3D ANIMATION FOR HAND PRESHAPING

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

BY

MEHMET SONER GÜLER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

MARCH 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet Erkmen

Head of Department

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

Assist. Prof. Dr. İlkay ULUSOY	Prof. Dr. Uğur Halıcı
Co-Supervisor	Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU	(METU, EE)	
Prof. Dr. Uğur HALICI	(METU, EE)	
Prof. Dr. Aydan ERKMEN	(METU, EE)	
Assist. Prof. Dr. İlkay ULUSOY	(METU, EE)	
Dr. Didem GÖKÇAY	(METU, II)	

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: Mehmet Soner GÜLER

Signature :

ABSTRACT

3D ANIMATION FOR HAND PRESHAPING

Güler, Mehmet Soner
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Uğur Halıcı
Co-Supervisor: Assist. Prof. Dr. İlkay Ulusoy

March 2006, 101 Pages

The human hand is an essential part of human body, capable of making complex and expressive motions. Its complicated structure makes it a formidable challenge for animators to animate hand motions. Most computer graphics research on hand motion has focused on preshaping, preshaping and gestures with application to areas of human computer interaction and sign language. There are also a number of educational applications such as typing, playing of musical instruments etc. From a computer graphics standpoint, these applications are difficult in animation of hand.

This thesis aims to animate 3D hand preshaping activity for a chosen virtual 3D object in real-time. Researches on human hand kinematics, structure and geometric stability analysis on preshaping are the main motivation for the algorithms developed in this thesis for animating 3D preshaping.

The algorithm that we developed is made of two main parts. The first part is related with the precision type preshaping requiring the finger-tips positioning for a given object such as the cube, cylinder or sphere. First part is completed by procedural approach which is based on kinematics to generate the motion of the hand for the given virtual object at the determined finger-tip positions.

Second part related with the wrap type preshaping aims to have maximum interaction between hand and object. For this purpose, we have developed the collision detection algorithm to find intersection surfaces between hand and object.

Even though developed algorithm based on the kinematics was used for the precision type preshaping application, it can also be used for many other applications requiring hand animation given the positions of finger tips.

Keywords: Animation, Hand Preshaping, Real-Time, JAVA 3D

ÖZ

EL ÖNŞEKİLLENDİRMESİ İÇİN 3B ANİMASYONU

Güler, Mehmet Soner

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Uğur Halıcı

Tez Yardımcı Danışmanı: Yrd. Doç. Dr. İlkay Ulusoy

Mart 2006, 101 Sayfa

İnsan eli, insan vucudunun, karmaşık ve anlamlı hareketler yapmakta yetenekli temel bir parçasıdır. Karmaşık yapısından dolayı el hareketlerinin animasyonu animatörler tarafından gerçekleştirilmesi zor bir problemdir. El hareketi üzerine yapılan birçok bilgisayar grafik araştırması, bilgisayar arayüzü ve işaret dili alanlarındaki uygulamalarda tutma, tutma önşekillendirmesi ve el jestlerine odaklanmıştır. Bununla birlikte daktilo kullanmak, müzik aleti çalmak vb. gibi birçok eğitim uygulamaları da bulunmaktadır. Bilgisayar grafiği bakış açısıyla bu uygulamalar elin gerçekçi animasyonu bakımından zordur.

Bu tez; seçilen sanal bir nesnenin 3D el ile tutma önşekillendirmesi aktivitesinin gerçek zamanda animasyonunu amaçlamaktadır. İnsan elinin kinematiği, yapısı ve tutmanın geometrik kararlılık analizi üzerine yapılan araştırmalar bu tezdeki 3D tutma önşekillendirmesi animasyonu algoritması geliştirilmesi için motivasyon olmuştur.

Geliştirdiğimiz algoritma iki ana bölümden oluşmaktadır. Birinci bölüm verilen küp, silindir veya küre gibi nesneler için parmak ucu konumlandırılması isteyen hassas tipli tutma önşekillendirmesiyle ilgilidir. Birinci bölüm sahnede bulunan sanal bir nesnenin

belirlenen parmak ucu noktalarından el hareketini gerçekleştiren, kinematiğe dayanan yordamsal bir yaklaşımla tamamlanır.

İkinci bölüm nesne ve el arasında maksimum etkileşimi amaçlayan güçlü tutma önşekillendirmesi ile ilgilidir. Bu amaç için, el ve nesne arasındaki yüzeylerin etkileşimini bulmak için çarpışma belirleme algoritmasını geliştirdik.

Geliştirilen kinematik temelli algoritma bu çalışmada hassas tutma önşekillendirme uygulaması için kullanılmasına rağmen, parmak uç noktalarının verildiği el animasyonuna gerek duyulan bir çok başka uygulamalarda da kullanılabilir.

Anahtar Kelimeler : Animasyon, Elle Tutma Önşekillendirmesi, Gerçek Zamanlı, 3B JAVA

To My Family

ACKNOWLEDGEMENTS

I express my sincere appreciation to my thesis supervisor Prof. Dr. Uğur Halıcı and co-supervisor Ass. Prof. Dr. İlkay Ulusoy for their guidance and insight throughout the research.

I wish to thank to my sister Derya Bengü Güler for their support and encouragement throughout the years of my education.

I would like to thank to all of the Computer Vision and Artificial Intelligence Research Lab. Group members. Particularly, Erdem Akagündüz, and Aykut Tokatlı for their invaluable feedback, support, help and motivations.

I also would like to thank to ASELSAN A.Ş. Communication Division System Engineering Department for the support during the thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1. Motivation

Human animation is still one of the most important research fields after a decade of development in animation techniques. In the near future, we expect any human individual to be modelled and placed in a virtual environment in which any human behaviour can be simulated. Moreover, the synthetic humans created by using computer animation tools and living in virtual world can communicate with people in the real world. The use of the hands is one of the most significant aspects of a human being. The large degrees of freedom in the hands are one of the major problems for their motion control.

Most computer graphics research on hand motion has focused on preshaping and gestures with application to areas of human computer interaction and sign language. There are also a number of educational applications such as sign languages, typing, playing of musical instruments etc. From a computer graphics standpoint, these applications are difficult in animation of hand realistically.

This thesis aims to animate 3D hand preshaping activity for a chosen virtual 3D object in real-time and to show hand configurations and hand joint angles. Researches on human hand kinematics, structure and geometric stability analysis on preshaping are the main motivation for the algorithms developed in this work for animating 3D virtual hand preshaping.

1.2. The Scope of The Thesis

The aim of this thesis is to animate 3D hand preshaping activity realistically for a chosen virtual 3D object in real-time. In order to develop a realistic animation algorithm, human hand, classification of preshaping types, human hand kinematic constraints, kinematics, optimization, geometric stability analysis and geometric constraints of preshaping have been investigated.

The algorithm that we developed is made of two main parts. The first part is related with the precision type preshaping. The first step in the precision type preshaping algorithm is to find proper contact points, restricted by the constraints, on the object. After finding a set of contact points, program will search new contact points set which is the best suited for the chosen preshaping. This search based on the preshaping quality index. These contact points are compared and ranked according to preshaping quality measure. After determination of contact points, hand joint angles and hand base location with respect to these points are calculated using kinematics techniques. Constructed 3D skeleton hand model is animated in JAVA 3D by using the calculated angles and positions.

Human hand preshaping dexterity is very complex and cumbersome, therefore preshaping classification is needed to adapt this behaviour to computer graphics. Cutkosky preshaping taxonomy (1986) was used to classify preshaping types. According to Cutkosky, there are four preshaping types: wrap, lateral, pinch and tripod. These are used to preshape primitive objects such as box, cylinder and sphere in the scene. Preshaping type decision is left to the user so that user can decide appropriate type of preshaping for a chosen object. Indeed, preshaping type is needed in order to determine which fingers will be used to preshape the object.

Wrap type preshaping differs from other types. Therefore new strategies are investigated. Wrap is one of the powerful preshaping types, requires maximum contact between hand and object. Algorithm to be developed should maximize the intersection between hand and object. Hence, we use collision detection schema to animate this type of preshaping. In this part of software, collision information between object and hand is used as a feedback to develop algorithm.

Our concern is to show hand's joint angles, finger positions, bone structure when making realistic preshaping therefore skeleton hand model is used as a 3D hand model.

1.3. Organization of The Thesis

The organization of the thesis is as follows;

- Chapter 2 presents a literature review on the human preshaping analysis and animation.
- Chapter 3 gives the necessary information related with 3D hand model. Also the anatomy of a generic hand is explained. Kinematics transformation matrix is presented.
- Chapter 4 gives the necessary information related with the preshaping analysis, and quality measure index in this thesis.
- Chapter 5 explains developed animation algorithm.
- Chapter 6 gives the details of implementation together with all generated outputs.
- Chapter 7 summary and conclusion and also possible future enhancements are presented.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Realistic human animation is a relatively new topic. The walking, preshaping and also some organ animations studies appeared in computer graphics area recently. On the other hand, preshaping is an established topic in robotics. The industrial applications of robot hands met several decades ago. There is a number of works on robotics explaining how to preshape for a given object. Many of these works are based on and inspired by human preshaping analysis. Robotics studies mainly concerns with the stability analysis, preshaping planning, and object recognition and robot hand design.

Mishra and Silver (1989) separated the previous work on preshaping into higher level physiological studies of the human hand and lower level studies of the robotic hands.

2.2. High Level Preshaping Analysis

One of the major difficulty of preshaping is the high degree of freedom of hands. This flexibility gives rise to an enormous set of possible hand configurations. A lot of studies in the medical and robotics community on the preshaping capabilities of the human hand, from the anatomical and functional points of view have been performed. However, in choosing their own preshaping, humans unconsciously simplify the task to selecting one of only a few different prehensile postures appropriate for the object and for the task to be performed. Medical literature has attempted to classify these postures into preshaping taxonomies as seen in Schlesinger (1919) and Taylor and Schwarz

(1955) associate human preshaping primarily with the object shape in their categorization of six preshaping (cylindrical, fingertip, hook, palmar, spherical and lateral). Griffiths's (1943) preshaping classification is also based on objects of varying form. He partitions the functions of the hand into cylinder preshape, ball preshape, ring preshape, pincer preshaping and plier preshape. McBride (1942) took a different approach in dividing the function of the hand: his classification depends on the parts of the hand which participate in the preshaping (preshaping with the whole hand, preshaping with thumb and fingers, and preshaping with finger and palm).

These classifications, while expressive and intuitively informative, do not reflect a fundamental analysis of the hand as an entity. They are also dependent on the shape of the preshaped object.

Napier (1956) proposed well known preshaping taxonomy taking into the considerations missing parts of previous studies. His work divides preshaping into two main parts, power and precision preshaping. His classification of preshaping is based on the purpose of the task, shape and size of the object, and the posture of the fingers. This division of preshaping into precision and power preshaping is the most widely accepted on today and used by researchers in the medical, biomechanical and robotic fields.

A power preshaping is used for higher stability and security at the expense of object manoeuvrability, while the converse is true for a precision preshape. A precision preshaping is characterized by a small degree of contact between the hand and the object. In this type of preshape, the object is normally pinched between the thumb and the flexor aspects of at least one finger. In a power preshape, however, the object is held tight by the fingers and the palm. The major classifications of a power preshaping are the cylindrical power preshaping and the spherical power preshape. In a cylindrical power preshape, the thumb can either be adducted for some element of precision, or abducted for more clamping action on the object. Henceforth the cylindrical power preshaping refers to the former type while the "coal-hammer" preshaping refers to the latter type.

Cutkosky and Wright (1986) extended this classification to the types of preshaping needed in a manufacturing environment and examined how the task and object geometry affect the choice of preshape. Their tree-like classification can be seen in Figure 2-1. At the lowest level, a preshaping is chosen based on object geometric details and task requirements. However, not only is the taxonomy incomplete, but also there may exist problems in categorizing preshaping in intermediate cases (e.g., the shape of the object is somewhere between being strictly prismatic and strictly spherical) because the preshaping classification is discrete. In these cases, determination of the type of preshaping will then be dependent mostly on human judgment rather than on reasoning.

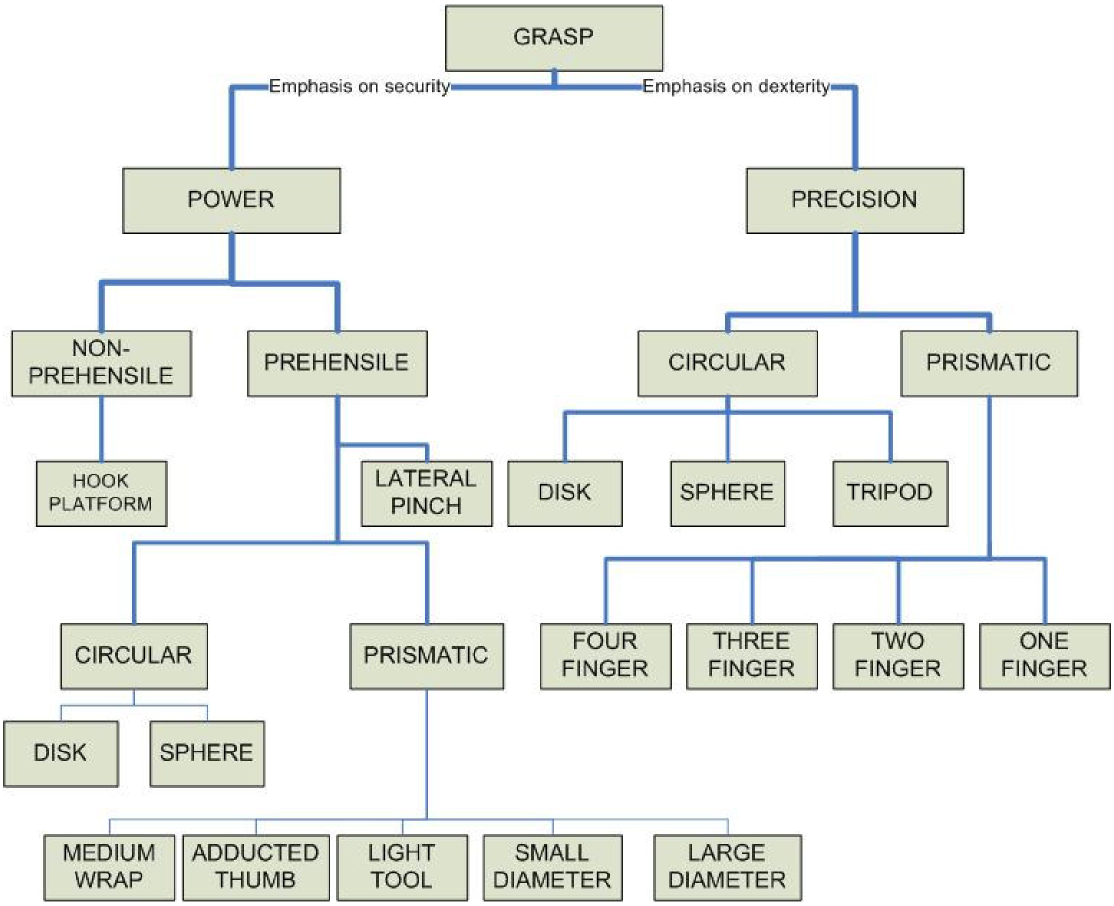


Figure 2-1 Cutkosky and Wright's (1986) taxonomy of human preshaping

Iberall (1987), (1997) observed that this classification too rigid, since in practice, the human hand often uses a combination of preshaping to accomplish a task. She defined preshaping with respect to two virtual fingers which apply opposing forces on the objects, and only later maps these virtual fingers onto physical fingers based on object characteristics. According to Iberall, human preshaping can be analysed by three oppositions;

1. Pad opposition, which is between the thumb and finger pads and used for precision type preshaping.
2. Palm opposition, which is between the palm and the finger bones and used for power type preshaping.
3. Side opposition, which is between the thumb and the side of the index finger. It constitutes compromise between the flexibility of the pad opposition preshaping and the stability of the palm opposition preshaping.

Lyons (1985) uses the concept of the virtual fingers in his development of a preshaping index that selects a preshaping on the basis of two object characteristics, shape and size, whether the preshaping should be firm or not and whether the preshaping should be precise or not. Unfortunately, his categories are quite broad and make it difficult to create a preshaping to specific objects.

Stansfield (1991) built these classifications into a rule based system that, when given a simplified object description from a vision subsystem, will provide a set of possible hand preshapes and reach directions for the pre-contact stage of preshaping. However, many problems are left unsolved. She only examines five possible approach directions, she does not try to choose the best preshaping from this set of possibilities, and for any preshaping that is chosen, the hand simply closes its fingers; no attempt is made to optimize the preshaping for stability.

Pao and Speeter (1989) developed a method which transforms that human hand poses to poses of the robotic hand by using a DataGlove to measure the joint angles of a human hand. They were able to recreate a variety of poses with the model hand. Speeter (1991) later created HPL, Hand Programming Language, which simplifies the problem of coding robotic preshaping and dextrous manipulation tasks. The language consists of a number of motion primitives that are related to common human preshaping and manipulation motions, providing a high-level abstraction of the preshaping process.

2.3. Lower Level Preshaping Analysis

Humans have evolved as the dominant species on the planet in part because of their skills in dexterous manipulation using their multifingered hands. The human hand is used in a variety of ways. In particular, the three most important functions are to explore, to restrain objects, and to manipulate objects. Because of the dexterity of hands, preshaping has a huge interest in robotics area. Some of works on preshaping in robotics area based on human behaviours, some of based on analytical analysis.

This thesis addresses several areas so it is worth to mention important issues of multifingered robotic manipulation. We have used geometric stability and quality measure index studies to form algorithm in this thesis.

In this section, we give a brief overview of the developed artificial robotic hands and the related research works.

2.3.1. Multifingered Robotic Hands

Over the past decades, there have been many activities in the design, analysis, and control of artificial multifingered hands to emulate the functions of the human hand.

In most of the traditional industrial applications, a robot arm is equipped with a simple parallel jaw gripper as its end effector. Such gripper has three major limitations in view of its ability to perform advanced autonomous tasks including:

1. Lack of flexibility: A simple gripper can only preshaping those objects with parallel and planar surfaces. It cannot easily preshaping the objects of arbitrary shapes with uneven surfaces.

2. Lack of dexterity: A simple gripper can preshaping an object but cannot manipulate it. Small reorientations of the object cannot be performed by the gripper alone. Hence, the entire arm has to be moved. Making fine motions by moving the entire arm is often difficult and time consuming due to the dominant effects of inertia and friction.
3. Lack of sensing ability: There are no sensors on the surface of the gripper. Structural properties of the preshaped object, such as the surface texture, cannot be inferred via such gripper.

The motivation of studying multifingered robotic hand comes from the desire to overcome the limitations of parallel jaw grippers and admiration of the dexterity, sensing ability, and versatility of human hand. To emulate the dexterity and sensing ability of human hand, several articulated multifingered robotic hands have been developed as research tools to study multifingered manipulation ((T. Okada, 1979), (M. Buss and K. Kleinmann, 1996), (G. Bekey, R. Tomovic, and I. Ziljkovic, 1990), (S. Jacobsen, J. Wood, K. Bigger, and E. Inversen, 1986) and (J. K. Salisbury, 1985)). The famous anthropomorphic (resembling human characteristics) hands are Salisbury hand (J. K. Salisbury, 1985) and Utah/MIT hand (S. Jacobsen, J. Wood, K. Bigger, and E. Inversen, 1986).

Figure 2-2 Salisbury Hand

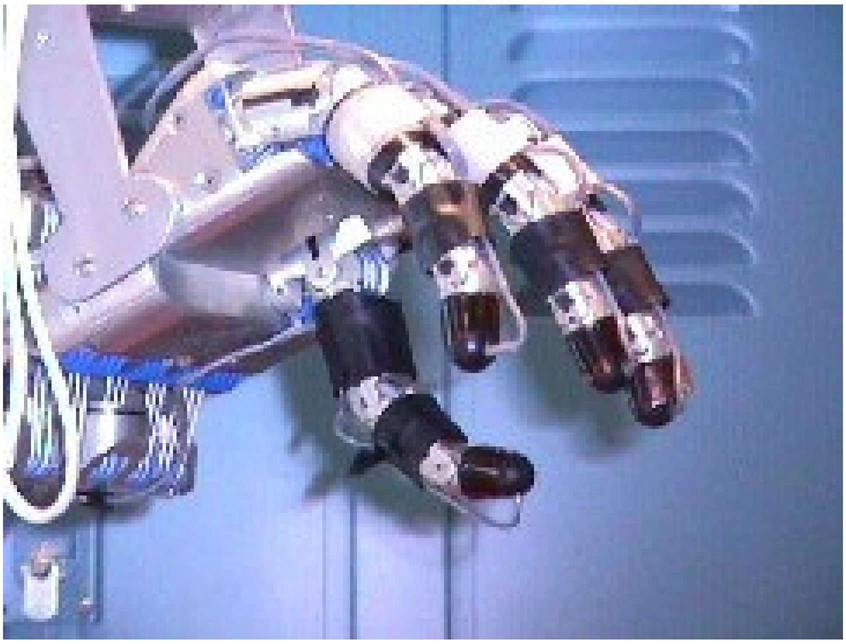


Figure 2-3 UTAH/MIT Hand

These two hands have become the standards for researchers involved in robot hand algorithm development and laboratory experimentation, particularly in the USA. The Salisbury hand (Figure 2-2) has three fingers and each finger has three degree-of-freedom. Its joints are all tendon driven. The placement of the fingers consists of one finger (the thumb) opposing the other two. The Utah/MIT hand (Figure 2-3) has four fingers (three fingers and a thumb) in a very mankind configuration; each finger has four degree-of-freedom and the joints are also tendon driven. Other hands of note include the Darmstadt hand ((M. Buss and K. Kleinmann, 1996)), the Karlsurhe hand ((G. Wohlke, 1990)), the Belgrade-USC hand ((G. Bekey, R. Tomovic, and I. Ziljkovic, 1990)), the DLR hand ((H. Liu, P. Meusel, J. Butterfass, and G. Hizinger, 1998), (H. Liu, J. Butterfass, S. Knoch, P. Meusel, and G. Hirzinge, 1999) and (J. Butterfass, G. Hizinger, S. Knoch, and H. Liu., 1998)) and the MEL hand ((H. Maekawa, 1992), (H. Maekawa, K. Tanie, and K. Komoriya, 1995, 1997)).

However, the aforementioned robotic hands are all of small size and can only preshaping and manipulate small objects. It is difficult to actuate the joints of the hands and to install sensors on these hands. In view of this, a large three-fingered robotic hand consisted of three industrial Motoman robots are developed in the Robot Manipulation Laboratory of the Hong Kong University of Science and Technology. It is called the HKUST hand.

Numerous types of sensors have been developed and implemented on robotic hands. For multifingered manipulation, in addition to joint position sensors (encoders, potentiometers, etc.), tactile, force/torque and vision sensors have been developed or utilized to sense contact location and contact forces ((I. McCammon and S. Jacobsen, 1990), (R. Howe and M. Cutkosky,1993), (R. Fearing, 1989) and (P. Allen, A. Miller, P. Oh, and B. Leibowitz., 1997)). Although tactile sensing technology is improving rapidly, it will be a long time before robotic hands can rival human hands for sensor quantity and variety. This lack of sensor richness has proved an obstacle to robotic hand development. However, numerous creative solutions are being developed. Different types of tactile sensors have been developed. A tactile sensor is defined to be a device which measures parameters of a contact interaction between the device and some physical stimulus. The interaction is normally confined to a touch sensitive region of the device's surface. Such a sensor may simply detect presence or absence of touch, whilst a more complex tactile sensor may provide data on the size, shape, position, thermal conductivity or distribution of a contacting object. Nicholls and Lee (1989, 1992) gave good summaries on tactile sensors. Among all kind of tactile sensors, capacitive tactile sensor is relatively robust, easy to construct and inexpensive. Three capacitive tactile sensors using capacitive strain measurement techniques have been developed. This development of the tactile sensors is modified from an earlier design of Fearing (1986).

2.3.2. Overview of Dexterous Manipulation

Preshaping and manipulation of object are two fundamental problems in study of robotic hand. Over the past decades, significant strides have been made in realizing features of multifingered manipulation. Shimoga presented a good survey on preshaping synthesis in (K. Shimoga, 1996). Bicchi and Kumar reviewed robotic preshaping and contact in (A. Bicchi and V. Kumar, 2000). Okamura, Smaby and Cutkosky gave an overview of research in dexterous manipulation in (A.M. Okamura, N. Smaby, and M.R. Cutkosky, 2000). Walker gave a survey of design, analysis, and control of artificial multifingered hands and corresponding research in the area of machine dexterity in (I.D. Walker, 1998). Bicchi summarized the evolution and the state-of-art in the field of robot hands in (A. Bicchi, 1996). He discussed in what state of the art of building artificial hands is at present times, and argued about possible directions it may take in the future.

2.3.2.1. Interaction Between Hand and Object

Given a particular robotic hand, the kinematic and dynamic (if desired) models of each finger can be readily obtained using techniques previously established for robot manipulators. However, modelling dextrous multifingered manipulation itself is not a trivial undertaking because of the essential difficulty in modelling the interaction between the finger and the object. The essential difference lies in the nature of the contacts between the fingers and the object. Multifingered manipulation is complicated by the fact that the fingertips are not solidly attached to the object, as in the typical multi-arm coordinated problem. The whole essence of dextrous manipulation lies in the ability of the fingertips to move relative to the object. This causes extra complications in the analysis of multifingered manipulation.

The first problem is to model the interaction between the finger and the object. Salisbury (1985) presented the suitable sets of unit basis twists and unit basis wrenches for each of the commonly encountered types of contact. There are three main contact models between fingertip and object including:

1. Point contact without friction
2. Point contact with friction
3. Soft-finger contact

Point contact without friction can only resist a unidirectional force normal to the surface. Adding friction allows fingertip to resist tangential forces, up to the friction limits. A soft fingertip can additionally resist moments about the surface normal. For point contact with friction and soft-finger contact, the standard “friction cone" defined by Coulomb friction determines the ratio of tangential to normal force that can be sustained without slipping. These contact models have been experimentally validated ((M. Cutkosky, P. Akella, R. Howe, and I. Kao, 1987)). A somewhat more complicated friction limit surface can similarly be defined for soft contacts ((N. Xydas and I. Kao., 1999), (R.D. Howe and M.S. Cutkosky, 1996)).

2.3.2.2. Kinematics

Dexterous manipulation is an area of robotics in which multiple fingers cooperate to preshaping and manipulate an object from an initial configuration to another. A distinguishing characteristic of dexterous manipulation is that it is object-centred. That is, the problem is formulated in terms of how the object is to be manipulated, how it should behave, and what forces should be exerted upon it. In keeping with an object-centred approach, the dexterous manipulation problem sets the framework for determining the required actuator force/torque to produce the desired motions of the object. This development requires knowledge of the geometric relationships of the dexterous manipulator-object system, including the contact locations, the object, fingertip and link geometries, and the finger kinematics.

Three important classes of kinematic relations underlying a multifingered manipulation system, among which (a) finger kinematics; (b) the preshaping map; and (c) the kinematics of contact, have been identified and thoroughly analyzed ((J. K. Salisbury, 1985), (C. Cai and B. Roth, 1987), (J. Trinkle, 1987), (D. Montana, 1998), (R. Murray, Z.X. Li, and S. Sastry, 1994),(J. Kerr and B. Roth, 1986) and (M. Cutkosky, 1985)).

Salisbury (1985) first defined the preshaping map which transforms the fingertip forces to the object frame such that the exerted fingertip forces can balance the object wrenches. Cai and Roth (1987) investigated the spatial motion of rigid bodies with point contact. The first investigation of manipulation with rolling contact was conducted by Kerr (1986). He discussed how to compute the movement of the fingers in order to produce a given displacement of the object. Kinematic equations are derived from the constraint that the fingertip and object velocities are equal at the point of contact. He formulated the kinematics of manipulation with rolling contact, namely the relationships between the motion of the fingers, manipulated object and the contact locations on both surfaces of the fingertips and the object. Cole et al. (1988, 1994) derived the kinematics of rolling contact for two surfaces of arbitrary shape rolling on each other. Maekawa et al. (1992, 1995, and 1997) investigated a new motion control system using tactile feedback for the manipulation of an object by a multifingered hand where the fingertip and the object make rolling contact. From a geometric point of view, Montana (1988, 1991, and 1995) formulated the kinematics of contact between the fingertips and object, which relates the contact velocities to the change rates of the local, coordinates of the fingertips and the object using their geometric parameters.

