It does not mean that the animals are aware of the correspondence problem; they just partially solve the correspondence problem between their bodies and the model demonstrator.

In [34], three female dolphins and one male dolphin were tested for imitation of human demonstrators using an unstated correspondence and showed that they are largely able to imitate their trainers after approximately 8 - 10 hours of dolphin human contacts each day: human forearms waving with bent elbow relate to wiggling pectoral fins; a human propelling her body partially out of water corresponds to a leap out of the water by the dolphin; human underwater somersaults correspond to that of the dolphin underwater too. Human head and dolphin head correspond, but human legs correspond to the dolphin's tail when raising them out of the water or slapping them on its surface. It can be speculated that possibly the dolphin understands how its body plan relates to that of a human.

Figure 2.1 shows the dolphin imitating its trainer at different body poses. This research was performed in Kewalo Basin Marine Mammal Laboratory in Honolulu [34]. Each action is characterized by the human demonstrator and the dolphin imitates those actions in real time. As we can see from Figure 2.1, the dolphin can relate its body to trainer's body by adopting postural or behavioral responses analogous to those of the trainer. As we can see easily, the body shapes of the two species (dolphin and human) differ considerably, the dolphin has to establish the analogy between its body and the trainer's body.

In particular, in Figure 2.1-a, the dolphin is imitating the human trainer who is leaning over backwards. In Figure 2.1-b, the human trainer stands facing the dolphin and leans her body and head backward. The dolphin imitates that posture, standing on its tail and leaning its own body and head backwards. In Figure 2.1-c, the human first faces sideways with her arms bent and raised at chest level. The dolphin also assumes this sideways posture, its body erected and pectoral fins out of water. In Figure 2.1-d, the human trainer raised her leg, and as the correspondence the dolphin raises its tail in the air. In Figure 2.1-e, the human leans forward and bends her head and upper body downward. The dolphin lowers its own head, hunching forward at the water surface. In Figure 2.1-f, the human jumps into the air. The human trainer's knees are

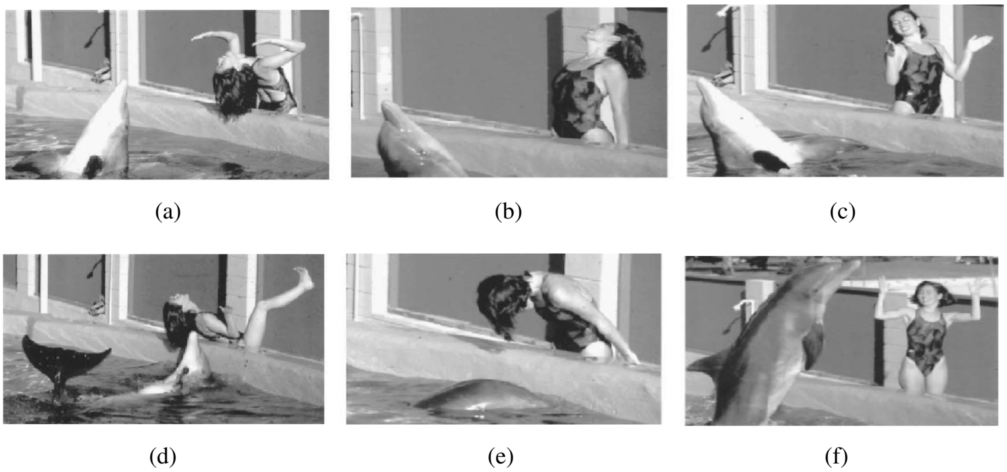


Figure 2.1: Examples of motor imitation of a human by a dolphin

about the level of the top of the tank wall. In correspondence, the dolphin quickly jumps itself upward, raising its body well above the surface of the water.

2.6 Fluidics in Robotics

2.6.1 Controlling the swarms

In our life we see swarm of biological creatures such as ants, fishes, birds etc. All of these swarms have aims on being together or working in a colony. The leaf cutter ants work together to harvest fresh plant matter to grow food, to sustain and expand the colony. The fish swim in schools as a defense mechanism from predators and to aid in foraging for food, and the birds fly in formation to reduce the drag force that each bird experiences compared to if each bird was flying alone. While the objectives of each group of organisms are quite different, they share the same underlying theme. That is, by working together, the task or objective at hand can be completed more quickly and efficiently than if the task were undertaken alone. Recognizing these benefits, researchers have applied the lessons learned from nature to the control and coordination of multiple agents which include robots and unmanned aerial vehicles (UAVs), and the coordination of multiple agents has become known as robotic formation control where each robot or UAV in the group seeks to orient itself relative to its neighbor or a leader.

The multi-robot systems (MRS) or cooperative robotic systems are used for solving the great variety of the tasks. These systems are dedicated to study techniques that allow robots in a team to cooperate with the other robots in the team or with humans in order to complete given task. The reason of using such a collective robotic system, for the solution of the given task can be summarized as follows:

- might be more reliable,
- more fault-tolerant,
- more flexible and
- less expensive.

Controlling multi-robot systems has a growing interest among the robotics community over the past decade [35]-[40]. We have seen recently the creation of a number of closely related research areas under the name of cooperative robotics, collective robotics and swarm robotics. Most of approaches can be categorized into centralized or decentralized approaches. Centralized approaches assume the existence of a central entity which is able to plan actions for each robot and also to obtain information of the whole group in order to perform the required task in an optimal way. Although centralized approaches, in general, guarantee completeness of the task, they are not scalable to large groups of robots due to computational limitations. On the other hand, decentralized approaches may use a divide-and-conquer strategy to provide more scalable solutions. In fact, decentralized approaches advocate that each robot should be responsible for planning its own actions based only on the local information available.

In swarm robotics or swarm of autonomous agents, the main idea is to control very large groups of primitive robots (ten to hundreds). The robots or agents in the swarm have limited capacity of communication, sensation and actuation. The important characteristics of the swarm are

- being flexible in order to overcome different tasks by using different coordination mechanisms

- being decentralized methodology, in the swarm all robots are anonymous
- being robust for dynamical change of the environment or addition/deletion of new robots.

The main application areas of the swarm robotics are search and rescue operations in hazardous environments or in places where humans cannot have access, contention of oil spills in the ocean, transport of heavy objects, environment monitoring, surveillance, etc. Moreover with the development of nanotechnology, the application areas of the swarm robotics are increasing in the nanorobotics field such as insect inspired robots.

The other application area of the swarm robotics is mobile sensor networks. The ultimate goal of such a system is to do surveillance by distributing sensor nodes over the environment. Starting from an initial configuration of the nodes, sensors are deployed in such a way as to maximize the total area covered by the network [41].

2.6.2 Fluidics Approaches

Robotics finds a grooving interest all over the world in the last half century as a response to the evolution of human social needs, from the industrial robotics that released the human operator from dangerous or risky tasks to the recent explosion of field and service robotics to assist human [42]. The researchers in robotics are generally working on making human life easier. For this purpose they design humanoid robots for helping older or disabled people, robotic manipulators and dexterous robotic hands for grasping and manipulating dangerous objects. Since the robotics has a growing interest, most of the other scientific branches are trying to contribute to this area new idea like biology (biologically inspired robotics), physics, and philosophy (cognitive science) etc.

Fluidics approaches have appeared recently in robotics. In the literature, most of the works about both robotics and fluidics are related with the hydro-mechanic or hydro-elastic actuators that are used in robot manipulators.

Fluid based modeling has been used in robotics, generally for swarms and recently in the formation of geometric patterns with multiple robots [43] - [44]. In these works, mobile robots are modeled as fluid particles and are controlled by the help of fluid dynamics parameters. To avoid particles from colliding with obstacles they generate some external forces by the help of finite element method (FEM) [44]. FEM is used for the computation of a vector field which models the external forces applied to the fluid. The particle approximation is done by Smoothed Particle Hydrodynamics in the model. SPH algorithm is solved for each particle robot in the colony and the velocity control input is applied after that external forces are calculated. They assumed that each robot in the colony is a SPH particle. Since the derived controller is decentralized, for solving the governing equation of the particle robots, local information is necessary: the external field at the location of the robot (computed using FEM) and position and velocity of neighboring robots. Since each robot in the colony has the information of itself and of neighboring robots, it is able to compute its own external force to escape from the obstacle. In other words they established a weak coupling between SPH and FEM, where FEM is used to compute static external forces to SPH equations. Their main contributions are pattern generation in the not obstacle free environment like circle, star, square etc. and maximization of the coverage in an unknown environment, solving the deployment problem. The clear advantage of the proposed method is that, there is no need for labeling robots. Therefore, all the robots run the same software and the success of the given task execution does not depend on specific members of the group.

In another work in which the fluid particles are modeled as a member of the colony, particle robots are controlled by the help of fluid dynamics parameters [43]. They mentioned in their work a way to optimally cover an environment by mobile sensor networks. The nodes in the mobile sensor networks are modeled as fluid particles. The motion of the particles are modeled both in compressible and incompressible fluid dynamics. There are various characteristics of fluid flows which are desirable in swarm robotics, such as obstacle avoidance and source to sink optimal path finding, path following behaviors of the fluid.

In grasping, for determining the continuum between hand preshape and the object, the fluidic approach is introduced in [45]. They modeled the human hand, medium and fluid by SPH to see the effect of the human hand on fluid particles while the hand is entering the fluidic environment.

Another research area related with fluids in robotics is underwater robotics. During the last few years, the use of underwater robotic vehicles has rapidly increased since such vehicles can be operated in deeper, riskier areas unreachable by divers. The potential applications of such vehicles include fishing, underwater pollution monitoring, rescue, and waste cleaning and handling in the ocean as well as at nuclear sites.

CHAPTER 3

FLUIDIC FORMATION CONTROL

3.1 Control Architecture

The formation control architecture of the fluidic swarm that has dynamics modeled by SPH is shown in Figure 3.1. This architecture realizes the mimicking human hand gestures and body poses captured from a camera.

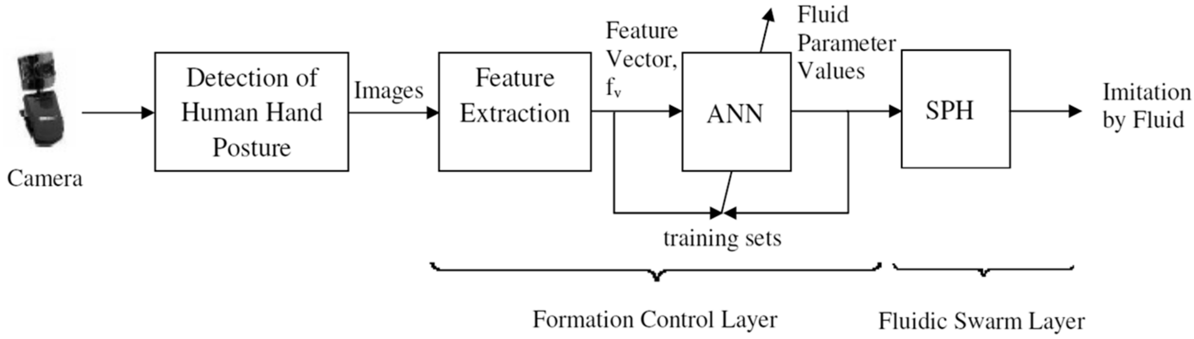


Figure 3.1: Fluidic formation controller for swarms

The need for imitation by swarm is triggered upon comparison of the initial formation of the swarm with the hand shape captured by a camera. The difference between these two shapes needs to be eliminated by imitation. After this triggering,feature extraction (section 3.2) is performed on the hand posture image that needs to be imitated. Feature extraction generates the feature vector which forms the input to the formation controller. The fluidic formation controller adjusts the parameters of the SPH swarm dynamic model (the plant, section 3.3) according to the input and the supervisor. The supervisor of the controller is based on an Artificial Neural Network (ANN) based learning module that is a feed forward neural network with single hidden layer composed of 5 nodes. Its input layer has as many neurons as the feature vector components. The number of neurons in its output layer represents the number of parameter to be controlled in the plant. In this work we only control the body force parameter, thus the ANN has only a single output. Examples of training sets and controller behavior are introduced in section 3.4.

3.2 Feature vector extraction of human hand postures

In this section we introduce examples of feature extractions based on captured images of hand preshapes. Consider the case where the colony has to imitate different branching of the fingers, corresponding to different hand postures, as shown in Figure 3.2. Images of the hand are skeletonized by first-filtering to avoid noisy artifacts coming from skin texture, then passing the hand images through a sobel operator for a binary output on which edge detection can easily be conducted. The obtained binary image is then dilated in order to obtain continuous distinct edges. Then skeleton of the hand preshapes is determined by finding the equidistant pixels from the boundaries. After skeletonizing, the characteristic features differentiating each hand postures are extracted. As an example we give in Figure 3.2 the feature vector extracted from scissor like behaviors.

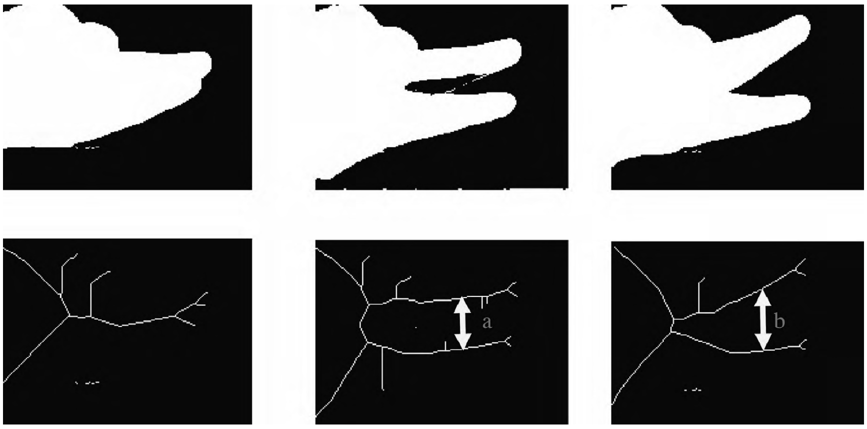


Figure 3.2: Feature extraction of scissor movement-like hand behavior

The general heuristic template we determined at the initial phase of this thesis work is given below.The constructed feature vector for scissor like had preshape is given below.

$$f_v = \begin{bmatrix} \text{Line-edge} \\ \text{U shape fitting} \\ \text{V shape fitting} \\ \text{Two curves fitting} \\ \text{One curve fitting} \\ \text{Distance between branches} \end{bmatrix}$$

This template summarizes the characteristics that can be found in human hand preshapes in general. The feature vectors corresponding to the hand preshapes given in Figure 3.2 and 3.3 (a to c) are given bellow respectively.

$$fv_a = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, fv_b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ a \end{pmatrix}, fv_c = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ b \end{pmatrix}$$

In Figure 3.2-a, the fingers are positioned as a fat line and no other characteristic of the preshape can be evaluated other than the presence of a line edge in the skeleton. In Figure 3.2-b, a U-shape characteristic appears with an aperture of a units which become the second entry and the last entry of the feature vector. The orientation of the hand has not changed. In Figure 3.2-c, the hand preshape turns into a clear V-shape that determines its apparition by a 1 in the third entry of the feature vector. Its aperture changes to b units while the hand direction of the base finger has not changed.

The feature vectors of the pinching fingers and C-shaped finger postures are given in Figure 3.3, after skeletonizing the preshapes. The feature vector template for this case is developed as characterized by curve fitting parameters and aperture of curved fingers.

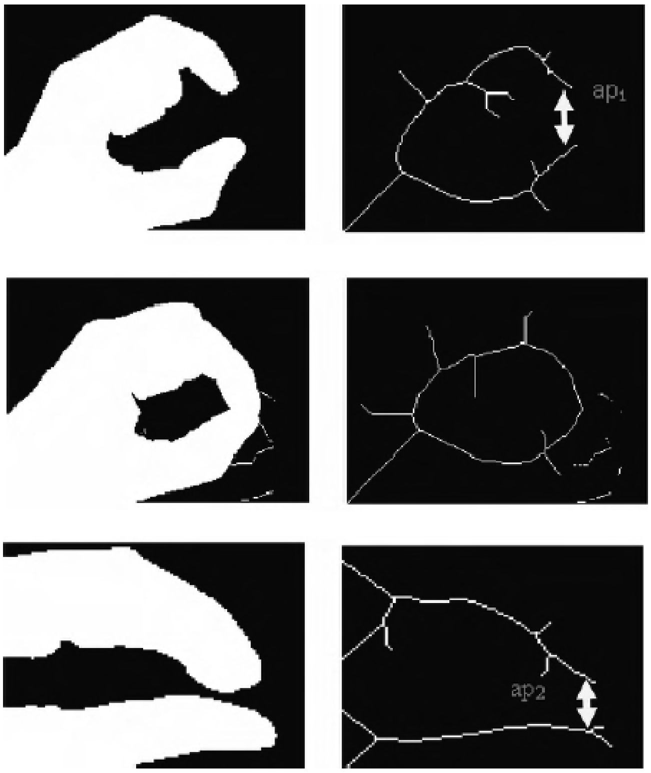


Figure 3.3: Feature extraction for pinching and C-shaped hand gestures

For the pinching and C-shaped hand gesture, the created feature vector is given below.

$$f_v = \begin{bmatrix} \text{Line-edge} \\ \text{U shape fitting} \\ \text{V shape fitting} \\ \text{Two curves fitting} \\ \text{One curve fitting} \\ \text{Distance between branches} \end{bmatrix}$$

The feature vectors corresponding to each frame a, b, or c of Figure 3.3 are given individually as:

$$fv_a = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ ap_1 \end{pmatrix}, fv_b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, fv_c = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ ap_2 \end{pmatrix}$$

In this preshape of Figure 3.3-c only 1 curve can be fitted for the upper finger, the lower one being a line. Therefor for this preshape the first and fifth entries are 1. ap_2 represents the value of the aperture in pixels. These feature vectors $fv_{a,b,c}$ form the input of the ANN either in training mode or testing mode (which is the case for an actual imitation run).

3.3 Modeling the swarm imitator as a colony of fluid particles

While compressible fluids (like gases) are spread around the environment homogeneously, incompressible fluid motions (like liquids) have directional characteristics. Since in human hand gestures, preshape features are generated from directional movements, we model our fluid particles as element of an incompressible fluid flow. We control these incompressible fluid particles by changing parameters of the fluid flow dynamics to get the desired hand preshapes. The fluid flow dynamical formalism based on SPH is introduced in detail in this section, in which parameters are controlled by the fluidics controller architecture introduced in section 3.1. In Section 3.4 the training sets of the controller will demonstrate which parameter change of SPH will guide the colony of particles into mimicking features of human finger behaviors such as the scissor-like or pinching behaviors analyzed by their feature vector in section 3.2.

Our SPH formulation of the dynamics of our fluid based imitator is adapted from [46], solving the momentum equation to determine the particle accelerations based on parameters, such as density, pressure viscosity, obtained from neighbor particles in the support domain.

In the SPH method, the problem domain is represented by a set of arbitrary distributed particles and no specific discretization connectivity for these particles is needed (mesh-free). In our work, these particles are controlled by the help of fluid parameters to get the desired shapes looking like human hand preshapes. In our colony each particle is affected by a finite set of neighboring particles forming the "support domain" of that particle (Figure 3.4).

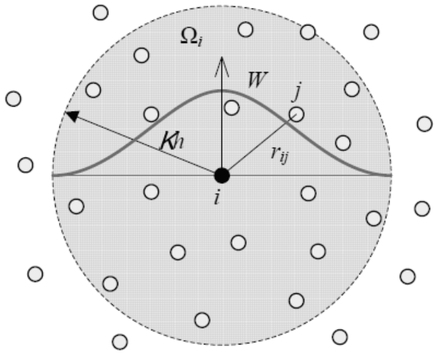


Figure 3.4: The support domain for particle i and 1D projection of a smoothing function over it.

All calculation of the field variables depend on these neighboring particles with the support domain that determines the resolution of discretization. Since SPH is an approximation method, integrals are approximated based on field functions. This is also known as kernel approximation in the SPH method. The kernel approximation is then discretized based on particle units (particle approximation). It is done by replacing the integration in the integral representation of the field function with summations over all the corresponding values at the neighboring particles in the support domain.

More specifically, consider the integral representation of a function $f(x)$ in the SPH method:

$$f(x) = \int_{\Omega} f(x^{'})\,\delta(x - x^{'})\mathrm{d}x^{'} \tag{3.1}$$

where Ω is the volume of integration, f is a field function like viscosity or pressure, and $\delta(x - x^{'})$ is the Dirac delta function. For the approximation of the integral representation in equation 3.1, Dirac delta function is replaced by a smoothing (kernel) function W such as:

$$< f(x_i) >= \int_{\Omega_w} f(x^{'})\, W(x - x^{'}, h)\mathrm{d}x^{'} \tag{3.2}$$

where the angle brackets indicate the approximation and W is the smoothing function which not only determines the pattern for the function approximation but also defines the dimensions of the support domain of particles.Ω_w is the volume of integration confined to W. Parameter h is the smoothing length defining the influence area of the smoothing function. This smoothing function is used in the calculation of fluid variables approximation. The smoothing function plays a very important role in the SPH approximations, as it determines the accuracy of the function and efficiency of the computation. For the smoothing function, there are various choices such as Gaussian kernel, the cubic spline kernel, the quadratic smoothing function etc. Due to its smoothness, stability, and accuracy we choose in this thesis work the Gaussian kernel given in Equation 3.3:

$$W(R_{ij}) = \begin{cases} \alpha_d e^{-R_{ij}^2} & R_{ij} \leq \kappa h \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Here $\alpha_d = \left(\frac{1}{\pi h^2}\right)$ for a two-dimensional space, $\kappa = 2$, $R_{ij} = |x_i - x_j|/h$. For the particle approximation, the continuous integral representation of the kernel approximation is converted to a discretized form of summation over all the particles in the support domain shown in Figure 3.4.

The particle approximation of the function is obtained as

$$< f(x_i) >= \sum_{j \in \Omega_i} \frac{m_j}{\rho_j} f(x_j) W(x_i - x_j, h) \tag{3.4}$$

where m_j and ρ_j are the mass and density of particle j. Equation (3.4) states that the value of a function at particle i is approximated using the average of function values at all particles in the support domain of particle i, weighted by the smoothing function.

Density approximation is computed via the relation

$$\rho_i = \sum_{j=1}^{N} m_j W_{ij} \tag{3.5}$$

where ρ is density, N is the number of particles which are in the support domain of particle i, m_j is the mass of particle j and W_{ij} is the smoothing function of particle i evaluated at particle j computed as

$$W_{ij} = W(x_i - x_j, h) = W(|x_i - x_j|, h) = W(R_{ij}, h) \tag{3.6}$$

where R_{ij} is the relative distance between particle i and j . Since density approximation (equation (3.5)) determines the particle distribution and the smoothing length evolution, it is really important in the SPH method, and simply states that the density of a particle can be approximated by the weighted average of the densities of the particles in the support domain of that particle.

As the density equation is one of the important equations for fluid flow, so is acceleration and momentum. The momentum equations calculate the time rate of change of velocity using substantial derivative D/Dt. Momentum and acceleration couple to density via the relations

$$\frac{Du_i}{Dt} = -\sum_{j=1}^{N} m_j\left(\frac{p_i + p_j}{\rho_i\rho_j} + \Pi_{ij}\right)\frac{dW_{ij}}{dx_i} + \sum_{j=1}^{N} m_j\left(\frac{\mu_i\varepsilon_i^{xx} + \mu_j\varepsilon_j^{xx}}{\rho_i\rho_j}\frac{dW_{ij}}{dx_i} + \frac{\mu_i\varepsilon_i^{xy} + \mu_j\varepsilon_j^{xy}}{\rho_i\rho_j}\frac{dW_{ij}}{dy_i} + f_i^x\right) \tag{3.7}$$

$$\frac{Dv_i}{Dt} = -\sum_{j=1}^{N} m_j\left(\frac{p_i + p_j}{\rho_i\rho_j} + \Pi_{ij}\right)\frac{dW_{ij}}{dy_i} + \sum_{j=1}^{N} m_j\left(\frac{\mu_i\varepsilon_i^{xy} + \mu_j\varepsilon_j^{xx}}{\rho_i\rho_j}\frac{dW_{ij}}{dx_i} + \frac{\mu_i\varepsilon_i^{yy} + \mu_j\varepsilon_j^{yy}}{\rho_i\rho_j}\frac{dW_{ij}}{dy_i} + f_i^y\right) \tag{3.8}$$

where u_i and v_i denote velocities for particle i along the x- and y- directions, respectively. p is the pressure, ρ is the density of the specified particle, Π_{ij} is the artificial viscosity (equation 3.9), W_{ij} is the smoothing kernel, and ε denotes the stress factor.