2.3.2.3. Preshaping Planning, Quality Measure and Optimization

Manipulators used for dexterous manipulation typically have kinematic redundancy. In addition, there are usually multiple choices for contact locations that will achieve force closure on an object. Furthermore, for each choice of contact locations, there are many solutions for applying contact forces that will satisfy the external force requirements while providing sufficient internal forces to prevent slipping. Therefore, there can be an infinite number of possible preshaping for a manipulation. The task of picking the "best" preshaping has resulted in a rich area of research. There are many different ways to choose the optimal contact locations, contact forces, and finger configurations for a particular hand, object and task combination. Preshaping planning and characterization of optimal preshaping incorporating task requirement have been extensively studied in (J. Trinkle, A. Farahat, and P. Stiller, 1995), (E. Rimon and J. Burdick, 1996), (D. Montana, 1991), (B. Mishra, J.T. Schwartz, and M. Sharir, 1987), (V. Nguyen, 1986), (A. Bicchi, C. Melchiorri, and D. Ablluchi, 1995) , (V. Nguyen, 1988), (Z.X. Li and S. Sastry, 1988). Dextrous manipulation with rolling contact constraints or finger gaiting has been investigated in ((Z. Li and J. Canny, 1990), (Z.X. Li, J.F. Canny, and S.S. Sastry, 1989), (D. Montana., 1995), (R. Murray and S. Sastry, 1990)) along with several useful algorithms for finger motion planning. Li, Canny and Sastry (1989) formulated the motion planning problem for dextrous manipulation and defined the hand map which maps the finger motion onto the object motion. The hand map gives an intrinsic characterization of the workspace of a multifingered robot hand. The defined hand workspace is an invariant associated with the kinematic structure of the hand and the object. Thus, it provides a criterion for evaluating designs of multifingered robot hands. Using the kinematic equation of contact, Li and Canny (1990) transformed contact constraints in the configuration manifold to a system of differential equations in the parameter space. They showed reachability for a sphere rolling on a plane, and for two spheres with different radius. They also proposed an algorithm to apply to adjusting contact configurations of a multifingered robot hand without slipping. Hong, Lafferriere, Mishra and Tan (1990) showed the existence of two and three finger preshaping in the presence of arbitrarily small friction for two and three dimensional smooth objects. They also proved the existence of finger gaiting for rotating a planar object using three and four fingers. Paljug, Yun and Kumar (1994) presented the planning and control for the coordination of multiple arms in manipulation tasks involving rolling contacts. They designed a planner to determine optimal contact point locations on the effector and the object for a given task. Based on nonlinear feedback that decouples and linearizes the system, they proposed a control algorithm which simultaneously controls the system trajectory, which includes the object trajectory as well as the trajectory of the contact points, and the constraint force in order to maintain rolling contact. Montana (1995) derived a configuration-space description of the kinematics of the fingers-plus-object system. He formulated contact kinematics as a "virtual" kinematic chain. The system can be viewed as one large closed kinematics chain composed of smaller chains, one for each finger and one for each contact point. He also proposed velocity-based approaches and discussed how to control the positions of the points of contacts for a two-fingered preshaping with soft-finger contact. Bicchi et. al. (1995) presented how to achieve dextrous manipulation capability of planning and controlling rolling motions of arbitrary objects.

Nguyen (1986, 1987) gave algorithms to find optimal planar preshaping and stable force closure preshaping. Mishra, Schwartz, and Sharir (1987) obtained bounds on the number of fingers needed to achieve positive and force closure preshaping on piecewise smooth objects. They assumed no friction but some of the results extend to arbitrarily small friction. Algorithms for the synthesis of such preshaping were also given in the case of polyhedral objects. Li and Sastry (1988) discussed the problem of optimal preshaping of an object by a multifingered robot hand. They also proposed three quality measures for evaluating a preshape. Montana (1995) derived a model of how the positions of the points of contact evolve in time on the surface of a preshaped object in the absence of any external force or active feedback. From this model, he obtained a general measure of the contact stability of any two-fingered preshape.

Based on rigid body mobility analysis, Rimon and Burdick (1996) addressed the problem on force and form closure for multiple finger preshaping. Bicchi, Melchiorri and Balluchi (1995) considered multiple robot systems for coordinated manipulation of objects. They analyzed mobility, different kinematics, velocity manipulability and velocity workspace of multiple robot system. Trinkle, Farahat and Stiller (1995) introduced the concept of first-order stability cells for spatial, quasi-static, multirigid-body systems with Coulomb friction acting at the contact points.

2.4. Kinematics in Computer Graphics

The area within computer graphics that makes extensive use of inverse kinematics is computer animation, in particular, the animation of articulated figures. An articulated figure is usually represented by a collection of kinematic chains connected together. Each joint in this articulated structure may have one, two, or three degrees of freedom. The degrees of freedom of an articulated structure increases with its complexity. As an example, a detailed approximation of the human skeleton may have in excess of two hundred DOF. Although well understood traditional animation techniques (J- Lasseter,

1987) help animators produce expressive motions in their animation, they require extensive manipulation of the figure to achieve the desired effects. It is obviously a very difficult task to create animation by manipulating joint angles to set up key frames that place end effectors of certain kinematic chains in desired locations. Multiple iteration of trial and error is generally required to produce the correct result. This approach is certainly very time consuming and error prone.

It is apparent that inverse kinematics offers an attractive solution to the above problem. Instead of letting the animator specify the joint angles that place the end effectors at a desired location, the computer automatically calculates these joint angles from the link configuration and the end effectors location specified by the animator. This technique was used by Girard and Maciejewski (1985) to build the PODA system which synthesizes the kinematic model of legged locomotion. Zhao and Badler (1989) proposed an algorithm that can incorporate various constraints and solve for simultaneous goals. Welman (1993) has presented two very distinct inverse kinematic algorithms suitable for real time manipulation and showed their effectiveness in a powerful interactive editor LifeForms. By formulating inverse kinematics into an optimization problem, Bawa (1995) has presented an algorithm which uses an iterative nonlinear constrained optimization algorithm for solving the inverse kinematics problem.

2.5. Computer Animation

Animation is the production of consecutive images, which, when displayed, convey a feeling of motion. Animated images are almost magical in their ability to capture our imagination. By telling a compelling story, astounding with special effects, or mesmerizing with abstract motion, animation can infuse a sequence of inert images with the illusion of motion and life. Creating this illusion, either by manually or with the assistance of computer software is not easy. Each individual image, or frame, in the animated sequence must blend seamlessly with the other images to create smooth and continuous motion that owns through time (Foley, 1990).

Traditionally, animation was created by drawing images of the characters for each frame in the action. At the start of the production, the animator is given storyboards, which are sketches depicting the sequence of major actions and illustrating the expressions of the characters. The animator also works from a finished soundtrack, which determines the timing of the piece. In older animations, the background scenery was often stationary and the characters were painted on cels, pieces of clear celluloid that could be stacked on top of the background. Most hand animation is created with keyframing where a lead animator creates the key, or most important frames, and a second animator creates the in between frames. Regardless of the medium, the challenge for the animator is to create images that impart expressiveness and life to the character.

The most basic computer animation tools assist the process of traditional animation by automatically generating some of the frames of animation. Animation tools have also been developed to composite together multiple layers of a scene in much the same way that layers of cels are used in manual animation. Other more powerful techniques make use of algorithms that render an image from a geometric description of the scene. These techniques change the task from drawing sequences of images to using computer tools to effectively specify how images should change over time.

In addition to providing tools that give the animator new capabilities, the computer also creates new applications for animation. Computer animations can be generated in real-time for use in video games and other interactive media. Combining puppeteering with computer animation allows a human operator to control an interactive character in a live performance. Realistic rendering and animation techniques enable the creation of digital actors that can be seamlessly blended with real world footage.

A wide variety of techniques are used in the process of creating a complex computer animation. These techniques can be grouped into two main classes: two dimensional (2D) and three dimensional (3D). Although there is some overlap between the two classes, 2D techniques tend to focus on image manipulation while 3D techniques usually build virtual worlds in which characters and objects move and interact (Taylor, 1996).

2.5.1. Two-dimensional Animation

Two-dimensional (2D) animation techniques contribute a great deal to computer animation by providing the tools used for sprite-based animation, blending or morphing between images, embedding graphical objects in video footage, or creating abstract patterns from mathematical equations.

The most common form of 2D animation is sprite animation. A sprite is a bitmap image or set of images that are composited over a background, producing the illusion of motion. They are usually small with respect to the size of the screen. For example, to animate a rabbit hopping across a meadow, the animator would create a sequence of images showing poses of the rabbit hopping. This sequence of images would then be composited one image per frame onto a background image of the meadow. Sprite-based animation can be done extremely quickly with current graphics hardware, and thus many elements of the scene can be moving simultaneously. The disadvantage of this technique is that the sprites come from a fixed library and subtle changes in lighting and depth cannot be reproduced. Consequently, sprite animation is most often used in interactive media where rendering speed is more important than realism.

Morphing refers to animations where an image or model of one object is metamorphosed into another. Morphing is remarkable because it provides a startling yet convincing transformation of one image into another. Unfortunately, morphing is labour intensive because the key elements of each image must be specified by manually, although automatic feature detection is an area of active research.

Mathematical equations are often used to create abstract motion sequences. When the values of the mathematical functions are mapped to colour values and varied with time, the motion of the underlying structures can be quite beautiful. Fractals are a well-known example of functions that create attractive patterns.

Morphing and the generation of abstract images from mathematical equations can be generalized for use in 3D. All of these 2D techniques can be used either on their own to create an animation or as a post-processing step to enhance images generated using other techniques.

2.5.2. Three-dimensional Animation

Three-dimensional animation involves constructing a virtual world in which characters and objects move and interact. The animator must model, animate, and render the 3D scene. Modelling involves describing the elements of a scene and placing them appropriately. Animation specifies how the objects should move in the 3D world (Kerlow, 1996). Rendering converts the description of the objects and their motion into images. Modelling and rendering are, for the most part, independent of their role in the animation process but a few necessary modifications are described below.

To animate motion, the user needs both a static description of an object and information about how that object moves. One common way to specify this additional information is to use an articulated model such as the one shown in Figure 2-4. An articulated model is a collection of objects connected together by joints in a hierarchical, tree-like structure. The location of an object is determined by the location of the objects above it in the hierarchy. For example, the motions of the elbow joint in a human model will affect not only the position of the lower arm but also the position of the hand and fingers. The object at the top of the hierarchy, or the root of the tree, can be moved arbitrarily to control the position and orientation of the entire model.

A second type of model used in animation is a particle system or collection of points. The motion of the particles through space is determined by a set of rules. The laws of physics often provide a basis for the motion so that the particles fall under gravity and collide with other objects in the environment. Systems that are modelled well by particle systems include water spray, smoke, and even flocks of birds.

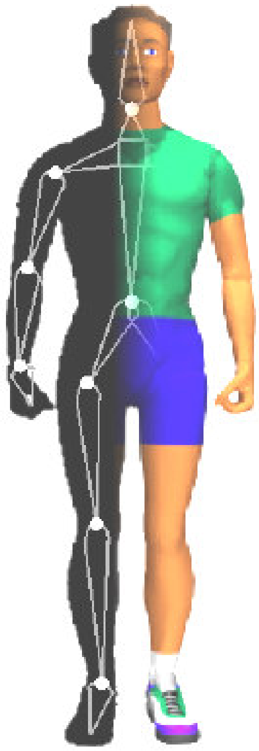


Figure 2-4 An articulated model of a human male
The structure of the joint hierarchy is shown on the left. The graphical model used for rendering is shown on the right. Image courtesy of the Graphics, Visualization and Usability Centre, Georgia Institute of Technology

Deformable objects are a third type of model and include objects that do not have well-defined articulated joints but nevertheless have too much structure to be easily represented with a particle system. Because of the broad nature of this class, there are several fundamentally different ways to represent deformable objects, including spring-mass lattices, volumetric models, and surface representations. Water, hair, clothing, and fish are among the systems that have been successfully modelled as deformable objects.

While each of these model types can be used to describe a wide variety of objects, complex systems often require hybrid models that combine two or more types. This approach allows each part of the system to be modelled by the most appropriate technique.

2.5.3. Motion Generation

The task of specifying the motion of an animated object to the computer is surprisingly difficult. Even animating a simple object like a bouncing ball can present problems. In part, this task is difficult because humans are very skilled at observing motion and quickly detect motion that is unnatural or implausible. The animator must be able to specify subtle details of the motion to convey the personality of a character or the mood of an animation in a compelling fashion.

A number of techniques have been developed for specifying motion, but all available tools require a tradeoff between automation and control. Keyframing allows fine control but does little to automatically insure the naturalness of the result. Procedural methods and motion capture generate motion in a fairly automatic fashion but offer little control over fine details.

2.5.3.1. Keyframing

Borrowing its name from the traditional manual animation technique, keyframing requires the animator to outline the motion by specifying key positions for the objects being animated. In a process known as in-betweening, the computer interpolates to determine the positions for the intermediate frames. The interpolation algorithm is an important factor in the appearance of the final motion. The simplest form of interpolation, linear interpolation, often results in motion that appears jerky because the velocities of the moving objects are discontinuous. To correct this problem, better interpolation techniques, such as splines, are used to produce smoothly interpolated curves.

The specification of keyframes can be made easier with techniques such as inverse kinematics. This technique aids in the placement of articulated models by allowing the animator to specify the position of one object and have the positions of the objects above it in the articulated hierarchy computed automatically. For example, if the hand and torso of an animated character must be in particular locations, an inverse kinematics algorithm could determine the elbow and shoulder angles. Commercial animation packages include inverse kinematics and interpolation routines designed specifically for animating human figures. These tools take into consideration such factors as maintaining balance, joint angle limitations, and collisions between the limbs and the body. Although these techniques make animation easier, keyframed animation nevertheless requires that the animator intimately understand how the animated object should behave and have the talent to express that behaviour in keyframes.

2.5.3.2. Procedural Methods

Current technology is not capable of generating motion automatically for arbitrary objects; nevertheless, algorithms for specific types of motion can be built. These techniques are called procedural methods because a computer follows the steps in an algorithm to generate the motion. Procedural methods have two main advantages over keyframing techniques: they make it easy to generate a family of similar motions, and they can be used for systems that would be too complex to animate by manually, such as particle systems or flexible surfaces.

Physically based simulation refers to a class of procedural methods that makes use of the laws of physics, or an approximation to those laws, to generate motion. Simulated motion will be realistic if the model captures the salient physical characteristics of the situation. For many applications, this realism is an advantage. Unfortunately, building a new simulation is sometimes a difficult process requiring an in-depth understanding of the relevant physical laws. Once a simulation has been designed, however, the animator may use it without understanding the internals of the simulation.

Simulations can be divided into two categories: passive and active. Passive systems have no internal energy source and move only when an external force acts on them. Passive systems are well suited to physically based simulation because the motion is determined by the physical laws and the initial conditions of the system. Pools of water, clothing, hair, and leaves have been animated using passive simulations.

Active systems have an internal source of energy and can move of their own volition. People, animals, and robots are examples of active systems. These systems are more difficult to model because in addition to implementing the physical laws, the behaviour of the simulated muscles or motors must be specified. An additional algorithm, a control system, must be designed to allow the model to walk, run, or perform other actions. For example, a control system for standing contains laws that specify how the hips and knees should move to keep the figure balanced when one arm is extended out to the side. Control systems can be designed manually for figures with the complexity of a 3D model of a human. For slightly simpler systems, they can be designed automatically using optimization techniques. After a particular control system has been built, an animator can use it by giving high-level commands such as stand, walk fast, or jump without understanding its internal details. To compute the running motion, the animator specifies the desired velocity and a control system generates the motion. The runner's clothes are a passive cloth simulation. Procedural methods can also be used to generate motion for groups of objects that move together. Flocks of birds, schools of fish, herds of animals, or crowds of people are all situations where algorithms for group behaviours can be used.

The main advantage procedural methods have over other techniques is the potential for generating interactive behaviours that respond precisely to the actions of the user. In a video game, for example, predicting the behaviour of the game player in every situation is impossible, but the characters should appear to be reacting to the actions of the player. Procedural methods allow this capability by computing a response in real-time. While methods using keyframing can also respond to the player, they can only do so by picking from a fixed library of responses.

Although procedural methods are currently computationally too expensive to generate motion in real-time for complicated scenes, advances in computer technology may render this possible.

The automatic nature of simulation has a cost in that the animator is not able to control the fine details of the motion. As a result, characters often lack expressiveness or individuality in their motions. Creating tools to allow the animator to control these aspects of a character is a topic of current research.

CHAPTER 3

HUMAN HAND MODELLING

3.1. Human Hand Anatomy

Hand consists of five fingers and palm. Each of the index finger, middle finger, ring finger and little finger has three joints. The joint closest to the palm is called the metacarpophalangeal joint or the MCP joint for short. The remaining two joint are the proximal interphalangeal (PIP) joint and distal interphalangeal (DIP) joint respectively (Pernkopf's Anatomy).

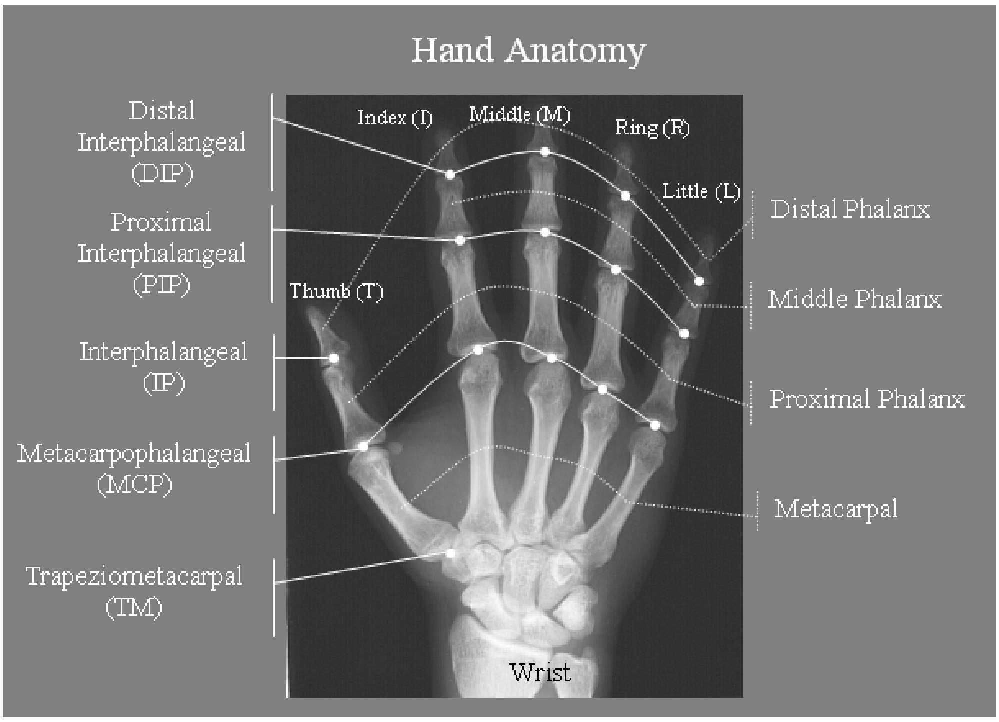


Figure 3-1 A hand skeleton observed from palmar side (Pernkopf's Anatomy)

The thumb is much more dexterous and therefore much more complex than the other fingers. The thumb's proximal joint is known as the Trapeziometacarpal (TM) joint. The next joint is the metacarpophalangeal (MCP) joint and the last joint is the interphalangeal (IP) joint (see Figure 3-1)

In the literature, the 9 IP joints are described as having only 1 DOF, flexion-extension. All 5 MCP points are described as saddle joints with 2 DOF's: abduction-adduction (e.g., spreading fingers apart) in the plane defined by the palm, and flexion-extension. The thumb is more difficult to model. Most of its flexibility lies in the Trapeziometacarpal (TM). This is another saddle point with 2 DOF's (same as above). As the last, the wrist's twist movement can be modelled by three DOF's (i.e., 3 DOF's for wrist rotation). The wrist's twist movement is included because the hand must be considered separately from lower arm. According to these joint movements' classifications, 24 DOF exist for the hand, including to position and orient it. (See Figure 3-2)

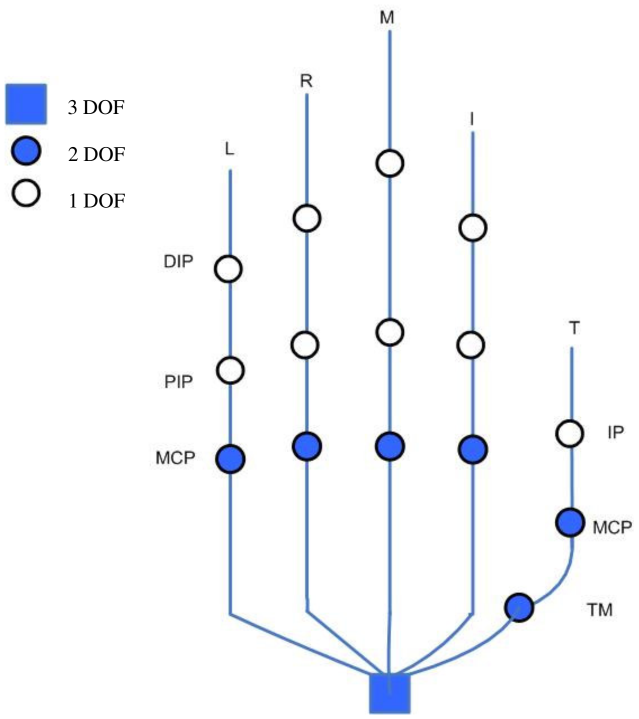


Figure 3-2 3D Skeletal Model with DOF's

The measurement and properties of fingers and hand are necessary for static and dynamic analysis of hand and finger movement, which include the lengths of the segments, the weights of the segments. Finger bone lengths ratios are shown Table 3-1.

Table 3-1 Averaged phalangeal lengths as % of hand length (Davidoff 1993)

Phalanx	Proximal	Medial	Distal
Thumb	17.1	-	12.1
Index	21.8	14.1	8.6
Middle	24.5	15.8	9.8
Ring	22.2	15.3	9.7
Little	17.2	10.8	8.6

Human hand modelled with respect to flexion/extension and adduction/abduction motions. These motions can be explained as follows ;

- Flexion - Bending movement that decreases the angle between two parts. Bending the elbow, or clenching a hand into a fist, is examples of flexion. When sitting down, the knees are flexed. Flexion of the hip or shoulder moves the limb forward (towards the anterior side of the body).
- Extension - The opposite of flexion; a straightening movement that increases the angle between body parts. In a conventional handshake, the fingers are fully extended. When standing up, the knees are extended. Extension of the hip or shoulder moves the limb backward (towards the posterior side of the body).
- Adduction - A motion that pulls a structure or part towards the midline of the body, or towards the midline of a limb. Dropping the arms to the sides, or

bringing the knees together, is examples of adduction. In the case of the fingers or toes, adduction is closing the digits together. Adduction of the wrist is called ulnar deviation (Figure 3-4).

- Abduction - A motion that pulls a structure or part away from the midline of the body (or, in the case of fingers and toes, spreading the digits apart, away from the centerline of the hand or foot). Abduction of the wrist is called radial deviation. Raising the arms to the sides is an example of abduction (Figure 3-5)

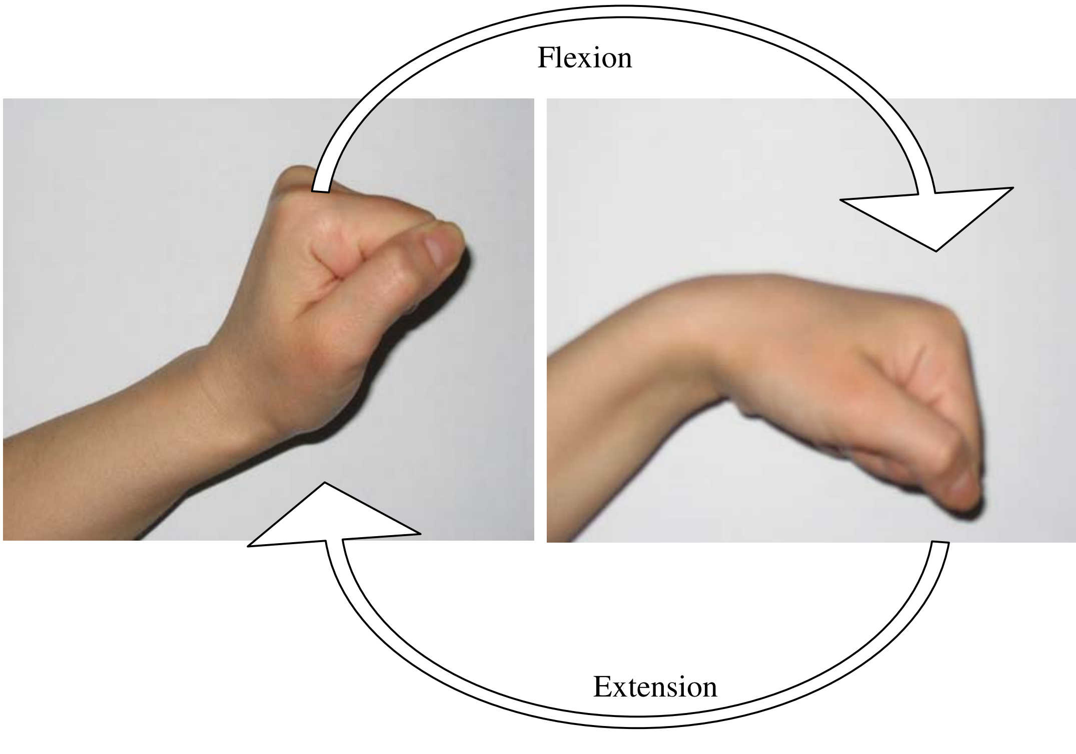


Figure 3-3 Wrist flexion/extension motion

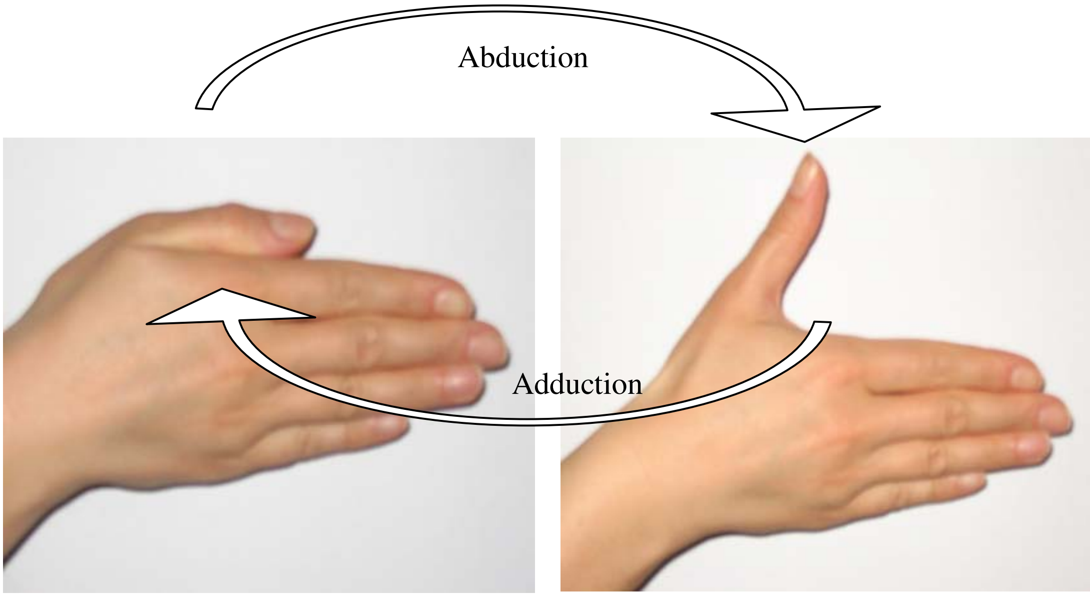


Figure 3-4 Thumb abduction/adduction motion

3.2. Human Hand Constraints

Human hand is complex mechanical structure comprising bones, ligaments loosely connecting bones, muscles serving as tension motors, tendons acting as cables connecting muscles to bone, and a covering of protective soft tissue and skin. The bones are linked at the joints and do not change in size. Muscles produce torque and/or joint movements through tension and for every muscle there exists one or more muscles that serve to opposite it through counter torque and/or opposing motion. Figure 3-6 illustrates the skeleton of right hand. In the modelling human hand, analysis of constraints is essential for the avoiding unrealistic motions during hand animation.

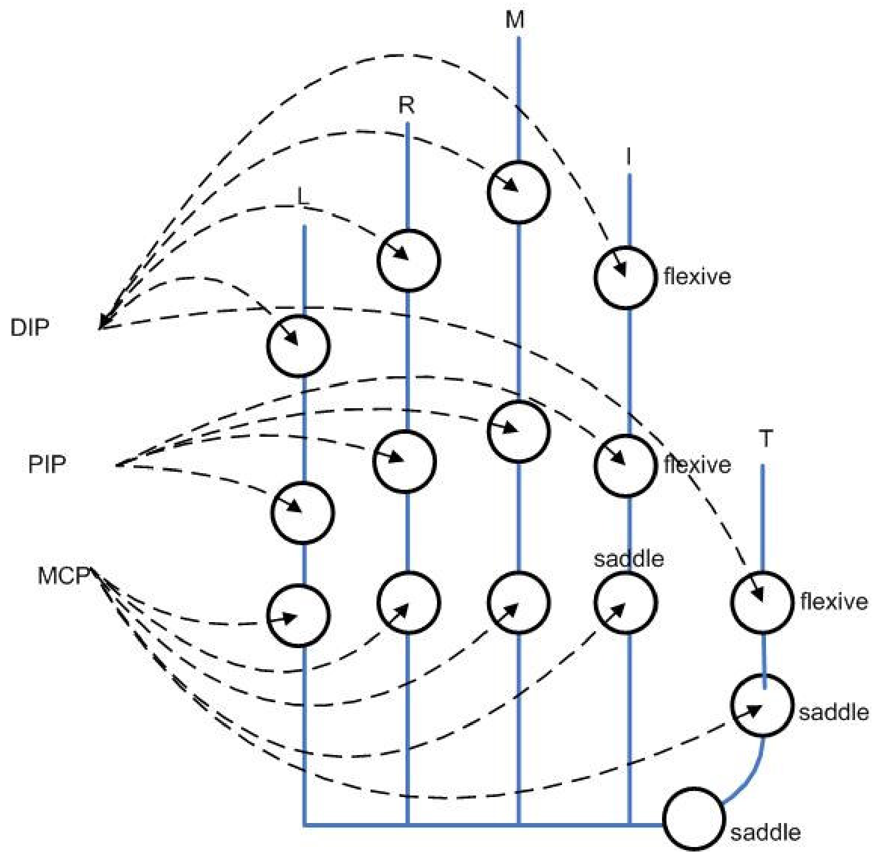


Figure 3-5 Joints of the hand and their movement types

Normally, movements of the finger joints are coordinated by constraints that make some configurations impossible. We therefore analyzed and incorporated some prominent constraints into the hand model, broadly classified as follows:

1. Joint angle limits and movement types

In Figure 3-1, note that possible movement of the MCP joint of fingers I, M, R and L are only flexion / extension or abduction / adduction and that of the PIP and DIP joints are only flexion / extension in the same direction. Hence the Distal, Middle and Proximal Phalanx segments occupy the same plane.