In momentum equations (eq.3.7) and (eq.3.8) we have three terms on the right hand side. The first term is the major portion of the equations due to the pressure gradient with the dissipative artificial viscosity which is mainly used in order to model the shock waves in a tube in fluid flow simulations. The second term shows the viscosity and stress parameters. One of the particles starts to move, the other particles are affected because of this motion. ε_{xx} and ε_{yy} denote normal stress, and ε_{xy} denotes shearing deformation, for generating dragging effect (equation 3.11). The last term, f , is the body force. Since it directly enters to the momentum equation, it has a direct effect on the flow. In fluid physics the typical body force is the gravitational force. It is suitable for the guidance of the particles. In this thesis we demonstrate the control of the body force to get the desired trajectories of the particles in the imitation learning of human hand gestures.

The artificial viscosity term in equations (3.7) and (3.8) can be written as

$$\Pi_{ij} = \begin{cases} \frac{\beta_{\pi}\varphi_{ij}^{2}}{\bar{\rho}_{ij}} & v_{ij}x_{ij} < 0 \\ 0 & v_{ij}x_{ij} \geq 0 \end{cases} \tag{3.9}$$

where

$$\varphi_{ij} = \frac{h_{ij}v_{ij}x_{ij}}{|x_{ij}|^{2} + \phi^{2}}, \qquad \phi = 0.1h_{ij}, \qquad \bar{\rho}_{ij} = \frac{1}{2}(\rho_i + \rho_j), \qquad h_{ij} = \frac{1}{2}(h_i + h_j) \tag{3.10}$$

The shear stress rates in equations (3.7) and (3.8) have a form similar to

$$\begin{aligned} \varepsilon_i^{xx} &= \tfrac{2}{3} \textstyle\sum_{j\epsilon\Omega_i} \tfrac{m_j}{\rho_j}\Big(2u_{ji}\tfrac{dW_{ij}}{dx_i} - v_{ji}\tfrac{dW_{ij}}{dy_i}\Big) \\ \varepsilon_i^{xy} &= \textstyle\sum_{j\epsilon\Omega_i} \tfrac{m_j}{\rho_j}\Big(v_{ji}\tfrac{dW_{ij}}{dx_i} + u_{ji}\tfrac{dW_{ij}}{dy_i}\Big) \\ \varepsilon_i^{yy} &= \tfrac{2}{3} \textstyle\sum_{j\epsilon\Omega_i} \tfrac{m_j}{\rho_j}\Big(2v_{ji}\tfrac{dW_{ij}}{dy_i} - u_{ji}\tfrac{dW_{ij}}{dx_i}\Big) \end{aligned} \tag{3.11}$$

Besides these differential equations, there is a suitable state equation between pressure p and density ρ for modeling compressible and incompressible fluid flow in the form of the equations

$$p_i = \rho_i R_i T_i \qquad \text{for compressible (gas like behavior) flow} \tag{3.12}$$

$$p_i = \beta_i((\frac{\rho_i}{\rho_0})^{\gamma} - 1) \qquad \text{for incompressible (liquid like behavior) flow} \tag{3.13}$$

where R is the specific gas constant, T is the temperature, β is the stiffness constant, ρ_0 is the reference density and γ is a constant around 7.

As a state equation, we use equation (3.13), since incompressible flow is much more suitable for our purpose. We are mainly interested in the directional trajectories of the particles, to get the desired shapes. The particle acceleration is calculated from the momentum equation (3.7) and (3.8) by using equations (3.9)-(3.11). Since the Navier-Stokes Equations do not have any analytical solution, time marching method is used for calculation of the particle velocity.

Up to this point we gave the mathematical background about our swarm modified SPH methodology. These equations are based on the Navier-Stokes equations and since the Navier-Stokes equations cannot be solved analytically, the SPH kernel and particle approximations are used for discretization of partial differential equations. The aforementioned SPH formulation is derived by discretizing the Navier-Stokes equations spatially, leading to a set of ordinary differential equations with respect to time that can be solved via time integration.

Our ultimate goal is to learn to control the SPH-based flow model parameters for the colony to resemble the hand gesture features of a human.

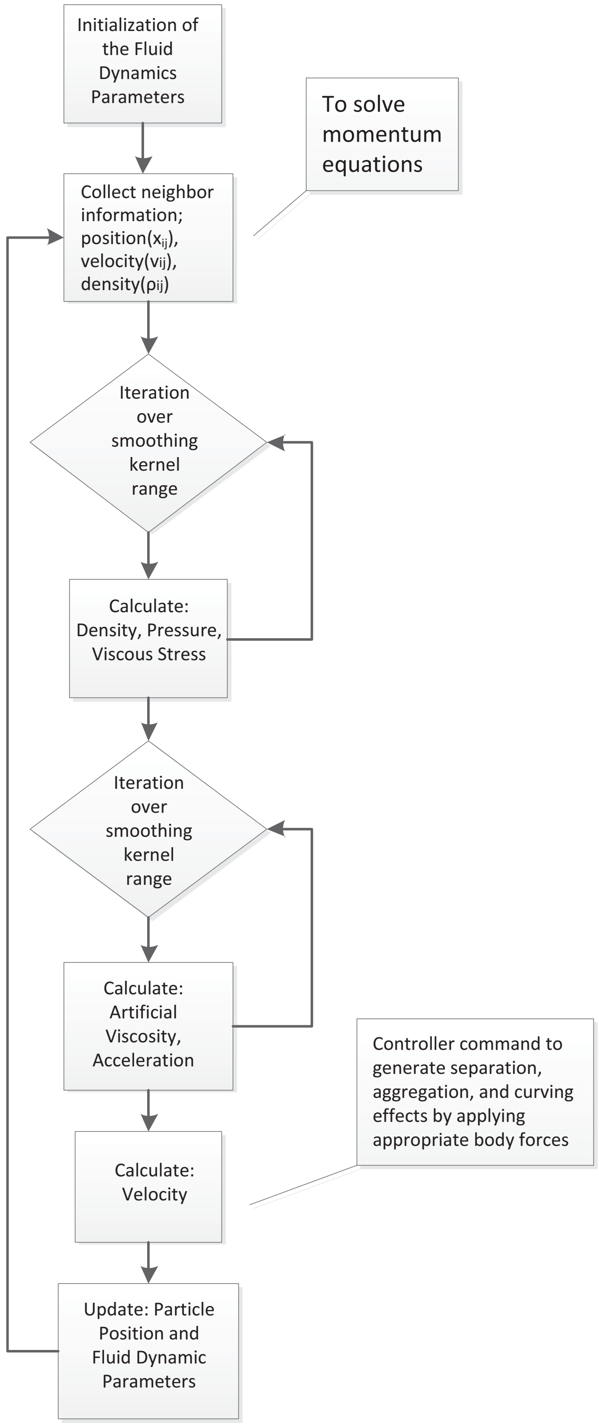

Initialization of the Fluid Dynamics Parameters
To solve momentum equations
Collect neighbor information; position(x_ij), velocity(v_ij), density(ρ_ij)
Iteration over smoothing kernel range
Calculate: Density, Pressure, Viscous Stress
Iteration over smoothing kernel range
Calculate: Artificial Viscosity, Acceleration
Controller command to generate separation, aggregation, and curving effects by applying appropriate body forces
Calculate: Velocity
Update: Particle Position and Fluid Dynamic Parameters

Figure 3.5: Flow chart diagramming the instance of control algorithm, performed for each particle *i*.

An example of formation control action performed by the controller which is given here for demonstration purposes is the flow chart of Figure 3.5. Here, the controller uses the body force term in the momentum equations. After updating the particle positions, the output of the controller gives the body force set for all particles for that iteration.

The flow chart of this instance of control action algorithm is provided here as an example in Figure 3.5. First, for particle *i* , the controller initializes its fluid dynamic parameters for setting the control commands. Then particle *i* needs to know the fluid variables of its neighbor particles which are inside the support domain. To solve the governing equations, it collects the information of position, velocity, and density of its neighboring particles. To solve the momentum equations in (Eq. 3.7), and (Eq. 3.8), particle *i* needs the values of pressure (Eq. 3.13), density (Eq. 3.5), and viscous stress (Eq. 3.11) and collects these information from its neighboring particles which are within the smoothing function volume. After calculation of these fluid variables, the acceleration of particle i is calculated from the momentum equation. After updating the position of the particle with time integration, the flow chart goes back and gets new commands from the controller to form the desired formation.

3.4 Generating the training sets of our controller

The ANN shown in Figure 3.1 has been trained by I/O pairs which are preshape features and fluid parameters. The principal fluid parameter demonstrated in this work is the body force. The body forces are generated with the idea of artificial potential fields in the training mood. Some attraction waypoints are determined on the hand images, and the whole particles are attracted by these potential field locations. After particles reach one of them, the next waypoint is activated and generated body forces are stored in a matrix. The function of a body force is given in the following equation (3.14):

$$F_{x,y} = \begin{cases} F_x = \alpha(d-r).cos(\theta) & \\ F_y = \alpha(d-r)sin(\theta) & r \leq d \leq s+r \\ F_x = \alpha s cos(\theta) & \\ F_y = \alpha s sin(\theta) & d > s+r \end{cases} \tag{3.14}$$

In this equation, d is the distance between particle position and waypoint, θ is the angle between particle and the waypoint, r is the waypoint radius, s represents the spread of the field, α is the scaling factor which scales the magnitude of the body force vector. If particle i has a journey every point in a two-dimensional simulation environment, the collection of the body force vectors would look something like the diagram illustrated in Figure 3.6.

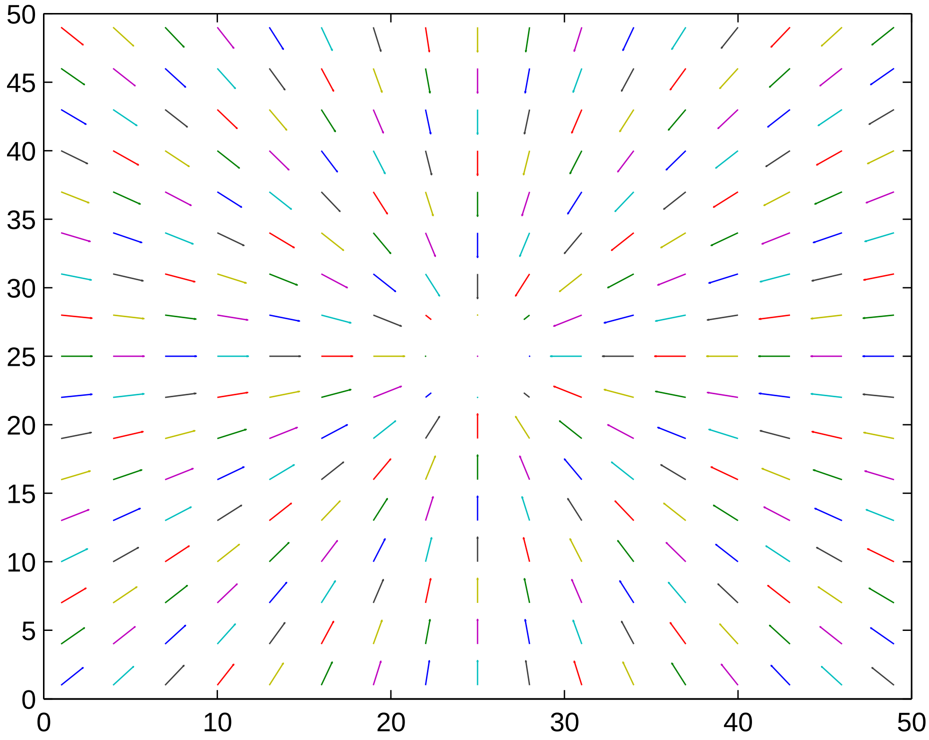


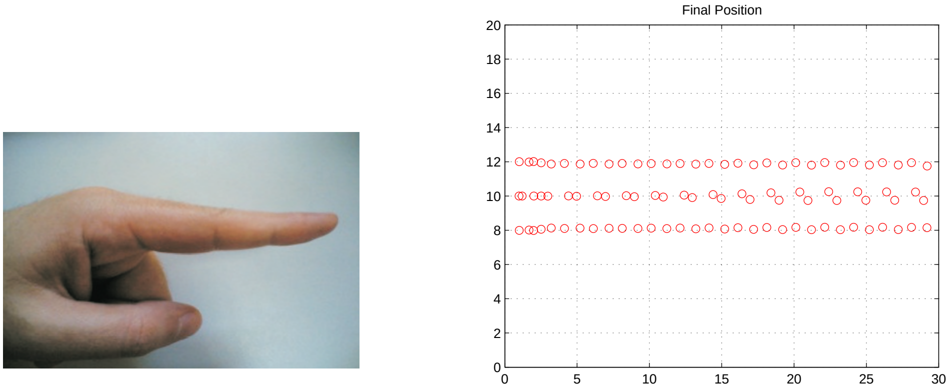
Figure 3.6: Body force vectors for an attraction point.

The body force generation approach is valid for the rest of this thesis work.

The results given in Figures 3.7, 3.8 and 3.11 demonstrate the importance of controlling body force in fluid flow and the influence that this parameter has, in the orientation and fingering effects of formation control. In each example, first we show the captured hand posture image for which a feature vector is extracted. Then the fluidic controller output in terms of body force value is given. And finally the fluidic swarm formation outcome of the system is given. The I / O training pairs of the ANN is f_v feature vector and body force vector.

3.4.1 Example-1:Horizontal Hand Gesture

In Figure 3.7-a, the captured human hand preshape is given. In this human hand preshape, tip of the human finger points a horizontal point.



(a) Horizontal hand preshape (b) Horizontal hand behavior with particles

Figure 3.7: Horizontal Pointing Training Set

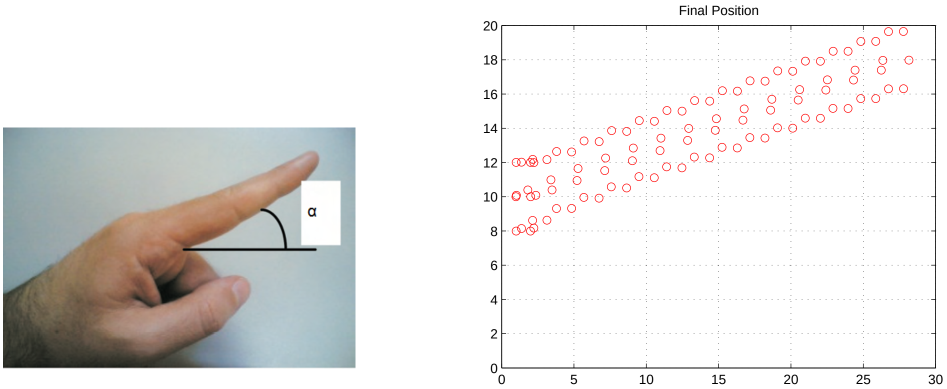
Since there is only 1-line edge in this hand preshape, the generated feature vector for this hand preshape is $[\,1\ 0\ 0\ 0\ 0\ 0\,]^T$.

The corresponding ANN output in terms of the body forces is $[\,2\ 0\,]^T$ giving an intensity of 2 for force in the same orientation x as the finger and no force in y direction.

The fluidic swarm formation outcome of the system is given in Figure 3.7-b where the horizontal axis is the position in x direction and the perpendicular one represents position in y direction.

3.4.2 Example-2:Pointing at angle hand preshape

In this example we use a human hand preshape that points at angle. The captured human hand with one finger points at an angle α. In Figure 3.8-a, the captured human hand preshape is given.



(a) Pointing hand preshape

(b) Pointing at angle behavior with particles

Figure 3.8: Pointing at angle training set

The generated feature vector for this hand preshape is $[\,1\ 0\ 0\ 0\ 0\ \alpha\,]^T$. The corresponding ANN output in terms of the body forces is $[\,1\ 0.3\,]^T$ where now force f_x has reduced intensity and f_y is now small but no zero to generate a linear angular flow.

The fluidic swarm formation outcome of the system is given in Figure 3.8-b.

3.4.3 Example-3:Separation of fingers training set

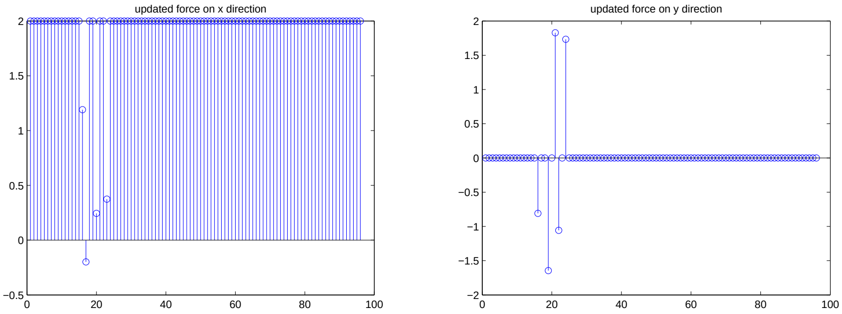
Here we use a two finger preshape where the hand motion separates the two fingers in a V shaped alignment. Figure 3.9 gives the captured human hand preshape.

The feature vector template is the one given in section 3.2. Since this hand preshape looks like a V-shape, the third entity of the feature vector is 1. Another property of the hand preshape is the approach distance between fingers. The generated feature vector for separation like behavior of hand preshape is $[\,0\ 0\ 1\ 0\ 0\ b\,]^T$.

The corresponding ANN output in terms of the body forces along x and y directions

Figure 3.9: Separation of fingers

are given in Figure 3.10.



(a) force distribution on x axis

(b) force distribution on y axis

Figure 3.10: Body force distribution for separation

As we can see from the plot of the body forces, after simulation is started, we applied the corresponding forces. In order to generate two different pathways, applied body forces are changed and the particles are divided into two branches for fingering effect. After dividing the particles into two branches, we applied again constant body forces which give the particles a directional motion along *x*-axis.

The fluidic swarm formation outcome of the system is given in Figure 3.11.

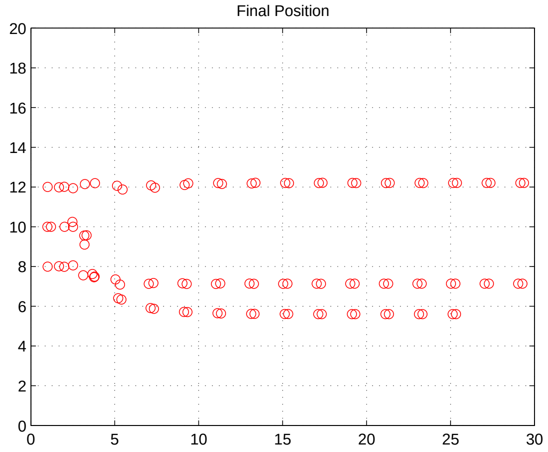


Figure 3.11: Separation of fingers with particles

3.4.4 Example-4:Pinching of fingers training set

The next example is shown in Figure 3.12 as a pinching grasp.In this grasping preshape the hand has a posture where the two fingers close at the tip. Thus the swarm particles have to be separate first into two groups and then converge at a further point.

As we can see from the Figure 3.12 the aperture between finger tips is ap_3, there is 1 line edge with θ angle for thumb, and 1 curve can be fitted on the index finger. The corresponding feature vector for this hand preshape is then extracted as $[\,1\;0\;0\;0\;1\;{\scriptstyle ap_3}\,]^T$.

To get the desired hand preshape given in Figure 3.12 ANN output sequences in terms of body forces can be seen on graphs 3.13. After starting of the simulation, first particles are divide into two groups, and then two branches are collected in a point with the applied body forces.

The fluidic swarm formation outcome of the system is given in Figure 3.14.

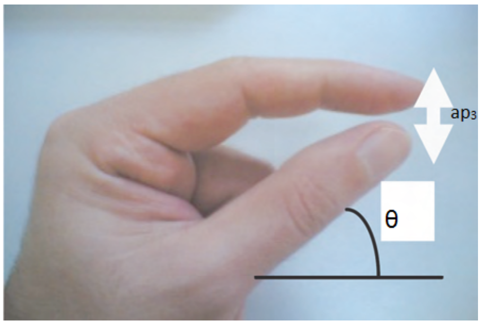
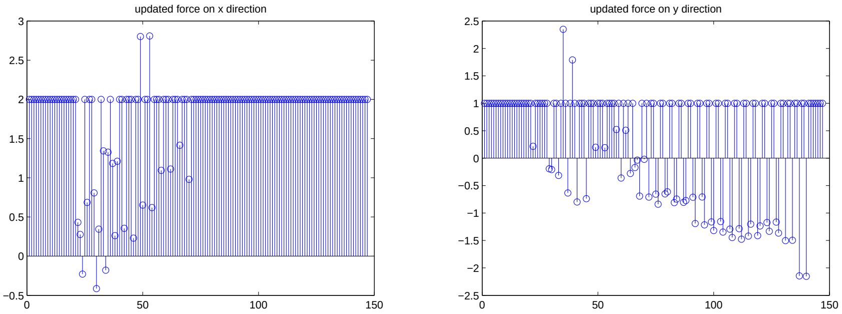


Figure 3.12: Pinching hand gesture



(a) force distribution on x axis

(b) force distribution on y axis

Figure 3.13: Body force distribution for pinching hand preshape

3.5 Imitation Results

We provide here simulation results of an SPH fluid swarm formation that mimics some basic hand preshapes based on body force control commands.

We give here full sequence from preshape feature vectors to swarm imitation for two examples, namely that of scissor like hand behavior and that of pinching preshape.

Figure 3.15-a gives the captured hand preshape image together with its skeletonized rendering in Figure 3.15-b. In this hand preshape we see a U shape separation behavior of the human fingers.

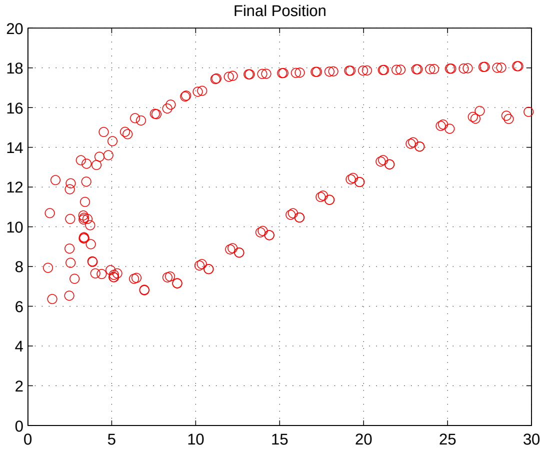
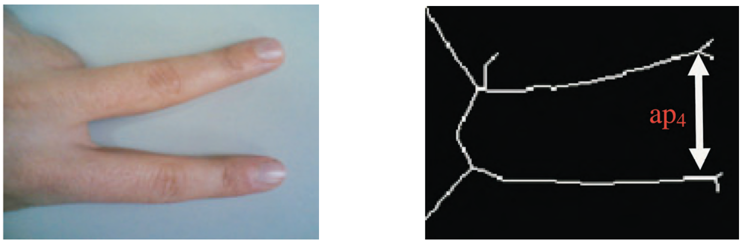


Figure 3.14: Pinching hand preshape with particles



(a) Test Image: Separation of two finges (b) Skeleton of separation of fingers

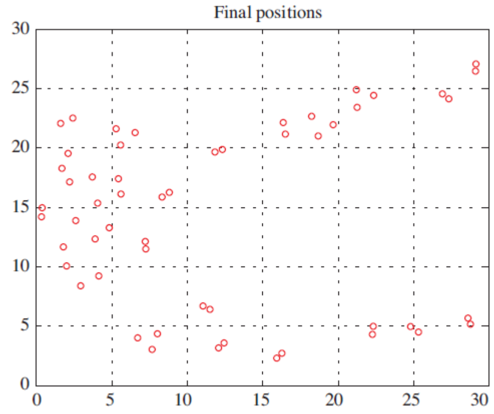
Figure 3.15: Imitation: Separation of fingers

For this hand preshape, using the previous approach developed in section 3.2, the generated feature vector is $[\,0\ 1\ 0\ 0\ 0\ {}_{ap_4}\,]^T$. As we understand from this feature vector, there is one U-shape finger posture, and the aperture between two finger tips is demonstrated by ap_4.

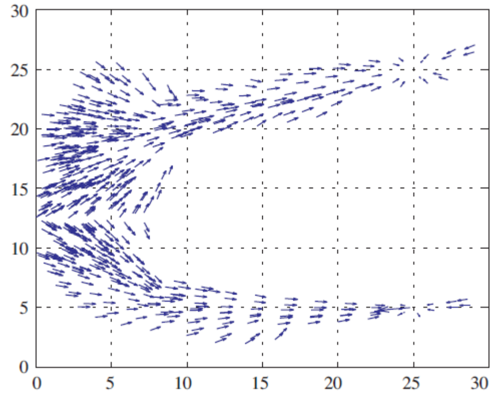
Particle distribution imitating the hand preshape is given in Figure 3.16-a together with the particle force distributions 3.16-b.

In another imitation test, pinching like hand preshape is used given in Figure 3.18-a and the skeleton of this hand preshape is given in Figure 3.18-b.

The force distribution that corresponds to the scissor like behavior clearly shows a



(a) Particle distribution for separation



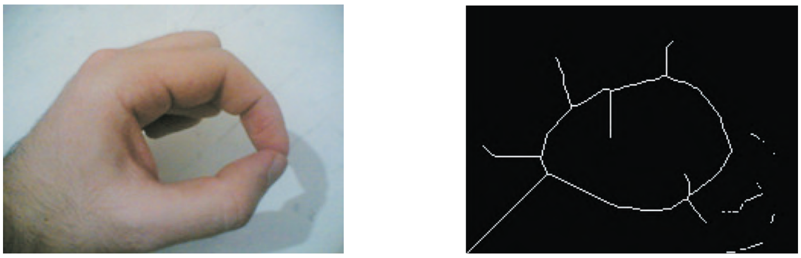
(b) Force distribution for separation

Figure 3.16: Imitation of separation of fingers

close imitation of the skeleton of the captured image, imitating the basic feature of the hand posture.

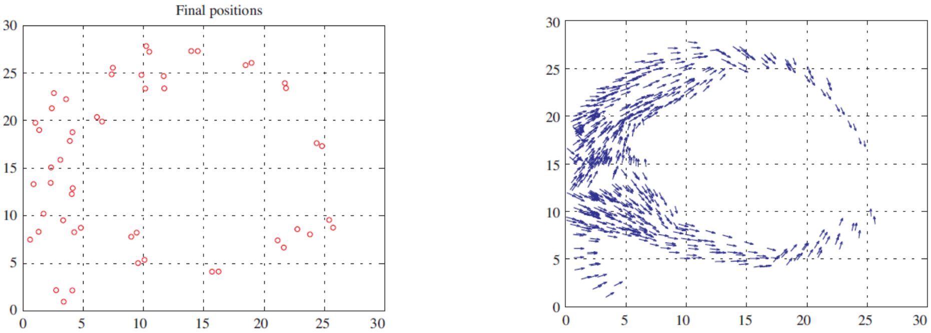
For a pinching example, its hand preshape, and skeletonized hand preshape are given in Figure 3.17a-b, particle and force distribution figures are given in figure below respectively Figure 3.18-a, and 3.18-b, the desired imitation should lead to separation of particles in two opposing curve like feature converting to a point at each tip of the curves.

The force distribution of the particles found for the pinching hand preshape shows a



(a) Test Image: Pinching hand preshape (b) Skeleton of separation of fingers

Figure 3.17: Pinching hand posture



(a) Particle distribution for pinching (b) Force distribution for pinching hand preshape

Figure 3.18: Imitation of pinching hand preshape

close resemblance to feature of the hand posture where an aperture error only occurs. This error is easily minimized if more particles are injected in the swarm colony.

CHAPTER 4

HUMAN HAND PRESHAPE CLASSIFICATION WITH PRINCIPAL COMPONENT ANALYSIS FOR IMITATION

4.1 Principal Component Analysis

In pattern recognition, the performance of a system heavily depends on the choice of the classifier, grouping feature vectors. And for this classification, efficiency is critically based on the feature extraction process for a good recognition process. A feature extraction method usually starts with a given set of features and then attempts to derive an optimal subset of features while maintaining the classification accuracy. It means that the process of feature selection should involve the derivation of salient features while reducing the redundant information of data and providing enhanced discriminatory power [47].

Among various feature extraction approaches for feature selection and dimensionality reduction, principal component analysis (PCA) is an effective method of linear dimensional reduction, which seeks to find the orthonormal transformation to maximize the scatter of all the samples, i.e., to generate a set of orthonormal basis vectors and use the projection onto the bases to represent the data.

Because of its simplicity in theory and implementation, besides decreasing redundancy information and computational burden, it is widely used in object recognition or classification. However, due to utilizing only the global information of images, it is inadequate for PCA to describe the complex nonlinear variations of real images, such as illumination, distortion. To overcome such ineffectiveness, many kernel-based methods have been proposed to describe nonlinear relations of input data during the last decade. The idea of the kernel approaches is to map the input data into a higher-dimensional implicit feature space in a nonlinear way and then use linear techniques to processing the new data.

Geometrical interpretation of PCA method can be summarized as follows:

- PCA projects the data along the directions where the data varies the most.
- These directions are determined by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues.
- The magnitude of the eigenvalues corresponds to the variance of the data along the directions of eigenvectors.

Before giving the mathematical bases of the PCA method, we will first provide and intuitive explanation. We assume that now we have a multi dimensional data set. In our case, multi dimensional data set is composed of human hand preshape images. Traditionally, principal component analysis is performed on symmetric covariance matrices or on symmetric correlation matrices. These matrices can be calculated from the data matrix. The covariance matrix contains scaled sums of squares and cross products. A correlation matrix is like a covariance matrix but first the variables, in the columns, are standardized. Thus we will have to standardize the data first if the variances of variables differ much, or if the units of measurement of the variables differ. One them solves for the eigenvalues and eigenvectors of a square symmetric matrix with sums of squares and cross products. The eigenvector associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component. The sum of the eigenvalues equals the trace of the square matrix and the maximum number of eigenvectors equals the number of rows (or columns) of this matrix.

4.2 Related Works

Principal Component Analysis (PCA) is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. This technique is generally used for human face recognition. Besides face recognition problem, PCA is applied to different problems such as human body and gesture recognition, face expression recognition, classification of unknown objects etc.

Since PCA is a method that reduces the dimension of the data by performing a covariance analysis, this technique is useful, when there is some redundancy in the data set, where, redundancy means that some of the variables are correlated with one another, possibly because they are measuring the same construct. Minimizing this redundancy is to reduce the observed variables into a smaller number of principal components (artificial variables) that will account for most of the variance in the observed variables.

Face recognition has attracted research community during the last few decades as it is the most common visual pattern in our environment. Significant development in this area has facilitated emergence of a wide range of recognition systems for commercial and law enforcement applications. Typical applications include driver's license, passports, voter registration card, human-computer interaction, database security, law enforcement, and virtual reality. Developing a computational model for human face recognition is quite difficult, because faces are complex, multidimensional, and meaningful visual stimuli. The PCA method is widely used for detection and identification of human faces [48]-[52].

Turk et. all. [53] introduced the eigenfaces which are the small set of characteristic feature images. These eigenfaces are the principal components of the training set of human face images. Recognition process is performed by projecting test image into the subspace whose bases are the eigenfaces and the classification of human face images is done by comparing the position of the test image with the position of known images.

In [54], Tzimiropoulos et al. employed the PCA for face recognition, but instead of making the operation on pixel intensities in the images, they replaced pixel intensities with gradient orientation. The proposed approach provides a consistent way to measure the image dissimilarity which is then used for face classification.

Another human face recognition method is developed recently by A.A. Mohammed et al. [55]. Their approach is based on bidirectional two dimensional principal Component Analysis (B2DPCA) and extreme learning machine (ELM). The dimension of the feature vectors is reduced by using B2DPCA. And these feature vectors are input to an ELM to analytically learn an optimal model. Experimental results demonstrated that the proposed method achieves improved recognition at a substantially faster rate against existing techniques.

PCA is not only used for the face recognition or pattern recognition problem in the literature. With the help of the abovementioned benefits about dimension reduction, PCA is also used in different areas. In [56], human grasping behaviors are analyzed by means of PCA. The aim of that work is to obtain a space with lower dimensionality from the most important degrees of freedom (DoF) involved in the human grasping behavior. By the help of this analysis it is easier to decide which DoF is the most important to control simplified hand models. As experiments, 200 samples of human hand model with 24 DoF are used. A significant simplification is done by PCA, for the obtain 5 principal components of each sample.

In another work [57] a distance kernel PCA used for object recognition problem is proposed. The distance kernel is a kernel function which sets up a distance based correspondence relation between the higher-dimensional feature space and the input space. In the validation of this work, two data sets are employed: one comprises of real face images with complex variations by means of illumination, pose and expression, and the other one is made of object images collected by robot camera containing slight variations of translation, rotations, scale and other imaging parameters.

Another dimension reduction approach with PCA is used for determining good grasping points from images [58]. The author's approach learns to identify good grasping points in the image-space of a novel object by computing a high dimensional feature vector for every pixel in the image, and applying logistic regression for classification. Since the dimension reduction with PCA decreases the number of feature vectors which need to be computed for classifying whether the given pixels in the image is associated with a good or bad grasping point, it significantly improves the speed while retaining classification accuracy.

Obtaining a realistic hand model which is suitable for manipulation tasks is an important issue. The hand model which is defined by Cabos et. all. [59] has 24 DoF and is based upon a morphological, physiological and anatomical study of the human hand. The hand model is used for developing a gesture recognition procedure which uses principal component analysis. By means of PCA, it is possible to identify how many variables are required to represent the information of the 24 DoF. Therefore, PCA analysis was used to identify the effective DoF more precisely.

4.3 Mathematical Background of PCA

PCA is mathematically defined as an orthogonal linear transformation that transforms the data into a new coordinate system such that the largest variance of any projection of the data lies on the first coordinate (called the first principal component), the second largest variance on the second coordinate, and so on.

In the first step of PCA, the training data set is generated. Since we employed this method for classification of human hand preshapes, in the training set we have human hand gestures where gestures are sequences of hand presahpes defined by horizontal and vertical line edges of fingers, separation of fingers, 1-line with 1 curvature hand preshape (semi pinch posture), and 2 curves hand preshape (pinching posture). The training data matrix (S) is composed of vectors of training images in each of its columns as:

$$S = \begin{bmatrix} img_1 & img_2 & \cdots & img_M \end{bmatrix}$$

where img_i corresponds to each image in the training data. M is the number of images in the training set.

After we obtain the training set, we need to calculate the mean image as:

$$\psi = \frac{1}{M}\sum_{n=1}^{M} img_i$$

Taking the difference between the input image and mean image, we determine the "difference image". Mean subtraction is necessary for performing PCA to ensure that the first principal component describes the direction of maximum variance. If mean subtraction is not performed, the first principal component might instead correspond more or less to the mean of the data. A mean of zero is needed for finding a basis that minimizes the mean square error of the approximation of the data.

$$\phi_i = img_i - \psi$$

In the next step, we need to find a set of M orthonormal vectors, u_n, which best describes the distribution of the training data set. The k^{th} vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T\phi_n\right)^2$$

is maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{if l=k} \\ 0 & \text{otherwise} \end{cases}$$

where u_k and λ_k are respectively the eigenvectors and eigenvalues of the Covariance matrix, C which is defined as.

$$C = \frac{1}{M}\sum_{n=1}^{M}\left(\phi_n\phi_n^T\right) = AA^T$$

where

$$A = \begin{bmatrix}\phi_1 & \phi_2 & \cdots & \phi_n\end{bmatrix}$$

and ϕ_i is the difference image.

Since this technique is generally used for human face recognition or face expression recognition problems in image processing, the set of images generated by using Principal Component Analysis is called "eigenfaces". Recently, this algorithm is also used for handwriting analysis, lip reading, voice recognition, sign language, hand gesture interpretation, and medical imaging analysis. Therefore, instead of the term "eigenfaces", we can call it "eigenimages". The eigenvectors of the covariance matrix C are the eigenimages of the training data set. These eigenimages are used for the recognition process. The recognition process based on the minimum distance from the projected data and training sets.

A new test image Γ is transformed into its eigenimage components. First the comparison between test image and mean image is performed. Multiplication of this difference with eigenvector of covariance matrix gives the weight values for the test image.

$$\omega_k = u_k^T(\Gamma - \psi)$$

And the Ω matrix contains the weight values of test images.

$$\Omega^T = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_m \end{bmatrix}$$

Determination of the class of the test image is done by minimization of the Euclidean Distance.

$$\varepsilon_k = ||\Omega - \Omega_k||$$

The class which has the minimum Euclidean distance is labeled as the image in that class.

4.4 Hand Preshape Classification

PCA is performed to reduce the dimension of the data set from higher dimensional space to lower dimensional space for hand preshape images and then, for decision mechanism, the nearest neighbor strategy is used.

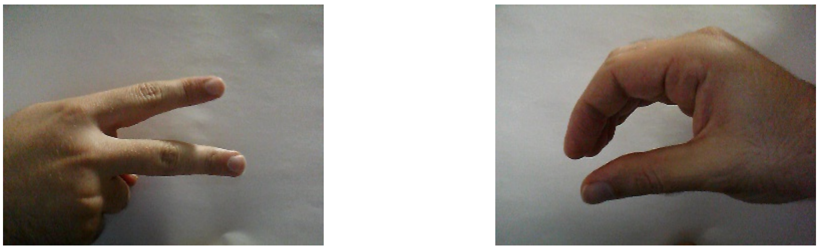
Our aim here is to classify the human hand preshapes by using the PCA technique. In the previous chapter, this classification is done by Artificial Neural Networks. The input-set of the neural network was the feature vectors of the skeleton of the human hand gestures. These input sets (feature vectors) are based on the shapes of the skeletons like U-shape, V-shape, curvatures etc. and generated by the user in a binary manner. By the help of the PCA we do not need to extract a feature vector based on the shapes of the human hand gestures. The feature vector is composed of eigenvectors of the covariance matrix starting from the most significant to the least significant.

To demonstrate the validation of the PCA algorithm that we used, we generated our training data set from the human hand preshape images. In the training data set we have 5 different human hand preshapes classes (horizontal, vertical, separation, 2 curvatures, and 1 line with curvature). For each hand preshape class, we used 10 different human hand images. The difference of the hand images are for separation the distance between fingers, for 1 line with curvature and 2 curvatures the aperture distance between finger tips, for horizontal and vertical finger shapes the thickness of the fingers. All the images are taken under the white background, same illumination conditions, and same image size.

The training data set is composed of two distinct persons hand preshapes. Some of the example training hand images are shown in Figure 4.1. We take 50 hand images for each person's hand, and used 40 of them for training and the rest images are used for testing. So, in the training data set, we have 80 hand images and in the test data set 20 hand images.

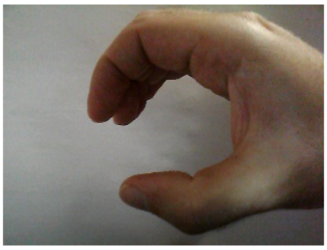
(a) Horizontal 2 fingers pointing

(b) Vertical 2 fingers pointing



(c) Scissor like hand preshape

(d) Hook preshape



(e) Cylindrical preshape

Figure 4.1: Examples of hand preshape classes

The algorithm for classification of the human hand preshapes is given in Figure 4.2. The modules in the flow chart are subsequently explained.

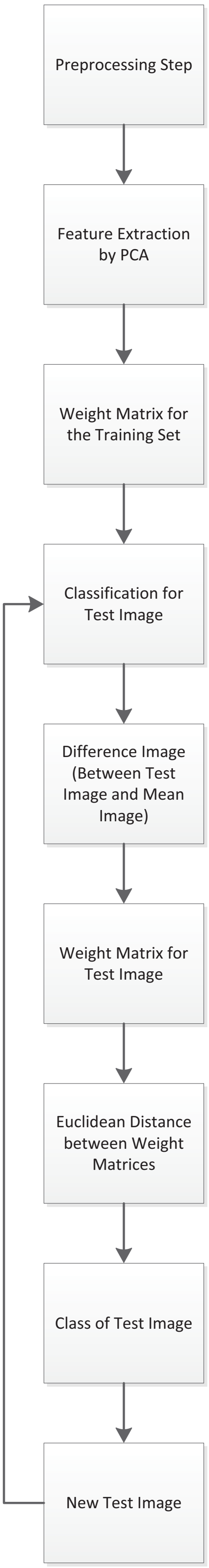

Preprocessing Step
Feature Extraction by PCA
Weight Matrix for the Training Set
Classification for Test Image
Difference Image (Between Test Image and Mean Image)
Weight Matrix for Test Image
Euclidean Distance between Weight Matrices
Class of Test Image
New Test Image

Figure 4.2: Flowchart of the PCA for classification

4.4.1 Preprocessing Module

In this module, the training images are collected in a matrix S. Each column of the matrix corresponds to a training image. The pixel resolution of each image in the training set is 240×320. Those images are taken under the same lighting condition.

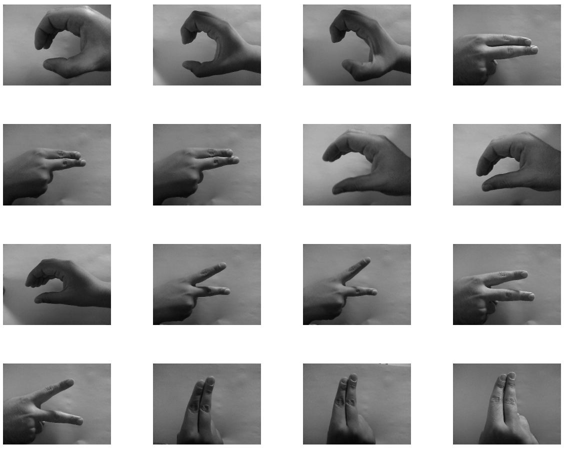
$$S = \begin{bmatrix} img_1 & img_2 & \cdots & img_M \end{bmatrix}$$

Image normalization to reduce the error because of the lighting condition is also done in the preprocessing step. All the training images in the training set are normalized. After that a mean image M is obtained from the training data set. An example of the training set and normalized version of this training set can be seen in Figure 4.3 - a and b.

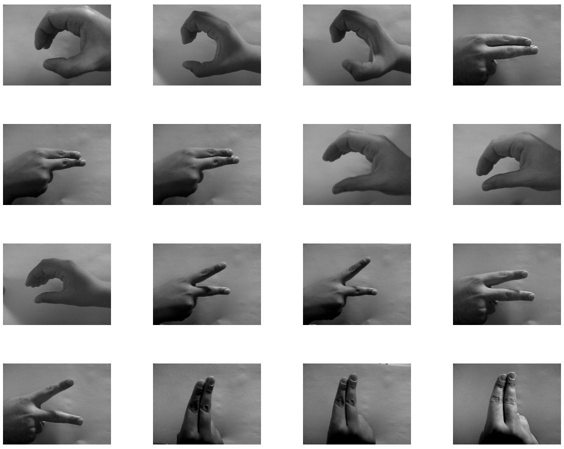
4.4.2 PCA Module

In this module, first the covariance matrix, C, for the training data set,S, is calculated. After that, the eigenvectors and eigenvalues of the covariance matrix is determined. The eigenvectors associated with the highest eigenvalues are taken as the feature vectors.

The rest of the eigenvectors are neglected. Since this step is computationally heavy, only some of the eigenvectors are taken as the feature vector. These eigenvectors called as eigenimages and some of the examples of the eigenimages for different classes can be seen in Figure 4.4. They are used for the classification step for comparison.

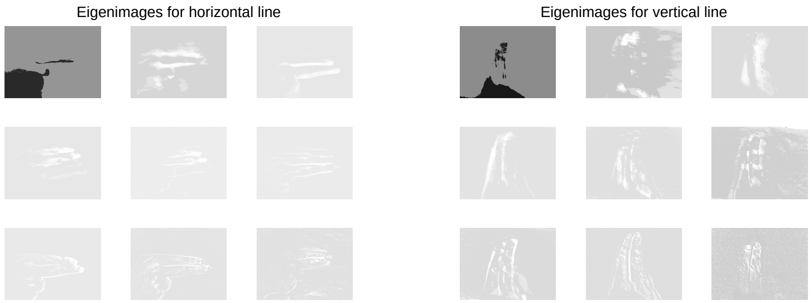


(a) Training set

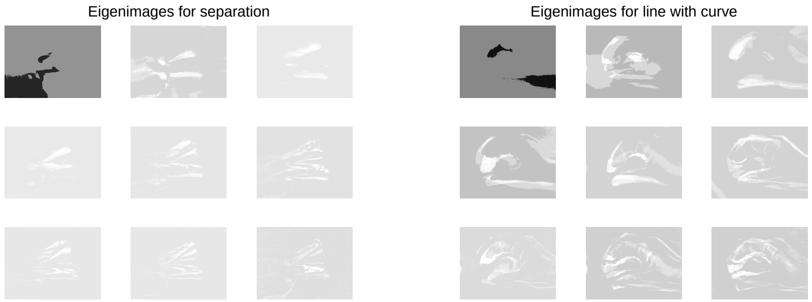


(b) Normalized training set

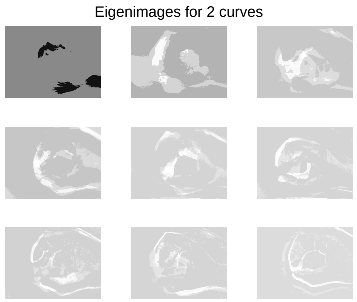
Figure 4.3: Example of training set


Eigenimages for horizontal line
Eigenimages for vertical line

(a) Horizontal 2 fingers pointing (b) Vertical 2 fingers pointing



(c) Scissor like preshape (d) Hook preshape



(e) Two curvatures

Figure 4.4: Cylindrical preshape

Second, the weight matrix is found for each image by simply taking the dot product of the images in the training set and eigenimages (i.e. feature vectors).

4.4.3 Test Module

For testing the algorithm, a new test image is used. For test images, the steps which are in the preprocessing step are repeated. First the test image is converted into a single vector, and then normalized version of test image is calculated. After that, difference image is constructed by taking the difference between input image and the mean image of the training set. In the second step, weight matrix for test image is calculated by taking the dot product of difference image and eigenimages.

4.4.4 Classification

This step is the decision making step, based on the Euclidean distance between the weight matrix of the training data set and weight matrix of the test image. The class name which has the minimum euclidean distance is given as the label of that test image.

An example of the test image is shown in Figure 4.5 below. When we compare the test image with the training image set, some similar images can be found but not an identical one. Our algorithm determines the most similar image which is in the training data set.