Although the allowable ranges of the joint angles vary slightly form person to person, they fall into general ranges (J. Lee and T. Kunii, 1995). Here, we distinguish between passive and active movements. The former movement is externally forced, whereas the latter is activated by tendons and muscles of the hand without external interaction. Joints generally have a greater range for passive movement. We only consider active hand motions.

2. Flexion of interphalangeal joints

In the human finger, it is nearly impossible to move the DIP without moving the adjacent PIP joint vice versa. Namely, active movement involves a dependency between the DIP and PIP joints (Landsmeer J., 1963). Anatomical studies have been directed at this phenomena and an empirical study (H. Rijpkema and M. Girard, 1991) revealed that an almost linear relationship exists between these two joints.

The joint angles of the DIP and PIP joints have a dependency represented by

$$\theta_{DIP} = 2/3\ \theta_{PIP}$$

3. Flexion of the metacarpophalangeal joints

The MCP joint has a flexion range of 90 degrees, slightly less for index finger (I) and progressively increasing for fingers M, R and L.

However, since isolated flexion of a finger is restricted by accompanying tension in the palmar interdigital ligament, such flexion might be cause flexion of the adjacent fingers. In the same way, a finger's extension is hindered by flexion of others. After measuring several people, this behaviour could be reasonably approximated through inequalities. The joint angle limits of the MCP joints depend on those of the neighbouring fingers according to the following inequalities.

$$\theta_{MCP(M)}+25 > \theta_{MCP(I)} > \theta_{MCP(M)}-54$$

$$\theta_{MCP(I)}+54 > \theta_{MCP(M)} > \theta_{MCP(I)}-25$$

$$\theta_{MCP(R)}+20 > \theta_{MCP(M)} > \theta_{MCP(R)}-45$$

$$\theta_{MCP(M)}+45 > \theta_{MCP(R)} > \theta_{MCP(M)}-20$$

$$\theta_{MCP(L)}+48 > \theta_{MCP(R)} > \theta_{MCP(L)}-44$$

$$\theta_{MCP(R)}+44 > \theta_{MCP(L)} > \theta_{MCP(R)}-48$$

I, M, R and L in formulas denote fingers. Finger's MCP joint angles should satisfy the above inequalities (J. Lee and T. Kunii, 1995).

4. The limits of the range of finger motions

We will only consider the range of motion of each finger that can be achieved without applying external forces such as bending fingers backward using the other hand. This type of constraints is usually represented using the following inequalities (Lin, Wu, and Huang, 1998).

Table 3-2 Finger Joint Angle Limits

	Flexion/Extension	Flexion/Extension	Flexion/Extension	Abduction/Adduction
Little	$-30° \leq \theta_{MCP} \leq 90°$	$0° \leq \theta_{PIP} \leq 110°$	$0° \leq \theta_{DIP} \leq 90°$	$-10° \leq \theta_{MCP} \leq 40°$
Ring	$-30° \leq \theta_{MCP} \leq 90°$	$0° \leq \theta_{PIP} \leq 110°$	$0° \leq \theta_{DIP} \leq 90°$	$-10° \leq \theta_{MCP} \leq 20°$
Middle	$-30° \leq \theta_{MCP} \leq 90°$	$0° \leq \theta_{PIP} \leq 110°$	$0° \leq \theta_{DIP} \leq 90°$	$-15° \leq \theta_{MCP} \leq 15°$
Index	$-30° \leq \theta_{MCP} \leq 90°$	$0° \leq \theta_{PIP} \leq 110°$	$0° \leq \theta_{DIP} \leq 90°$	$-20° \leq \theta_{MCP} \leq 10°$
Thumb	$-10° \leq \theta_{TM} \leq 30°$	$0° \leq \theta_{MCP} \leq 60°$	$0° \leq \theta_{IP} \leq 90°$	$-10° \leq \theta_{TM} \leq 100°$

3.3. Human Hand Kinematics

The human hand is a remarkably complex mechanism, and researchers have made various approximations when modelling it, depending on the application. This section investigates a human kinematic hand model forming a base for our precision type preshaping study. Human hand kinematics structure should be investigated because joint angles and finger-tip positions of human hand will be calculated with respect to kinematics structure of hand. An overview of research into modelling the human hand is presented in previous chapter. For the preshaping application here, in order to simplifying the problem, modelling of the tendons or external appearance is left and out of scope of this thesis.

This section develops a sufficiently accurate kinematics model of the human hand for preshaping. Using the models developed by Rohling and Hollerbach (1993) and Kramer (1996), we have developed a kinematics model suited for measuring and displaying fine fingertip preshaping. In this model, the human hand is converted to a mechanical linkage, with finger bones (as the links) connected by pin joints. The model does not take into account effects such as soft tissue deformation or bone-on-bone sliding, because these effects are not observable and assumed to cause little error in the estimated tip position. For convenience, the base coordinate system shown in Figure 3-7 is located in the hand at the point where the thumb and the index metacarpal meet. (In the Figure, the X_0, Y_0, Z_0 system is displaced from this point for clarity.) The base frame x-axis points along the index metacarpal bone, the y-axis is directed outward from a flat open palm, and the z-axis is defined by the right hand rule.

The index finger is defined similarly to that presented in Rohling and Hollerbach (1994). The index metacarpophalangeal joint has two orthogonal collocated degrees of freedom, abduction (I_{ABD}) and flexion (I_{MCP}). The I_{MCP}, I_{PIP} and I_{DIP} joints are all defined such that the axes of rotation are parallel. The middle, ring and little fingers are kinematically identical to the index finger, with the bases of the fingers offset along the z-axis.

Modelling the thumb is more challenging. It has five degree of freedom, two of them at TM joint, two of them at MCP and last is DIP joint. It has abduction/adduction, flexion /extension and twist motion. The T_{TR} joint is located at the base of the thumb with the axis of rotation along the index metacarpal. Figure 3-7 shows the model for the index and thumb, with the link lengths designated.

The homogenous transforms are denoted such that;

$$d^B = D_B^A . d^A$$

Where d^A is the homogenous position vector of a point with respect to frame A, d^B is the homogenous position vector of the same point with respect to frame B, D_B^A and is the homogeneous transformation from frame A to frame B.

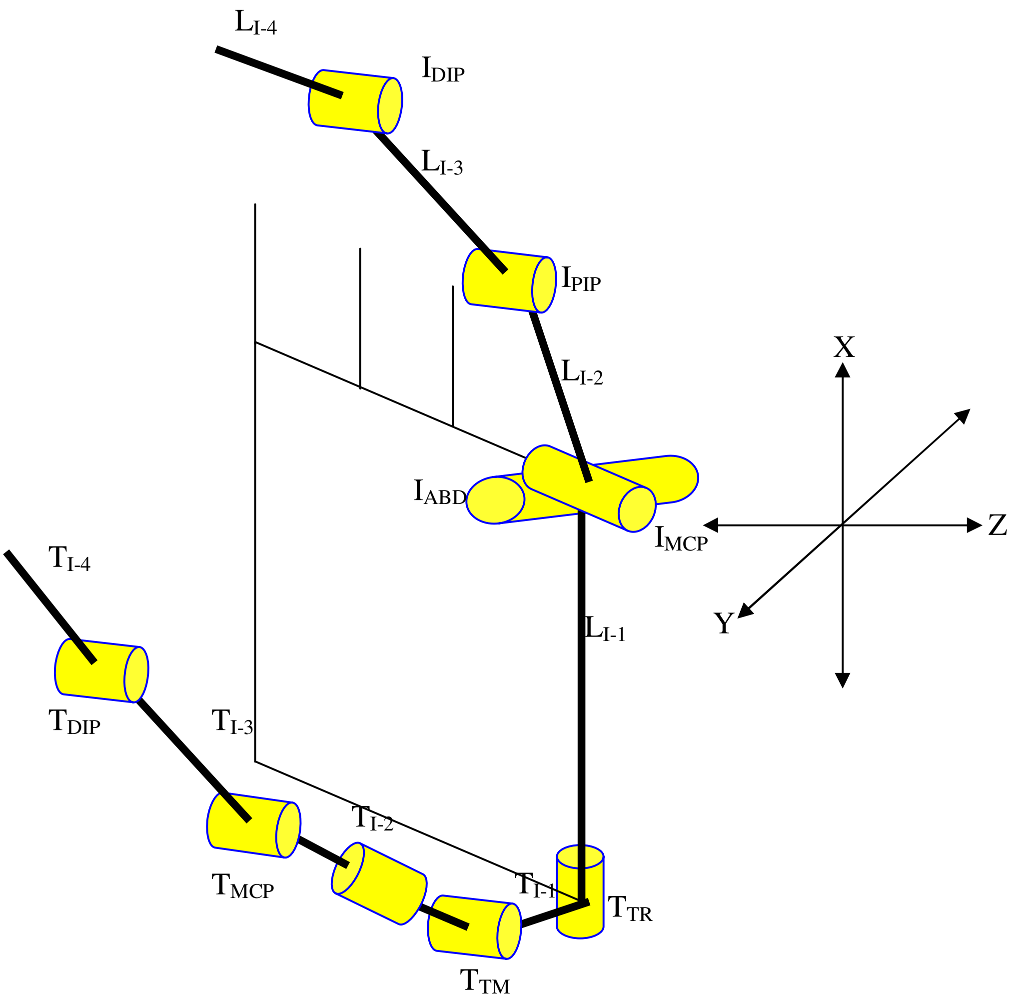


Figure 3-6 Human Hand Model with Link Length and Reference Frames Defined

3.3.1. Thumb Finger

The T_{TR} frame of the thumb has the z-axis pointing along the axis of rotation, the x-axis points toward the T_{TM} axis along the common normal and the y-axis is defined by the right hand rule. The homogeneous transformation from the T_{TR} frame to the base frame is defined by a change of axes and rotation of ϕ_{TR} about the z-axis. For, $\phi_{TR} = 0$ the x-axis of the T_{TR} frame is coincident with the z-axis of the base frame.

$$D^{0}_{T_{TR}} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ \sin(\phi_{TR}) & \cos(\phi_{TR}) & 0 & 0 \\ \cos(\phi_{TR}) & -\sin(\phi_{TR}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The thumb T_{TM} frame is defined with the z-axis pointing along the abduction axis of rotation, the x-axis pointing toward the T_{MCP} axis along the common normal (the thumb metacarpal bone) and the y-axis defined by the right hand rule. The transformation from the T_{TM} frame to the T_{TR} frame involves a rotation ϕ_{TM} about the T_{ABD} z-axis, a change of axes and a translation L_{T-1} along the T_{TR} x-axis. For $\phi_{TM} = 0$, the x-axis of the T_{TM} frame is parallel to the x-axis of the T_{TR} frame.

$$D^{T_{TR}}_{T_{TM}} = \begin{bmatrix} \cos(\phi_{TM}) & -\sin(\phi_{TM}) & 0 & L_{T-1} \\ 0 & 0 & 1 & 0 \\ -\sin(\phi_{TM}) & -\cos(\phi_{TM}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The thumb T_{MCP} frame is defined with the z-axis pointing along the T_{MCP} axis of rotation, the x-axis pointing toward the T_{DIP} axis of rotation along the common normal (the thumb proximal phalange). The transformation from the T_{MCP} frame to the T_{TM} frame involves a rotation ϕ_{MCP} about the T_{MCP} z-axis, a translation L_{T-2} along the T_{TM} x-axis and rotation ϕ_{TW} about the T_{TM} x-axis. For $\phi_{MCP} = 0$, the x-axis of the T_{MCP} frame is parallel to the x-axis of the T_{TM} frame.

$$D_{T_{MCP}}^{T_{TM}} = \begin{bmatrix} \cos(\phi_{MCP}) & -\sin(\phi_{MCP}) & 0 & L_{T-2} \\ \sin(\phi_{MCP})\cos(\phi_{TW}) & \cos(\phi_{MCP})\cos(\phi_{TW}) & -\cos(\phi_{TW}) & 0 \\ \sin(\phi_{MCP})\sin(\phi_{TW}) & \cos(\phi_{MCP})\sin(\phi_{TW}) & \sin(\phi_{TW}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The T_{DIP} frame is defined with the z-axis pointing along the T_{DIP} axis of rotation, the x-axis pointing toward the finger tip and the y-axis defined by the right hand rule. The transformation from the T_{DIP} frame to the T_{MCP} frame is a rotation ϕ_{DIP} about the T_{DIP} z-axis and a translation L_{T-3} along the T_{MCP} x-axis. For $\phi_{MCP}=0$, the x-axis of the T_{DIP} frame is parallel to the x-axis of the T_{MCP} frame.

$$D_{T_{DIP}}^{T_{MCP}} = \begin{bmatrix} \cos(\phi_{DIP}) & -\sin(\phi_{DIP}) & 0 & L_{T-3} \\ \sin(\phi_{DIP}) & \cos(\phi_{DIP}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The finger tip frame is defined with the same orientation as the T_{DIP} frame, located at the tip of the finger. The transformation from the tip frame to the T_{DIP} frame is a pure translation along the T_{DIP} x-axis.

$$D_{T_{TIP}}^{T_{DIP}} = \begin{bmatrix} 1 & 0 & 0 & L_{T-4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation from each frame to the base frame can be found by matrix multiplication.

$$D_{T_{TM}}^{0} = D_{T_{TR}}^{0} \,.\, D_{T_{TM}}^{TR}$$

$$D_{T_{MCP}}^{0} = D_{T_{TM}}^{0} \,.\, D_{T_{MCP}}^{TM}$$

$$D_{T_{DIP}}^{0} = D_{T_{MCP}}^{0} \,.\, D_{T_{DIP}}^{MCP}$$

$$D_{T_{TIP}}^{0} = D_{T_{DIP}}^{0} \,.\, D_{T_{TIP}}^{DIP}$$

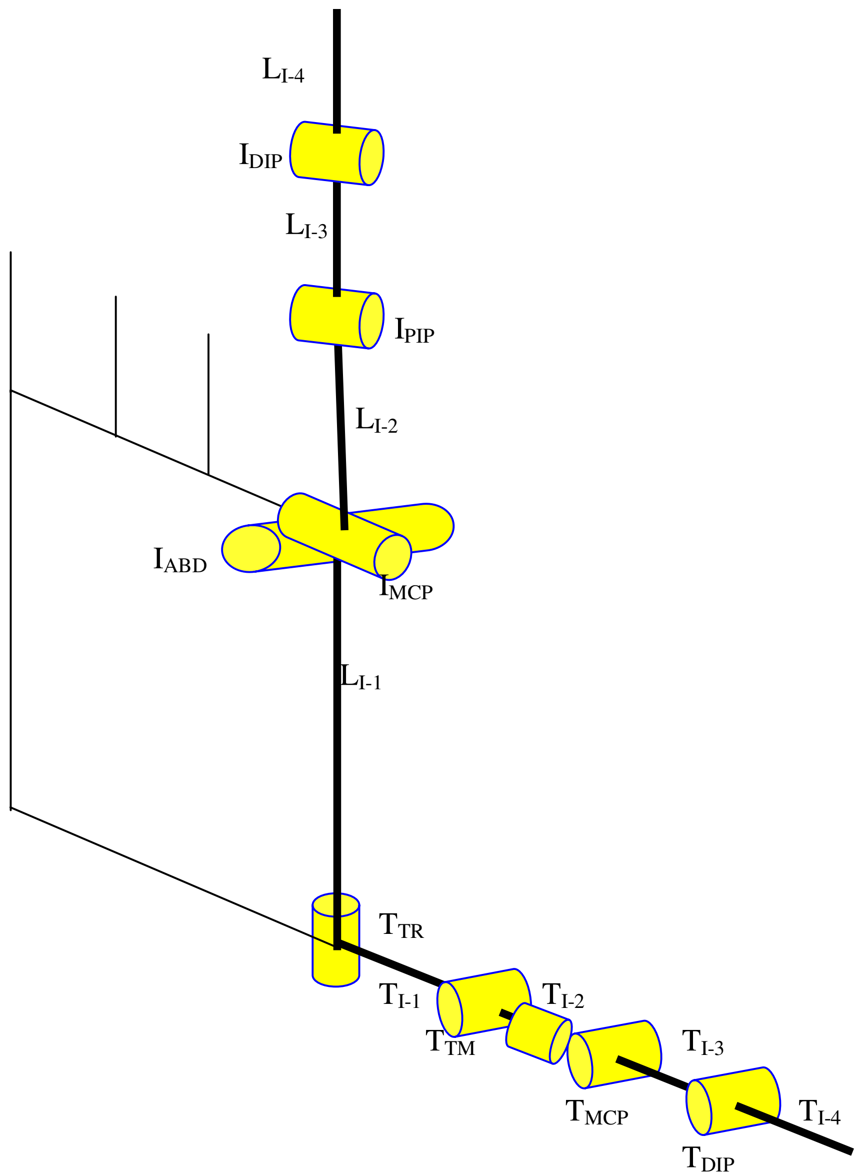


Figure 3-7 Hand model and all joint angles at zero

3.3.2. Index Finger

The I_{ABD} frame is defined with the y-axis pointing along the axis of rotation, the x-axis is parallel to the base frame x-axis for $\phi_{ABD} = 0$ and the z-axis according to the right hand rule. The transformation from the I_{ABD} frame to the base frame is a rotation ϕ_{ABD} about the y-axis in the I_{ABD} frame and a translation L_{I-1} along the x-axis of the base frame.

$$D_{I_{ABD}}^{0} = \begin{bmatrix} \cos(\phi_{ABD}) & 0 & \sin(\phi_{ABD}) & L_{I-1} \\ 0 & 1 & 0 & 0 \\ -\sin(\phi_{ABD}) & 0 & \cos(\phi_{ABD}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The I_{MCP} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the I_{PIP} axis of rotation along the common normal and the y-axis defined by the right hand rule. The transformation from the I_{MCP} frame to the I_{ABD} frame is a rotation ϕ_{MCP} about the I_{MCP} z-axis. For $\phi_{MCP} = 0$, the I_{MCP} frame and I_{ABD} frame are identical.

$$D_{T_{MCP}}^{T_{ABD}} = \begin{bmatrix} \cos(\phi_{MCP}) & -\sin(\phi_{MCP}) & 0 & 0 \\ \sin(\phi_{MCP}) & \cos(\phi_{MCP}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The I_{PIP} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the I_{DIP} axis of rotation along the common normal and the y-axis defined by the right hand rule. The transformation from the I_{PIP} frame to the I_{MCP} frame is a rotation ϕ_{PIP} about the I_{PIP} z-axis and a translation along the I_{MCP} x-axis. For $\phi_{PIP} = 0$, the I_{PIP} frame is oriented parallel to the I_{MCP} frame.

$$D_{T_{PIP}}^{T_{MCP}} = \begin{bmatrix} \cos(\phi_{PIP}) & -\sin(\phi_{PIP}) & 0 & L_{I-2} \\ \sin(\phi_{PIP}) & \cos(\phi_{PIP}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The I_{DIP} frame is defined with the z-axis pointing along the axis of rotation, the x-axis pointing toward the index finger tip and the y-axis defined by the right hand rule. The transformation from the I_{DIP} frame to the I_{PIP} frame is a rotation ϕ_{DIP} about the z-axis of the I_{DIP} frame and a translation L_{I-3} along the x-axis of the I_{PIP} frame. For $\phi_{DIP} = 0$, the orientation of the I_{DIP} frame is parallel to the I_{PIP} frame.

$$D_{T_{DIP}}^{T_{PIP}} = \begin{bmatrix} \cos(\phi_{DIP}) & -\sin(\phi_{DIP}) & 0 & L_{I-3} \\ \sin(\phi_{DIP}) & \cos(\phi_{DIP}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The index tip frame is defined with the same orientation as the I_{DIP} frame located at the finger tip. The transformation from the I_{TIP} frame to the I_{DIP} frame is a translation along L_{I-4} the I_{DIP} x-axis.

$$D_{T_{TIP}}^{T_{DIP}} = \begin{bmatrix} 1 & 0 & 0 & L_{I-4} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The transformation from each frame to the base frame can be found by matrix multiplication.

$$D_{T_{MCP}}^{0} = D_{T_{ABD}}^{0} \,.\, D_{T_{MPC}}^{ABD}$$

$$D_{T_{PIP}}^{0} = D_{T_{MCP}}^{0} \,.\, D_{T_{PIP}}^{MCP}$$

$$D_{T_{DIP}}^{0} = D_{T_{PIP}}^{0} \,.\, D_{T_{DIP}}^{PIP}$$

$$D_{T_{TIP}}^{0} = D_{T_{DIP}}^{0} \,.\, D_{T_{TIP}}^{DIP}$$

3.3.3. Implementation of Hand Model on JAVA 3D

Java 3D is a client-side Java application programming interface (API) developed at Sun Microsystems for rendering interactive 3D graphics using Java. Using Java 3D you will be able to develop richly interactive 3D applications, ranging from immersive games to scientific visualization applications.

JAVA 3D applications define a complex scenegraph hierarchy. Scenegraph can be defined as hierarchical data structure that captures the elements of spatial relationships between objects. Just as when the wrist joint was moved its constituent parts were also moved. This principle is central to applications that require hierarchical control. At the scenegraph level, the key to specifying relative positions for Nodes within the scenegraph is the TransformGroup[1] Node. A TransformGroup encapsulates a Transform3D instance, which in turn encodes a 4×4 scaling, rotation, and translation matrix. The important principle is that a scenegraph Node's rotation, scaling, and translation is always specified relative to its parent Node's rotation, scaling, and translation.

We have created a human hand as a hierarchic model of joints and bones with the given actual size ratio defined Tablo 3-1. An important principle of the scenegraph is that the position of a child Node only depends upon the positions of its parent Nodes. In other words, the position of the end of the little finger depends upon;

- Length of little finger bones
- Rotation of little finger joints
- Length of palm
- Rotation of the wrist joint

Figure 3-9 shows human hand skeleton model on JAVA used in this thesis. Each joint has own constraints such that each joints movements capability is in the range that defined in section 3.2.

[1] TransformGroup: Group node that contains a transform. The TransformGroup node specifies a single spatial transformation, via a Transform3D object, that can position, orient, and scale all of its children.

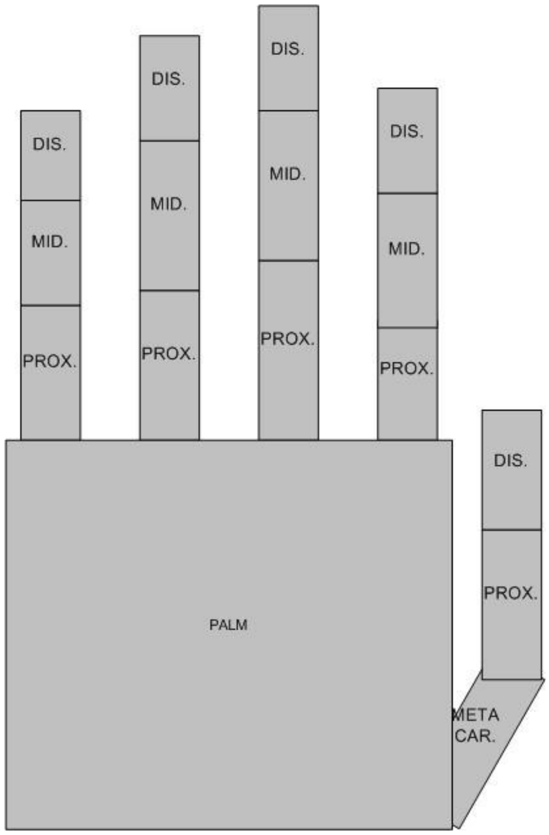


Figure 3-8 Java 3D Hand Model

Each joint has four TransformGroup, TG Joint stores the rotation of each joint and a bone can be added to hand by using TG Joint. TG Joint has three children.

> TG Trans: Shifts the geometry for the cylinder upward by L/2. TG Trans adjusts the base of rotation for each bone. Rotation base point of each bone is the joint which connects the bone to the hand. For example, proximal bone rotation point is MCP joint; distal bone rotation point is DIP joint.
>
> TG Rotation: TG Rotation modifies its parent TG Joint to rotate the joints of the model. Our algorithm computes the joints angles. TG Rotation uses these angles and interpolates the start and final angle (Computed angle) of each bone. It can be thought that TG Rotation computes the frames in which between two end points.

TG Offset: Contains the length of the bone, and hence shifts the coordinate system of the next bone (its child). TG offset adds the bones (cylinders) to each other by shifting the length of the bone.

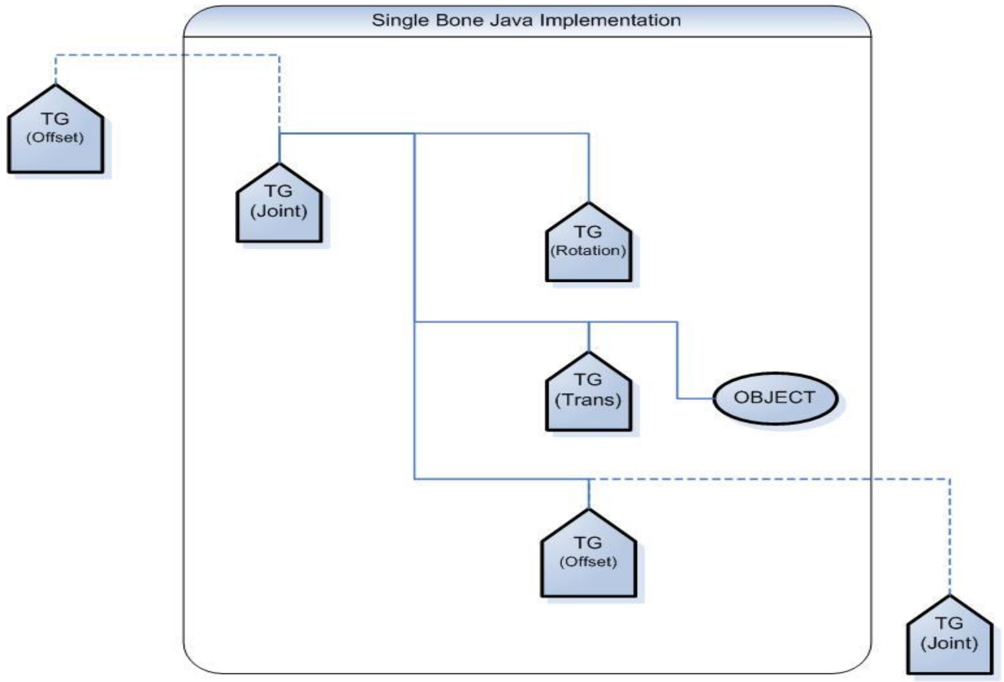


Figure 3-9 The completed scenegraph for the single finger bone model.

Our hand model is developed by using object oriented methods. Figure 3-9 shows the bone class structure. Each bone can be added to previous bone by using TG Joint TransformGroup of the bone.