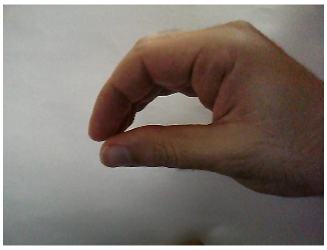


Figure 4.5: Test image: closed hook

As discussed previously, for decision of the hand preshape, we have to look the Euclidean distances between test image and the eigenimages. Euclidean distance of the test image is shown in Figure 4.6. As we can see from Figure 4.6, the minimum euclidean distance is for 8^{th} image in the training set. When we compare the test image with the 8^{th} image from the training set, we see that it is the most similar image to the test image. The most similar image to the test image and the least similar image (5th image in the training set) according to the graphic given in Figure 4.6 are given in Figure 4.7

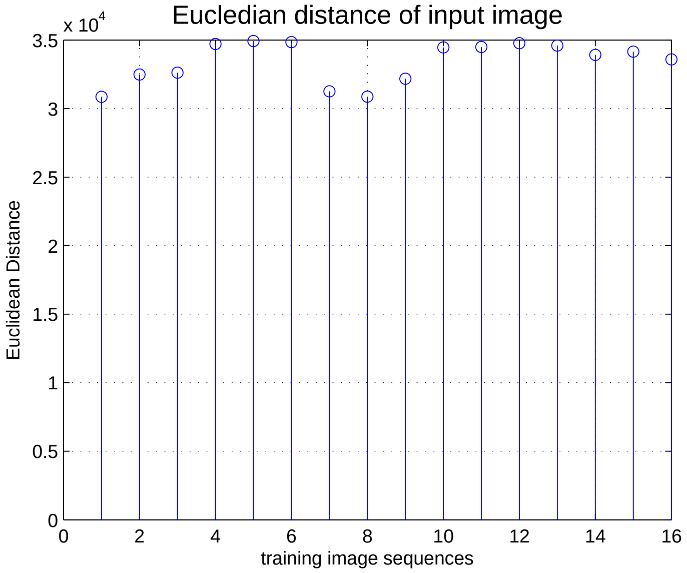
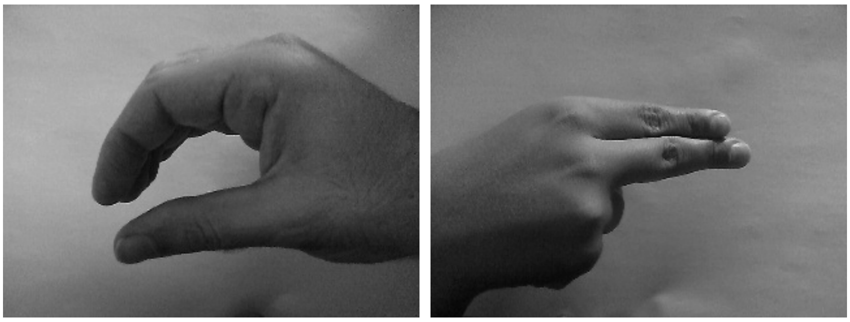


Figure 4.6: Euclidean distances for the test image



(a) Minimum Euclidean Distance Image (b) Maximum Euclidean Distance Image

Figure 4.7: Most and Least Similar Images

4.5 Statistical Analysis of Principal Components for Classification

In this section, we demonstrate the results of statistical analysis of principal components which are used for classification. This statistical analysis is independent from the previous classification test. In this experiment, we used 50 hand preshape images which belong to 5 different classes given in previous section. The training data set is constructed from hand images which are collected from two distinct persons with 25 images per person. The test images, which are used for demonstration of the importance of the number of principal components, are taken one day later from the training data set. In the test images, again the same 2 persons are used. The image size both in the training data set and test data set is set to 240 × 320.

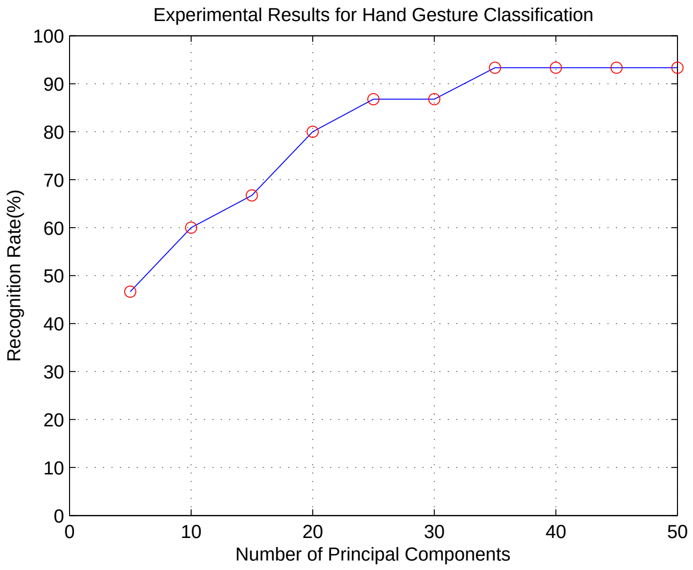


Figure 4.8: Recognition rate for classification

The experimental results are shown in Figure 4.8 and tables 4.1 and 4.2. Figure 4.8 shows the classification accuracy of human hand preshapes based on the number of principal components that are used for classification. The recognition rate is improved with the increase of the number of principal components. After the usage of 70% of the principal components, the wrong classified number of image is only 1. From Figure 4.8, we can also see that the best classification performance (%93.33) is attained with at least 35 principal components.

Table 4.1 shows the recognition rate of the test data implicitly for some principal components.

The results of the average consumed training time, testing time and total time are shown in Table 4.2. In the recognition process, the computing time is an important performance measure to evaluate the recognition system's performance. It includes both the training time and the testing time. Another important performance measure is recognition accuracy.

Table 4.1: PCA Recognition Rate

# of Principal Components	# Wrong Classified	# Correct Classified	Recognition Rate(%)
5	8	7	46.67
15	5	10	66.67
30	2	13	86.67
35	1	14	93.33
50	1	14	93.33

Table 4.2: Average Consumed Time for Training and Testing

# of Principal Components	Training Time	Testing Time	Total Time (sec)
5	9.91	0.49	10.40
15	10.92	0.51	11.43
30	13.00	0.52	13.52
50	16.12	0.64	16.76

Finally, by analyzing these performance criteria, we can conclude that the performance of the PCA for hand gesture classification is much more efficient than the previously discussed Artificial Neural Network based classification: higher recognition accuracy and less feature extraction time.

4.6 Imitation of Hand Gestures with Stiffness Coefficient

In the previous imitations of human hand gestures, as the output of the controller we used body force term. The body force term directly enters the equation of motion which can be seen in (3.7) and (3.8) and this term has a direct effect on the flow of the fluid body. Therefore it is suitable to use this term for guiding the motion of particles toward desired formation.

Since the solutions of these equations of motions or in other words momentum equations give the accelerations of each particle along x and y directions, each terms in these equations have important effect on the formation generation. The first term on the right-hand-side (RHS) of the momentum equation has an effect on particles due to pressure gradient along with the dissipative artificial viscosity in the specified direction. Because of the minus sign in front of this term, increasing the stiffness coefficient causes decreasing the velocity and it also provides particle directions along perpendicular direction. It means that this term slows down the particles. When we consider this term, there are two pressure components, one is for the particle itself and the other one is for the neighboring particle. We know that the pressure component is related with the stiffness coefficient β from equation 3.13.

In Chapter 3, in the training part of the Artificial Neural Network, we experienced the importance of the body forces. When we applied constant body force along x direction, we saw that particles move on the surface without showing any branching effect. In order to generate branching of the particles to mimic the separation of fingers, we changed the applied body forces at moments suitable for branching (Chapter 3). Now we want to show this branching effect without using any body forces. In this implementation, we employ the stiffness coefficient instead of the body forces. We applied a constant velocity along the x-direction to all the particles. This time the stiffness coefficient is not same for all particles. For showing the branching effect on the particles, we need to slow down the middle particles and the rest will move faster that the middle ones, so the fluid body has a formation that undergoes a division into two branches.

The initial fluid particle distribution is given in Figure 4.9. For mimicking scissor like hand gesture, we have to achieve this branching in the fluid body. During the simulation, if we slow down the middle particles while they are moving along the x axis, the particles which are above and under the middle particles move faster and these particles will construct the two branches.

When we applied the stiffness coefficient to the middle particles much larger than the edge particles, the middle particles move slowly and the edge particles generate the scissor like fingering effect on this fluid body. Specifically we choose the stiffness coefficient for this formation, for middle particles as;

$$\beta_{middle} = 2$$

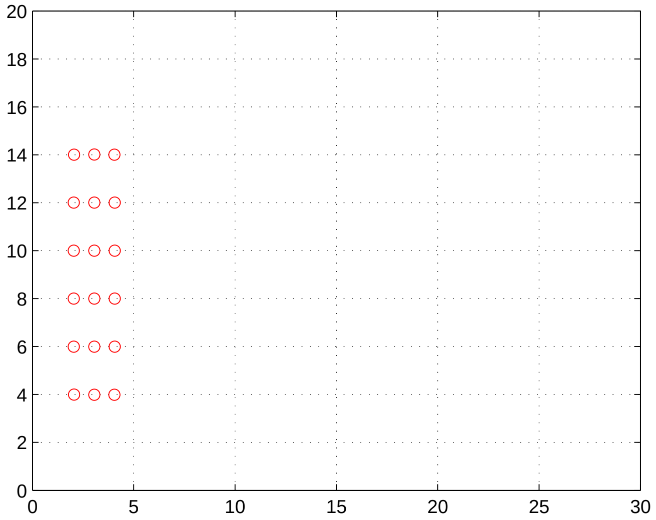
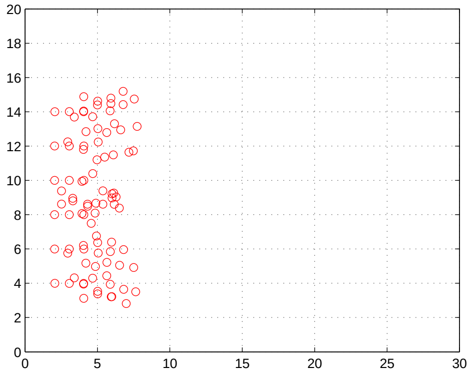


Figure 4.9: Initial particle distribution

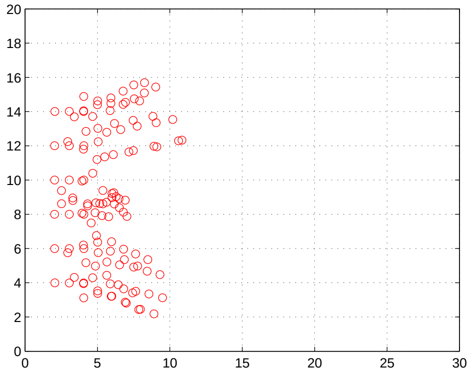
and for the edge particles we set this coefficient to

$$\beta_{edge} = 20.$$

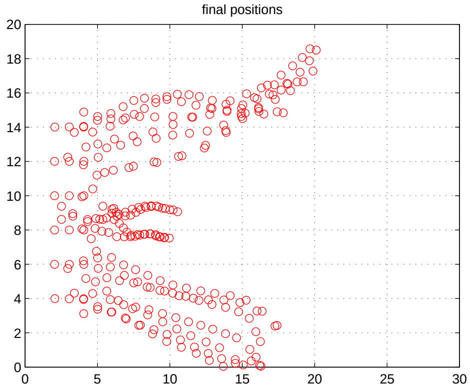
The result of this effect can be seen in the Figures 4.10 given below . We take the snapshots to see the effect of the stiffness coefficient during the simulation in the following figure when $t = 80$, $t = 120$, and at the end of the simulation.



(a) $t = 80$



(b) $t = 120$



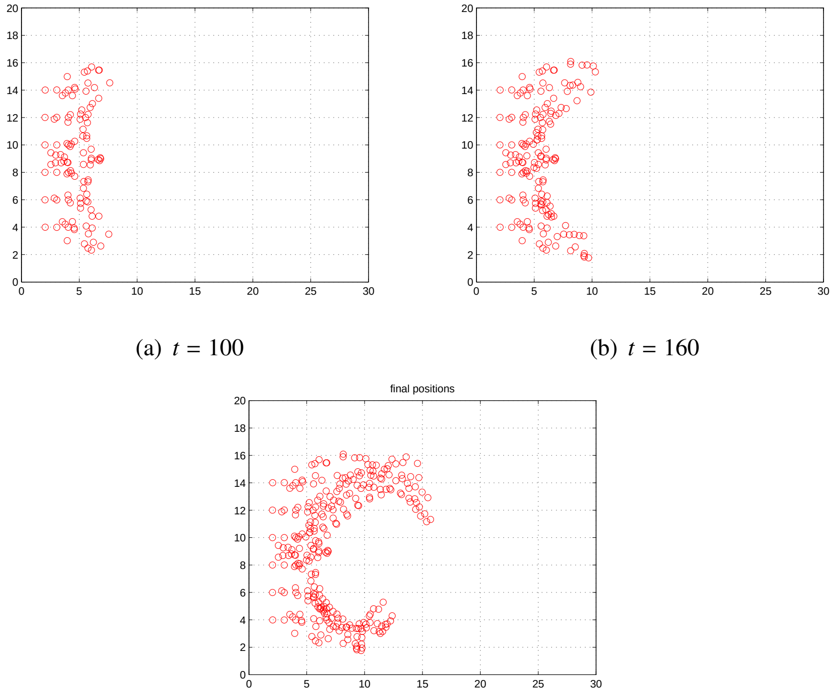
(c) $t = 300$ end of the simulation

Figure 4.10: Separation of particles with stiffness coefficient

As we can see from the simulation results, in Figure 4.10-c, the scissor like fingers gesture is imitated using only the stiffness coefficient. The cause is mainly the slowing down of the middle particles while the faster edge particles generate the fingering effect.

After observing the effect of the stiffness coefficient, this time the value of this coefficient is set to numerically $\beta = 50$. Again we applied constant velocity along the x direction with the same initial particle distribution, since we want a motion on the x direction. Our aim is to increase the stiffness in the x direction and make the particles escape to other directions, namely the y direction.

The result of this implementation can be seen in Figure 4.11. The snapshots are taken from $t = 100$, $t = 160$, and from the end of simulation ($t = 300$).



(a) $t = 100$ (b) $t = 160$

(c) $t = 300$ end of the simulation

Figure 4.11: Curving of two fingers with stiffness coefficient

The simulation results showed us that, by increasing the stiffness coefficient for all particles, we make the particles escape along y direction and in this way we obtained the cylindrical preshape alike a corresponding human hand gesture.

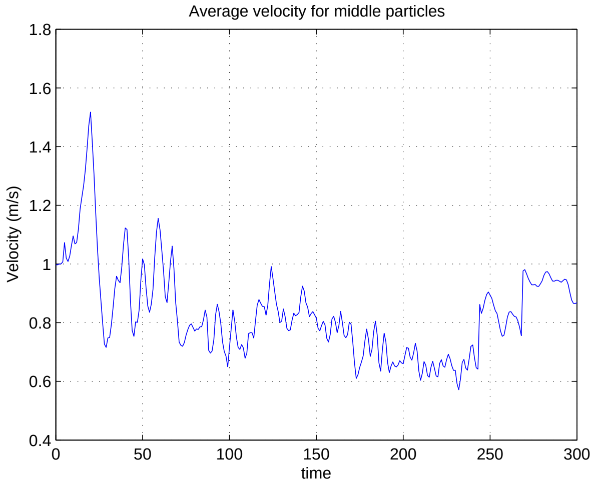
4.7 Sensitivity Analysis

Although there are some widely used fluid simulation platforms such as Fluent from ANSYS cooperation, ANSYS CFX simulator for solving the computational fluid dynamics equations in mesh based environment etc., they are not suitable for our aim, to imitate human gestures with fluid body. They are mainly physical based fluid simulator to solve the physical problems, and visualize the simulation environment. We preferred to develop our own simulator to implement the proposed fluid dynamics model to imitate human hand gestures.

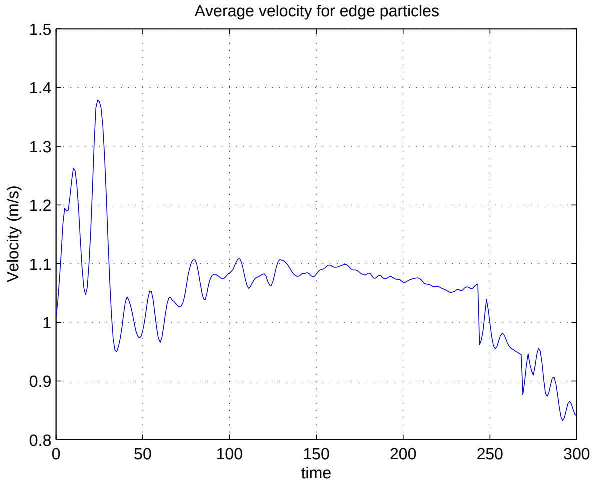
In this section, we give the results of how the other parameters change while the fluid particles imitate human hand gestures. In other words we conduct a sensitivity analysis of the fluid parameters. In the previous section we showed both scissor like shaping of fluid particles and the formation imitating cylindrical preshape using only the stiffness coefficient β. In this section first we demonstrate the average velocity of particles during the separation simulation.

Figure 4.12-a and 4.12-b are the average velocities of middle particles and edge particles. As we expected, the average velocity of the middle particles are less than the edge particles. Numerically the average velocity for middle particles is $V_{middle} = 0.832$m/s and for edge particles it is $V_{edge} = 1.052$m/s.

When we compare both graphics, we see that at the end of the simulation (exactly starting from time $t = 250$) the average velocity for edge particles has a sudden decrease although this decrease cannot be seen in middle particles. The reason of this sudden decrease is when the fluid particles exceeds the environment borders, the fluid parameters are set to 0 for that particles.



(a) Average velocity for middle particles



(b) Average velocity for edge particles

Figure 4.12: Average particle velocity for separation hand gesture

For cylindrical hand gesture, we set the stiffness coefficient $\beta = 50$ for all particles. The initial velocity is $1m/s$ along x direction and 0 for y direction. Since the particle motions is along x direction and we set the stiffness coefficient to high value, the particles have a high resistance to move on x axis. So the particles escape and construct the curvature of fingers in the cylindrical hand gesture.

In Figure 4.13 average velocities of the particles are given for the imitation of the cylindrical hand gesture where 2 curvatures are fitted to the human fingers. This time average particle velocity is lower than the previous case since the higher valued stiffness coefficient is applied for all particles.

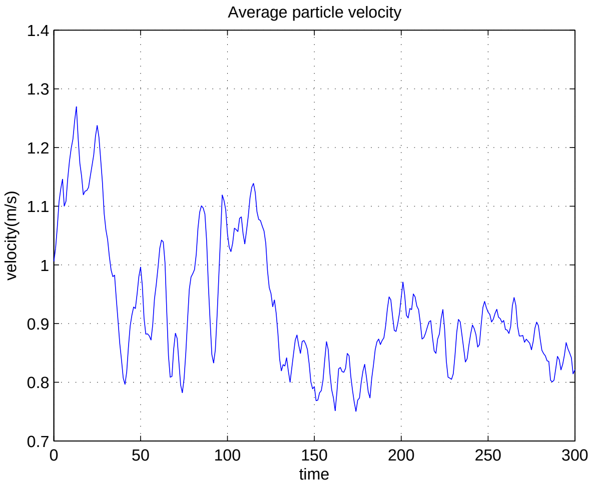


Figure 4.13: Average particle velocity for curving hand gesture

When we look at the equation of motion given in (3.7) and (3.8), another fluid parameter besides pressure gradient and body force is the physical viscosity. This parameter provides interactions and coherence among fluid particles such that when one of the particle in the system starts to move, then the rest of the fluid particles is affected by this motion and the neighboring particles also start accelerating in the same direction.

To illustrate this effect in a simulation environment, we used the following fluid swarm of Figure 4.14. We applied $(1,0)$ body force to the middle particle whose coordinate is $(5,4)$ in order to start the swarm to move (shown by a symbolic horizontal line in the figure).

After the simulation run, the middle particle in the swarm starts to move along the x direction because of the body force that we applied to that particle. The velocity field of the swarm and also the physical viscosity field distributions are given in Figure 4.15 and 4.16, respectively.

These results demonstrate that movement of one particle in the fluid swarm generates a similar motions on the surrounding particles. This similar motion is result of the physical viscosity term in the equation of motion. Physical viscosity is effective among neighboring particles which are inside the smoothing kernel such that move-

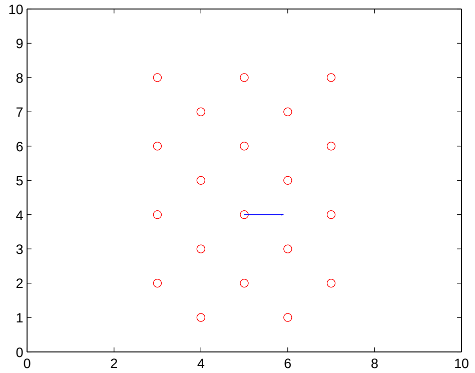


Figure 4.14: Initial particle position

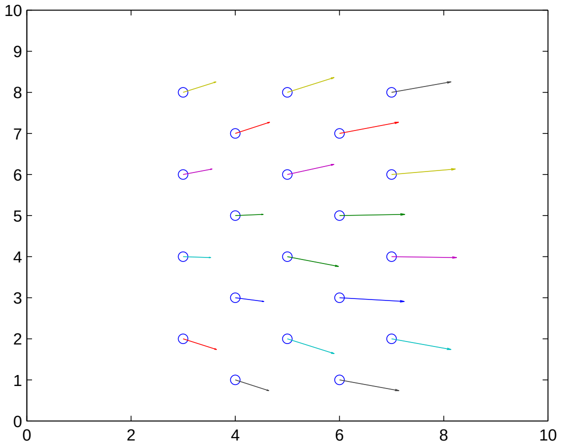


Figure 4.15: Velocity field of the particles

ment of one particle induces a similar motion on the surrounding particles. On the other hand, movement of that specific particle causes a dragging effects along the perpendicular axis. This effect can be seen in Figure 4.15, all the particles have small amount of velocity field along y direction (upward drifts), although the motion is along x direction.

Therefore, viscosity demonstrates an implicit coordination mechanism among particles which is used to generate a desired collective motion of the system such as in formation control.

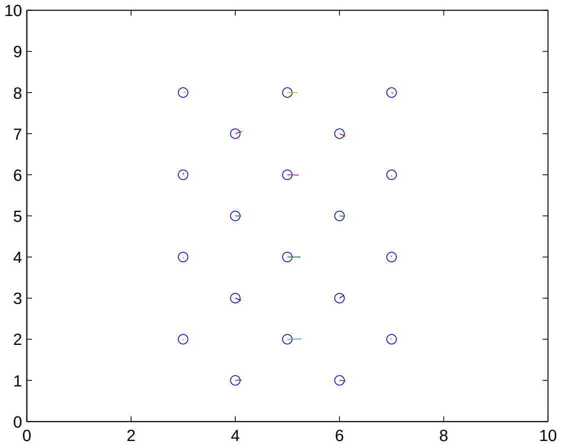


Figure 4.16: Physical viscosity field on the particles

Moreover, in the previous section, it is mentioned that if we increase the stiffness coefficient β, we start loosing the coherence between fluid particles. The following Figures 4.17 and 4.18 show that higher value of stiffness coefficient causes the loss of the particle coherence. And no directional drag can be seen in the directions of the velocity fields. In the first experiment, we applied constant stiffness coefficient $\beta = 2$ and in the second case, we applied $\beta = 50$ to the fluid particles during the simulation. From the velocity field of the particles, we see that the coherence between particles also depends on the stiffness coefficient.

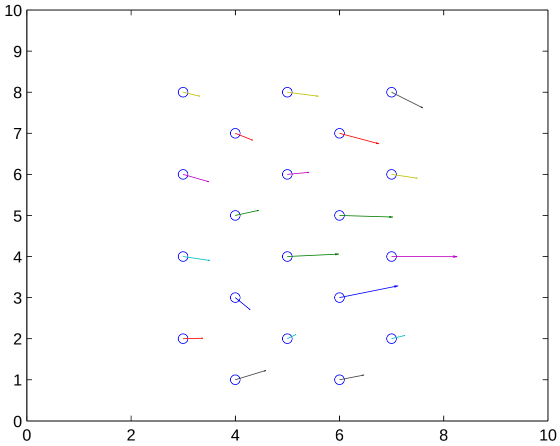


Figure 4.17: Velocity field for $\beta = 2$

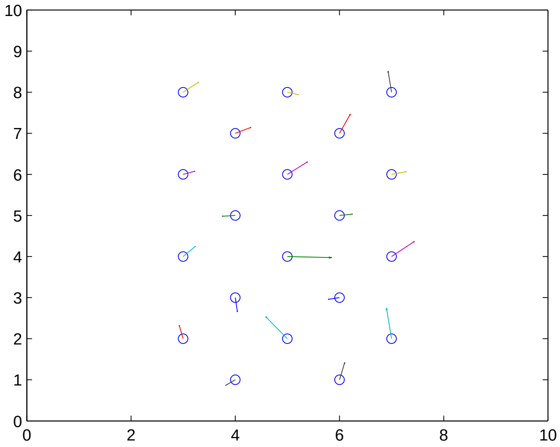


Figure 4.18: Velocity field for $\beta = 50$

CHAPTER 5

HAND GESTURE AND HUMAN BODY POSE IMITATON

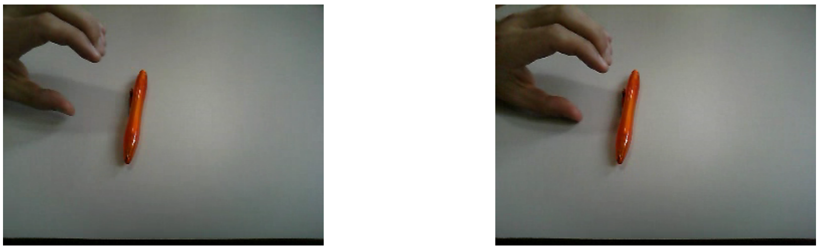
5.1 Human Hand Gesture Imitation

In the previous sections we showed the examples of imitation of human hand preshapes. As the input data set we used the single hand preshape images, and those hand preshapes are imitated by the body of fluid particles by adjusting appropriately fluid parameters. Mainly two fluid parameters are used for those examples: One is the fluid body forces and the other one is the stiffness coefficient.

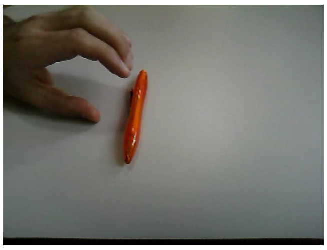
In the first part of this section we give the examples of human hand gesture imitation. In these examples we used the previously generated human hand preshapes as the training set. But this time, as the test image we use a sequence of hand gestures. We will hereby demonstrate the frame by frame imitation by a fluidic swarm of a sequence of hand gestures as well as a sequence of body poses. The hand gestures and body poses considered in the sequence is a combination of more than one basic hand preshapes that we have considered in the training sets.

5.1.1 Example-1

In the first example we combined 2 hand preshapes, namely the scissor like hand behavior, with the hook preshape. This hand gesture resembles the gesture made for grasping an object placed on the table by a 2 fingers pinching grasp and then putting the object forward on the table. The following Figures 5.1 and 5.3 explain this hand gesture clearly. In the first frames of this hand gesture (Figure 5.1), the hand gesture is scissor like then hook preshape towards the object lying on the table. These images are captured from the video of a human hand grasping an object and they are used for testing our algorithm.



(a) frame-85 (b) frame-95

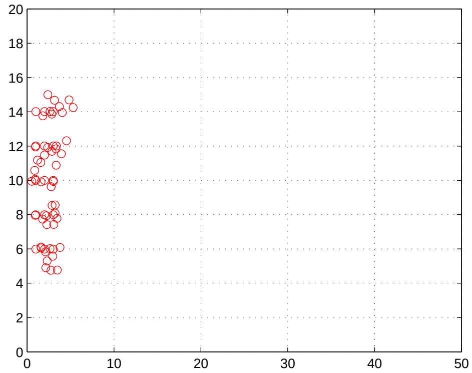


(c) frame-105

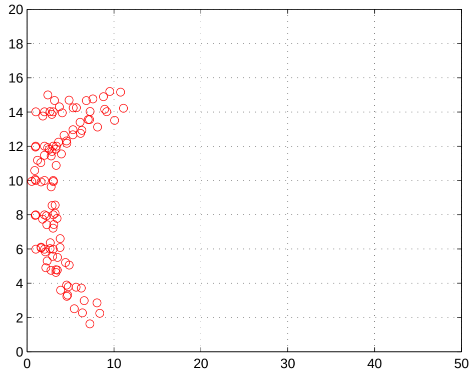
Figure 5.1: Hand preshapes combining scissor + hook basic preshapes.

The images, which are taken randomly in different frames, are tested with PCA discussed in previous chapter and the appropriate fluid parameters, namely body forces and stiffness coefficients are applied to the particles during the simulation.

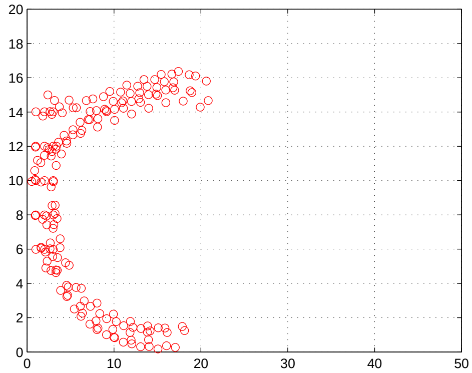
In the first part, since the test images correspond to the scissor like hand behavior, for the middle particles the stiffness coefficient of the fluid parameters are set to higher value than the edge particles. By the help of this parameter, the middle particles move slowly, and we get scissor like hand gesture. The imitation result of this hand gesture is given in Figure 5.2. It can be seen from the figure that the fluid particles start to separating in to two sub-branches.



(a) Separation of particles-1



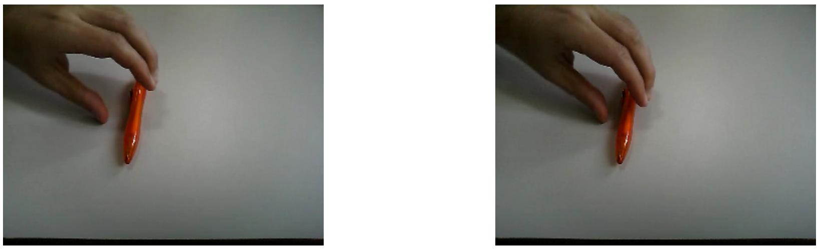
(b) Separation of particles-2



(c) Separation of particles-3

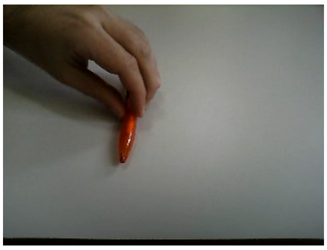
Figure 5.2: Separation of particles

After the scissor like separation of the fingers, when we get close to the object lying on the table we close the finger tips on the object. In Figure 5.3 the hand images which are taken from the testing video are given.



(a) frame-120

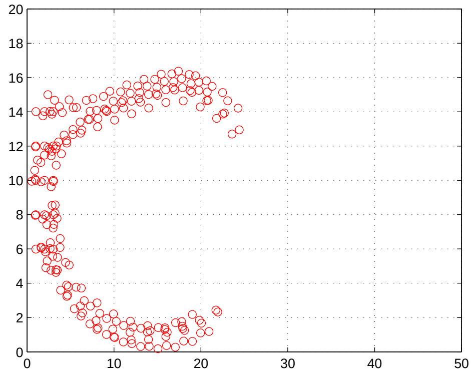
(b) frame-130



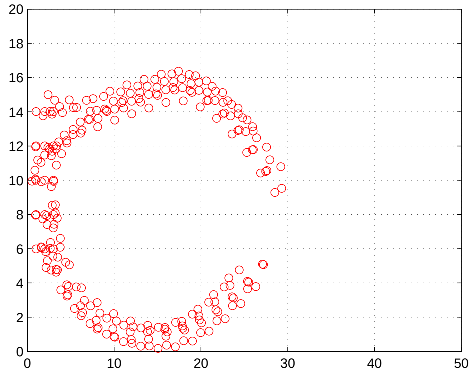
(c) frame-150

Figure 5.3: Closing of the finger tips on the object

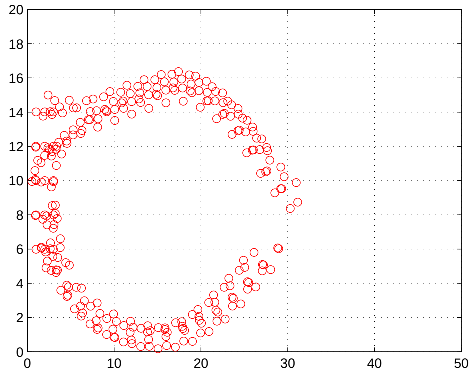
As we told previously we see that finger tips are closing on the object. These hand images again used as the testing images, according to the class of this test group, the body force term which is in the equation of motion are set to appropriate values and the fluid particles imitate this hand gesture as seen in Figure 5.4.



(a) Aggregation of particles-1



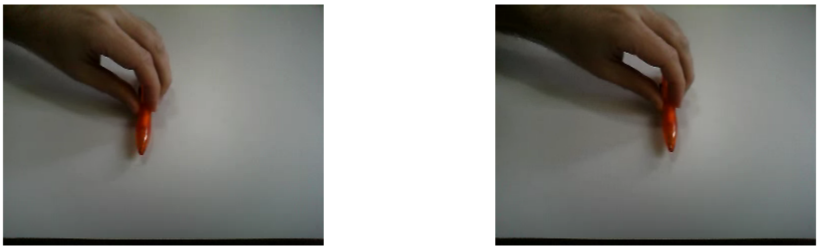
(b) Aggregation of particles-2



(c) Aggregation of particles-3

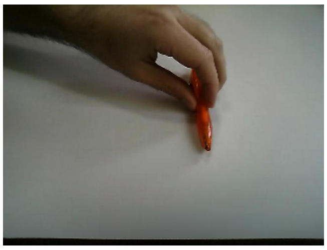
Figure 5.4: Aggregation of particles

After aggregation of the fluid particles on a point in the environment, the next testing images are given in Figure 5.5. In these images the object is grasped and it is put at a further point on the table.



(a) frame-160

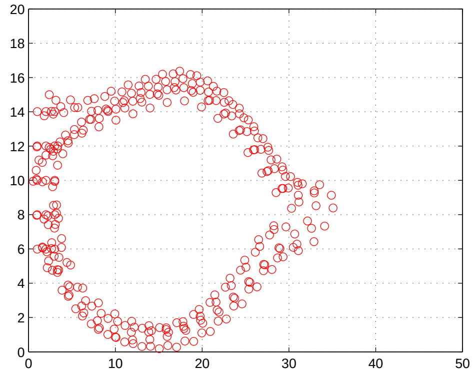
(b) frame-170



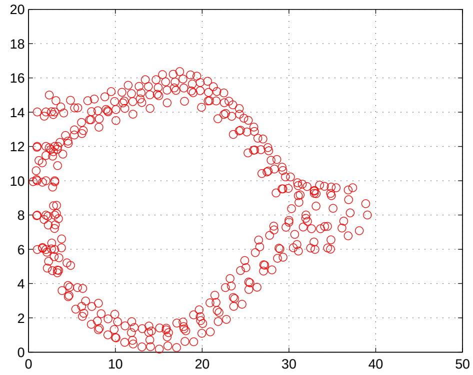
(c) frame-190

Figure 5.5: Moving forward of the object

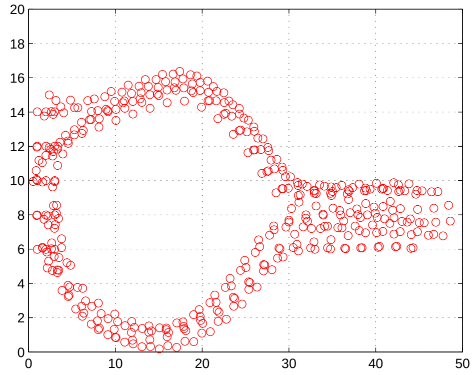
This hand gesture looks like the previously given example, hook preshape with a hand motion along the x axis like horizontal pointing of fingers. If we use the previously given fluid parameters for horizontal pointing of fingers we get the following particle positions during the simulation (Figure 5.6). After the aggregation of the particles, particles need to flow along the x axis and this motion corresponds to the movement on horizontal edge.



(a) Moving forward of particles-1



(b) Moving forward of particles-2

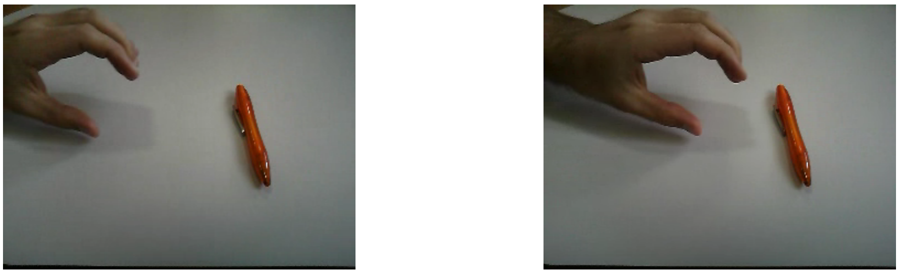


(c) Moving forward of particles-3

Figure 5.6: Moving forward of particles

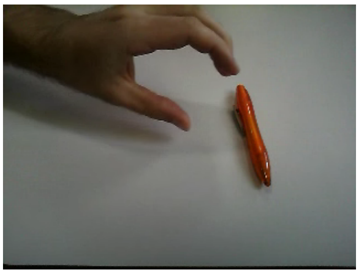
5.1.2 Example-2

In the second example, we take another imitation demonstration with fluid particles. This time, the human hand moves with the aim of grasping an object lying on the table. The grasping strategy for this purpose is the cylindrical preshape followed by a pinching like grasping of the object. When we analyze this grasp strategy we see that the first thumb and the rest of the fingers (index, middle, ring and little fingers) are separated into two branches. This hand gesture looks like the previously given cylindrical preshape. After a while when the human hand get close to the object lying on the table, the aperture between the thumb and the rest of fingers decreases. Thumb finger has a horizontal line edge preshape and the other fingers have curving like behavior and wrap around the object. This hand gesture looks like the previously given “hook” preshape.



(a) frame-80

(b) frame-100



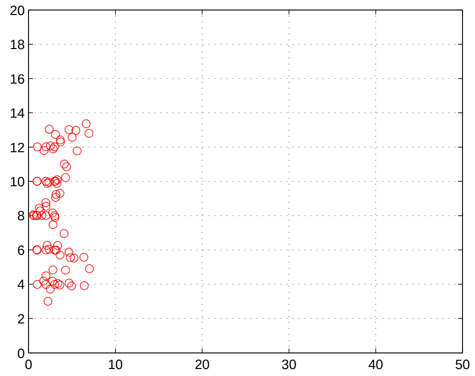
(c) frame-110

Figure 5.7: First group frames of hand gesture

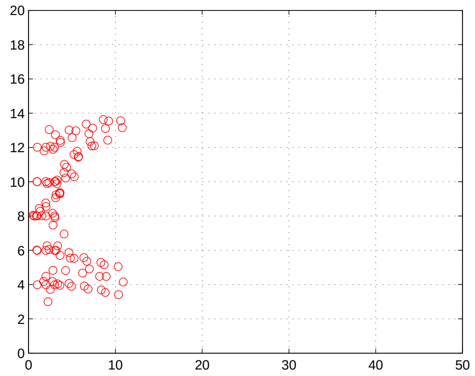
Again with the same manner as in example-1 we constructed the test images from the video input. The following figures, in Figure 5.7 and Figure 5.9, are used for the test images of our system. And according to the classes of the test images, we set the fluid parameters to imitate the human hand gesture.

In the first group of images (Figure 5.7), which are taken early frames from the video input, the fingers are starting to show the branching effect of the hook hand gesture. Human hand starts to move towards the object. There are two sub-branches in the human hand gestures and between these branches there is a certain amount of apertures. This hand gesture corresponds to the branching effect in the hook preshape. To imitate this hand gesture, we need to divide the fluid particles into two groups. In previous sections we generated this hand preshape by using the stiffness coefficient. In order to slow down the middle particles, we used higher valued stiffness coefficient for them, and for edge particles since they need to flow faster than the edge particles, we assigned lower valued stiffness coefficient. Moreover, in order to prevent penetration of the middle particles we applied constant body forces to the middle particles along $-x$ direction so we obtain empty space between the finger branches.

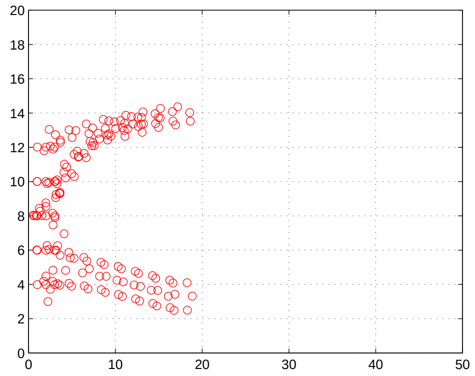
The following Figure 5.8 demonstrates the branching effect of the particles, the snap shots which are taken during the simulation. In the first snap shot, the particles have just started to separate, by the controller applying higher valued stiffness coefficient to the middle particles. Besides that by the help of body force along $-x$ direction, middle particles are taken close to the initial positions and the empty space is created between finger branches. The following snapshots demonstrates the complete separation of the fluid particles.



(a) Snapshot-1



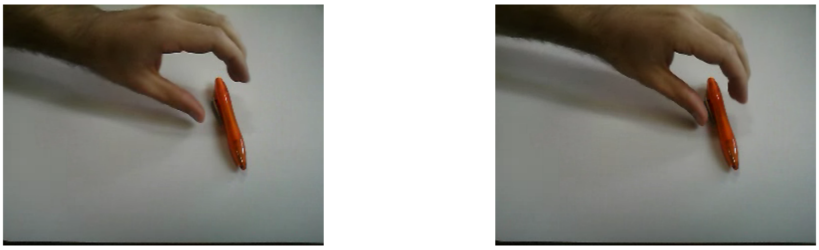
(b) Snapshot-2



(c) Snapshot-3

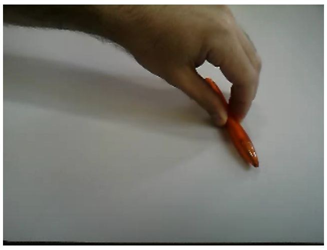
Figure 5.8: Snapshots for separation of particles

In the second part of this hand gesture, the following images which are given in Figure 5.9 are taken from the video input. These images are used for testing the proposed system. The principal component analysis based classifier decide the class name of the test image according to the minimum euclidean distance between test images and the training data sets. As we can see from the figure, when the human hand gets close to the object, the index finger (upper branch of the finger set) has a curvy feature. The thumb finger has a line edge shape. The combination of these two branches one is curvature of index finger and the other one is line edge correspond to the "hook preshape".



(a) frame-125

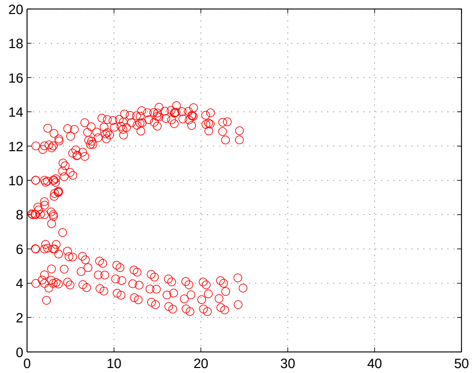
(b) frame-135



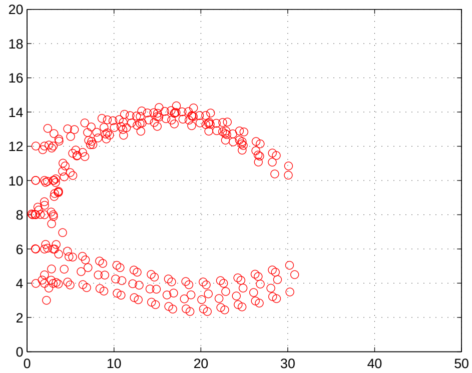
(c) frame-155

Figure 5.9: Second phase of the hand gesture

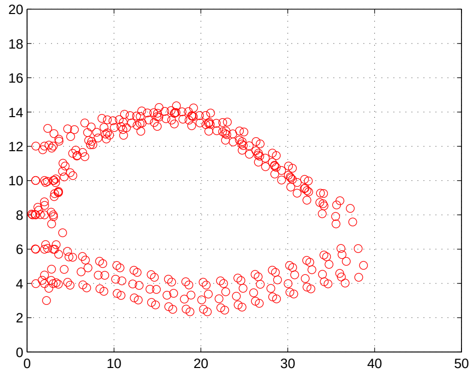
In the simulation part of the particles (Figure 5.10), to show these curving like behavior of the index finger and horizontal line edge for thumb, we again used the body forces. In here we need curvature for index finger and the tip of the index finger get close the tip of the thumb. Initially mentioned aperture distance between two branches disappears.



(a) Snapshot-1



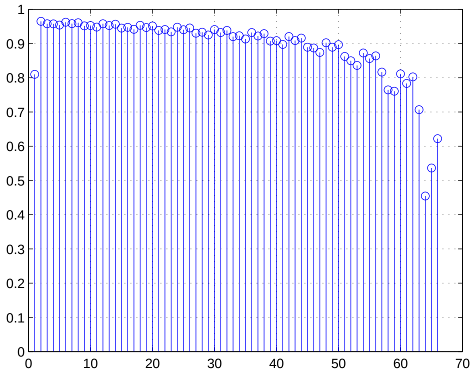
(b) Snapshot-2



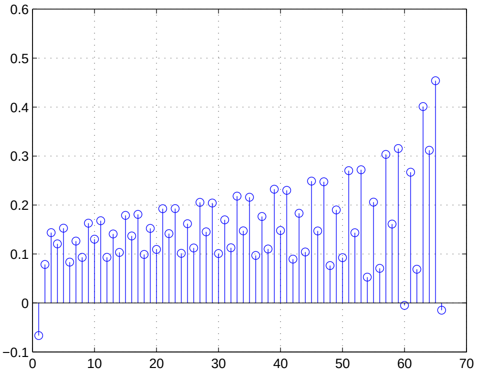
(c) Snapshot-3

Figure 5.10: Snapshots for curvature with horizontal edge with particles

For this purpose to construct directional motion of the thumb finger the controller applies constant body force along x direction, and for index finger body forces along x and $-y$ directions as given in Figures 5.11 and 5.12.



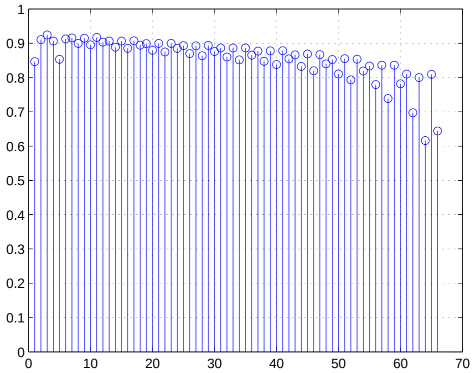
(a) Body forces along x direction



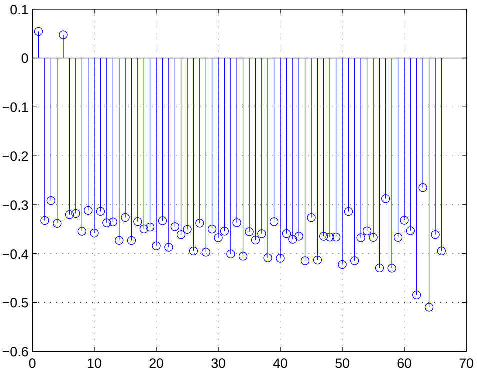
(b) Body forces along y direction

Figure 5.11: Body force distribution for line edge hand gesture

To construct the horizontal line edge for thumb finger, the applied body forces along x and y directions are given in Figure 5.11 a and b, respectively. On the other hand to construct the curving like behavior for index finger, the applied body forces to the particles along x and y directions are given in Figure 5.12 a and b, respectively.