CHAPTER 4

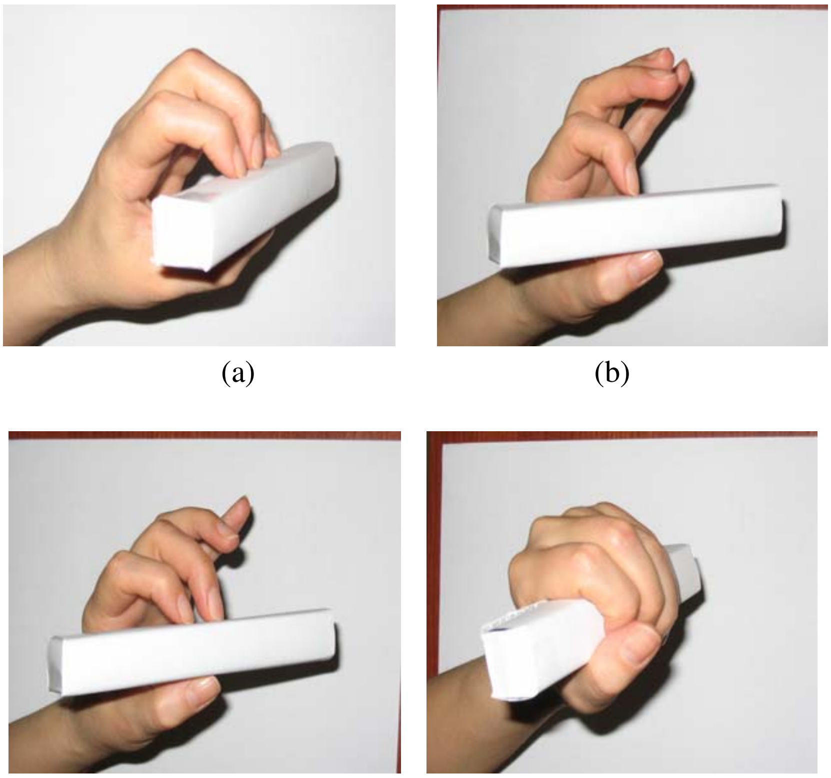
PRESHAPING ANAYSIS

4.1 Introduction

We have investigated grasp analysis to analyze human preshaping behaviour. Napier (1956) analyzed the preshaping movements of the human hand. The human preshapings are divided into two primary categories: Precision and power. In precision preshaping the object is usually held by the fingertips. Hence manipulability is more important than the ability to resist large external forces. On the other hand in power preshaping the object is usually constrained by the palm and both the proximal and distal surface of the fingers. The first step of the preshaping planning processes is to find a set preshaping points or finger tip points. To achieve this, the system requires the object geometry that consists only of shape primitives such as spheres, cylinders, and boxes. The choice of primitives will determine the different strategies used to preshape the object. For each shape, we have defined a set of preshaping strategies to limit the huge number of possible preshaping. Cutkosky and Wright's (1996) taxonomy of human preshaping is generalized form of preshaping analysis. In the previous work of taxonomy, preshaping was categorized according to the object's shape, weight and size. Preshaping types, which are summarized in the Table 4-1, are explained below.

Table 4-1Preshaping Types According to Object's Features

Type	Size	Preshaping Way
Prismatic	Thin	Wrap/Pinch/Tripod/All Finger
	Medium	Wrap/Pinch/Tripod/All Finger
	Thick	Wrap/Pinch/Tripod/All Finger
Sphere	Small Diameter	Wrap/Disk/Tripod/All Finger
	Large Diameter	Wrap/Disk/Tripod/All Finger
Cylinder	Small Diameter	Wrap/Disk/Tripod/All Finger
	Large Diameter	Wrap/Disk/Tripod/All Finger



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

Figure 4-1 (a) Prismatic Lateral, (b) Prismatic Pinch, (c) Prismatic Tripod, (d) Prismatic Wrap

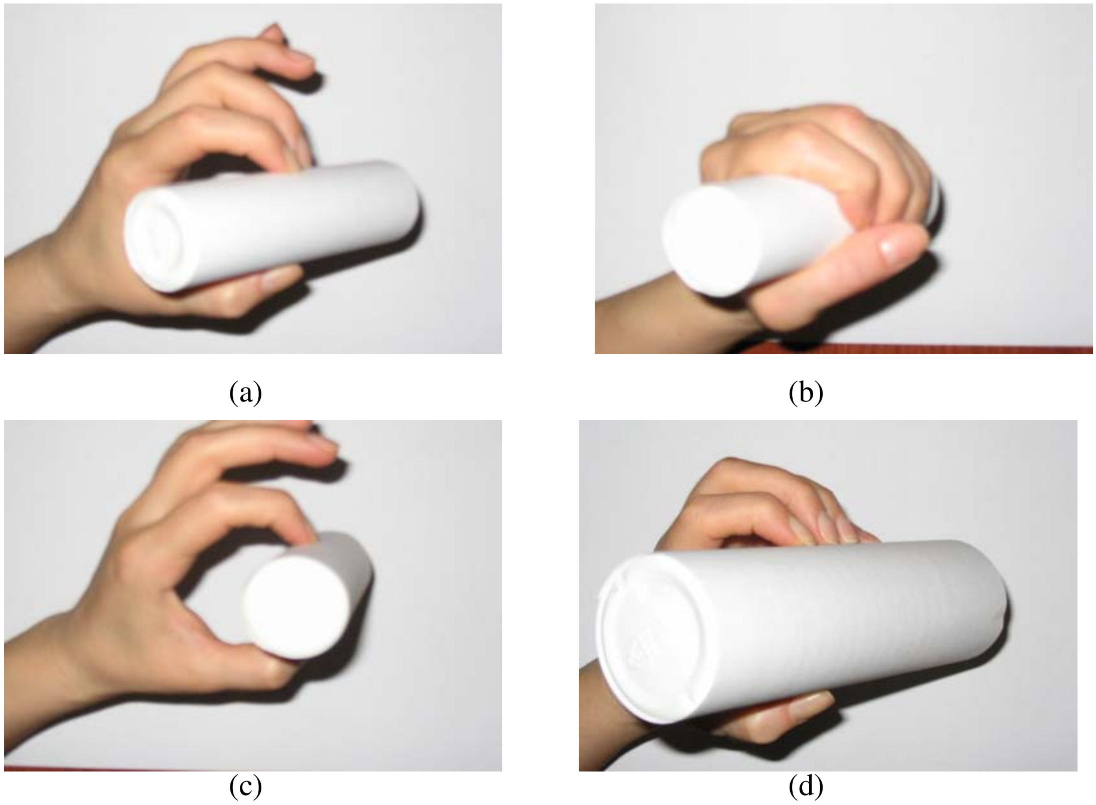


Figure 4-2 (a)Cylindrical Tripod (b)Cylindrical Wrap (c)Cylindrical Pinch (d)Cylindrical Lateral

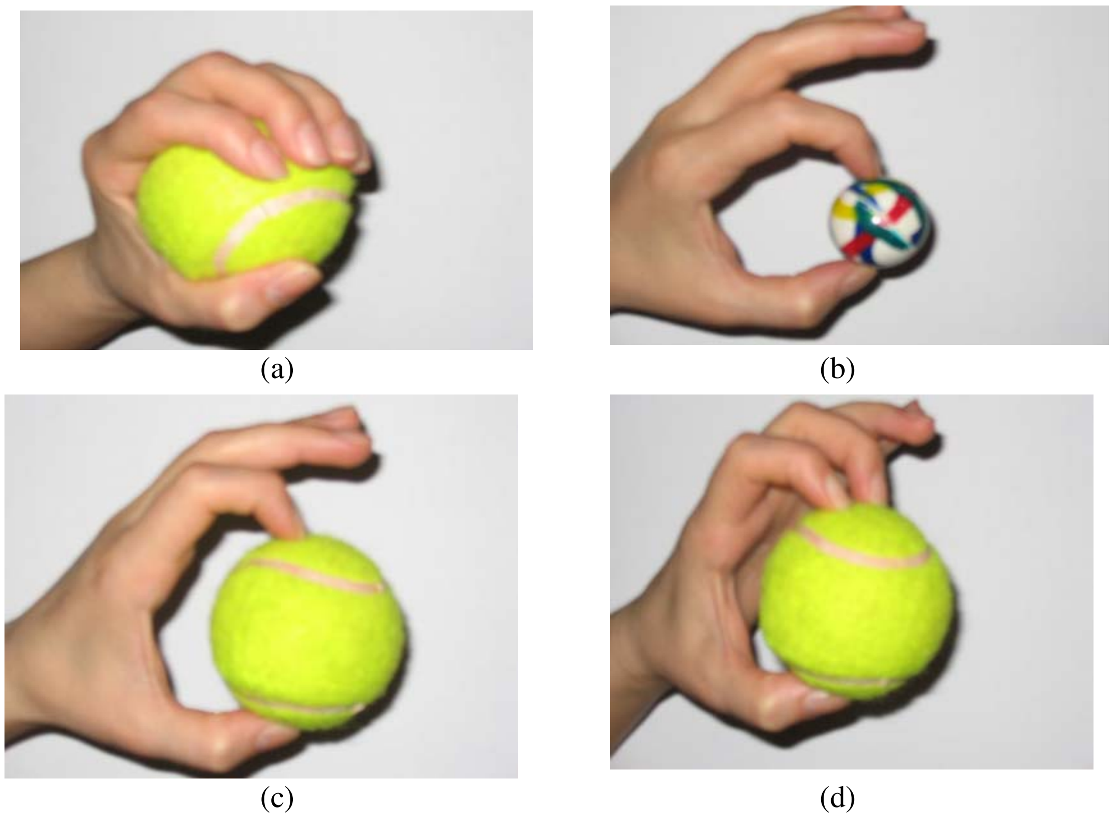


Figure 4-3 (a) Sphere Wrap (b)Sphere Pinch (c)Sphere Pinch (d)Sphere Tripod

In our work, wrap, pinch, tripod and five fingers preshaping types are chosen and animated. Figure 4-1, Figure 4-2, Figure 4-3 shows the hand position and configuration of these types. Explanations of them are given below.

Wrap Preshaping: All five fingers in the preshape are used to preshape the object. The object is preshaped between the thumb and the opposing four fingers. The aim is to attain maximum contact area between the object and the hand, including the palm of the hand. This preshaping is used for applying force to the object. This preshaping provides maximum stability, but minimum dexterity for further manipulation.

Five Finger Preshaping: All five fingers in the preshape are used for preshaping the object. The index finger is used as the master finger in shaping the four fingers. The object is preshaped between the thumb and the opposing four fingers such that, the four fingers carry the weight of the object. This preshaping is used for optimizing the manipulability and stability criteria and a given task. Five finger preshaping can be called lateral in literature (Schlesinger, 1919) but there also lateral pinch (key preshaping) type preshaping (subtype of lateral) such that thumb and side of index finger are used to preshape object. This type was not chosen to show hand configuration.

Pinch Preshaping: Only two fingers of the preshape are used to preshape the object. The first finger is the thumb. The second finger is the index finger. The object is preshaped between the tips of the thumb and the opposing index finger. The aim is to achieve maximum manipulability on the preshaped object.

Tripod Preshaping: Three fingers of the preshape are used for preshaping the object. Tripod preshaping uses three fingers. The first finger is the thumb. The other fingers are the index and middle fingers. The index finger is used as the master finger. The object is preshaped between the thumb and the opposing two fingers. The first one or two links of the fingers are in contact with the object. This preshaping is used to optimize the manipulability and criteria of a given task

Lateral, pinch, tripod and wrap are main preshaping types defined by Cutkosky and preshaping approach of wrap is different form others. Wrap is a power preshaping type and whole hand covers the object.

In our study, two different approaches for preshaping have been investigated. First approach is for manipulative preshaping or precision preshaping (Lateral, pinch and tripod). It requires highly computation and analytical work. Computation of joint angles and finger tip positions are based on inverse kinematics. Obtaining best configurations depend on many parameters so it requires time and memory. Second one is used for power preshaping (Wrap). It requires animation and software know-how. Collision detection is used to animate realistic human hand preshaping for the wrap type.

4.2 Preshaping Quality Measure

One of the key features of this work is that it can be used with any form of preshaping evaluation. In order to animate preshaping behaviour of human hand, some quality metrics should be used in animation algorithm. By using quality metric, we can find proper finger tip positions on the object. In the literature, there are many quality metric studies on robotics area. Preshaping metrics usually emphasize one or more aspects of the resulting preshaping configuration: Capacity to resisting external disturbances (Guo, G., Gruver, W., and Jin, K., 1989), distance between contacts (Chen, I., and Burdick, J., 1983), ratio between resulting wrenches and applied forces, or minimization of total forces applied to the object (Ferrari, C., and Canny, J., 1995), preshaping stability (Jameson, J., and Leifer, L., 1987) and others. Generally, primary objective of quality measure is the stability of preshaping such that human brain was learned before this motion becomes a reflex for human anymore.

In our study quality measure is defined for preshaping by measuring the distance between the center of mass of the preshaped object and the center of the contact points.

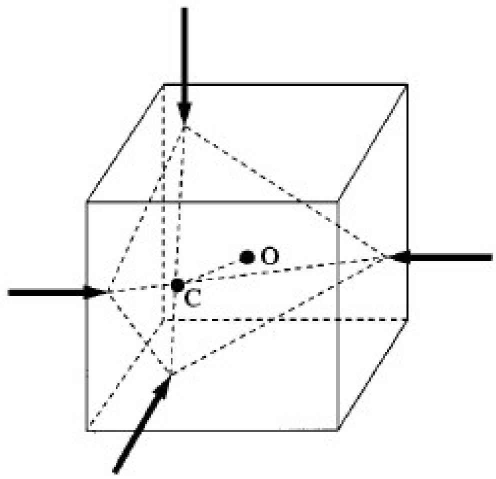


Figure 4-4 Definition of Quality Measure, C is the centre of the preshaped points

We have introduced a performance index similar to that of Ponce (1997) by measuring the distance between the centre of mass g_o (x_o; y_o; z_o) of the preshaped object and the centroid of the contact points g_o(x_c; y_c; z_c) (Figure. 4-4). The centre of mass is chosen to be the origin of the object coordinate frame and hence g_o is (0, 0, 0). The centroid of the contact points can be calculated;

$$g_c = \frac{1}{N}\sum_{i=1}^{N} Pi$$

g_c= Centroid of contact points

Pi=Position of i^{th} finger

N=Number of contact points

Then the objective function can be represented as: min $u(z) = \|g_o - g_c\|^2$

Notice that to shorten the distance between the center of mass of the object and the centroid of the contact points enables us to decrease the effect of gravitational and inertial forces during the motion of the hand and thus achieves a more realistic preshaping.

4.3 Geometric Stability of Preshaping

For pinch, lateral and tripod types preshaping are made by thumb and other fingers such that one, two, three or four fingers can be used. There exist two main force sources; one is thumb, second is other fingers (See Figure 4-5). To make a geometric stable preshaping, two force sources should be balanced. But our concern is to show hand configurations of preshaping behaviour of human, so that positions of fingers on object are main considerations. Hand configurations depend on constraints of human hand, base and finger positions.

At this point, we need to calculate combined position resulted from four fingers (little, middle, point, and ring).

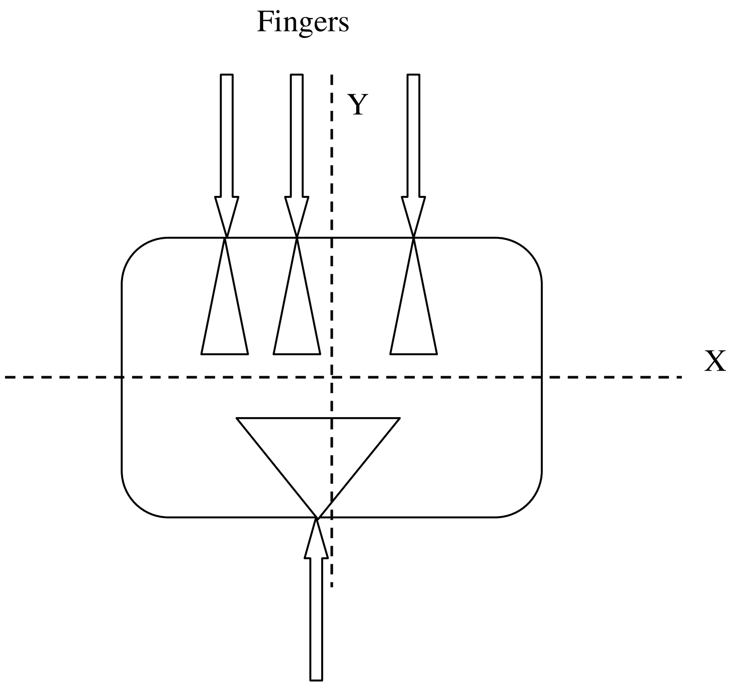


Figure 4-5 Preshaping points

Combined force position of fingers (index, middle, ring and little) for preshaping application can be written as centroid of finger positions.

$$P_c = \frac{1}{N}\sum_{i=1}^{N} Pi$$

N: The number of fingers,

Pi=Position of ith finger

After finding combined position of fingers, according to Iberall (1997), two virtual fingers should be in opposite directions to each other (Figure 4-6).

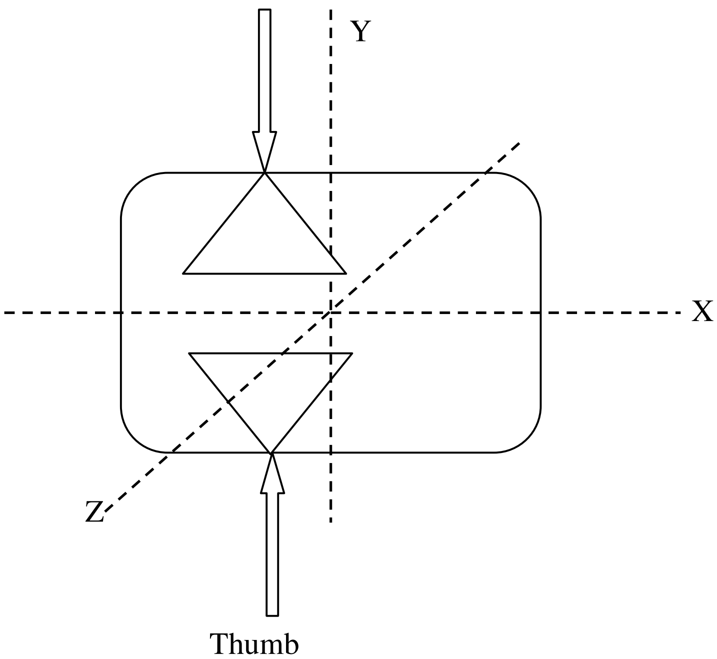


Figure 4-6 Two Virtual Fingers at Opposing Sides

Position of thumb finger should be opposite side of combined position of other fingers.

$$P_{CX} = T_{PX}$$

$$P_{CZ} = T_{PZ}$$

T_{PX}=Thumb finger position at X direction

T_{PZ}=Thumb finger position at Z direction

P_{CX}=Combined Position at X direction

P_{CZ}=Combined Position at Z direction

In order to animate preshaping behaviour of human, thumb finger position should satisfy the above equality (Figure 4-6). From above equality, thumb position depends on the preshaping types and object size. Preshaping types mainly determine that how many fingers will be used. For pinch preshaping, two fingers will preshaping the object, these are index and thumb. Positions of these fingers will be located on the object and satisfying above equality.

CHAPTER 5

PRESHAPING ALGORITHM

5.1. Precision Type Preshaping

Lateral, pinch and tripod preshaping types need finger tip positions whereas wrap type only needs collision information between hand and object because of aims of preshaping types which mentioned before. We have developed an iterative algorithm to find finger positions on the chosen object in the scene. Developed algorithm for precision type preshaping activity considers many parameters and intersection of these parameters constructs our solution. Figure 5-1 shows constraints and intersection of these constraints give us a solution.

There are some restrictions to compute finger tip positions. The constraints that restrict the finger positions are the geometric constraints that the finger must have be in contact with an object, kinematics constraints that each finger must have a feasible inverse kinematics solution appropriate human hand kinematics constraints to a given finger-tip position, static constraint that the fingers must maintain equilibrium (Figure 4-6).

Computed finger tip positions should be reachable by fingers. All points should be covered by fingers. In some case, there are some points which satisfy all restrictions but it is not appropriate by the fingers joint restriction, so it can not be used as finger tip positions. This is the geometric constraints.

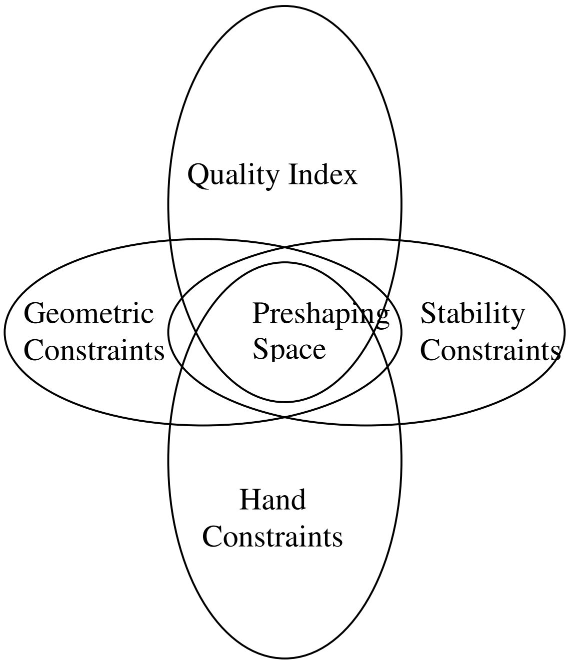


Figure 5-1 Restrictions of preshaping planning

Human hand kinematics constraints were mentioned previous chapter. The constraints are related with the joint angles of fingers and correlation of these angles. Each joint has a maximum flexion angle so this should be taken into account when computing finger tip positions.

Above restrictions mainly related with the inverse kinematics solution of finger tip position. Static restriction was explained previous chapter, fingers must maintain the geometric equilibrium. Contact points should satisfy this criterion otherwise resulted preshaping would be seen in unrealistic appearance.

Quality index was used as a feedback or optimization for the developed algorithm. It is possible to obtain many contact points sets satisfying the above restriction. Quality index selects the most appropriate sets from obtained contact points sets. Closed-loop solution was generated with the quality index to form preshaping activity.

The algorithm that we developed for preshaping is presented in Figure 5-2.

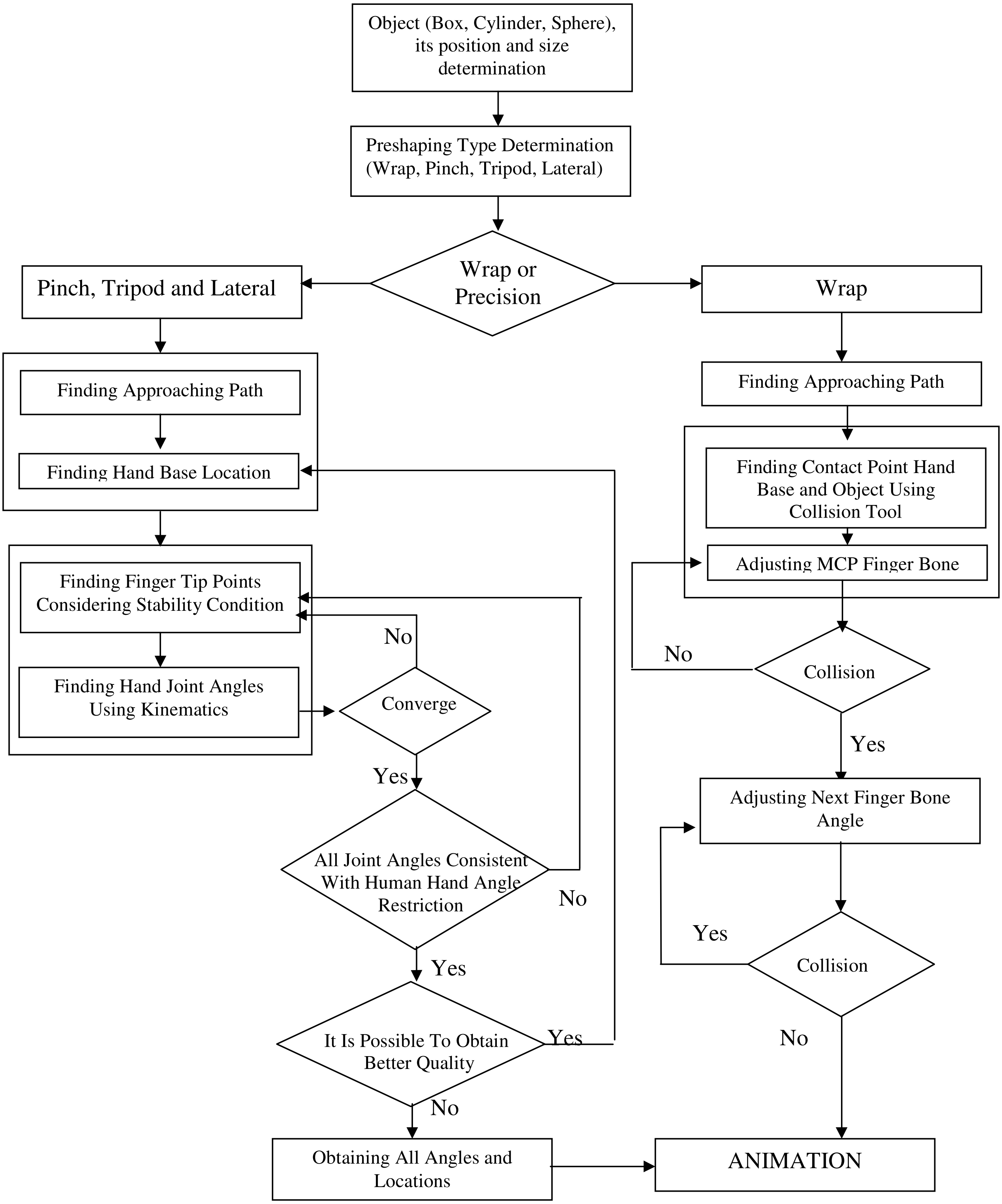


Figure 5-2 Developed Realistic Preshaping Animation Algorithm

To preshaping object we assume that all information is known about object that is dimension, position, orientation. To preshape the object, we should develop an algorithm to find preshaping points on the object satisfying stability of object, kinematics of hand.

5.2. Wrap Type Preshaping

We have used different approach for wrap type preshaping. In Figure 5-2, right part of algorithm based on collision detection has been developed for wrap type preshaping. Power preshaping is a stable preshaping and there is no need to take into account to stability of the object. So that we have used collision detection to animate wrap type preshaping.

Collision detection includes algorithms from checking for collision, i.e. intersection, of two given solids, to calculating trajectories, impact times and impact points in a physical simulation. Collision test can be done by two ways, these are bound and geometry test. Bounding assumes that each object on scene have active region bounded by sphere (Figure 5-3). If two bounds intersect each other, collision can be detected. Geometry test uses object geometry, if two object’s occupied region intersect each other, collision can be detected.

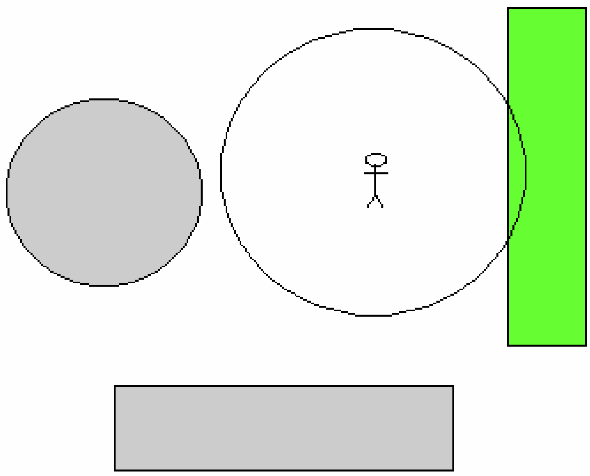


Figure 5-3 The basic collision areas and the user views. The grey areas are removed from the collision detection because they do not intersect with the user's view area

In our algorithm, we have used geometry based collision detection in which we need accurate collision information between two objects. Geometry based collision information can be used by accurate systems. To detect collisions, new hand model is developed. All bones of new hand was inserted to CollisionDetector class which gives us collision information at anywhere at any time. Java 3D is powerful 3D programming platform but collision detection library is not powerful by many aspects. So that collision detection class has been developed and can be seen from Figure 5-4.

Algorithm, firstly, finds approaching path to the object, to be collision it is needed to rotate and close hand to the object. After collision of the palm, all bones moves orderly to the preshaped object. Each bone has own collision detector, if bone collides with the object, bone stops moving and next bone starts moving. Animation is completed with the thumb finger collisions.

```
 class CollisionDetector  extends Behavior {
 private final Color3f highlightColor = new Color3f(0.0f, 1.0f, 0.0f);
 private final ColoringAttributes highlight = new ColoringAttributes(
 highlightColor, ColoringAttributes.SHADE_GOURAUD);
 static boolean inCollision = false;
 private Shape3D shape1;
 private ColoringAttributes shapeColoring;
 private Appearance shapeAppearance;
 private WakeupOnCollisionEntry wEnter;
 private WakeupOnCollisionExit wExit;

public Cd1(Shape3D s) {
  shape1 = s;
  shapeAppearance = shape1.getAppearance();
  shapeColoring = shapeAppearance.getColoringAttributes();
  inCollision = false;
 }
 public void initialize() {
  wEnter = new WakeupOnCollisionEntry(shape1,  WakeupOnCollisionEntry.USE_GEOMETRY);
  wExit = new WakeupOnCollisionExit(shape1,      WakeupOnCollisionEntry.USE_GEOMETRY);
  wakeupOn(wEnter);
 }
 public void processStimulus(Enumeration criteria) {
  inCollision = !inCollision;
  if (inCollision) {
    if (!Hand_Coll.ap1.isPaused()) {
    Hand_Coll.ap1.pause();
  }
    inCollision = false;
    shapeAppearance.setColoringAttributes(highlight);
    wakeupOn(wExit);
  }
```

Figure 5-4 The Developed Collision Detector Class

CHAPTER 6

IMPLEMENTATION OF THE ALGORITHM, SAMPLE OUTPUTS AND PERFORMANCE ANALYSIS

In this thesis, animation of preshaping behavior of human hand is aimed and studied. Several experiments were conducted to illustrate preshaping animation for both precision and power preshaping. The algorithm explained in Chapter 5 is implemented using Java 3D. Experiments made on Sun Java Development Kit (JDK) 1.4 with library of OpenGL-SDK. View of graphical user interface is shown in figure below.