(a) Body forces along x direction



(b) Body forces along y direction

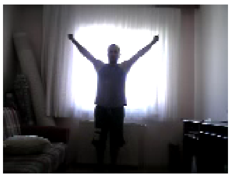
Figure 5.12: Body force distribution for curvature hand gesture

5.2 Human Body Pose Imitation

In the previous sections, we gave the examples of imitation of human hand gestures and preshapes with fluid particles by adjusting the computational fluid dynamic parameters according to the classification of the human hand gestures. In this part of the thesis, we use the human body motions as the test image to be imitated by fluid particles. The test images are taken from a video, in which a person is performing body motions. Namely these movements are standing up, bending to the right and left, opening the arms, opening the arms and moving them upside and can be classified into six different group. In Figure 5.13 examples for different classes can be seen.

(a) Standing up (b) Bending to the right

(c) Bending to the left (d) Opening the arms



(e) Opening the arms towards upside

Figure 5.13: Human body pose classes

Instead of using the whole images as both training and testing images, we tracked the human body poses and used them in our system. For detecting and tracking the human body motions from the video sequence, we used a stationary background. The steps of the procedure as follows:

1. Use the first few frames of the video to estimate the background image.
2. Separate the pixels that represent the people from the pixels that represent the background.
3. Group pixels that represent the person and calculate the appropriate bounding box for human body.

4. Match the people in the current frame with those in the previous frame by comparing the bounding boxes between frames.

The system for detecting and tracking person movements can be seen in Figure 5.14.

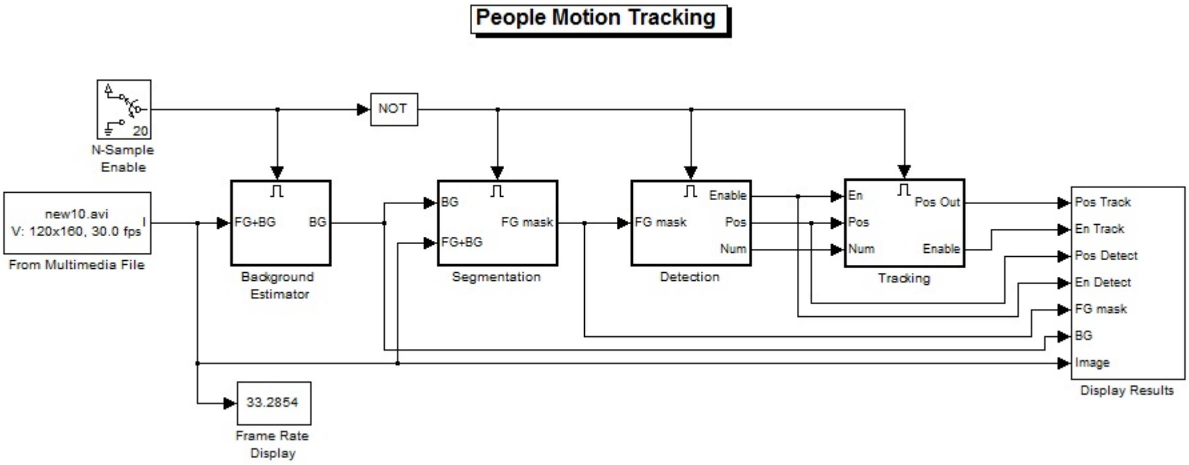
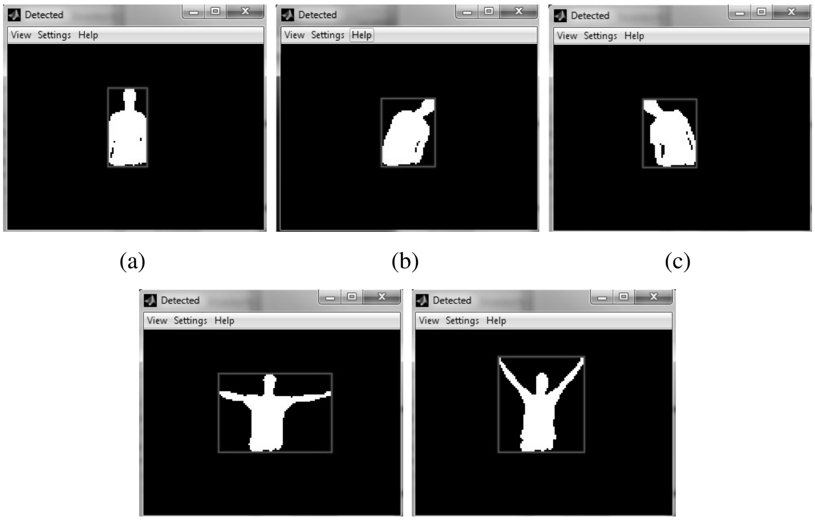


Figure 5.14: Human body tracking system on Simulink

In the Segmentation subsystem, the Autothreshold block uses the difference in pixel values between the normalized input image and the background image to determine which pixels correspond to the moving objects in the scene.

In the Detection subsystem, the Close block merges object pixels that are close to each other to create blobs. For example, pixels that represent a portion of a person's body are grouped together. Next, the Blob Analysis block calculates the bounding boxes of these blobs. In the final step, the Detection subsystem merges the individual bounding boxes so that each person is enclosed by a single bounding box.

In the Tracking subsystem, a Kalman Filter block uses the locations of the bounding boxes detected in the previous frames to predict the locations of these bounding boxes in the current frame. To determine the locations of specific people from one frame to another, the system compares the predicted locations of the bounding boxes with the detected locations. This enables the system to assign a unique color to each person. The system also uses the Kalman Filter block to reduce the effect of noise in the detection of the bounding box locations.



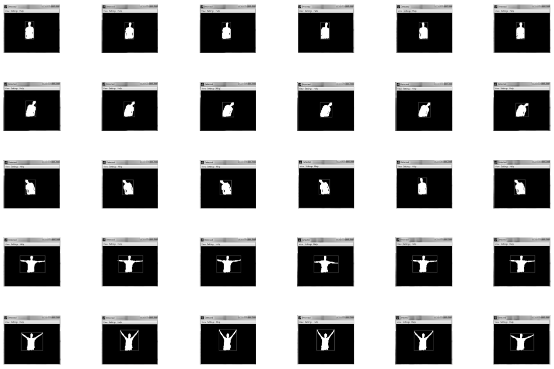
(a) (b) (c)

(d) (e)

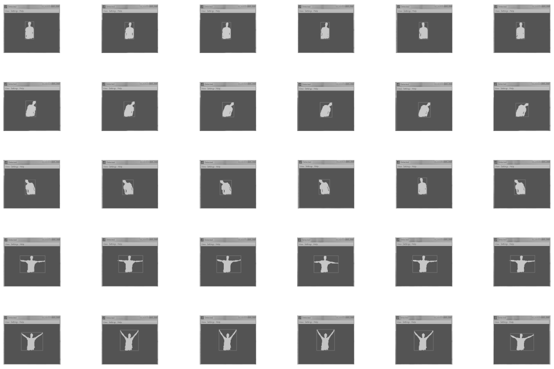
Figure 5.15: Examples of detected human body motions

In Figure 5.15 some examples of the tracked and detected human body motions can be seen. These images which are taken during the system is processed and used for training and testing data for the system. For the training data set, we used 6 images from each classes. Since we have 5 different classes, in the training data set we have 30 human body pose images. Also from the same video, we generated the test images. These test images are classified according to the euclidean distance between test images and the training data set, which is discussed in Chapter 4.

For training of the system we used the images which are given in Figure 5.16 a, and the normalized version of the training data set is given in Figure 5.16 b.



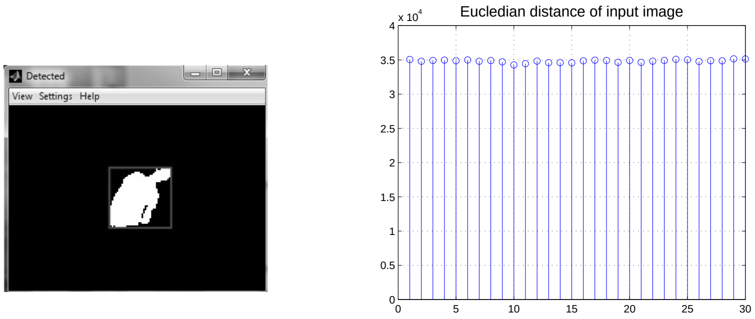
(a) Training data set



(b) Normalized version of the training data set

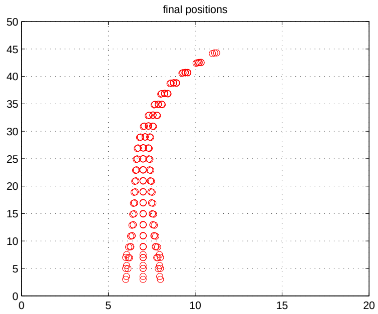
Figure 5.16: Training data set for human body poses

In the testing step, we used the images which are taken during the human body motion. For some examples of the test images, classification of the PCA based decision mechanism done according to the Euclidean distances is given in the following figures. On the left hand side of each figure the test image can be seen, in the middle resides the Euclidean distance graph, and on the right hand side the particle based simulation of the detected human poses. In the fist example (Figure 5.17), we take the test image as bending towards right.



(a) Test image

(b) Euclidean distance graphic

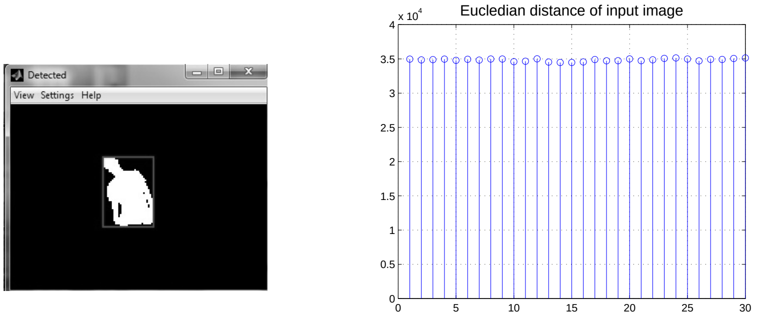


(c) Imitation of the test image with fluid particles

Figure 5.17: Training data set for human body poses

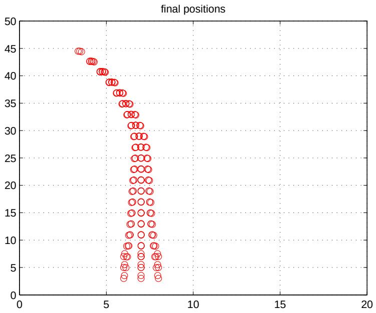
As we can see from the Euclidean distance graph, minimum distanced image is the 10^{th} image from the training data set which is given in Figure 5.17. The fluid particles are first accelerated along y direction and after some time steps, they bend towards right by the help of the appropriate body forces. The fluid particle motion is given in Figure 5.17 c and this motion is closely resemble the test image for this example 5.17 a.

In the second example, this time the human torso undergoes a bending towards left hand side is considered. This test image is symmetrical to the previous example. Again in the following figure, firstly we give the test image on the left hand side in Figure 5.18, secondly, in the middle the euclidean distance graph is given in order to see the minimum distance and decide which training image is close to the test image, and thirdly, on the right hand side the result of imitation can be seen with fluid particles.



(a) Test image

(b) Euclidean distance graphic



(c) Imitation of the test image with fluid particles

Figure 5.18: Training data set for human body poses

In this example, to imitate human body movement with fluid particles again the guidance effect of the body forces is employed. First the particles are accelerated along y axis with the initial velocity, after a while to bend the particle toward left hand side, body force is applied along $-x$ direction.

CHAPTER 6

IMITATION OF HUMAN BODY POSES AND HAND GESTURES BY REGIONAL CONTROLLER

6.1 Introduction

Up to know, different control algorithms were introduced, in order to generate desired fluid particle motions according to images that used as the input of the system. In chapter 3 ANN based controller strategy is introduced. In this controller, the user defined the feature vectors according to skeleton version of the human hand images. In chapter 4, the PCA was used for classification of the human hand gestures and body poses. This type of classifier generate its feature vector from the eigenvectors of the training set. According to the class of the test image, corresponding fluid parameters are applied to the fluid particles during the simulation. But this time, the proposed controller is not an adaptive strategy.

In this chapter, we introduced our region based controller which adaptively adjusts fluid parameters according to the shapes of the input images (desired images).

6.2 Fluidic Control Layer

6.2.1 Control Architecture

In this section, a region based formation control architecture of the swarm of fluid particles having an SPH modeled dynamics when imitating human body poses and hand gestures which are captured from a camera is developed and introduced. The

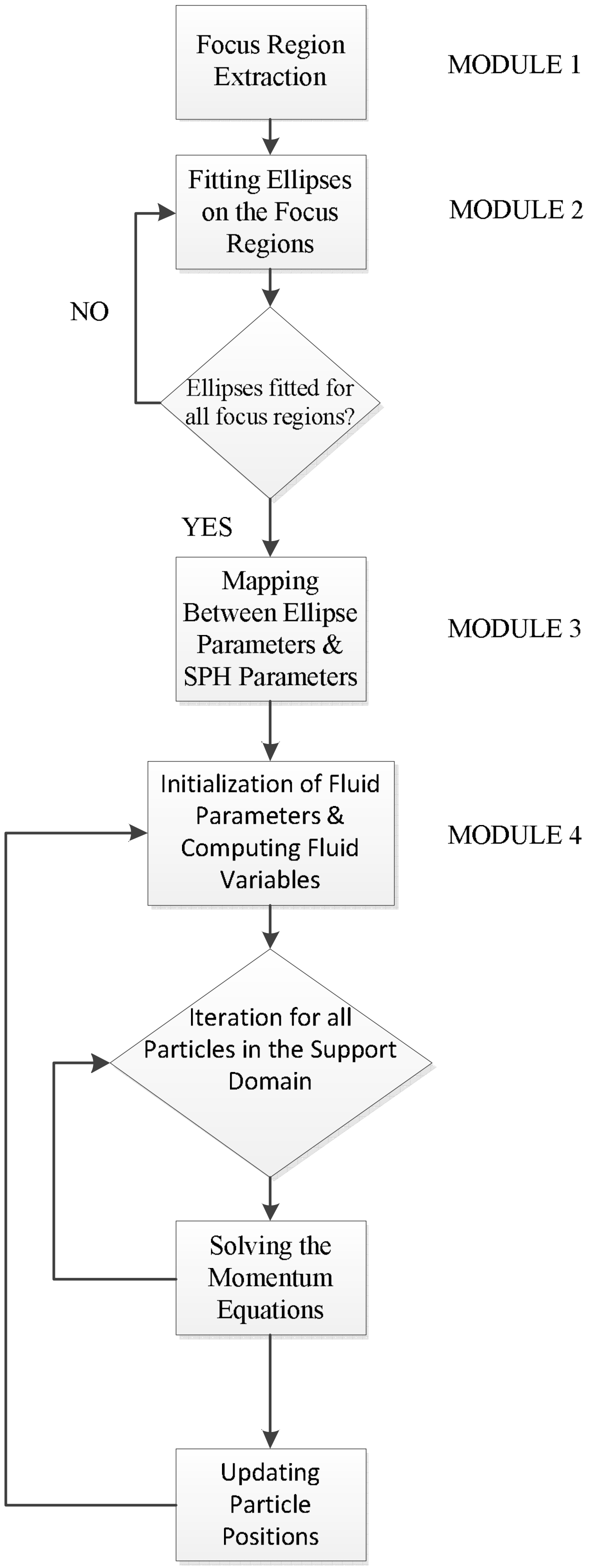

Focus Region Extraction
MODULE 1
Fitting Ellipses on the Focus Regions
MODULE 2
NO
Ellipses fitted for all focus regions?
YES
Mapping Between Ellipse Parameters & SPH Parameters
MODULE 3
Initialization of Fluid Parameters & Computing Fluid Variables
MODULE 4
Iteration for all Particles in the Support Domain
Solving the Momentum Equations
Updating Particle Positions

Figure 6.1: Flow chart of the proposed approach

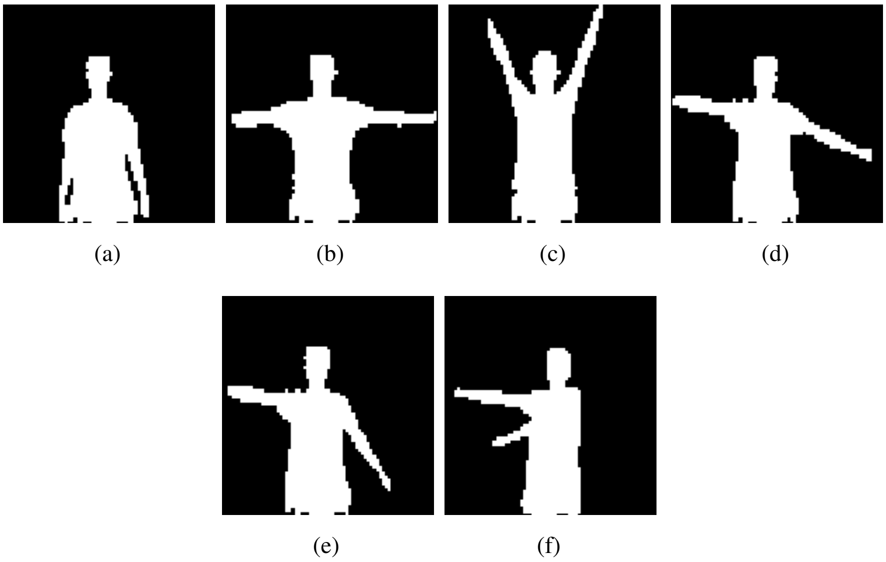
algorithm of the proposed method is demonstrated in Figure 6.1. The modules of this flow chart is explained in the following subsections.

6.2.2 Module 1: Focus Region Extraction

Module 1 is the responsible for extraction of the human body regions for torso, head, left and right arms and critical hand gesture points like center of image and center of hand in the image. In order to extract the regions of focus that has to capture attention in the imitation of human poses, background subtraction is first performed to filter out only human body poses and hand gestures in each video frame. In this work, we use a single camera to get the human motions videos of the upper body of that human. More specifically for upper body human pose imitation in that videos, first the human is standing and his/her arms are closed upon his/her torso. This standing human image is taken as the reference image which is given in Figure 6.2-a. A sequence sample of human upper body poses are given in Figure 6.2 b to f. There, the person extends both arms to the sides one right and left at shoulder level, and then extends both arms to the ceiling. The sequence continues with two arms descending along two sides in a slanted manner, the right arm stays horizontal at shoulder level while the left arm moves down at an angle then this extended arm draws a curve to approach the right arm. The representation of these motions are demonstrated in Figures 6.2 d, e and f.

In the human hand gesture example, the human hand performs different hand preshapes as a sequence of beginning with different variants of scissor like preshapes moving into a closed pinching that opens up into a hook preshpe. Similar to the human body posture, first background subtraction is performed. The representation of the this sequence is demonstrated in Figure 6.3.

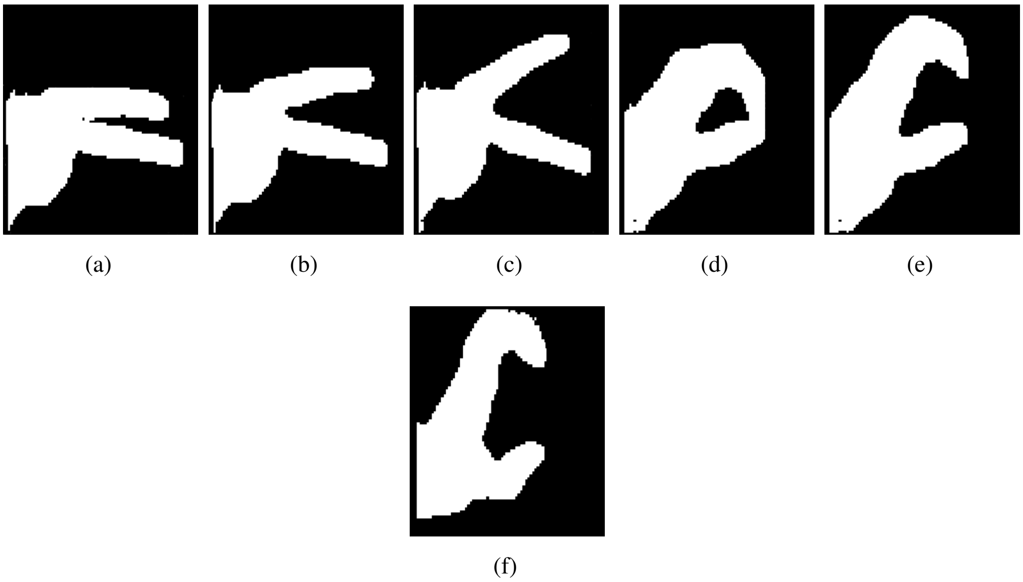
Module 1 is responsible for the generation of focus regions for both human body poses and human hand gestures. Focus regions are subparts of the pose either body or hand to imitate that can undergo change in action sequences. For human body pose focus regions, first we subtract the reference image from the input image. In segmentation of human body poses, Fig 6.2-a is used as the reference image. Since the maximum movement in this video frames occur around two arms, we need to determine two biggest connected component of this subtracted image. These two biggest



(a) (b) (c) (d)

(e) (f)

Figure 6.2: Human body movements for imitation



(a) (b) (c) (d) (e)

(f)

Figure 6.3: Hand gesture movements for imitation

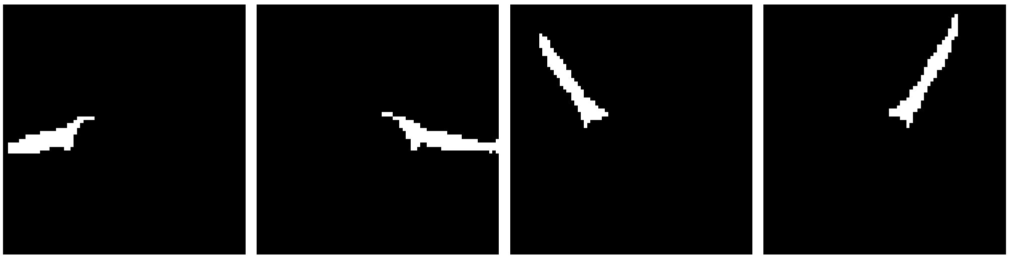

(a) (b) (c) (d)

Figure 6.4: Left and right arm segmentation for different human poses

regions that belong to the left and right arms that undergo the largest movements become the focus regions of the imitation. In Figure 6.4 the two biggest component regions which are found to belong to the left and right arm can be seen for the two frames of the video (Figure 6.2 a and b). The further elevation upward of both arms (Figure 6.2 c and d) became a change in the focus regions defined as the 2 arm regions. After determining the left and right arms of the human body which are the initial focus frames capturing the attention of the imitator, for frames sequences in Figure 6.2 a to d, the torso and the head become the next region of attention in the frames in Figure 6.2 d to f together with the arms, since the head and torso also move along with the arm. To determine these new regions of focus, our algorithm this time calculates the common regions of the reference image and the input image. Again in this case, there are two biggest connected component regions, one is for human torso and the other is for human head. The other connected component regions are negligible because of their small size which can be thought as the noisy regions. The example of segmented human body torso and head regions can be seen in Figure 6.5 a and b respectively.

Segmentation of human hand gestures is also done according to generated focus regions. Here again the focus regions are connected components undergoing changes relative to the each other in a gesture sequence. Focus regions are determined around critical points in terms of fitted ellipses. Module 1 extracts these critical hand position points which are demonstrated in Fig 6.6. The red star represents center of the gravity of the human hand, the center of the whole image is demonstrated as cyan star in same figure. The blue stars represent first ellipse fitting point which are correspond

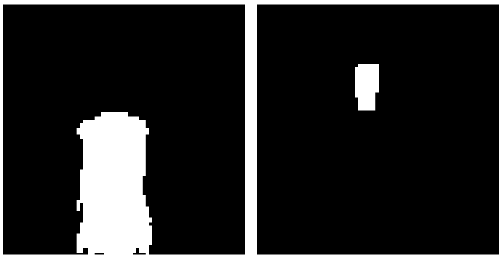

(a) TORSO REGION (b) HEAD REGION

Figure 6.5: Human body torso and head regions

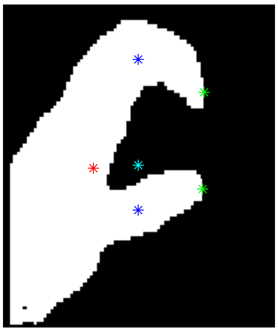


Figure 6.6: Focus region extraction for hand gesture

to the first joint of index and thumb fingers and the green stars represents finger tip locations. The blue star critical location is found by taking the derivative of the column which has the image center location. The green starts for finger tips are found by taking the derivative along the minus y-axis of the image. First intensity changed pixels which are below and upper side of the image center are taken as the finger tip locations.

6.2.3 Module 2: Ellipse Fitting to the Regions of Focus

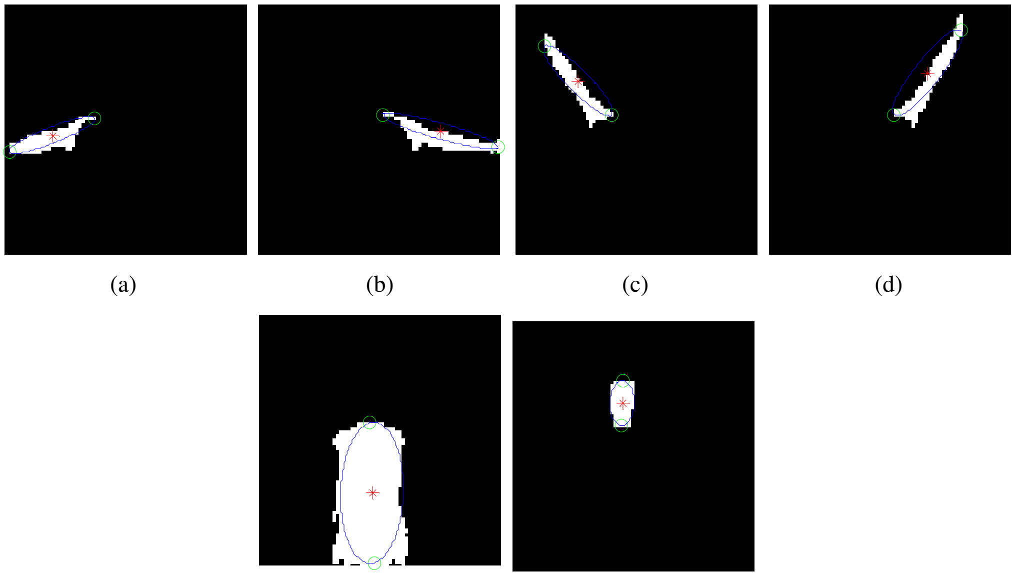
This module fits ellipses to the focus regions determined in module 1.

Ellipse fitting technique is widely applied in the analyses of human body motions and hand gestures. Lee et. all. [61] used ellipse parameters in their gait analysis for human gender classification. 3D ellipse fitting is also used for extraction of the gait features [62]. These gait features are again used for classification of the human gender. Moreover McColl et. all. [63] used ellipse model for interpretation and classification of the human hand gestures in human-robot interactions for socially assistive robots.

The best ellipse covering each of these regions is however not possible for all cases because of noisy parts of the extracted regions. Our proposed algorithm first finds the extreme pixels for arm regions along the x-direction (horizontal) while for head and torso regions along y direction (vertical) since the orientation of the head and torso regions of focused is perpendicular to the orientation of the arms. After that it calculates the distance between these extreme pixels. This distance corresponds to the major axis of the ellipse. And then the middle point is calculated between these extreme pixels. The pixels which are perpendicular to the middle points give the minor axis length of the ellipse. Finally we can calculate the orientation of the ellipse simply by comparing the major axis with the minor axis. Now we have 4 parameters, to represent an ellipse given in equation 6.1 which are namely length of major and minor axis (l_{major}, l_{minor}), the orientation of major axis of the ellipse (m), and center point of the ellipse (x_c, y_c), to represent an ellipse given by 6.1.

$$f(r_i) = (l_{major}, l_{minor}, m, x_c, y_c) \tag{6.1}$$

In Figure 6.7 the best matching ellipses are given for the regions demonstrated in Figure 6.4 and 6.5 for a human body pose sequence. In these figures, the green small circles demonstrate the extreme pixels of the ellipses and the red stars show the center pixel of the ellipses. With the same manner, the algorithm fits ellipses on human hand gesture images as shown in Figure 6.8. Now that the imitator has determined the regions to focus its attention by module 1 and the parameters of the best fitting ellipses by module 2. We will introduce the fluidic based swarm layer in module 4 in which the motions of the fluid particles are modeled by the equations of the Smoothed Particle Hydrodynamics (SPH).



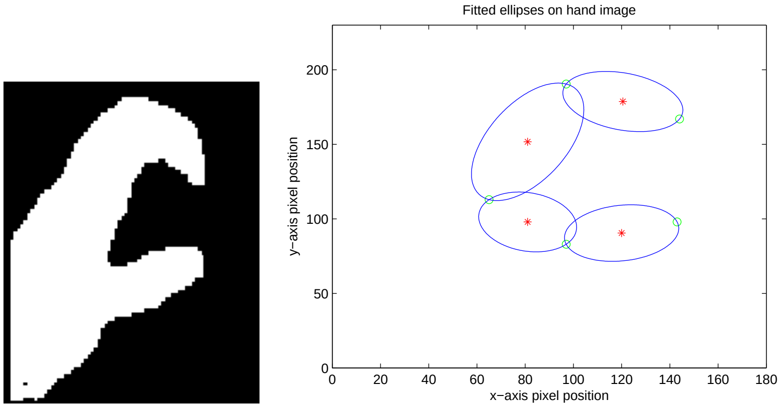
(a) (b) (c) (d)

(e) (f)

Figure 6.7: Fitted ellipses to the segmented human body regions

6.2.4 Module 4 Swarm Layer: Modeling The Fluid Particles With SPH

In nature, the fluid flow exists in 2 structures. One is compressible fluids (like gases) whose structure spreading around the environment and the other one is incompressible fluids (like liquids) whose structure having directional motions. We model the fluid particles as the elements of an incompressible fluid flow since human body motions and hand gestures are composed of directional movements. Our formation control algorithm is based on the mapping of the parameters of ellipses which are fitted on the focus regions of the human body and the parameters of fluid flow. In this work we do not consider a controller based on boundaries of regions of focus rather than the swarm flow is guided by critical points of the fitted ellipses. Filling a cavity with a fluid particles is not a strategy to solving the correspondence problems or in other words organ matching problem between human and fluid body. If it was the case, modeling the fluid particles as an element of the compressible fluid might be useful. In this work, formation control of these fluid particles as considers by assigning appropriate parameters according to the ellipse parameters which were discussed in the previous section. Formation control is accomplished by the modules 3 and 4.



(a) (b)

Figure 6.8: Focus region extraction for hand gesture

Our SPH formulation for fluid flow is adopted from [46] solving the momentum equation to determine the particle accelerations by using fluid parameters, such as density, pressure viscosity, obtained from neighbor particles in the support domain. In this method, the problem domain is represented by an arbitrary distributed particles and no specific discretization connectivity for these particles is needed. In our fluidic swarm, each particle is affected by a finite set of neighboring particles which are inside a “support domain” of that particle so the computational burden is decreased.

All calculation of the field variables depend on these neighboring particles. Since SPH is an approximation method, integrals are approximated based on field functions representation method is used for field function approximation. Details of SPH method to drive fluid particle colony were given in Chapter 3.

6.2.5 Module 3: Mapping Between Ellipse Parameters and SPH Parameters

Now that all the focus regions have been marked by the fitted ellipses, automatically to extracted human body and hand gesture regions (Module 1-2), and for all these regions we have the ellipse parameters, we then need to map to the SPH parameters. The algorithm converts the ellipse parameters to appropriate fluid parameters which are body forces, viscosity, stiffness coefficient according to the input image.

As mentioned earlier, the body forces have a guiding effect on the fluid particles since they directly enters the momentum equation. This property is useful for the construction of directional characteristics to imitate in human body poses or gestures. In imitation of the human body poses, since we have 4 ellipses for 4 different body regions (head, torso, left and right arms), in the simulation environment we have to divide our fluid swarm into 4 classes. Each swarm class will be responsible for the generation of the focus regions. The centers of the ellipses are used as waypoints, which are attractors so that the body forces are generated towards these locations by using the idea of potential field in order to pull the fluid particles. The used body force function is given in the following equation 6.2.

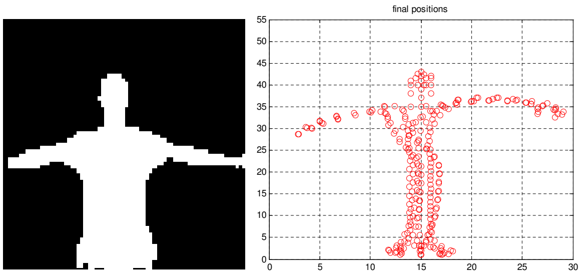
$$F_{x,y} = \begin{cases} F_x = \alpha(d-r).cos(\theta) & \\ F_y = \alpha(d-r)sin(\theta) & r \leq d \leq s+r \\ F_x = \alpha s cos(\theta) & \\ F_y = \alpha s sin(\theta) & d > s+r \end{cases} \tag{6.2}$$

In this equation, d is the distance between particle position and waypoint, θ is the angle between particle and the waypoint, r is the waypoint radius, s represents the spread of the field, α is the scaling factor which scales the magnitude of the body force vector. For example, since the starting point of the particles is below the human torso in human body pose imitation, the center of the ellipse for torso, is a common waypoint for all particles which are responsible for construction of head, left and right arm regions. This is how we map the ellipse centers to the SPH parameter. So center pixels of the ellipses are associated with the waypoints, in other words with the body forces. By completing the construction of the torso for human body pose imitation, fluid particles which are responsible for construction of the head, left and right arms reach the extreme pixel for torso. At this point, the particles for right and left arms need to be divided into two by the help of the higher valued stiffness coefficient while the particles for head, use the next waypoint which is the center of the ellipse for head region. Since the torso region is the common focus region, when the particles reach top of the torso, the stiffness coefficient β is changed form 2 to 20 for middle particles. While the middle particles slow down, the edge particles moves faster in both direction, namely the x and y axis. The edge particles are responsible for construction of the right and left arm regions, and middle particles are used for generation of the head region. After dividing the swarm flow into two for the arms, the particles for left and right arms use the center points of the ellipses forming the left and right arm focus regions as the next waypoints that attract them using the body force or in other words attractive potential function.

Similarly for hand gesture imitation, since human hand is represented by 4 ellipses according to the position of the fingers, the fluid swarm is divided into 4 classes again. During the imitation of hand gesture, the first critical point is used as the initial attraction point for all particles. After particles reach this point, the fluid colony is divided into 2 by applying higher valued stiffness and viscosity coefficients in order to generate fingering effect. When the particles reach first initial attraction point, in the algorithm the stiffness coefficient β is changed now from 2 to 50, since we want here there is no middle particles like imitation of human body poses. After that fluid separate into 2, the upper and lower ellipse center points are used as the waypoints for the fluid particle swarm. In order to generate closely similar hand gesture imitation with the input image, we use the second lower and upper ellipse center and finger tip locations as the waypoints. After dividing the fluid particles into 2, second ellipse center points and the finger tip waypoints are activated as the waypoints respectively. Our algorithm generates body forces to reach these waypoints with the same function given in equation 6.2.

6.3 Simulation Results

In the previous section we gave the mathematical background of our swarm modified SPH methodology. These equations are based on the Navier-Stokes equations and since the Navier-Stokes equations cannot be solved analytically, the SPH kernel and particle approximations are used for discretization of partial differential equations. The aforementioned SPH formulation is derived by discretizing the Navier-Stokes equations spatially, leading to a set of ordinary differential equations with respect to time that can be solved via time integration. Our ultimate goal is to learn to control the SPH-based flow model parameters by mapping the ellipse parameters to the simulation environment for the colony to resemble the human body poses and hand gestures.

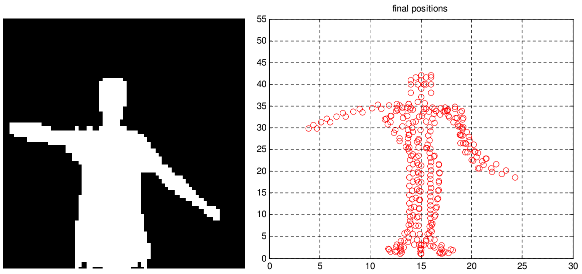


(a) (b)

Figure 6.9: Separation of arms two sides

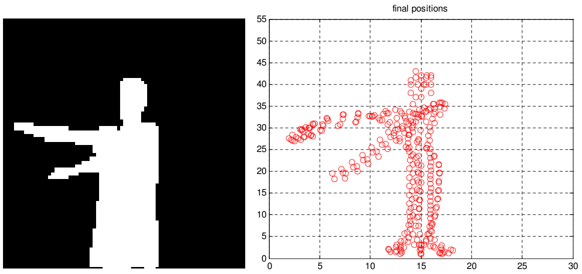
In this section, the result of the imitation of the human body poses and human hand gestures with fluid particles are given. In Figures 6.9, 6.10, and 6.11, the input image that has to be imitated and the imitation result of the fluid swarms are given. In the first example, the characteristic of the input image to be imitated is the extension of the left and right arm to two opposite sides. After fitting the ellipses of the human body regions, ellipse parameters are mapped to the fluid parameters. The ellipse parameters which are the length of the major and minor axes, center point of the ellipse and orientation of the major axis of the ellipse, are mapped to the body forces and stiffness coefficient in SPH parameters. According to the number of ellipses, the particles are assigned for generation of the ellipses. In each example, there are 4 ellipses, so in the simulation environment there are 4 classes of particles. The center point of the ellipse which is fitted on the torso is the initial common waypoints for all ellipses for all particles. According to these waypoints, body forces are generated by the algorithm using the idea of potential field as given analytically in the previous section. After forming the torso, for construction of left and right arms the particles need to be divided into two branches. In this case, higher values of the stiffness coefficient whose numerical examples are given in the previous section and used for the separation of the fluid colony particles into two to imitate the arms. After separation, again the center points of the corresponding arm ellipses are attractors generating appropriate body forces values. In a similar way, the head portion is constructed by the fluid particles. In the second and third examples, the left arm goes down and stays in the close vicinity of the right arm as seen in Figures 6.10 and 6.11.

Similarly, for imitation of hand gestures, each hand gesture is represented by 4 el-



(a) (b)

Figure 6.10: Moving left arm down side



(a) (b)

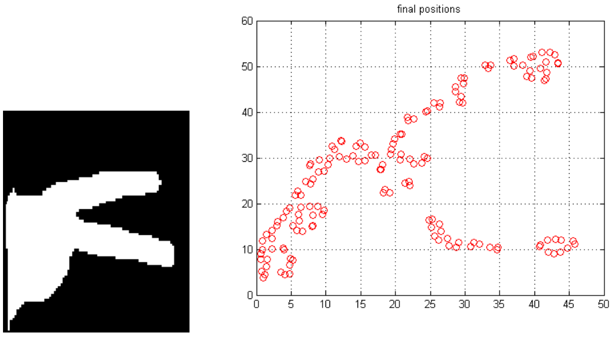
Figure 6.11: Left arm close to the right arm

lipses. The first critical ellipse point is used as the initial attraction point of the fluid particles. It means that all fluid particles start to move towards this first ellipse point and this motion generates the wrist and forehand of the imitated hand gesture. When the particles reach this critical location, fluid colony is divided into 2 by high values of stiffness coefficient, which is 50 numerically at this location. For conservation and generation of the formation, body forces play a vital importance. After division of particles and demonstration of the fingering effect next ellipse center points are activated as waypoints which are generated by using the idea of potential field given in equation 6.2. In Figures 6.12, 6.13, and 6.14 the desired hand gestures that have to be imitated are represented on the left, and the imitation results are given on right. In the first example scissor like hand gesture is imitated. First fluid particles are attracted by the first ellipse center, and every particle start to move on that direction. This is the critical point for the particles. Then they start to separate into 2 different paths to construct the middle and index finger due to applied higher valued stiffness coefficient. After fluid particles are divided into 2 different branches, they are attracted by the closest ellipse center point and generate the separate into index and middle fingers as seen in Figure 6.12-b. In Figure 6.13 the desired hand gesture is the hook preshape with a certain amount of aperture between thumb and index finger. The following hand gesture that has to be imitated and the imitation result are given in Figure 6.14-a and b, respectively. As seen in Figure 6.14-a the hand gesture is constructed by 1 line edge of thumb and the rest of the fingers are closing upon the thumb.

The sensitivity analysis of the fluid particles is done for the average particle velocities, and the average particle pressure values during the imitation of human body poses. In the three examples for human body postures, for separation of the human arms higher valued stiffness coefficient β is used. The effect of such a high valued stiffness coefficient occurs in the increase of the pressure value term which yields the slowing down of fluid particles. It also generates a velocity in a perpendicular direction to the motion due to the pressure gradient term in the momentum equation (eq:3.8).

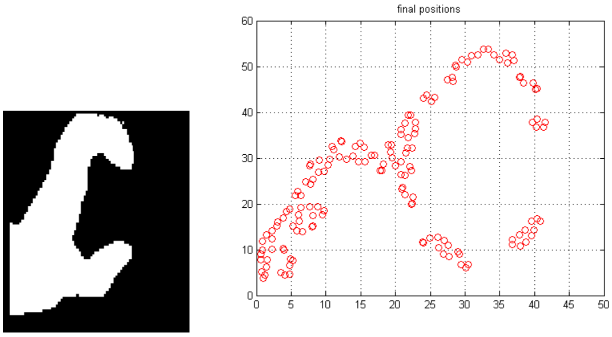
Figure 6.15 - a demonstrates the average pressure graph during the branching into the left and right arm imitation. As we expected, the average pressure is getting higher and stay almost at that level during the application of high value of the stiffness coefficient to the particles during the separation into the left and right arms. The average particle velocities for imitation of the left, right arms and for torso and head regions are given in Figure 6.15 - b and c respectively. It is clear that the average velocity for the focus region of the arm is less than the torso and head regions. Again the reason of these differences is that for the arm region stiffness coefficient is set to a higher value ($\beta = 20$) than for that of torso and head ($\beta = 2$) by the algorithm.

Interparticle distance analysis is done for imitation of hand gestures whose fluid swarm results are demonstrated in Figures 6.12, 6.13 and 6.14 while the interparticle distance results are given in Figure 6.16. The highest average distance which is 16.7, occurs during the imitation of scissor like hand gesture. The interparticle distances in the imitation of cylindrical to hook hand gesture and hook like pinching closing upon a thumb are 14.6 and 14.4, respectively. They are found to be close to each other since the finger tips closing upon themselves become similar to a pinching like preshape.



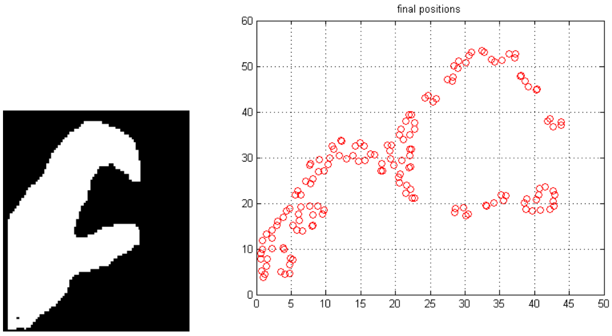
(a) (b)

Figure 6.12: Scissor like hand gesture



(a) (b)

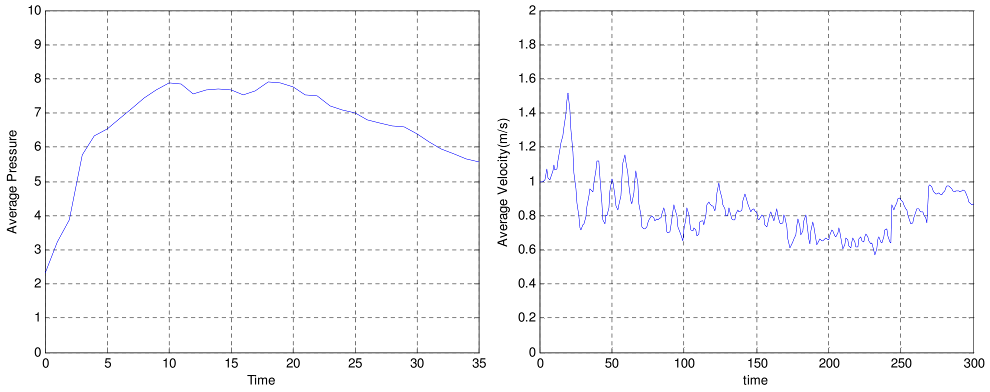
Figure 6.13: Cylindrical to hook preshape



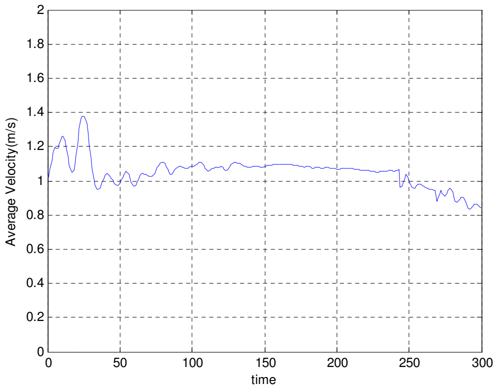
(a) (b)

Figure 6.14: Pinching like hand gesture

The pressure distributions during the imitation of hand gestures in scissor like pre-



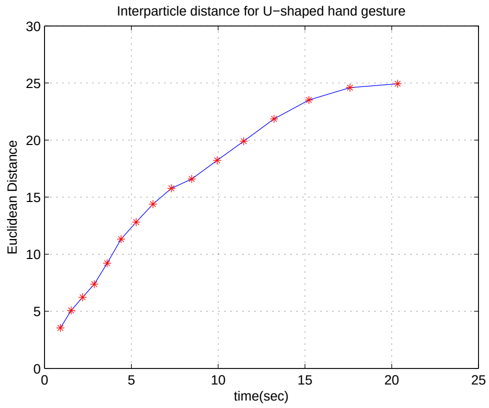
(a) Average pressure (b) Average velocity for arm particles



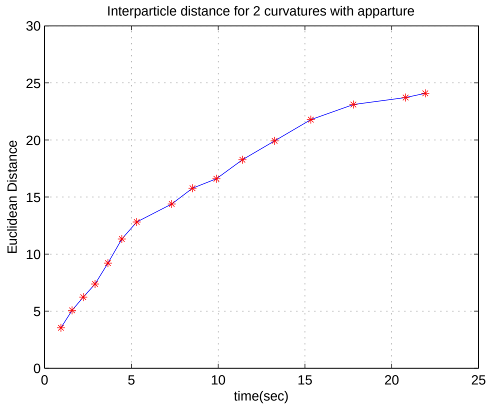
(c) Average velocity for head and torso particles

Figure 6.15: Sensitivity analysis of fluid parameters

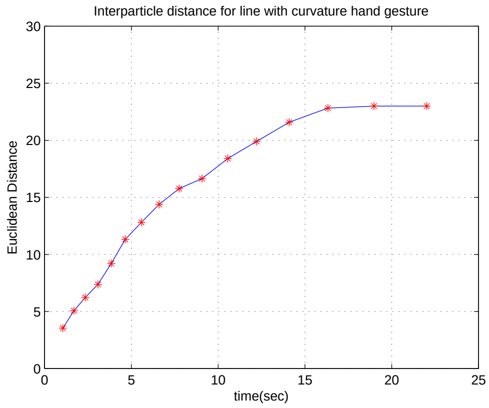
shape, pinching like grasping and hook preshape closing upon an erected thumb which are given in Figures 6.12, 6.13, and 6.14 respectively, are graphically demonstrated in Figure 6.17. If we compare the graphics in Figure 6.16 and Figure 6.17 we see that, when the average interparticle distance decreases, the pressure of fluid particles increase. As we aforementioned, the maximum interparticle distance occurs during the imitation of scissor like hand gesture and the average pressure is found to be 1.6124 for this hand gesture. For comparison purpose, the minimum interparticle distance is 14.4 when imitating hook preshape and the average pressure for this hand gesture is 1.8030. These results confirm what pressure equation (3.13) demonstrates which is that the particle pressure is directly proportional to the stiffness coefficient β and density ρ. When we consider the density equation (3.5), we can drive from that equation that when particles are close to each other, the density is higher than the case when the particles flow branch out into 2 directions.