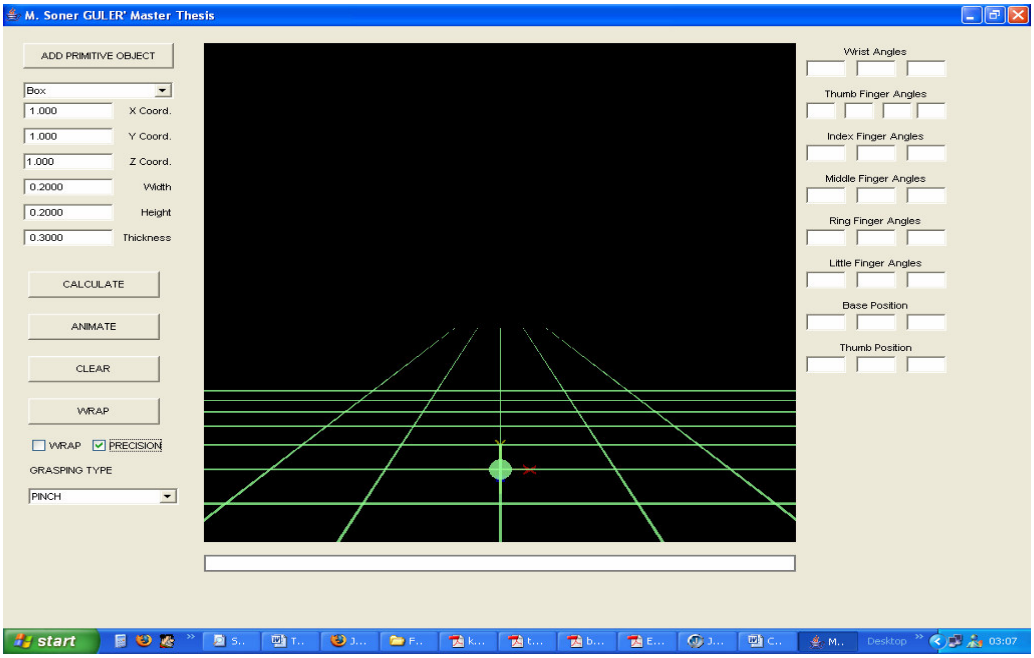


Figure 6-1 GUI of Preshaping Program

Graphical user interface defines Panel 3D object which contains Canvas 3D class. Canvas 3D class provides a drawing canvas for 3D rendering. It is used either for on-screen rendering or off-screen rendering. Canvas3D is an extension of the AWT Canvas class that users may further subclass to implement additional functionality.

The Canvas3D object extends the Canvas object to include 3D-related information such as the size of the canvas in pixels, the Canvas3D's location, also in pixels, within a Screen3D object, and whether or not the canvas has stereo enabled. Java 3D can convert a Canvas3D size in pixels to a physical world size in meters. It can also determine the Canvas3D's position and orientation in the physical world.

GUI consists of five buttons, two combo box and text fields. Buttons ADD PRIMITIVE OBJECT, CALCULATE, ANIMATE, CLEAR and WRAP.

ADD PRIMITIVE OBJECT: Inserts an object which position and size can be selected to the screen. Object selection can be done from combo box which is bellow the ADD PRIMITIVE OBJECT button. Position and size selection can be done from text fields which are below the object selection combo box. The developed algorithm takes these parameters (object position, size and type) as inputs.

CALCULATE: Used when precision type preshaping is selected. Precision type preshaping animation is requiring joint angles and finger tip positions so these parameters should be calculated by using inverse kinematics equations and hand constraints which are defined Section 5. After selecting object size position and type, if required animation is precision type preshaping, CALCULATE button is used to find angles and positions. Results of calculation process can be seen from test boxes which are located at right part of the GUI. The text boxes show the angles and positions of hand fingers.

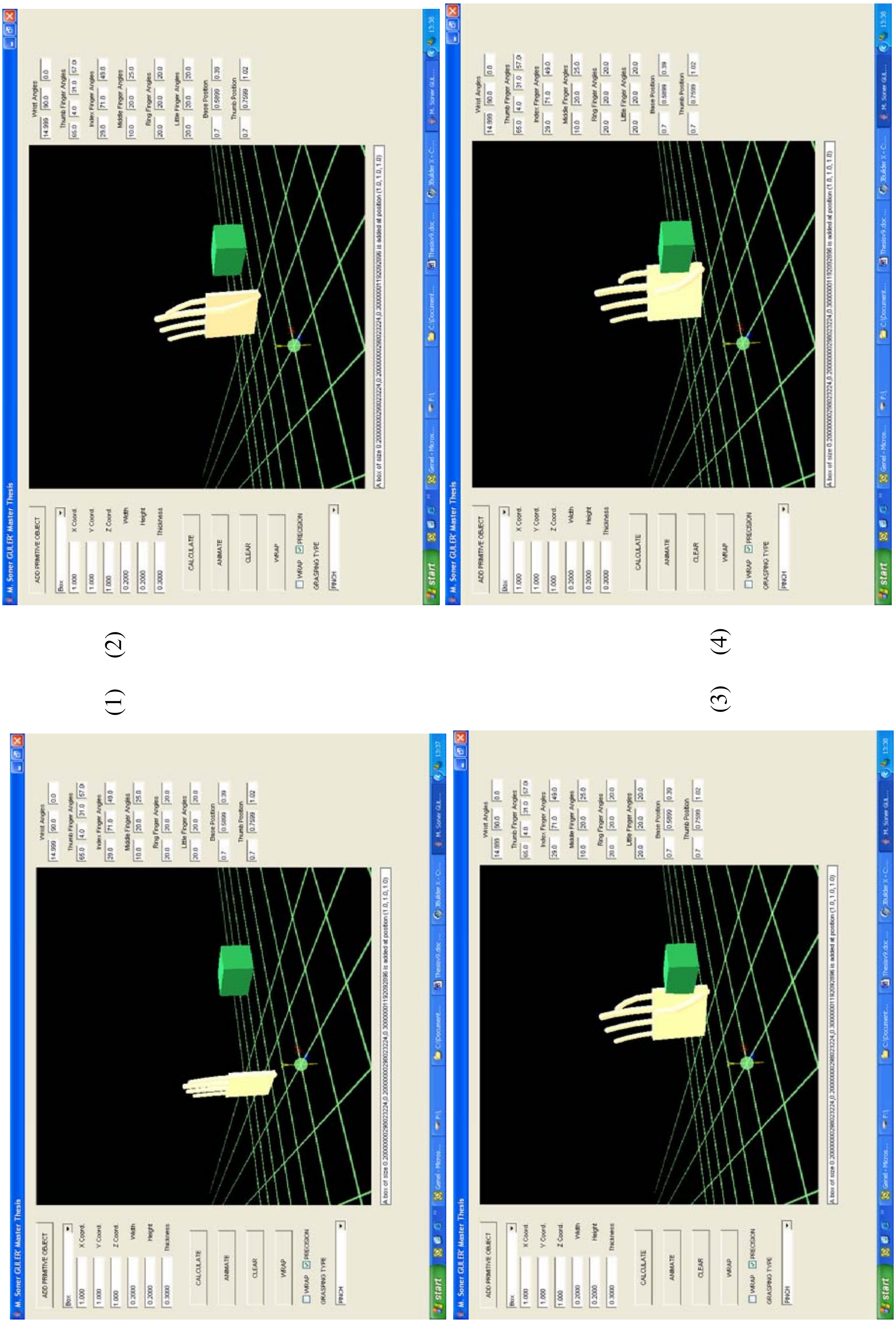
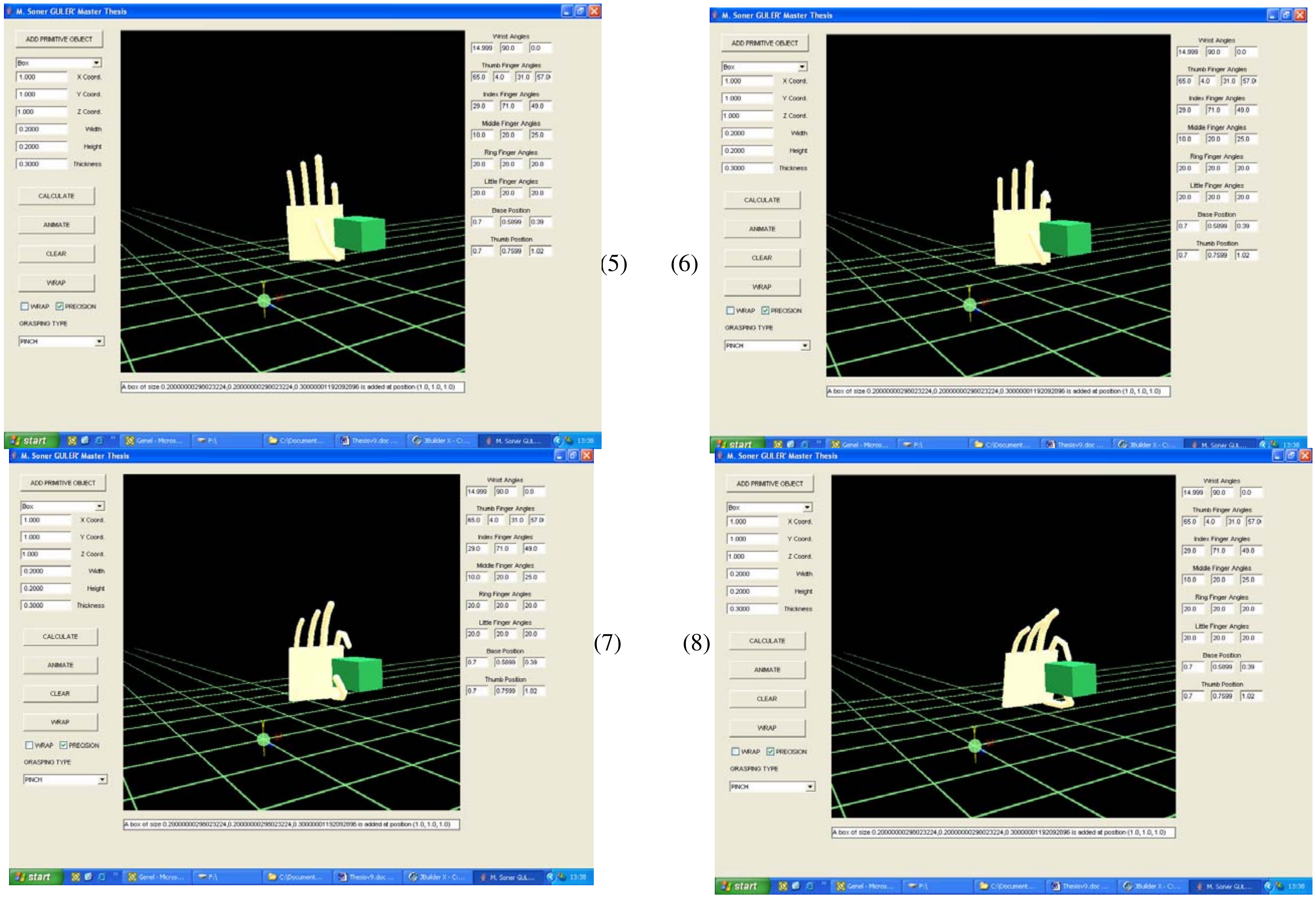


Figure 6-2 Sample Animation Frames



(5) (6)

(7) (8)

Figure 6-3 Sample Animation Frames (Cont.)

ANIMATE: Button can only be used for precision type preshaping animation after finding necessary positions and angles. Animation button animates the hand preshaping behavior with respect to chosen parameters. Hand starts the animation from the initial position to the end position. Animation samples can be seen from the Figure 6-2.

CLEAR: CLEAR button deletes the object which is previously inserted to the screen. If any object wrongly placed, CLEAR button can be used to delete this object.

WRAP: Two different approaches are combined into single GUI. Wrap preshaping animation can be animated by using WRAP button. Wrap type preshaping is not requiring angles or positions information so that there is not any calculation to animate this type preshaping. After selection object and inserting to the screen, wrap animation can be started by pressing this button. Java 3D continuously monitors the screen, in case of collision, appropriate task which is defined on collision algorithm is done.

Preshaping type selection can be done by using preshaping type combo box. There are three preshaping types for precision type preshaping which are pinch, tripod and all fingers defined Section 5.

6.1. Precision Type Preshaping

The developed software has been taking into account all possible configurations of objects and preshaping types. Table 6-1 summarizes the experiments.

All preshaping types can be applied to all predefined objects. Object size should be less than maximum holdable object size. Total length of hand including palm, thumb and index finger is 1,4 f. This hand length determines the maximum holdable object size. To preshape circular objects, length of the hand should be higher than half circumference of object. For prismatic objects, height of objects should be less than sum of palm and index proximal bone length.

Table 6-1 Precision Preshaping Experiments

Prismatic	Thin	Pinch/Tripod/All Fingers
	Medium	Pinch/Tripod/All Fingers
	Thick	Pinch/Tripod/All Fingers
Cylinder	Thin	Pinch/Tripod/All Fingers
	Medium	Pinch/Tripod/All Fingers
	Thick	Pinch/Tripod/All Fingers
Sphere	Thin	Pinch/Tripod/All Fingers
	Medium	Pinch/Tripod/All Fingers
	Thick	Pinch/Tripod/All Fingers

After selection of object type, size, one of the precision type preshaping type should be selected. These are pinch which uses index and thumb fingers, tripod which uses middle, index and thumb fingers and all fingers which uses index, middle, ring little and thumb fingers. Precision preshaping algorithm computes proper preshaping points and hand joint angles with respect to chosen parameters. Obtained end configurations and joint angles given below figures. Animation starts from initial position. Initial position of the hand is all finger angles are zero and hand is located at the origin. After animation start, all joints move to their end position with defined speed. In below figures, there are four views of single animation. First view shows the middle phase of the animation. Hand starts move from the initial position to the end position. Other three views show the end position configuration of the hand, left, right and front views. Used finger joint angles and positions can been from text box located at the right of the screen. For each preshaping, animation four screen shots are inserted. In each preshaping animation hand joint angles and hand finger and base position can be seen form the screen.