(a) Scissor like hand gesture

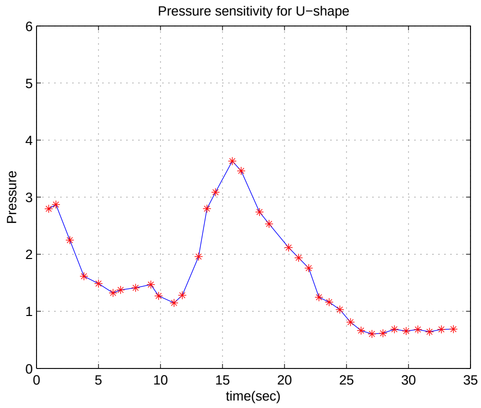


(b) Cylindrical hook hand gesture

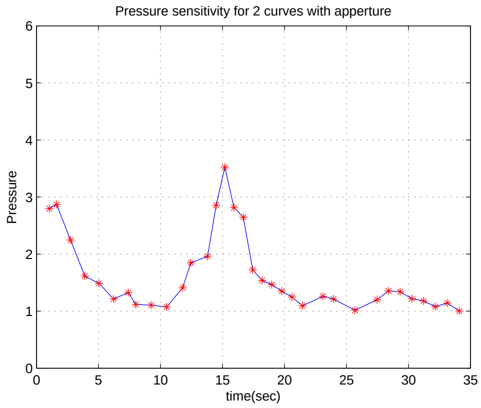


(c) Pinching like grasping

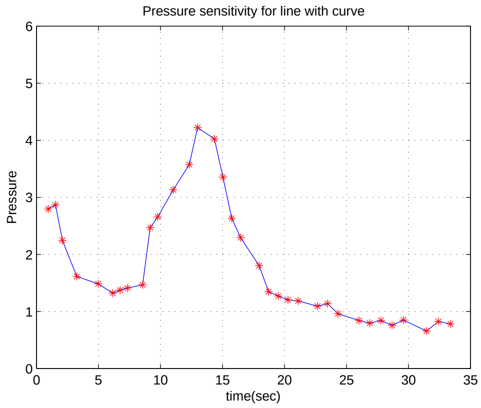
Figure 6.16: Interparticle distances for different hand gestures



(a) Pressure distribution for scissor like preshape



(b) Pressure distribution for cylindrical hook hand gesture



(c) Pressure distribution for pinching like hand gesture

Figure 6.17: Pressure distributions for different hand gestures

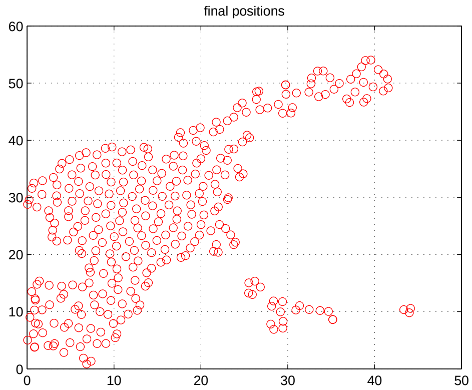
Another sensitivity analysis for imitation of human hand gestures is performed with respect to the number of fluid particles and step size of the algorithm. When the number of particles is approximately more than 300 it is not possible to see the branching effect for imitation and also the simulation results may not resemble the initially given hand gestures. For scissor like hand gesture, cylindrical and pinching like hand gestures, imitation results are given in Figure 6.18. At the end of the first imitation results of scissor like hand gesture in Figure 6.18-a use 351 fluid particles. As one can see from this result, the separation of particles into two branches cannot be seen as clearly as in Figure 6.12. Large number of fluid particles should then be used such that the branching can be clearly visible in imitations.

However, in the second and third imitation examples given in Figure 6.18-b and c, the problem is the thick wrist region because of the large number of fluid particles. When we compare these two imitation results with the first one of Figure 6.18-a, the branching is much more clearer. During the imitation of these two hand gestures, 321 and 319 fluid particles are used in each case.

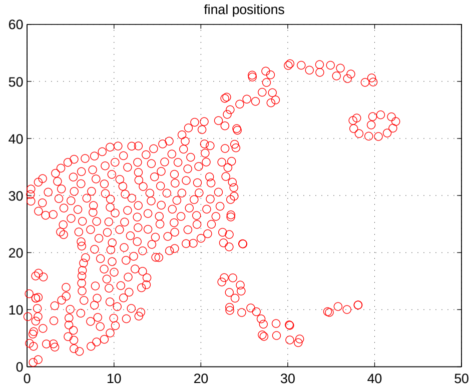
When the step size of the algorithm is decreased, the simulation time increases rapidly. Even in some cases, when especially the step size is less than 5 iterations, the simulation gets stuck before any convergence. The elapsed times towards solution for imitation of the above mentioned hand gestures are given in table (6.1).

Table 6.1: Elapsed Time According to the Step Size(sec)

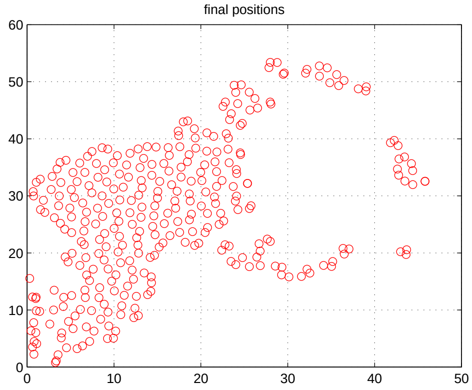
Step Size	Scissor like Hand Gesture	Cylindrical preshape	Pinching preshape
7	187	214	219
10	126	136	141



(a) Scissor like hand gesture



(b) Cylindrical hook hand gesture



(c) Pinching like grasping hand gesture

Figure 6.18: Particle number sensitivity for hand gestures

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

There are various characteristics of fluid flows which are desirable in swarm robotics, such as obstacle avoidance and source to sink optimal path finding behaviors of the fluid. In this work, we combine these behaviors to get the desired shape of a colony of particles for mimicking human hand postures based on fluidic formation control. Such a formation control is nonexistent in the literature where the work that resembles our approach by far is [43] In this reference work the swarm group is controlled by the help of the SPH parameters,for only obstacle avoidance issue, without considering at all swarm formation.

Our imitator is a fluid body imitating through sensing, the human body poses and hand gestures. We provide in this thesis, the “proof of concept” demonstration that our proposed fluidic formation control can make the fluid body assume postures that mimic basic human hand gestures and body poses. Our approach is based on generating the control of flow field variables in order to get desired behaviors and shapes of the fluid body, by observing human hand and body pose behaviors.

In this work, we tackled the problem of imitating human hand postures and body poses by a system that possesses a completely different dynamics, thus unable to initiate an imitational organ matching. In the balance of this thesis work, we introduce the novel architecture of a fluidics formation controller to tune fluid flow parameters to get the desired colony formations which resemble human hand gestures and human body motions.

The fluidics formation controller commands on SPH approximated dynamics of a fluidic swarm, and has the ability to learn imitation based on hand and body pose feature/fluid parameters I/O pairs which form its training sets.

This thesis provides the proof of concept imitating human body poses and hand gestures with different type of controllers commanding the body force vector and stiffness coefficient of swarm particles. Our present research work resides on incorporating more fluid parameters with the control action of the fluidics formation controller so as to imitate complex human hand postures and body poses.

The main contributions of this thesis work can be summarized in the following items.

- The main contribution of this thesis is imitation of human hand gestures and human body motions in the presence of correspondence problem. In our work, the imitator is the fluid particles which are used for the discretization of the problem domain.
- In this work we introduced the decentralized controller for fluid particles which imitate human hand gestures and human body motions.
- To the best of our knowledge it is the first time such an extreme embodiments are used for imitation problem. For example it is not like humanoid robot imitating a human (it is somehow possible to establish one to one organ matching since humanoid robots have similar organs with human beings like legs, arms, heads, hands etc.)

7.2 Future Work

In this dissertation, a novel method is proposed to eliminate the effect of the correspondence problems which are caused by the difference in embodiments. In our case, the correspondence problem is initially exist since as the demonstrator, human body poses and hand gestures are taken, while the imitator is the fluid body which is considered as combination of fluid particles.

Possible future work directions can be summarized as follows.

Imitation can be performed between two swarm of small sized robot colonies. These colonies can be composed of small sized robots which have different properties or they might have partially organ matching. By the help of imitation learning, the learning time between two robots can be decreased.

Another possible future direction might be usage of small sized robots as the member of fluid particles. These robots can be used for rescue operations in the unstructured environments. They might behave incompressible like fluid if they need to move faster, and in some cases for example if the maximization of the coverage area in order to reach to the all of the collapsed regions, they might behave like compressible fluid.

REFERENCES

[1] Thomas Cederborg, Ming Li, Adrien Baranes and Pierre-Yves Oudeyer, “Incremental Local Online Gaussian Mixture Regression for Imitation Learning of Multiple Tasks”, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2010, Taipei.

[2] K. Dautenhahn and C. L. Nehaniv., “An agent-based perspective on imitation” In K. Dautenhahn and C. L. Nehaniv, editors, ”Imitation in Animals and Artifacts”, pages 1-40. MIT Press, 2002.

[3] Brenna Argall, Brett Browning, Manuela Veloso, “Learning by Demonstration with Critique from a Human Teacher” HRI ’07 Proceedings of the ACM/IEEE international conference on Human-robot interaction.

[4] Pieter Abbeel and Andrew Y. Ng., “Exploration and apprenticeship learning in reinforcement learning”, In ICML ’05, Proceedings of the 22nd international conference on Machine learning 2005.

[5] A. Billard, “Learning motor skills by imitation: a biologically inspired robotic model”, in cybernetics & Systems , special issue on ”Imitation in animals and artifacts”, C. Nehaniv & K. Dautenhahn editors. Summer 2000.

[6] Raffaella I. Rumiati, Joana C. Carmo, Corrado Corradi-Dell’Acqua, “Neuropsychological perspectives on the mechanisms of imitation”, Phil.Trans. Soc. B 2009, 2337 - 2347.

[7] J. Demiris and G. Hayes, “Imitative Learning Mechanism in Robots and Humans”, in Proceedings of the 5th European Workshop on Learning Robots, pp. 9-16, Bari, Italy, July 1996.

[8] Lucy, L. B., “Numerical approach to testing the fission hypothesis”, Astronomical Journal, 82:1013-1024 1977.

[9] Gingold, R. A. and Monaghan, J. J., “Smoothed particle hydrodynamics: Theory and application to nonspherical stars.”, Monthly Notices of the Royal Astronomical Society, 181:375-389 1977.

[10] Billard, A., Siegwart, R. “Robot learning from demonstration”, Robotics and Autonomous Systems 47 (2-3), pp. 65-67.

[11] Anna Belardinelli, Fiora Pirri, “Bottom-Up Gaze Shifts and Fixations Learning by Imitation”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[12] Calinon, S. and Sauser, E. and Billard, A. and Caldwell, D., “Evaluation of a probabilistic approach to learn and reproduce gestures by imitation”, Proceedings of the IEEE Intl Conf. on Robotics and Automation (ICRA), 2010 Alaska USA.

[13] Mark A. Wood, “Skill Acquisition Through Program-Level Imitation in a Real-Time Domain”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[14] Sylvain Calinon, Florent Guenter, and Aude Billard, “On Learning, Representing, and Generalizing a Task in a Humanoid Robot” IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[15] Manuel Lopes, and José Santos-Victor, “A Developmental Roadmap for Learning by Imitation in Robots”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[16] Michael Pardowitz, Steffen Knoop, Ruediger Dillmann, and Raoul D. Zöllner, “Incremental Learning of Tasks From User Demonstrations, Past Experiences, and Vocal Comments”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[17] Krishnanand N. Kaipa , Josh C. Bongarda, Andrew N. Meltzoff, “Self discovery enables robot social cognition: Are you my teacher?”, Neural Netw. 2010 Oct-Nov;23(8-9):1113-24. Epub 2010 Aug 8.

[18] A. Billard, “Learning motor skills by imitation: A biologically inspired robotic model,” Cybern. Syst., vol. 32, no. 1/2, pp. 155-193, Jan. 2001.

[19] W. Erlhagen, A. Mukovskiy, E. Bicho, G. Panin, C. Kiss, A. Knoll, H. van Schie, and H. Bekkering, “Action understanding and imitation learning in a robot-human task,” in Proc. ICANN, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, Eds. New York: Springer-Verlag, Sep. 11-15, 2005, vol. 3696, pt. 1, pp. 261-268.

[20] M. N. Nicolescu and M. M. Matari´c, “Learning and interacting in humanrobot domains,” IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, vol. 31, no. 5, pp. 419-430, Sep. 2001.

[21] S. Schaal, “Is imitation learning the route to humanoid robots?” Trends Cogn. Sci., vol. 3, no. 6, pp. 233-242, 1999.

[22] Y. Kuniyoshi, M. Inaba, and H. Inoue, “Learning by watching: Extracting reusable task knowledge from visual observations of human performance,” IEEE Trans. Robot. Autom., vol. 10, no. 6, pp. 799-822, Dec. 1994.

[23] Aris Alissandrakis, Chrystopher L. Nehaniv, and Kerstin Dautenhahn, “Correspondence Mapping Induced State and Action Metrics for Robotic Imitation”, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 37, No. 2, April 2007.

[24] Schaal, S.; Ijspeert, A.; Billard, A. “Computational approaches to motor learning by imitation” in Frith, C.D.;Wolpert, D. (eds.), The Neuroscience of Social Interaction, 1431, pp. 199-218, Oxford University Press, 2004.

[25] Alissandrakis, A., Nehanic, C. and Dautenhahn, K. “Imitation with ALICE: Learning to Imitate Corresponding Actions across Dissimilar Embodiments”, IEEE Transaction on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol.32, No.4, July 2002.

[26] Glas, D.F., Miyashita, T., Ishiguro, H., Hagita, N. "Laser tracking of human body motion using adaptive shape modeling". IEEE/RSJ International Conference on, IROS 2007, San Diego, California, USA.

[27] Minato, T., Ishiguro, H. "Generating natural posture in an android by mapping human posture in three-dimensional position space". IEEE/RSJ International Conference on, IROS 2007, San Diego, California, USA.

[28] Baris Ozyer, Erhan Oztop, "Task Dependent Human-like Grasping", 8th IEEE-RAS International Conference on Humanoid Robots December 1-3, 2008 / Daejeon, Korea.

[29] Vishwanathan Mohan, Pietro Morasso, Jacopo Zenzeri, Giorgio Metta, V. Srinivasa Chakravarthy, Giulio Sandini "Teaching a humanoid robot to draw Shapes", Autonomous Robots (2011) 31:21-53 DOI 10.1007/s10514-011-9229-0 , pp. 1-33.

[30] Sylvain Calinon, Florent D'halluin, Eric L. Sauser, Darwin G. Caldwell and Aude G. Billard, "Learning and reproduction of gestures by imitation: An approach based on Hidden Markov Model and Gaussian Mixture Regression", IEEE Robotics and Automation Magazine, vol. 17, num. 2, 2010, p. 44–54.

[31] S. Hurley and N. Chater, Eds., ""Perspectives on Imitation: From Neuroscience to Social Science", Cambridge, MA: MIT Press, 2005.

[32] B. Petreska and A. Billard, "A Neurocomputational Model of an Imitation Deficit following Brain Lesion", In Proceedings of 16th International Conference on Artificial Neural Networks (ICANN 2006), Athens, Greece. Lecture Notes in Computer Science (LNCS), vol. 4131 (2006) pp. 770-779.

[33] B. Petreska, M. Adriani, O. Blanke, A.G. Billard, "Apraxia: a review", Progress in Brain Research 164, pp. 61-83.

[34] Louis M. Herman, "Vocal, Social, and Self-Imitation by Bottlenosed Dolphins" in Imitation in Animals and Artifacts, K. Dautenhahn and C. L. Nehaniv, Eds. Cambridge, MA: MIT Press, 2002, pp. 63-108.

[35] Y. U. Cao, A. S. Fukunaga, A. B. Khang, F. Meng, "Cooperative Mobile Robotics: Antecedents and Directions," in Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1995, pp. 226 - 234.

[36] G. Dudek, M. R. M. Jerkin, E. Milios, and D. Wilkes,"A Taxonomy for Multi-Agent Robotics," Autonomous Robots, Vol. 3(4), pp. 375 - 397, Dec. 1996.

[37] L. E. Parker, "Current State of the Art in Distributed Autonomous Mobile Robotics," in Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems, 2000, pp. 3 - 12.

[38] T. Arai, E. Pagello, L. E. Parker, "Editorial: Advances in Multi-Robot Systems," IEEE Transactions on Robotics and Automation, Vol. 18(5), pp. 655 - 661, Oct. 2002.

[39] G. Beni, “From Swarm Intelligence to Swarm Robotics,” Swarm Robotics: State-of-the-Art Survey, Lecture Notes in Computer Science 3342, Springer-Verlag, pp. 1 - 9, 2005.

[40] E. Sahin, “Swarm Robotics: From Sources of Inspiration to Domains of Application,” Swarm Robotics: State-of-the-Art Survey, Lecture Notes in Computer Science 3342, Springer-Verlag, pp. 10 - 20, 2005.

[41] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A Survey on Sensor Networks,” IEEE Communications Magazine, pp. 102 - 114, Aug. 2002.

[42] Elena Garcia, Maria Antonia Jimenez, Pablo Gonzalez De Santos, and Manuel Armada, “The Evolution of Robotics Research”, Volume: 14, Issue: 1 2007 , Page(s): 90 - 103.

[43] Pac, M.R., Erkmen, A.M., Erkmen, I. “Control of Robotic Swarm Behaviors Based on Smoothed Particle Hydrodynamics”, IEEE/RSJ International Conference on, IROS 2007, San Diego, California, USA.

[44] Pimenta, L.C.A., Mendes, M.L., Mesquita, R.C., Pereira, G.A.S. “Fluids in Electrostatic Fields: An Analogy for Multirobot Control”, IEEE Transactions on Magnetics, April 2007.

[45] Baris Ozyer , Ismet Erkmen, Aydan M. Erkmen, “Catching Continuum Between Preshape and Grasping Based on Fluidics” ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis (ESDA2010) July 12-14, 2010 , Istanbul, Turkey.

[46] G. R. Liu and M. B. Liu, “Smoothed Particle Hydrodynamics: A Meshfree Particle Method”, World Scientific, 2003.

[47] Chengjun Liu, and Harry Wechsler, “Evolutionary Pursuit and Its Application to Face Recognition”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 6, pp. 570-582, 2000.

[48] Choi, K., Toh, K.-A., Byun, H., “An efficient incremental face annotation for large scale web services”, Telecommunication Systems 47 (3-4), pp. 197-214, 2011.

[49] Kisku, D.R., Mehrotra, H., Gupta, P., Sing, J.K., “Robust multi-camera view face recognition”, International Journal of Computers and Applications 33 (3), pp. 211-219.

[50] AlEnzi, V., Alfiras, M., Alsaqre, F, “Face recognition algorithm using two dimensional principal component analysis based on discrete wavelet transform”, Communications in Computer and Information Science 188 CCIS (PART 1), pp. 426-438.

[51] Yang, J., Zhang, D., Frangi, A.F., Yang, J.-Y. , “Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1), pp. 131-137.

[52] Draper, B.A., Baek, K., Bartlett, M.S., Beveridge, J.R., “Recognizing faces with PCA and ICA”, Computer Vision and Image Understanding 91 (1-2), pp. 115-137.

[53] Matthew A. Turk and Alex P. Pentland, “Face Recognition Using Eigenfaces”, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR ’91, page(s): 586 - 591.

[54] Georgios Tzimiropoulos, Stefanos Zafeiriou and Maja Pantic, “Principal Component Analysis of Image Gradient Orientations for Face Recognition”, 2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011 , art. no. 5771457, pp. 553-558.

[55] A.A. Mohammed, R.Minhas, Q.M.JonathanWu, M.A.Sid-Ahmed, “Human face recognition based on multidimensional PCA and extreme learning machine”, Pattern Recognition 44 (10-11), pp. 2588-2597.

[56] Salvador Cobos, Rafael Aracil, and Manuel Ferre, “Low Dimensionality Space for Controlling Human Hand Models”, International Conference on Biomedical Robotics and Biomechatronics, The University of Tokyo, Tokyo, Japan, September 26-29, 2010.

[57] Jin-fu Yang, Min Song, Ming-ai Li, “Efficient Robot Object Recognition Technique Based on Distance Kernel PCA” Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics December 14-18, 2010, Tianjin, China.

[58] Benjamin Balaguer and Stefano Carpin, “Efficient grasping of novel objects through dimensionality reduction”, 2010 IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA.

[59] Salvador Cobos, Manuel Ferre, M. Angel Sanchez-Uran, Javier Ortego, Rafael Aracil, “Human hand descriptions and gesture recognition for object manipulation”, Computer Methods in Biomechanics and Biomedical Engineering 13 (3), pp. 305-317.

[60] U. Tilki, İ. Erkmen, and A. M. Erkmen, “Imitation of basic hand preshapes by fluid based method: fluidics formation control”, Turk J Elec Eng & Comp Sci, Vol.19, No.2, 2011, @ TÜBİITAK.

[61] L. Lee, W.E.L.Frimson, “Gait Analysis for Recognition and Classification”, Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FGR’02) Washington, D.C., USA.

[62] S. Sivapalan, D. Chen, S. Denman, S. Sridharan, C. Fookes, “3D Ellipsoid Fitting for Multi-view Gait Recognition”, 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance, Klagenfurt, Austria.

[63] D. McColl, Z. Zhang, G. Nejat, “Human Body Pose Interpretation and Classification for Human-Robot Interaction”, International Journal of Social Robotics(2011)3:313-332.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name : Tilki, Umut
Nationality : Turkish (TC)
Date and Place of Birth : 04 January 1980 , Karabük, TURKEY
email : utilki@metu.edu.tr, utilki@gmail.com

EDUCATION

Degree	**Institution**	**Year of Graduation**
PhD	: METU, Electrical and Electronics Engineering,	2012
BS	: Süleyman Demirel University, Electronics and Telecommunication Eng.,	2001
High School	: Karabük Demir Çelik Lisesi,	1997

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2002–Present	: METU, Department of Electrical and Electronics Eng.	Research and Teaching Assistant

FOREIGN LANGUAGES

English

PUBLICATIONS

1. U. Tilki, İ. Erkmen, and A. M. Erkmen *Human Hand Gesture Imitation by the Formation Control of a Fluid Swarm Modeled by Smoothed Particle Hydrodynamics (SPH)*, Computer and Information Science, Canadian Center of Science and Education 2012 (accepted with minor revision).

2. U. Tilki, İ. Erkmen, and A. M. Erkmen *Imitation of Human Body Poses and Hand Gestures by The Formation Control of a Fluidic Swarm*, 2012, (ready for submission).
3. U. Tilki, İ. Erkmen, and A. M. Erkmen *Imitation of basic hand preshapes by fluid based method: fluidics formation control*, Turk J Elec Eng & Comp Sci, Vol.19, No.2, 2011, @ TÜBİTAK.
4. U. Tilki, İ. Erkmen, and A. M. Erkmen, *Imitation of Human Body Poses by the Formation Control of a Fluidic Swarm*, Proceedings of the ASME 2012 11th Biennial Conference On Engineering Systems&Design And Analysis ASME / ESDA 2012 July 2-4, 2012, Nantes, France
5. U. Tilki, İ. Erkmen, and A. M. Erkmen, *Imitation of Basic Hand Preshapes by Fluid Based Method: Fluidics Formation Control*, 6th International Conference on Electrical and Electronics Engineering, November 2009, Bursa, Turkey (selected paper, best student paper award).