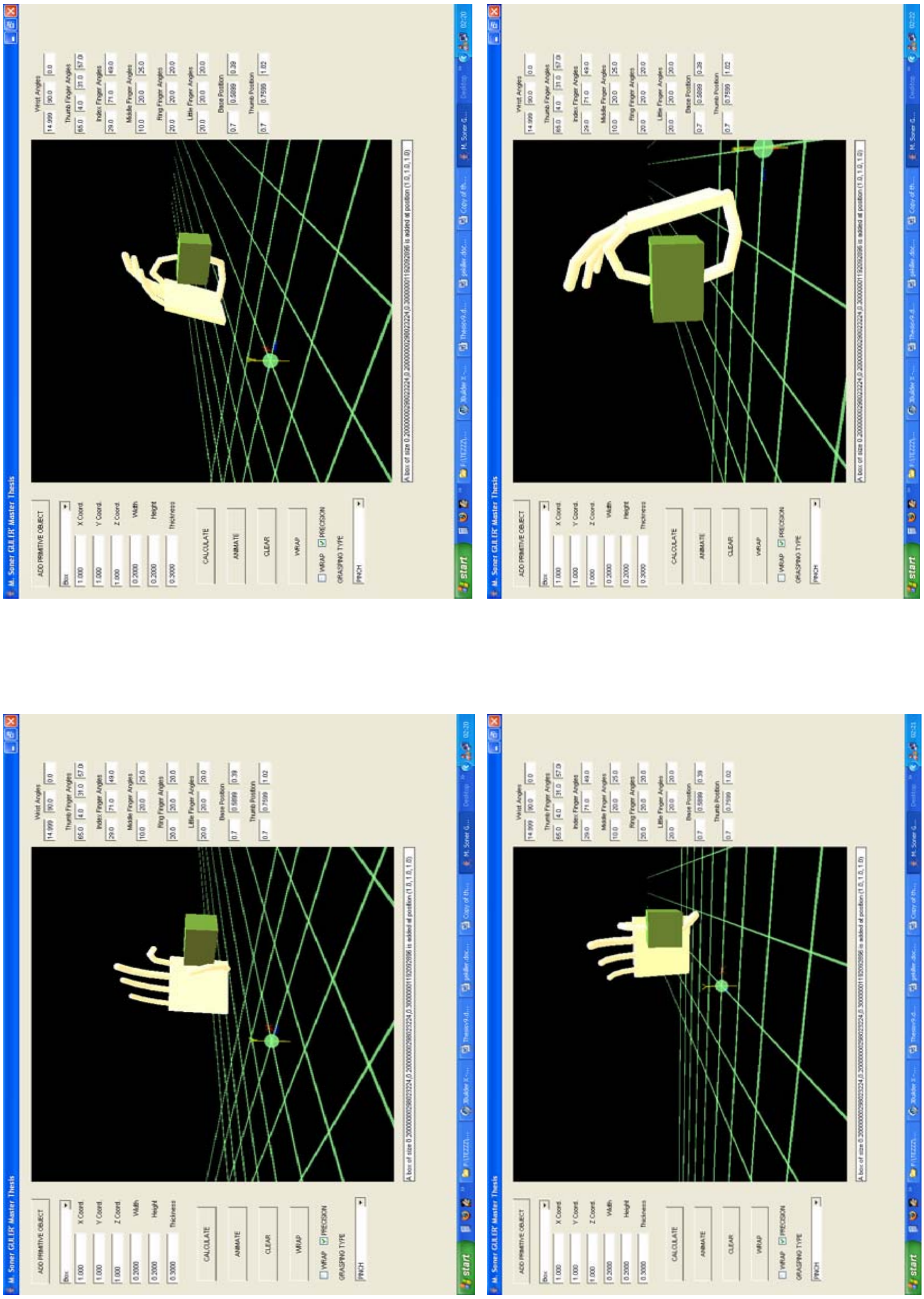


Figure 6-3 Pinch Preshaping of Middle Size Prismatic Objects

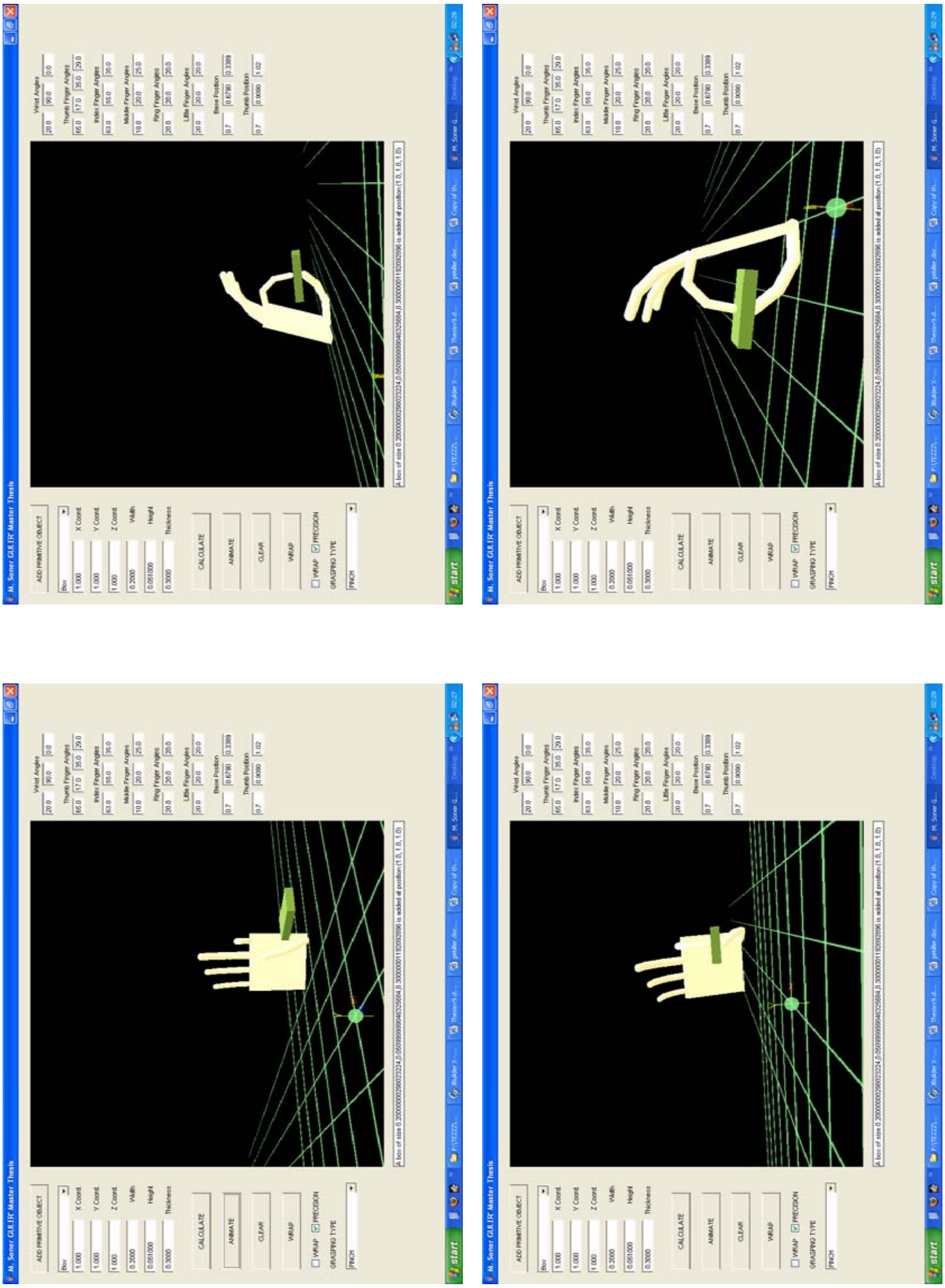


Figure 6-4 Pinch Preshaping of Small Size Prismatic Objects

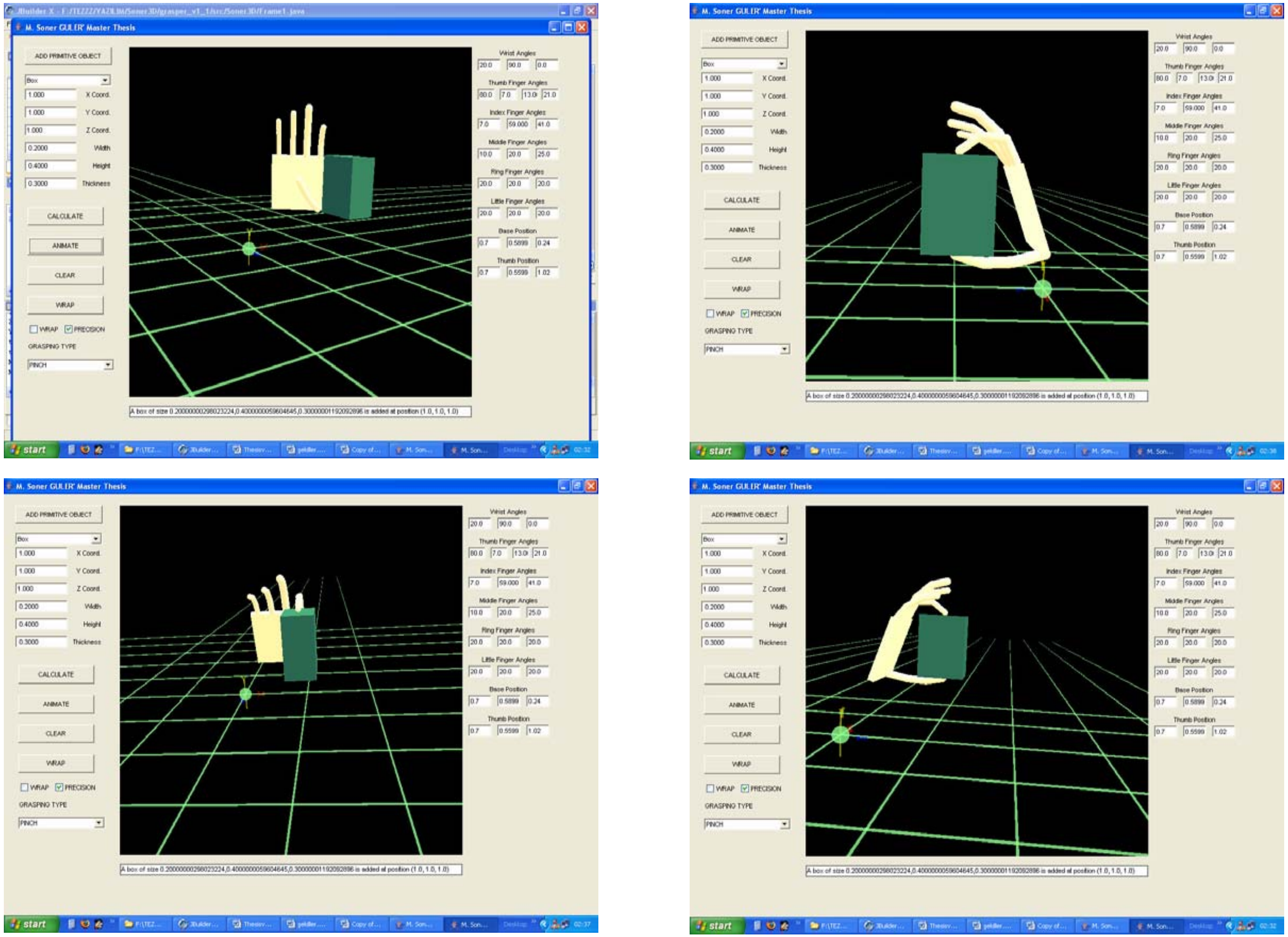


Figure 6-5 Pinch Preshaping of Large Size Prismatic Objects

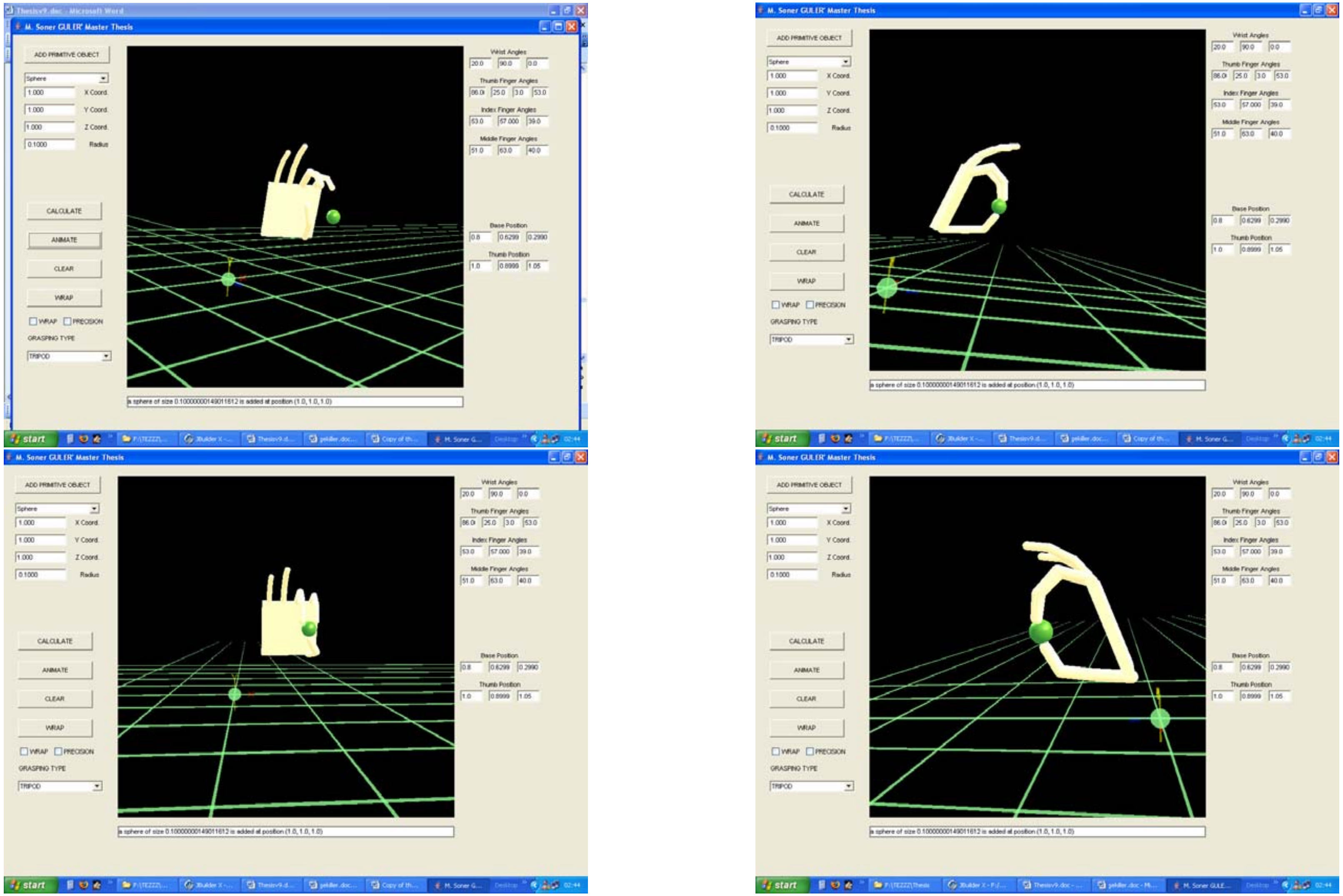


Figure 6-6 Tripod Preshaping of Small Size Cylindrical Objects

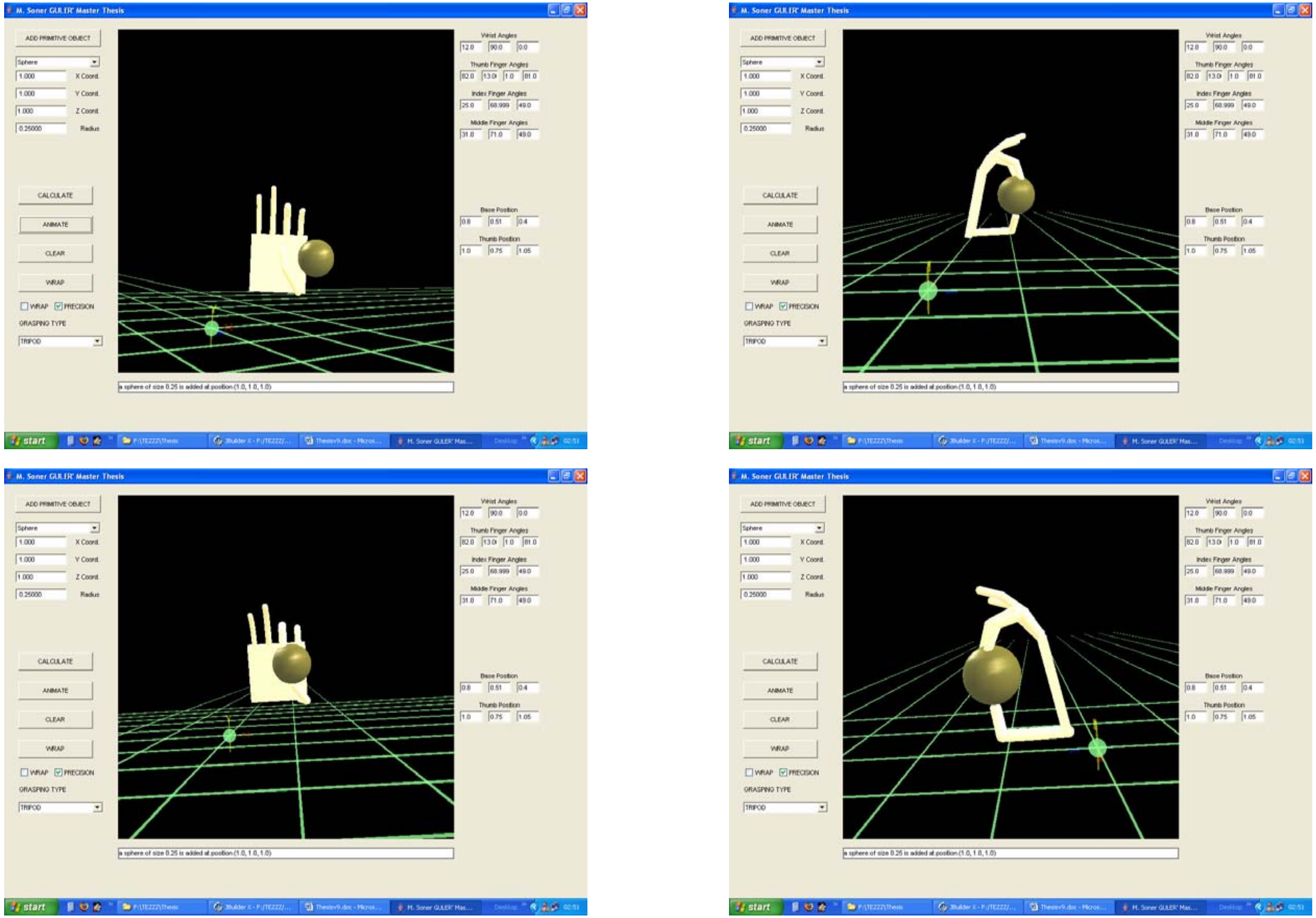


Figure 6-7 Tripod Preshaping of Middle Size Spherical Objects

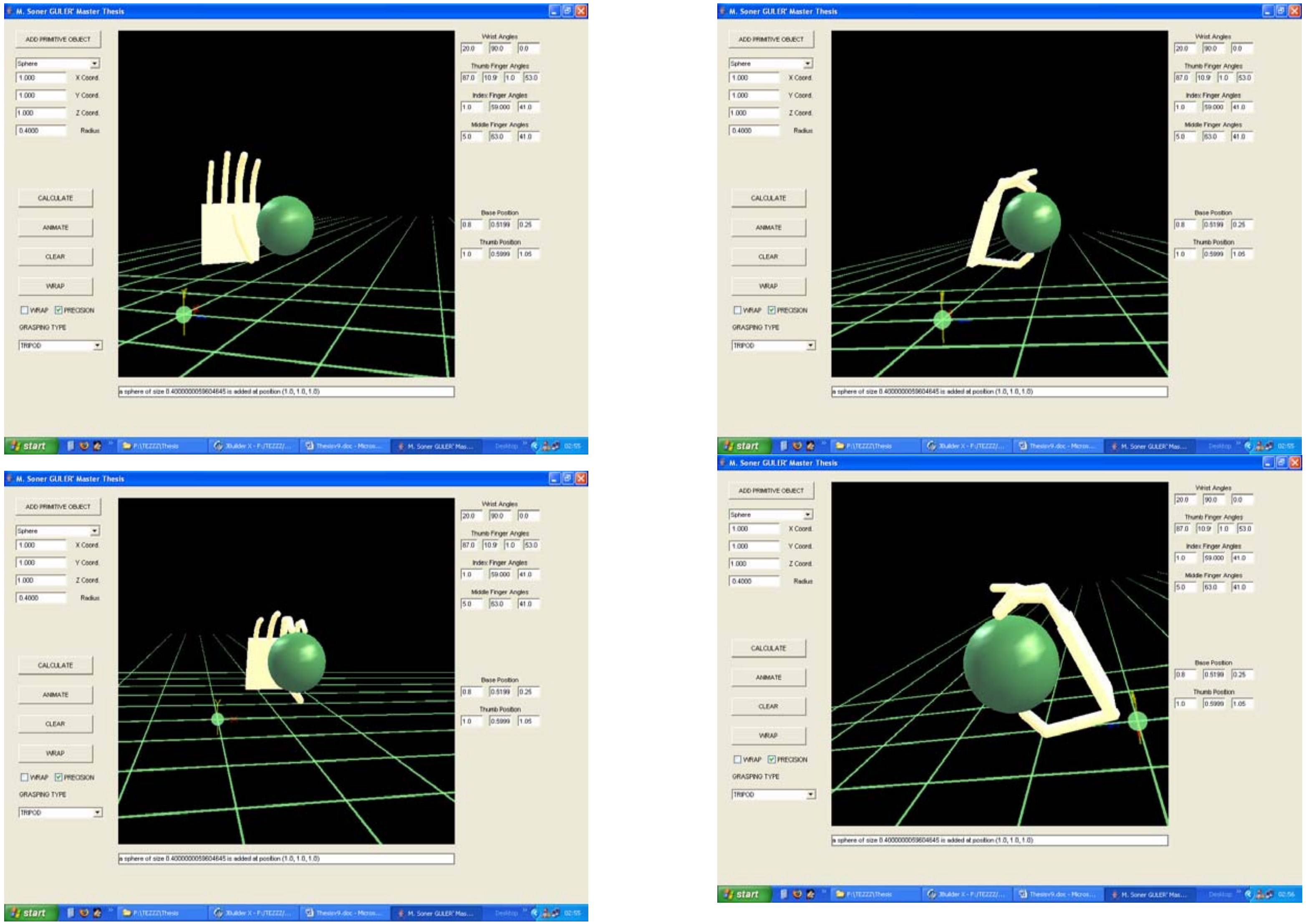


Figure 6-8 Tripod Preshaping of Large Size Spherical Objects

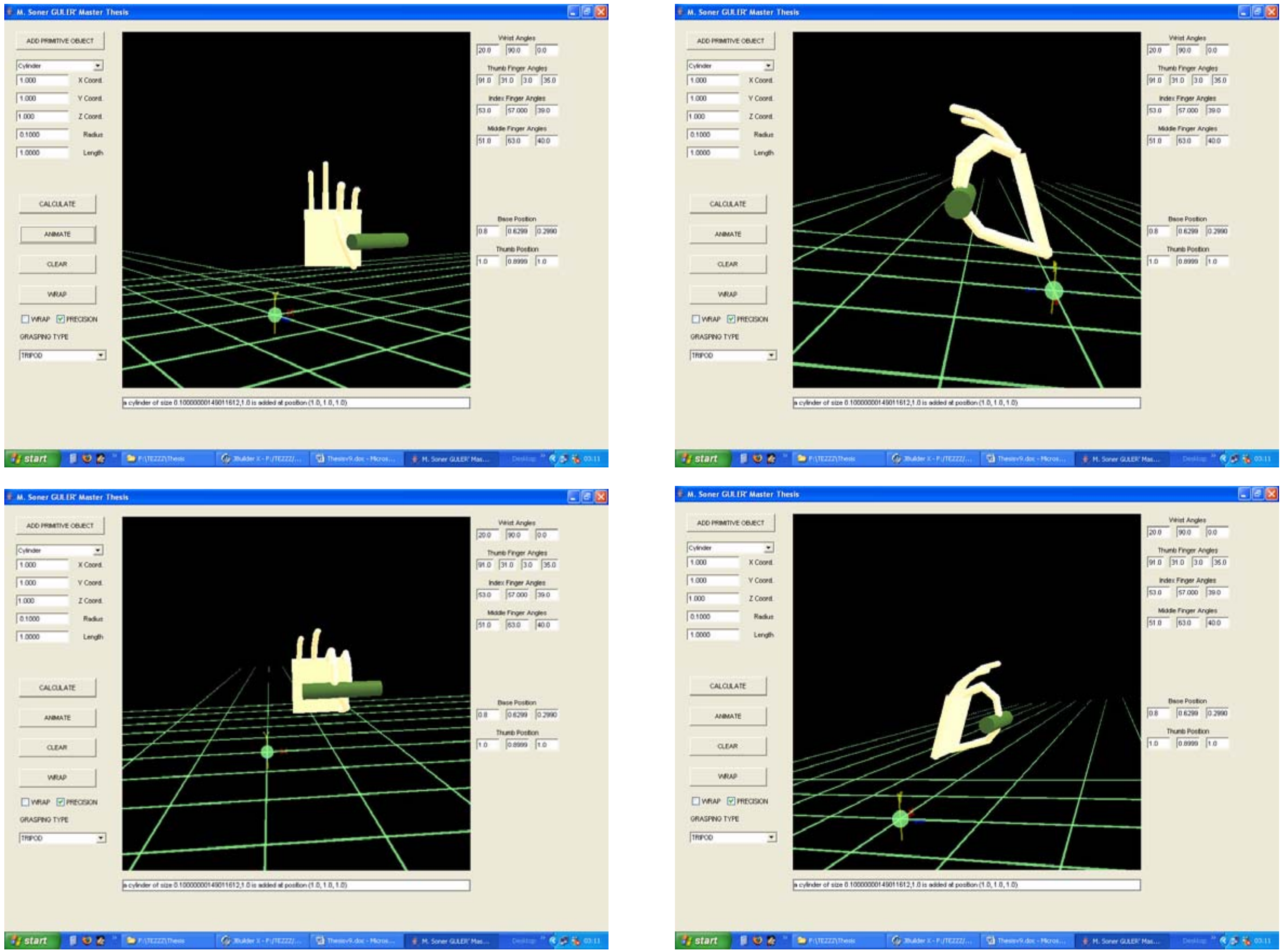


Figure 6-9 Tripod Preshaping of Small Size Cylindrical Object

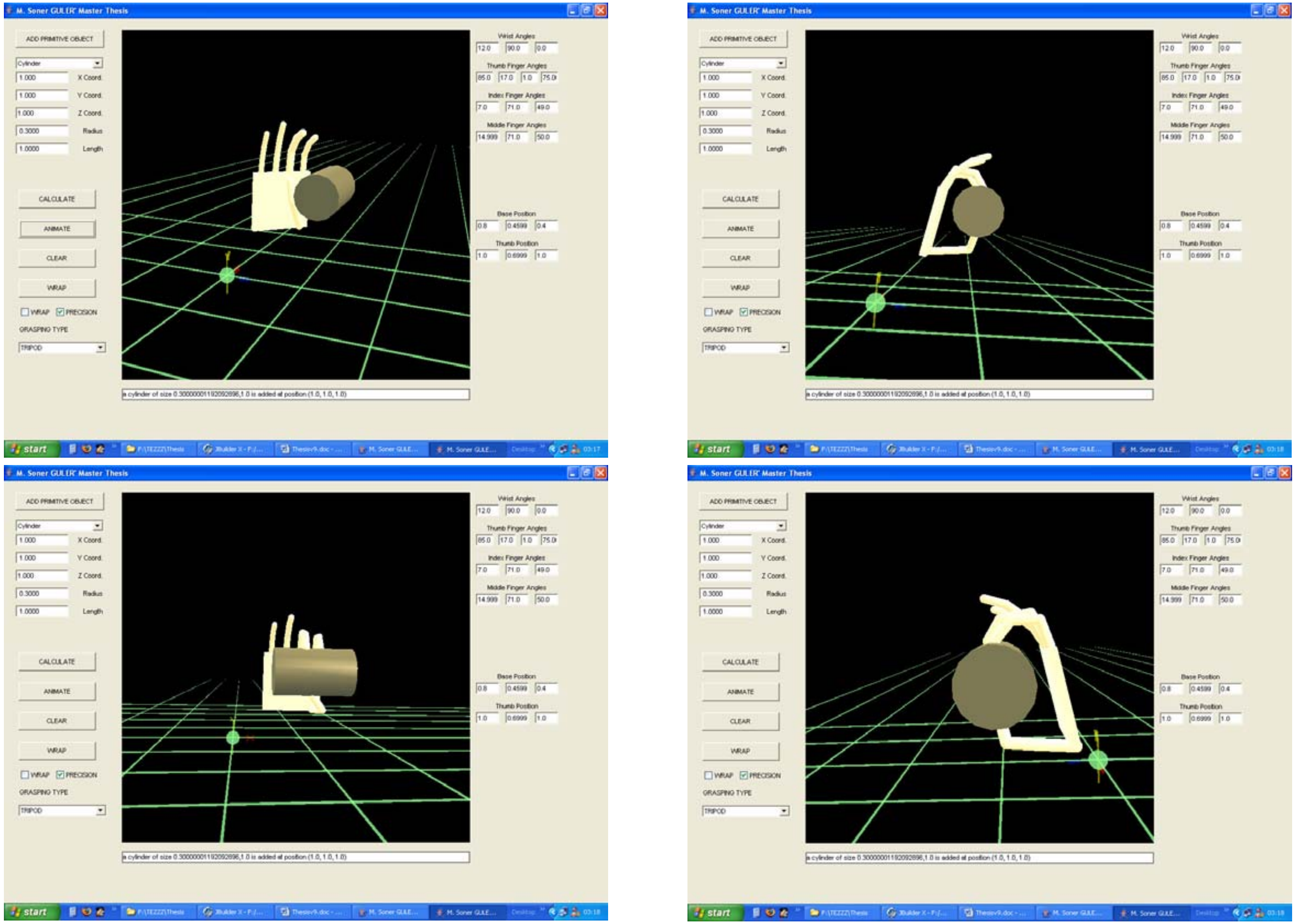


Figure 6-10 Tripod Preshaping of Middle Size Cylindrical Object

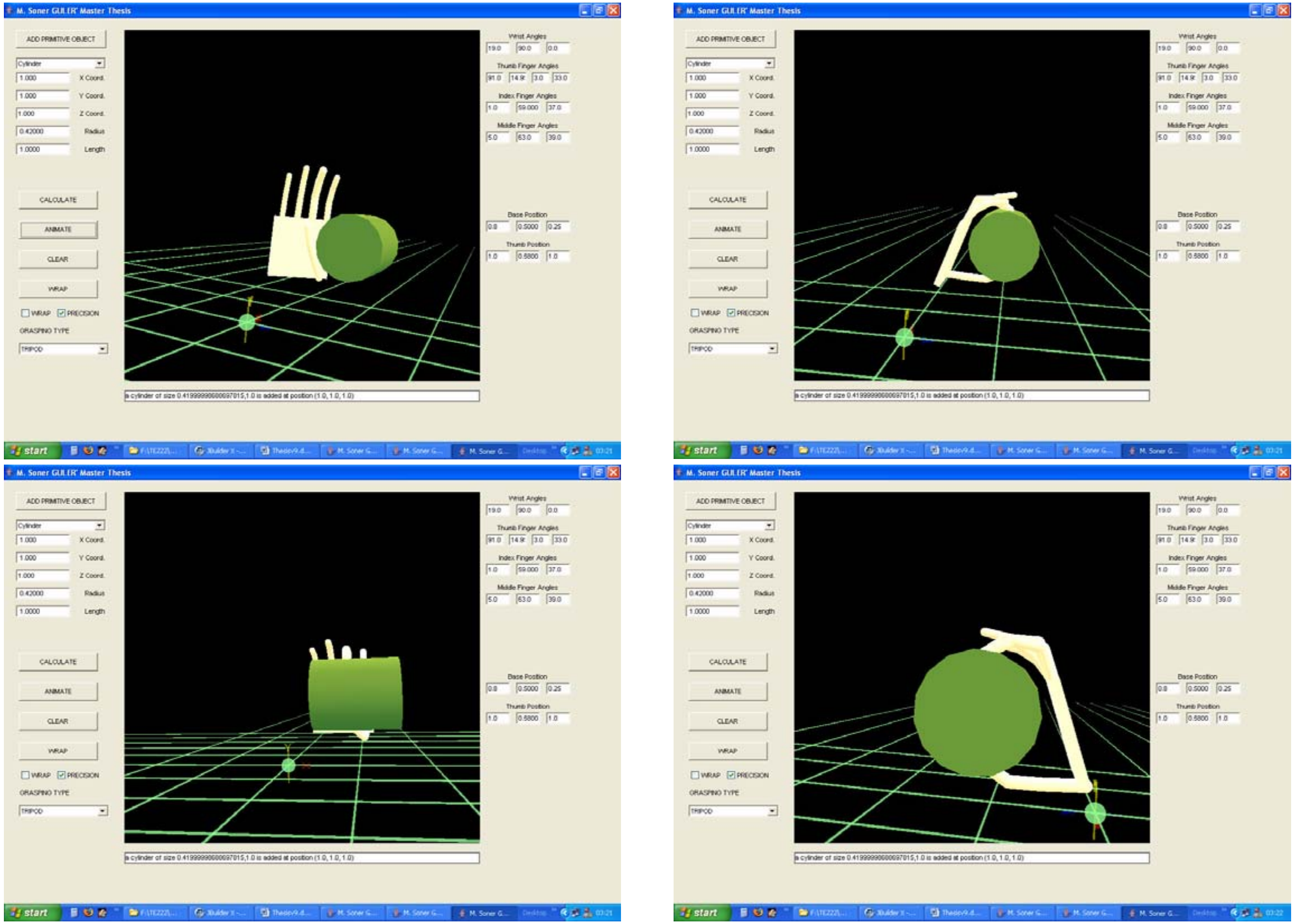


Figure 6-11 Tripod Preshaping of Large Size Cylindrical Object

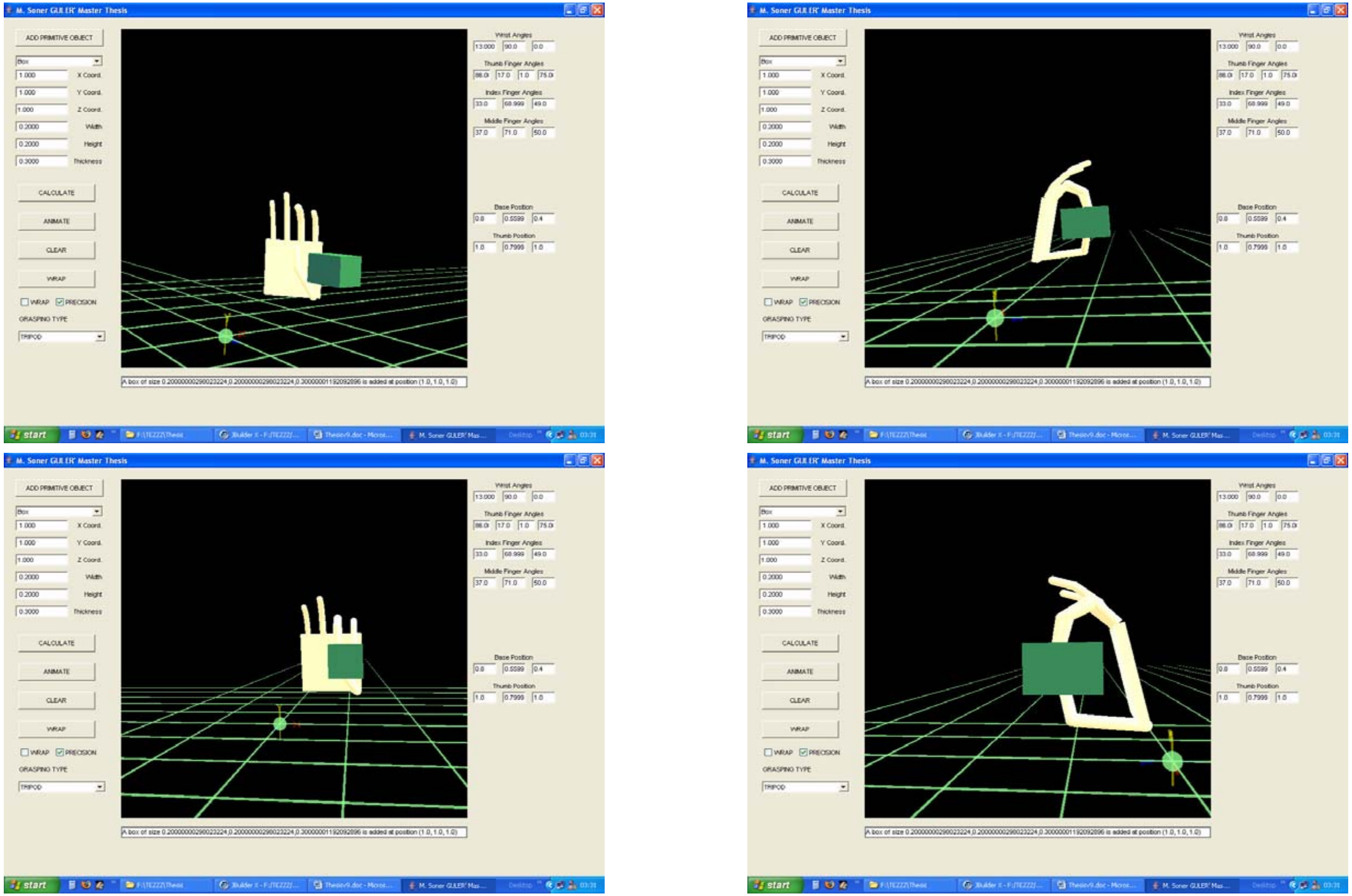


Figure 6-12 Tripod Preshaping of Middle Size Prismatic Object

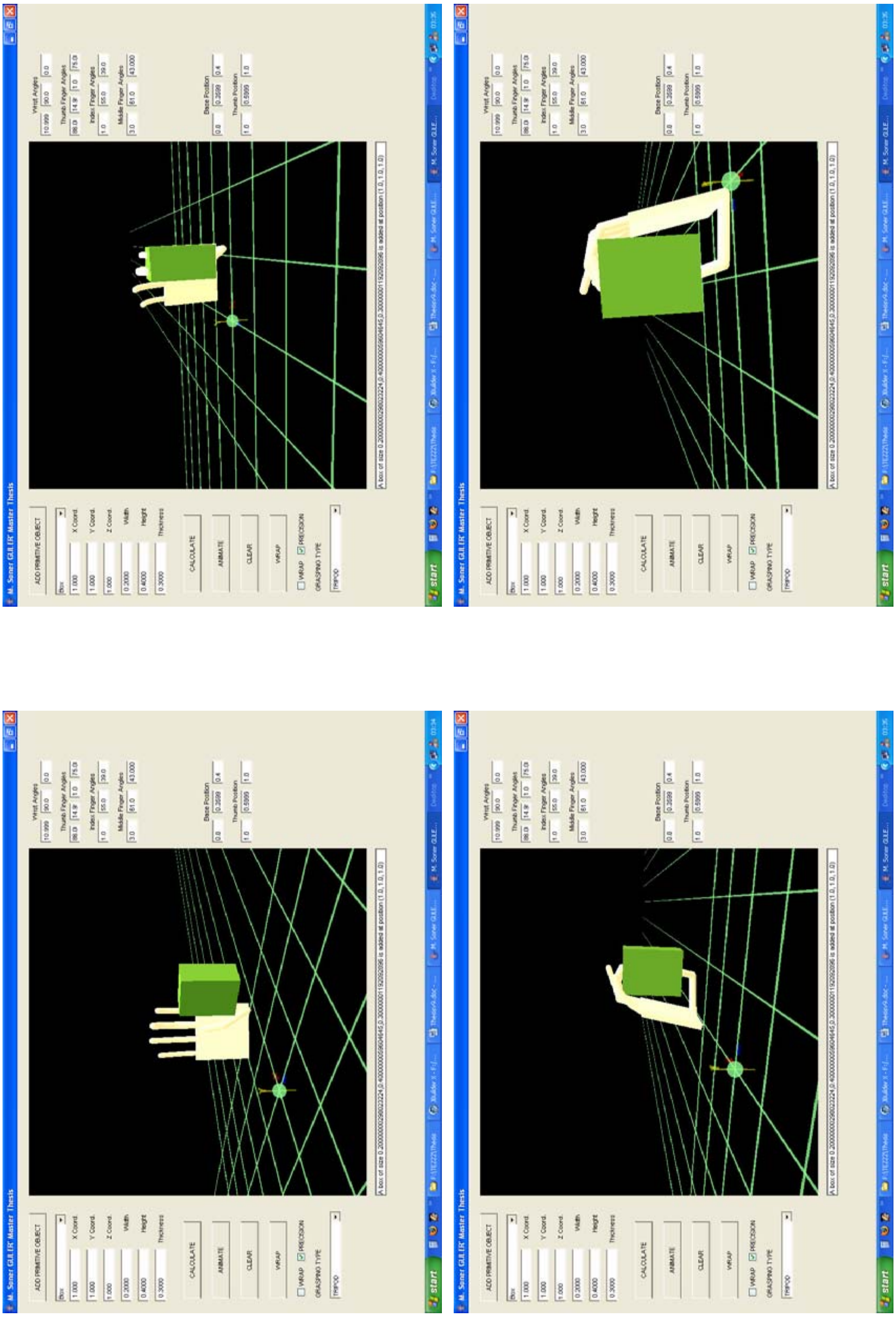


Figure 6-13 Tripod Preshaping of Large Size Prismatic Object

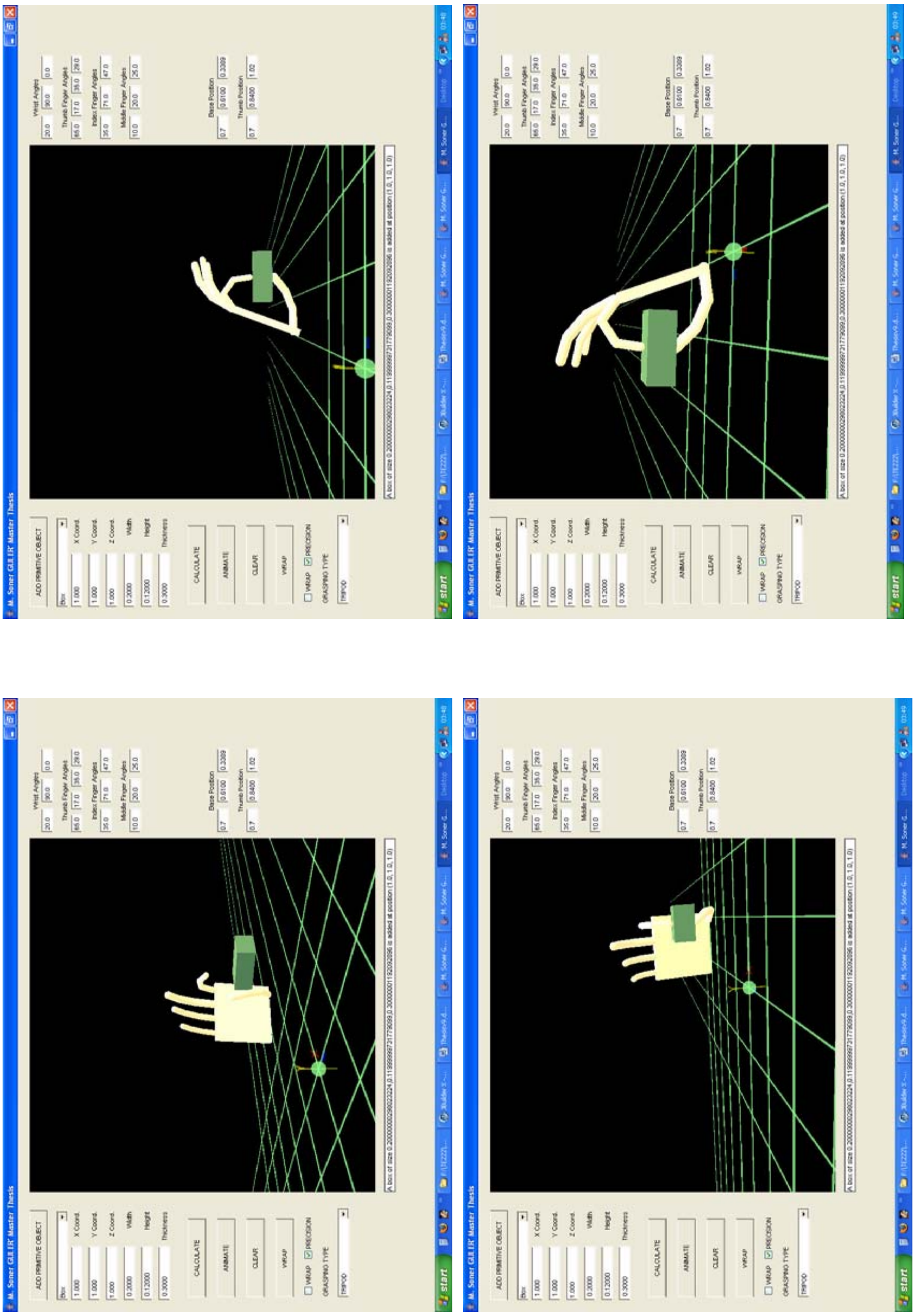


Figure 6-14 Tripod Preshaping of Middle Size Prismatic Object

6.2. Wrap Type Preshaping

Wrap type preshaping animation is based on collision detection. For this purpose new hand model has been developed to detect collision between object and hand (Figure 6.15). To prevent wrong collision information, hand joints has been left empty. If two bones intersect each other, JAVA 3D will give us collision every time. This leads us wrong motions.

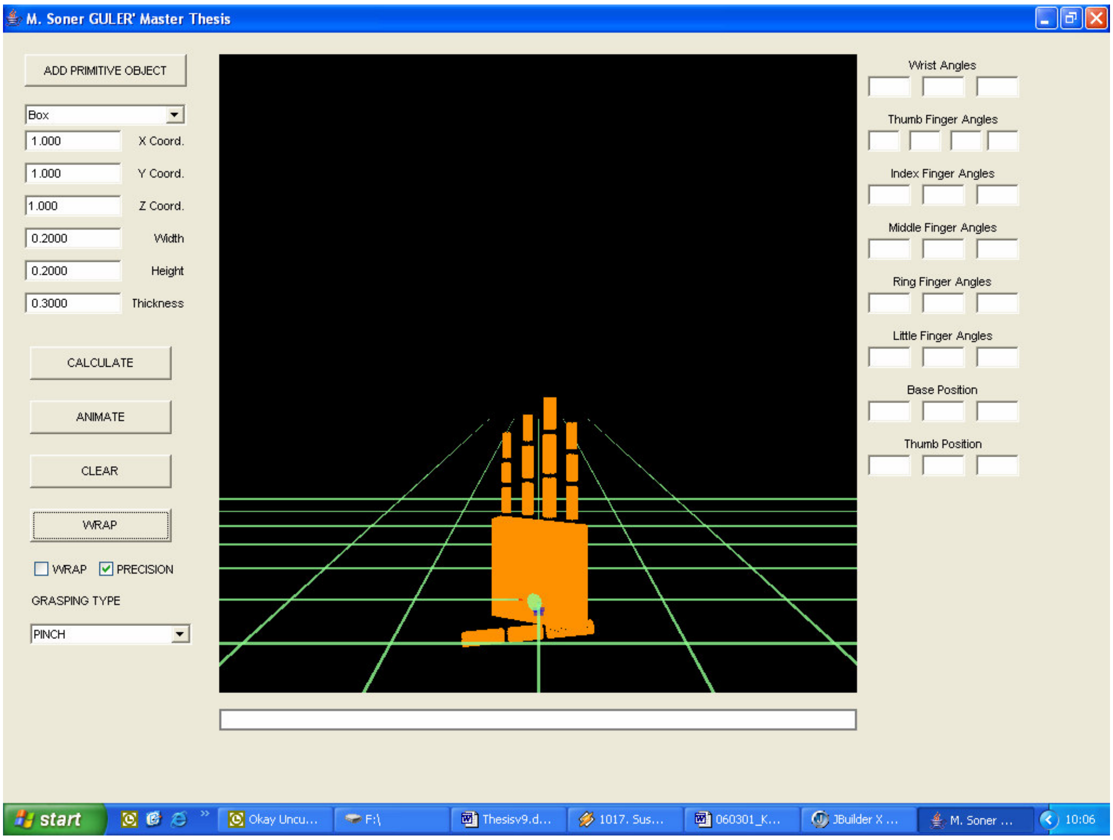


Figure 6-15 Collision Detection Hand

Each bone has own collision detection property. To visualize collision, bone' colour has been changed to green in case of collision. There is a hierarchical collision control mechanism. Upper bones has authority to below bones, that is if there is a collision on the proximal bone of hand, all bones below the proximal bone stop to move.

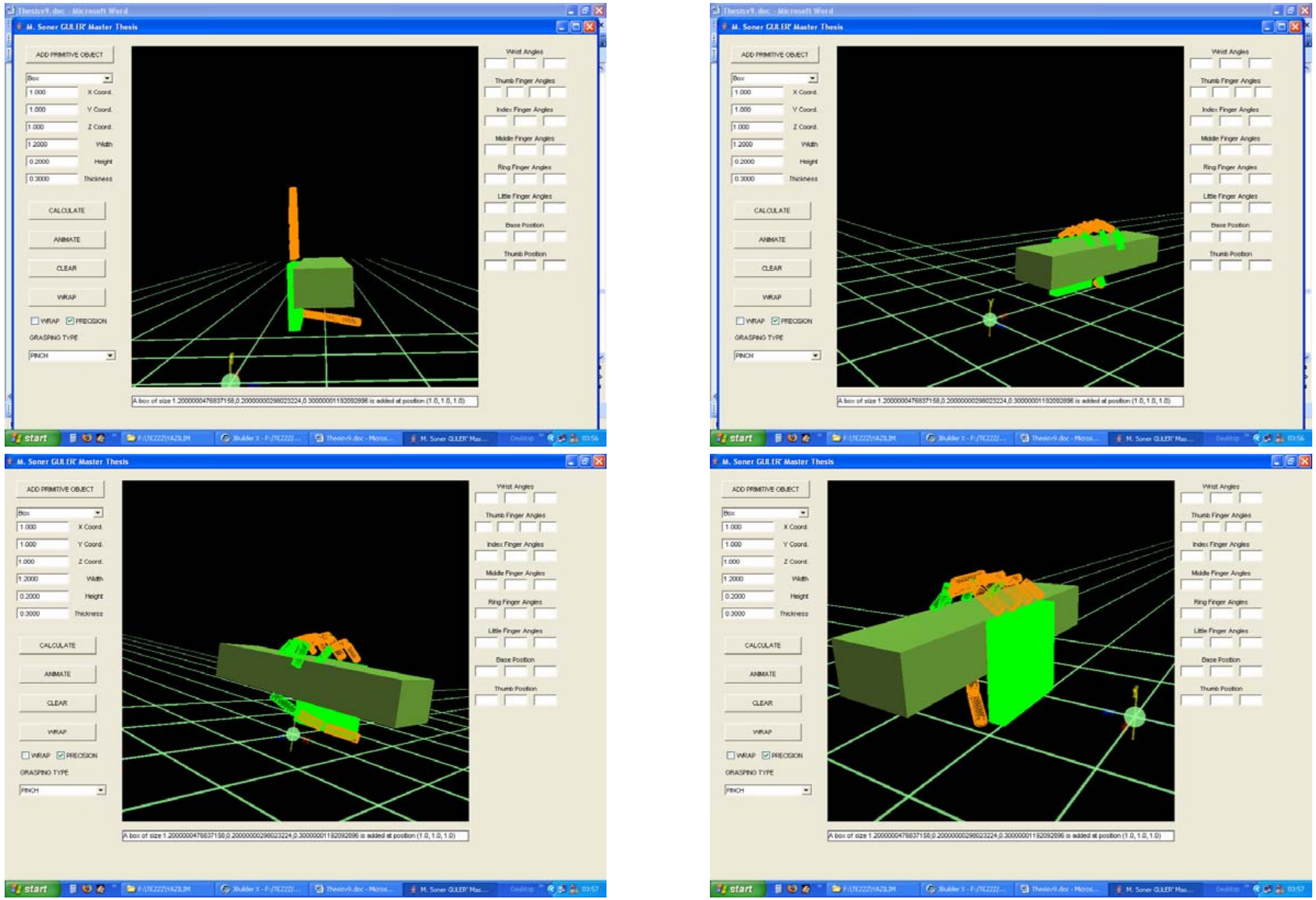


Figure Error! No text of specified style in document.**-1 Preshaping of Prismatic Object**

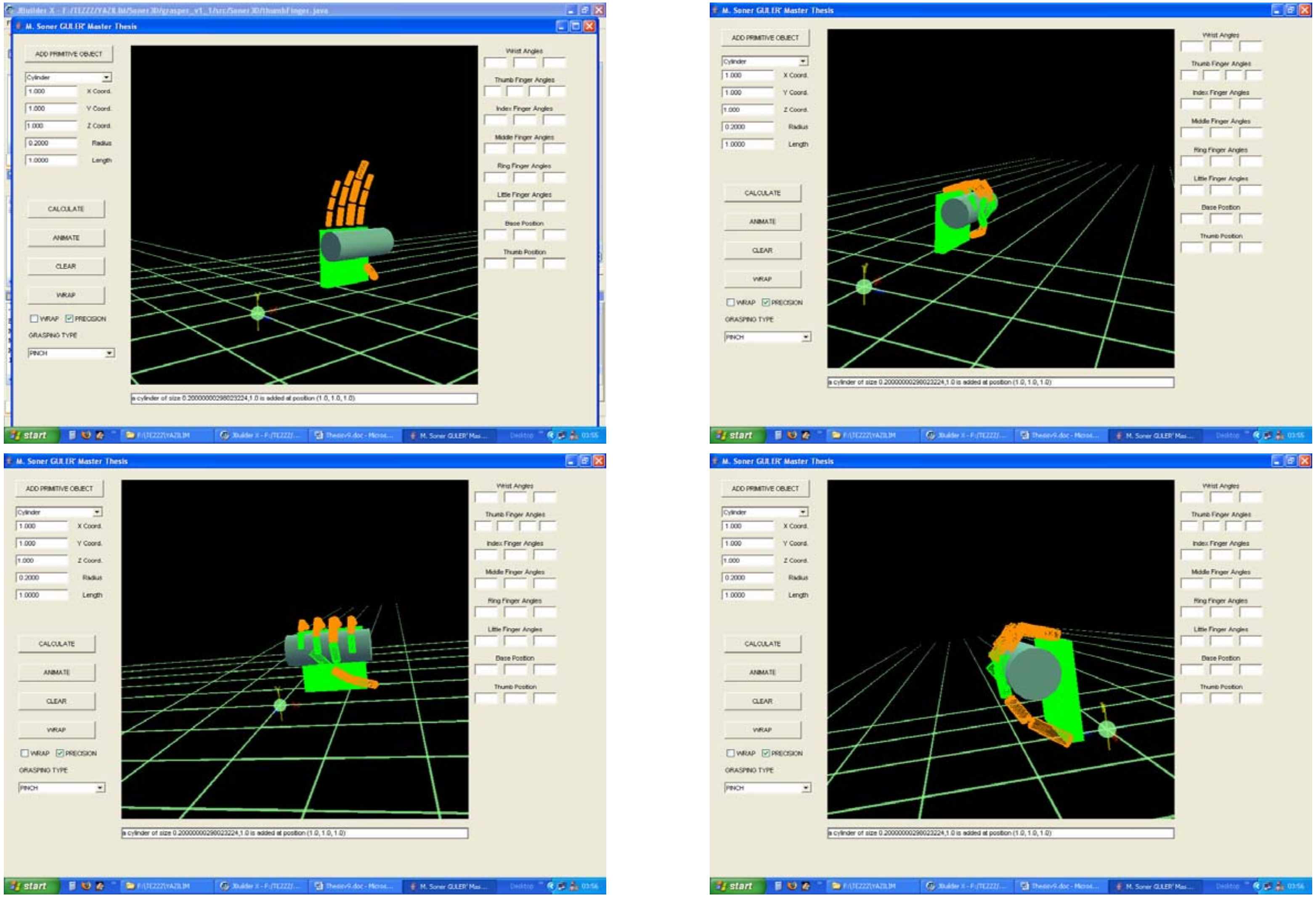


Figure Error! No text of specified style in document.**-2 Preshaping of Cylindrical Object**

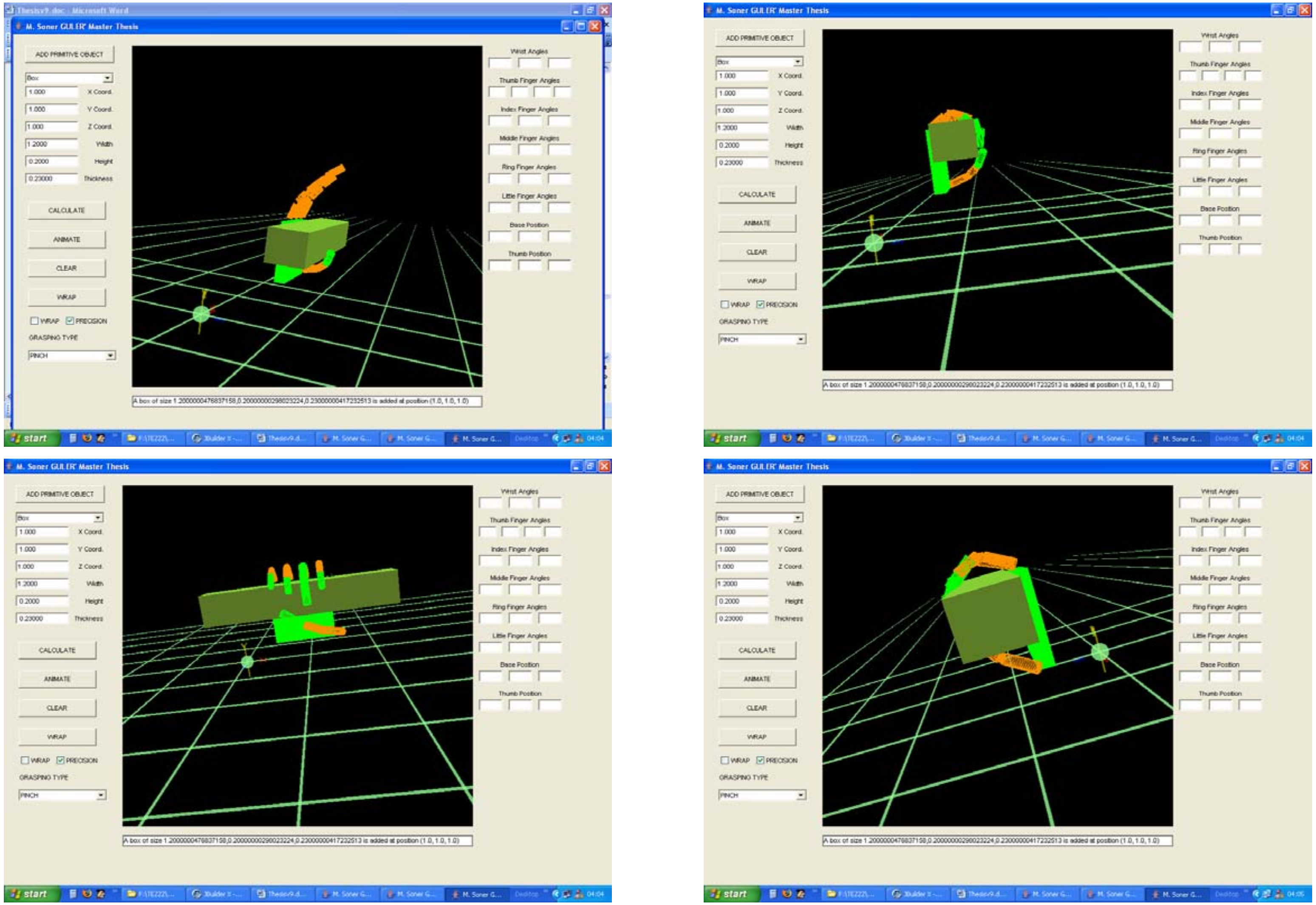


Figure Error! No text of specified style in document.**-3 Preshaping of Large Prismatic Object**

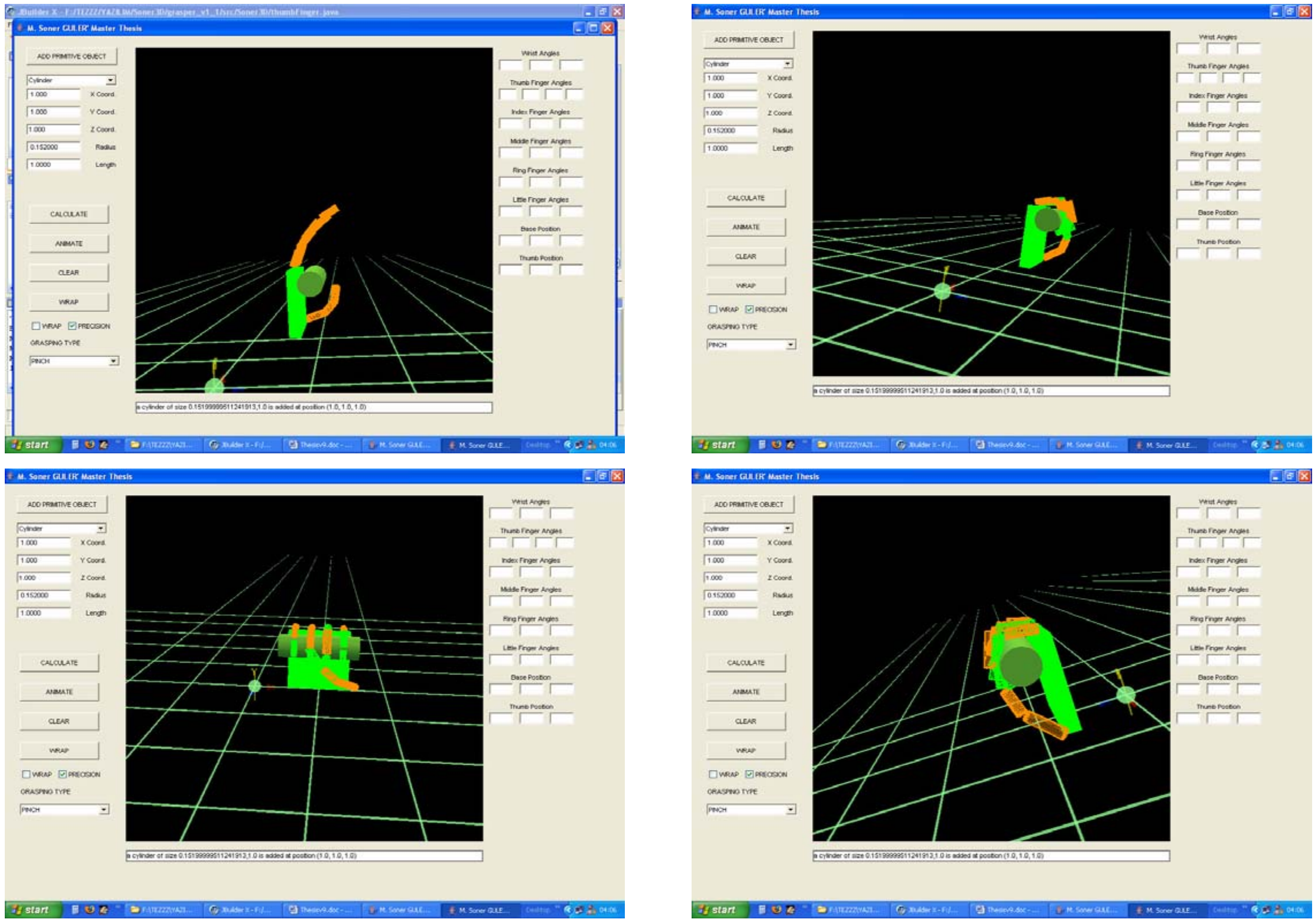


Figure Error! No text of specified style in document.**-4 Preshaping of Small Cylindrical Object**

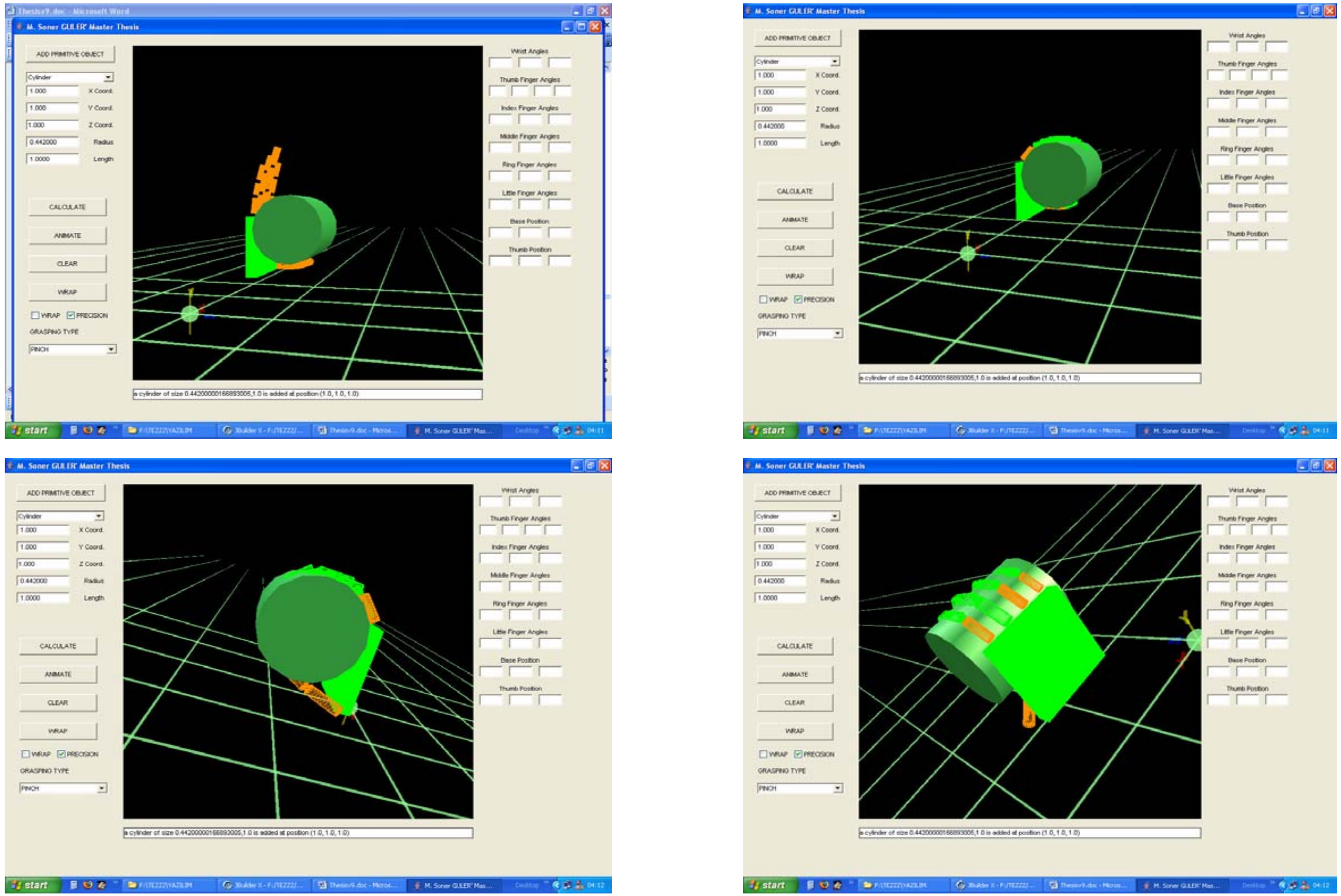


Figure Error! No text of specified style in document.**-5 Preshaping of Large Cylindrical Object**

6.3. Performance Analysis

To analyze the performance of the animation is worth to mention animation environments. Animation was made on Java 3D platform and using Pentium IV, 2.8 GHz Prescott processor. Figure of merit of the performance is time and object size for precision and wrap type preshaping animation, respectively.

Calculation time of the finger angles is related with size of the object. Object size affects the hand base position with respect to algorithm. Smaller objects need closer base position. In our algorithm, there are three loops for precision type grasp preshaping, outer loop determines the base position. If any change of the base position, calculation time of the finger angles takes much more time. In figures, relations of the object size and calculation times are given.

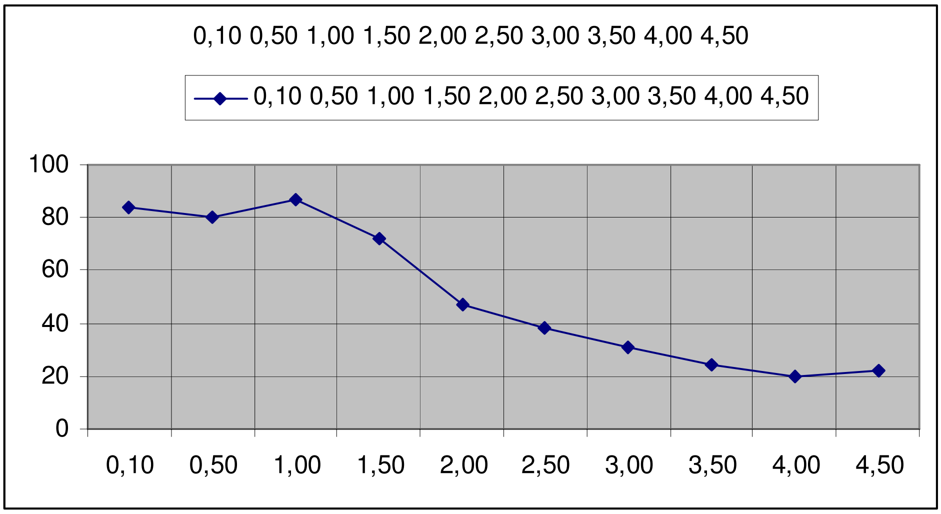


Figure 6-21 Calculation time of the finger joint angles for the spherical object

X axis of the graphs shows the object size (diameter of the sphere), Y axis shows the animation time in seconds.

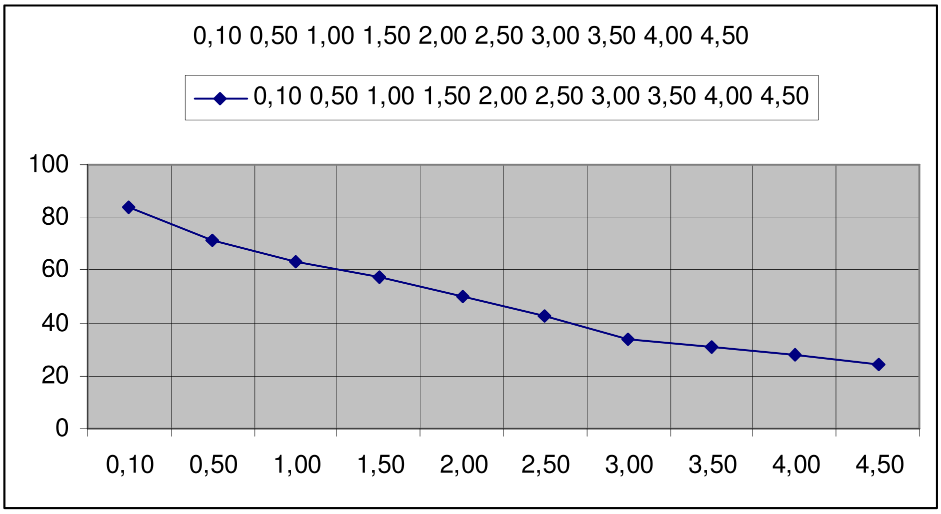


Figure 6-22 Calculation time of the finger joint angles for the cylindrical object

X axis of the graphs shows the object size (diameter of the cylinder), Y axis shows the animation time in seconds.

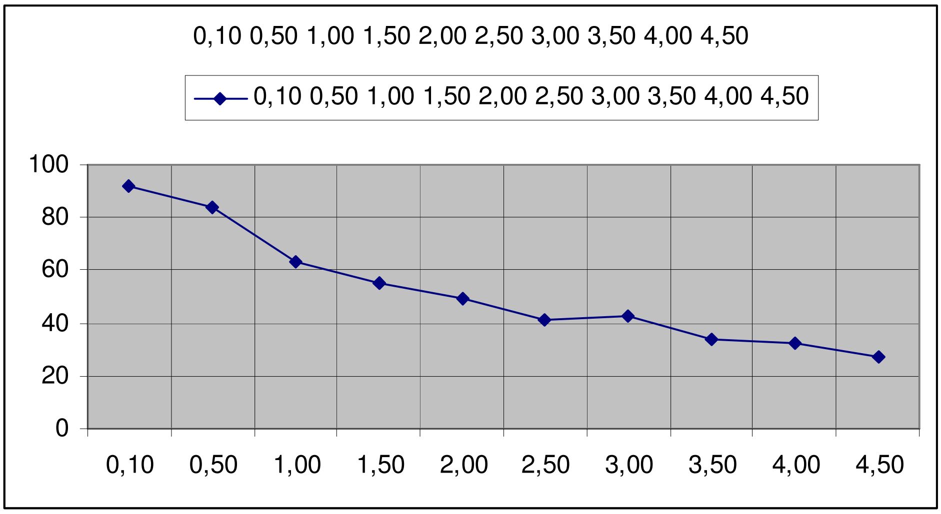


Figure 6-23 Calculation time of the finger joint angles for the prismatic object

X axis of the graphs shows the object size (height of the object), Y axis shows the animation time in seconds.

In case of wrap type preshaping, collision detection algorithm is developed. Collision detection is Java 3D tool and it is responsible for detecting of intersection of the objects on the scene. Collision detection library of the java is not so powerful and capable but it is possible to implement collision detection animation algorithms. One of problem of the collision library on java 3D is reorganization of the collisions. In some cases java 3d gives the early collision, source of the early collision is to use java primitive objects such as box, sphere and cylinder. Particularly, using sphere on scene causes early collisions. To prevent this malfunctionality, we have defined new objects without using the predefined objects. In spite of this, we got some wrong collision when making sphere preshaping. One of the screen shots of the wrap type sphere preshaping is given below.

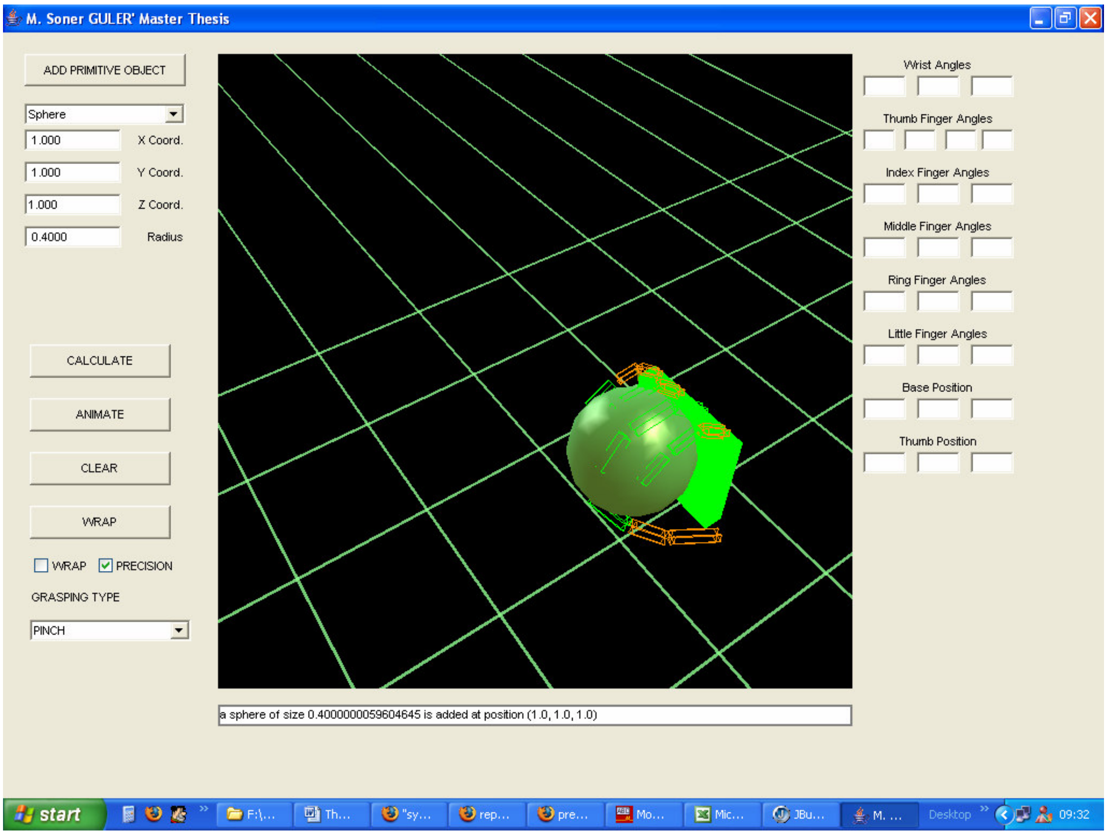


Figure 6-24 Sphere preshaping for wrap type

In figure 6-24, some fingers enters inside of the object, because of the malfunctionality of the Java 3d.

CHAPTER 7

SUMARRY AND CONCLUSION

7.1. Summary

In this thesis, we aimed to develop a hand preshaping animation algorithm for precision and wrap types in accordance with human hand constraints. The algorithm is implemented using Java language and Java 3D platform is used as implement two different approaches for precision and wrap types. Graphical user interface of program consists of scene in which animation can be seen, buttons and textbox. User can select one of the predefined objects which are box, sphere and cylinder, and preshaping type. With the chosen object and preshaping type, animation can be started.

Our hand model is based on human hand bone structure. To be more realistic, real human bone ratios are used. Hierarchical approach has been used to implement hand on JAVA 3D. Benefits of hierarchical approach are to minimize computational complexity. JAVA 3D TransformGroup Node made easy to find relative positions of finger bones to each others. To form similar human skeletal hand model, cylindrical objects are used to constitute finger bones and box object is used to constitute palm of hand. All joints have own constraints and features. Dependencies of these joints are controlled by the main program. Hand model is highly hierarchical and it can be easily scaled and modified.

Hand model is developed by using object-oriented software development methodology. So our hand model can be easily improved or new hand models can be easily integrated to our study. We have left skin deformation of the hand to further studies. New hands having skin deformation capability can also be easily integrated to our study.

We have implemented hand kinematics, hand movement constraints and geometric stability constraints to find joint angles and finger tip positions for precision type of preshaping. Precision type preshaping have high computational load. Algorithm should compute base location (wrist location), fingers tip positions on the object and inverse kinematics solution should be obtained with these positions.

The developed precision type of preshaping algorithm requires two inputs, base position and finger tip position. Hand joint angles are calculated with respect to these positions. If given positions are chosen such a way to play guitar, to type etc, animation program would run with respect to these positions and for example playing guitar animation can be easily realized.

In case of wrap type preshaping, we have implemented collision detection algorithm. Collision detection on JAVA 3D can detect intersection of objects in scene. If there is a collision or intersection between objects, collision detection gives us an input identifying collision between any two objects at anywhere on the scene. JAVA 3D collision library gives only one collision at the same time and it is not possible to recognize which object collides on scene. Other drawback of collision detection library of JAVA 3D is insensitive and rough collision information in case of using predefined objects (box, sphere and cylinder) from library of JAVA 3D. Solution of this problem is to define new objects consisting of lines and planes. These are main restriction of collision detection library on JAVA 3D. Because of these restrictions, collision detection algorithm has been designed so that its input consists of single collision at any time. Fingers move in order so that index, middle, ring, little and thumb fingers moves respectively. Because of need of define new objects on the scene, hand model of collision detection algorithm is not seem more realistic.

Animation phase of program is mainly procedural but precision type of preshaping is mixed type of keyframing and procedural approaches. Keyframing animation approaches uses in-betweening between two predefined key positions. The computer interpolates to determine the positions for the intermediate frames known as in-betweening. Precision type of preshaping defines two key frames, one is start, second is final position at the beginning of the animation. Computer interpolates two positions but final position of hand is purely computational, not from library or predefined so this stage of animation is procedural. Therefore precision type of preshaping is mixed form of procedural and keyframing approaches. Whereas wrap type of preshaping only consists of procedural, there is a collision algorithm running backside and final position is not definite at beginning of the animation.

In the light of this study, it is possible to show hand configurations, joint angles and finger positions for preshaping purposes. Hand joint constraints and geometric stability analysis have been implemented, so this gives more realistic results especially for precision type preshaping. Whereas collision detection algorithm has been implemented for the wrap type preshaping, because of application purposes of it. The former is more realistic, computational and time consuming in which convergence time is approximately two or three minutes to find joint angles and finger positions. But the latter depends only on object interaction at scene and there is no need to computation so it is possible to animate faster.

7.2. Conclusion and Remarks

Old traditional animation techniques are drawing consecutive frames and playing them at some speed. This approach is time consuming and tiring for animators. Automation is needed to decrease necessary human work for an animation. For this purpose, we have introduced the automation to animation studies and animators. Automation of the animation is relatively new topic in computer vision area. In spite of new topics, obtained results show that automation gives animators flexibility, time saving and lightened work. Preshaping animation results are satisfying from many aspects. With the developing of new hand model which have skin deformation, animation of hand will be fully realistically.

We have generated an animation programming library with object oriented approach so new classes can be included and available classes can be improved. We developed classes for this study, especially, transformation class which calculates joint angles or finger tip position of four linkage kinematics chain and collision detection class which detects the collision of the objects. These classes can be adapted to many applications with little effort. With these tools and methods, animation of articulated models can be easily achieved and adapted to the other areas such as robotics.

Finally, as the method which we have developed make us hopeful to deal with the automation of animation. We have implemented two powerful methods for animation. Collision based and inverse kinematics based approaches have own advantages and disadvantages.

7.3. Future Works

Collision based approach can be used more by interactive media such as robot environment simulation and games. There is no analytical background behind the collision detection so that it is faster and user dependent. Because of implementation feasibility of this approach and short animation time, it is useful for robotics area. Any robot can be modelled and inserted on the virtual scene, movement capability can be investigated in case there is obstacle in environment of robot by using collision approach. A computer model represents a real robot in a virtual world, which is a precise copy of the robot's surroundings. The robot may repeat moves, which were previously recognized safe by simulating them on robot's model, and of course, the robot should not perform an operation, if its model collided with some virtual object during simulation. With these applications modelled robot can be protected from any damage.

Whereas kinematics based approach can also be used for interactive systems. But theoretical backgrounds of this approach give us an opportunity to simulate real objects such as its structure is articulated. This approach can be used in many applications.

Kinematics based approach can be used to investigate a robot capabilities, real movements. It gives more accurate results so that it is appropriate method for animators and simulators. Articulated objects such as human animation based on the realism is one of general application areas of kinematics based animation.

REFERENCES

[1] (A. Bicchi, C. Melchiorri, and D. Ablluchi, 1995)

On the mobility and manipulability of general multiple limb robots. IEEE Transaction on Robotics & Automation, 11(2):215-228, 1995.

[2] (A. Bicchi, 1996)

Hands for dexterous manipualtion and powerful preshaping: A difficult road towards simplicity. In G.Giralt and G.Hirzinger, editors, Robotics Research: the seventh international symposium. Springer, 1996.

[3] (A. Bicchi and V. Kumar, 2000)

Robotic preshaping and contact: A review. In Proc. of IEEE International Conference on Robotics & Automation, pages 348-353, San Francisco, CA, 2000.

[4] (A. Cole, J. Hauser, and S. Sastry, 1988)

Kinematics and control of a multifingered robot hand with rolling contact. In Proc. of IEEE International Conference on Robotics & Automation, pages 228-233, 1988.

[5] (A.M. Okamura, N. Smaby, and M.R. Cutkosky, 2000)

An overview of dexterous manipulation. In Proc. of IEEE International Conference on Robotics & Automation, pages 255-262, San Francisco, CA, 2000.

[6] (A. P. Georgopoulos, G. Pellizzer, A. V. and M. H. Schieber, 1999)

Neural Coding of Finger and Wrist Movements. Journal of Computational Neuroscience, 6, 279.288,

[7] (Alan Watt and Mark Watt., 1991)

Advanced Rendering and Animation Techniques: Theory and Practice. Addison-Wesley, Reading, MA.

[8] (B. Mishra, J.T. Schwartz, and M. Sharir, 1987)

On the existence and synthesis of multifinger positive grips. Algorithmica, 2:541-558, 1987.

[9] (C. Cai and B. Roth, 1987)

On the spatial motion of rigid bodies with point contact. In Proc. of IEEE International Conference on Robotics & Automation, pages 686-695, 1987.

[10] (Chris Welman, 1993)

Inverse kinematics and geometric constraints for articulated figure manipulation. M.Sc. thesis, Simon Fraser University: April 1993.

[11] (C. K. Liu and Z. Popovi´c, 2002)

Synthesis of Complex Dynamic Character Motion from Simple Animations. ACM Computer Graphics (Proc. of SIGGRAPH 2002), 408.416, 2002.

[12] (C. Häger-Ross and M. H. Schieber, 2000)

Quantifying the Independence of Hand Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. The Journal of Neuroscience, 20(22)8542.8550,

[13] (Chen, I., and Burdick, J., 1983)

Finding antipodal point preshaping on irregularly shaped objects. In Proc. 1992 ZEEE Znt. Conf. Robotics Automat. (Nice, FRANCE, May 1992), vol. 3, pp. 2278-2283.

[14] (Cooney, W. P., M. J. Lucca, et al. 1981)

The kinesiology of the thumb trapeziometacarpal joint. Journal of Bone Joint Surgery 63A: 1371-1381.

[15] (Cutkosky and Wright, 1986)

M. R.Cutkosky and P.K. Wright. Modeling manufacturing grips and correlation with the design of robotic hands. In Proc. of the 1986 IEEE International Conference on Robotics and Automation, pages 1533-1539, San Francisco, CA, 1986.

[16] (D. Montana, 1988)

The kinematics of contact and preshap. International Journal of Robotics Research, 7(3), 1988.

[17] (D. Montana, 1991)

The condition of contact preshap stability. In Proc. of IEEE International Conference on Robotics & Automation, pages 412-417, 1991.

[18] (D. Montana, 1995)

The kinematics of multifingered manipulation. IEEE Transaction on Robotics & Automation, 11(4):491-503, 1995.

[19] Davidoff NA, Freivalds A (1993) A graphical model of the human hand using CATIA. International Journal of Industrial Ergonomics 12: 255-264

[20] (E. Paljug an X. Yun and V. Kummar, 1994)

Control of rolling contacts in multi-arm manipulation. IEEE Transaction on Robotics & Automation, 10(4):441-452,1994.

[21] (E. Rimon and J. Burdick, 1996)

On force and form closure for multiple finger preshaping. In Proc. of IEEE International Conference on Robotics & Automation, pages 1795-1800, 1996.

[22] (Ferrari, C., and Canny, J., 1995)

Planning optimal preshaping. In Proc. 1992 ZEEE Znt. Conf. Robotics Automat. (Nice, FRANCE, May 1992), vol. 3, pp. 2290-2295.

[23] (Frank Thomas and Ollie Johnston, 1984)

Disney Animation: The Illusion of Life. Abbeville Press, New York, NY.

[24] (G. Wohlke, 1990)

A programming and simulation environment for the karlsruhe dextrous hand. Journal of Robotics and Autonomous Systems, 9:243-263, 1990.

[25] (G. Bekey, R. Tomovic, and I. Ziljkovic, 1990)

Control architecture the belgrade/usc hand. In S.T. Venkataraman and T. Iberall, editors, Dextrous Robot Hands. Springer-Verlag, 1990.

[26] (Girard. 11. and Maciejewski.A.A, 1985)

Computational modeling for computer generation of Iegged figures. Computer Graphics, 19(3):263-270. 1995.

[27] (Griffiths, H.E., 1943)

Treatment of the Injured Workman, Lancet, 1943.

[28] (H. Liu, J. Butterfass, S. Knoch, P. Meusel, and G. Hirzinge, 1999)

A new control strategy for dlr's multisensory articulated hand. IEEE Control Systems, 19(2):47-54, April 1999.

[29] (Guo, G., Gruver, W., and Jin, K., 1989)

Preshap planning for multifingered robot hands. In Proc. 1992 ZEEE Znt. Conf. Robotics Automat. (Nice, FRANCE, May 1992), vol. 3, pp. 2284-2289.

[30] (H. Liu, P. Meusel, J. Butterfass, and G. Hizinger, 1998)

DLR's multisensory articulated hand part ii: Parallel torque/position control system. In Proc. of IEEE International Conference on Robotics & Automation, pages 2087-2093, 1998.

[31] (H. Maekawa, 1992)

Stable preshap and manipulation of 3d object by multifingered hands. In Proc. of Int. Symp. on Measurement and Control Robotics, pages 335-342, 1992.

[32] (H. Maekawa, K. Tanie, and K. Komoriya, 1995)

Tactile sensor based manipulation of an unknown object by a multi_ngered hand with rolling contact. In Proc. of IEEE International Conference on Robotics & Automation, pages 743-750, 1995.

[33] (H. Maekawa, K. Tanie, and K. Komoriya, 1997)

Tactile feedback for multifingered dynamic preshaping. IEEE Control Systems, 17(1):63-71, February 1997.

[34] (H. Maekawa, K. Tanie, and K. Komoriya, 1997)

Kinematics, static and stiffness effect of 3d preshap by multifingered hand with rolling contact at the fingertip. In Proc. of IEEE International Conference on Robotics & Automation, pages 78-85, Albuquerque, NM, 1997.

[35] (H.R. Nicholls, 1992)

Advanceed Tactile Sensing for Robotics, volume 5 of World Scientific Series in Robotics and Automated Systems. World Scientific, 1992.

[36] (H.R. Nicholls and M.H. Lee, 1989)

A survey of robot tactile sensing technology. International Journal of Robotics Research, 8(3):3-30, 1989.

[37] (H. Rijpkema and M. Girard, 1991)

Computer Animation of Knowledge-Based Human Preshaping. ACM Computer Graphics (Proc. of SIGGRAPH 91), 339.347, 1991

[38] (Hollister, A., W. L. Buford, et al., 1992).

The axes of rotation of the thumb carpometacarpal joint. Journal of Orthopaedic Research 10: 454-460.

[39] (I.D. Walker, 1998)

Multifingered hands: A survey. In P.Chiacchio and S.Chiaverini, editors, Complex Robotic Systems, pages 129-160. Springer, 1998.

[40] (I. McCammon and S. Jacobsen, 1990)

Tactile sensing and control for the utah/mit hand. In S.T. Venkataraman and T. Iberall, editors, Dextrous Robot Hands. Springer-Verlag, 1990.

[41] (Iberall, 1987)

The nature of human prehension: Three dexterous hands in one. In Proc. of the 1987 IEEE International Conference on Robotics and Automation, pages 396-401, 1987.

[42] (Iberall, 1997)

Human prehension and dexterous robot hands. The International Journal of Robotics Research, 16(3):285-299, June 1997.

[43] (Isaac Victor Kerlow, 1996)

The Art of 3-D Computer Animation and Imaging. Van Nostrand Reinhold, New York,NY, 1996.

[44] (J. Butterfass, G. Hizinger, S. Knoch, and H. Liu., 1998)

DLR's multisensory articulated hand part i: Hard- and software architecture. In Proc. of IEEE International Conference on Robotics & Automation, pages 2081-2086, 1998.

[45] (K. Shimoga, 1996)

Robot preshap synthesis algorithms: A survey. International Journal of Robotics Research, 15(3):230-264, 1996.

[46] (K. Pullen and C. Bregler, 2002)

Motion Capture Assisted Animation: Texturing and Synthesis. ACM Computer Graphics (Proc. of SIGGRAPH 91), 501.508,

[47] (Kramer, J. F. 1996)

Determination of thumb position using measurements of abduction and rotation. United States Patent #5482056.

[48] (Korein J, 1985)

Geometric investigation of reaching, MIT Press kinematics, Cambridge, 1985

[49] (Landsmeer J.,1963).

Coordination of Finger-Joint motions J. of Bone and Joint surgery, Vol 45-A, 1963

[50] (Lyons, 1985)

D. Lyons. A simple set of preshaping for a dexterous hand. In Proc. of the 1985 IEEE International Conference on Robotics and Automation, pages 588-593, 1985.

[51] (L. Kovar, M. Gleicher and F. Pighin., 2002)

Motion Graphs. ACM Computer Graphics (Proc. of SIGGRAPH 2002), 473.482,

[52] (L. Moccozet, 1996)

Hand Modelling and Animation for Virtual Humans. Ph. D. thesis, University of Geneva,

[52] (J. K. Salisbury, 1985)

Kinematics and force control of articulated robot hands. In M. Mason and J. K. Salisbury, editors, Robot Hands and the Mechanics of Manipulation. MIT Press, 1985.

[53] (J. Kerr and B. Roth, 1986)

Analysis of multifingered hands. International Journal of Robotics Research, 4(4):3-17, 1986.

[54] (J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, N. S. Pollard, 2002) Interactive Control of Avatars Animated with Human Motion Data. ACM Computer Graphics (Proc. of SIGGRAPH 2002), 491.500,.

[55] (J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat, and J.-P. Merlet, 1997) On computing four-finger equilibrium and force-closure preshaping of polyhedral objects," Int. J. Robot. Res., vol. 16, no. 1, pp. 11–35,

[56] (J. Trinkle, 1987)

Mechanics and planning of enveloping preshaping. PhD thesis, Dept of CIS, Univ. of Penn., 1987.

[57] (J. Trinkle, A. Farahat, and P. Stiller, 1995)

First-order stability cells of active multirigid-body systems. IEEE Transaction on Robotics & Automation, 11(4):545-557, 1995.

[58] (J.W. Hong, G. La_erriere, B. Mishra, and X.L. Tang, 1990)

Fine manipulation with multifinger hand. In Proc. of IEEE International Conference on Robotics & Automation, pages 1568-1573, 1990.

[59] (J- Lasseter, 1987)

Principles of traditional animation applied to: 3d computer animation. Computer Graphics. 21(4):35-44. Proceedings of SIGGRAPHS, 1987.

[60] (J. Lee and T. Kunii, 1995)

Model-based Analysis of Hand Posture. IEEE Computer Graphics and Applications, 15(5), 77-86, 1995

[61] (Jameson, J., and Leifer, L., 1987)

Automatic preshaping: An optimbation approach. ZEEE h n s . Syst., Man, Cybem. SMC-17, 5 (Sept./Oct. 1987).

[62] (James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes., 1990) Computer Graphics, Principles and Practice, Second Edition. Addison-Wesley, Reading, MA.

[63] (M. Cutkosky, 1985)

Preshaping and Fine Manipulation for Automated Manufacturing. Kulware-Academic Publisher, 1985.

[64] (M. Cutkosky, P. Akella, R. Howe, and I. Kao, 1987)

Preshaping as a contact of sports Technical report, Department ofMechanical Engineering, Stanford University, 1987.

[65] (M. Buss and K. Kleinmann, 1996)

Multifingered preshaping experiments using realtime preshaping force optimization. In Proc. of IEEE International Conference on Robotics & Automation, pages 1807-1812, 1996.

[66] (M. H. Schieber., 1996)

Individuated Finger Movements: Rejecting the Labeled-Line Hypothesis. Editors A. M. Wing, et al. Hand and Brain: The Neurophysiology and Psychology of Hand Movements. San Diego: Academic Press, Inc.

[67] (McBride, 1942)

E. D. McBride. Disability Evaluation. J.B. Lippincott Co., Philadelphia, third edition, 1942.

[68] (Mishra and Silver, 1989)

Bud Mishra and Naomi Silver. Some discussion of static gripping and its stability. IEEE Journal on Systems, Man, and Cybernetics, 19(4):783-796, August 1989

[69] (N. Cliblak and H. Lipkin, 1994)

Asymmetric cartesian sti_ness for the modeling of comliant robotic systems. Robotics: Kinematics, Dynamics and Controls, ASME1994, Vol.72, 1994.

[70] (N. Xydas and I. Kao., 1999)

Modeling of contact mechanics and friction limit surface for soft _ngers in robotics with experimental results. International Journal of Robotics Research, 18(9):941-950, September 1999.

[71] (N. S. Pollard and J. K. Hodgins, 2002)

Generalizing Demonstrated Manipulation Tasks. Workshop on the Algorithmic Foundations of Robotics (WAFR `02), Nice, France,

[72] (Napier, 1956)

J. Napier. The prehensile movements of the human hand. Journal of Bone and Joint Surgery, 38B(4):902-913, November 1956.

[73] (P. Allen, A. Miller, P. Oh, and B. Leibowitz., 1997)

Using tactile and visual sensing with a robotic hand. In Proc. of IEEE International Conference on Robotics & Automation, pages 676-681, 1997.

[74] (P. G. Kry, D. L. James and D. K. Pai. EigenSkin, 2002)

Real Time Large Deformation Character Skinning in Hardware. Proc. of ACM SIGGRAPH 2002 Symposium on Computer Animation,

[75] (Pao and Speeter, 1989)

Lucy Pao and Thomas H. Speeter. Transformation of human hand positions for robotic hand control. In Proc. of the 1989 IEEE International Conference on Robotics and Automation, pages 1758-1763, Scottsdale, AZ, May 1989.

[76] (R.D. Howe and M.S. Cutkosky, 1996)

Practical force motion models for sliding manipulation.International Journal of Robotics Research, 15(6):557-572, 1996.

[77] (R. Fearing, 1986)

Simplified preshaping and manipulation with dextrous robot hands. IEEE Transaction on Robotics & Automation, RA-2(4):188-195, 1986.

[78] (R. Fearing, 1989)

Tactile sensing for shape interpretation. In S.T. Venkatarman and T. Iberall, editors, Dextrous robot hands. Springer-Verlag, 1989.

[79] (R.A. Grupen, T.C. Henderson, and I.D. McCammon, 1989)

A survey of generalpurpose manipulation. International Journal of Robotics Research, 8(1):38-62, 1989.

[80] (R. Howe and M. Cutkosky, 1993)

Dynamica tactile sensing: Perception of fine surface features with stress rate sensing. IEEE Transaction on Robotics & Automation, 9(2):140-151, 1993.

[81] (R. Murray and S. Sastry, 1990)

Preshaping and manipulation using multifingered robot hands. In R.W. Brockett, editor, Robotics: Proceedings of Symposia in Applied Mathematics, volume 41, pages 91-128. American Mathematical Society, 1990.

[82] (R. Murray, Z.X. Li, and S. Sastry, 1994)

A Mathematical Introduction to Robotic Manipulation. CRC Press, 1994.

[83] (Richard Taylor, 1996)

The Encyclopedia of Animation Techniques. Quarto Inc., London, England,

[84] (Rohling, R. N. and J. M. Hollerbach,1994)

Modeling and parameter estimation of the human index finger. 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 223-230.

[85] (Rohling, R. N. and J. M. Hollerbach, 1993)

Calibrating the human hand for haptic interfaces. Presence 2(4): 281-296.

[86] (S. Jacobsen, J. Wood, K. Bigger, and E. Inversen, 1986)

The utah/mit hand: works in progress. International Journal of Robotics Research, 4(3):221-250, 1986.

[87] (Schlesinger, G., 1919)

Der Mechanische Aufbau der Kunstlichen Glieder, Borchardt, et al. (eds.), Ersatzglieder und Arbeitshilfen fur Kriegsbeschadigte und Unfallverletzte, Springer, 1919, pp. 321-699.

[88] (Slocum and Pratt, 1946)

D. B. Slocum and D. R. Pratt. Disability evaluation for the hand. Journal of Bone and Joint Surgery, 28:491, 1946.

[89] (Speeter, 1991)

Thomas H. Speeter. Primitive based control of the Utah/MIT dexterous hand. In Proc. of the 1991 IEEE International Conference on Robotics and Automation, pages 866-877, Sacramento, CA, April 1991.

[90] (Stanfield, 1991)

S. A. Stansfield. Robotic preshaping of unknown objects: A knowledge-based approach. International Journal of Robotics Research, 10(4):314-326, August 1991.

[91] (Sumeet Bawa, 1995).

Interactive creation of animations: An optimization approach to real-time inverse kinematics. M.Sc. t hesis. Simon Fraser University. April 1995.

[92] (Taylor, C.L., and Schwarz, R.J., 1955)

The Anatomy and Mechanics of the Human Hand," Artificial Limbs, No. 2, 1955, pp. 22-35.

[93] (T. Okada, 1979)

Object handling system for manual industry. IEEE Trans. on SMC, SMC 12:289-299, 1979.

[94] (V. Nguyen, 1986)

The synthesis of force-closure preshaping. Master's thesis, Department of Electrical Engineering and Computer Science, MIT, 1986.

[95] (V. Nguyen, 1988)

Constructing force-closure preshaping. International Journal of Robotics Research, 7(3):2-17, 1988.

[96] (Y. Koga, K. Kondo, J. Kuffner and J. C. Latombe., 1994)

Planning Motions with Intentions. ACM Computer Graphics (Proc. of SIGGRAPH 94), 395.408,

[97] (Y. Li, T. Wang and H. Y. Shum, 2002)

Motion Texture: A Two-Level Statistical Model for Character Motion Synthesis. ACM Computer Graphics (Proc. of SIGGRAPH 2002), 465.472,

[98] (Z.X. Li and S. Sastry, 1988)

Task oriented optimal preshaping by multifingered robot hand. IEEE Trans. on Robotics and Automation, 4(1):32-44, 1988.

[99] (Zhao J.and. Badler N. L, 1989).

Real time inverse kinematics with joint limits and spatial constraints. Technical Report MS-CIS-90-09. Department of Computer and Information Science, Gniversity of Pennsylvania, 1989.

[100] (Z. Li and J. Canny, 1990)

Motion of two rigid bodies with rolling constraint. IEEE Transaction on Robotics & Automation, RA2-06:62-72, 1990.

[101] (Z.X. Li, J.F. Canny, and S.S. Sastry, 1989)

On motion planning for dexterous manipulation. In IEEE International conference on robotics and automation, 1989.

[102] (John Lin, Ying Wu, Thomas S. Huang, 1998)

Modeling the Constraints of Human Hand Motion