2007

Grasp planning for digital humans

Faisal Amer Goussous
University of Iowa

GRASP PLANNING FOR DIGITAL HUMANS

by

Faisal Amer Goussous

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

July 2007

Thesis Supervisor: Professor Karim Abdel-Malek

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

MASTER'S THESIS

This is to certify that the Master's thesis of

Faisal Amer Goussous

has been approved by the Examining Committee
for the thesis requirement for the Master of Science
degree in Electrical and Computer Engineering at the July 2007 graduation.

Thesis Committee: ___________________________________
Karim Abdel-Malek, Thesis Supervisor

____
Jon Kuhl

____
Jasbir Arora

____
Soura Dasgupta

____
Tim Marler

To my Family: May, Amer and Osama.

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Karim Abdel-Malek for providing me with these great opportunities and for his constant encouragement. I would also like to thank Dr. Timothy Marler for his thorough critique of my work in every step along the way. Further thanks go to Dr. Jingzhou Yang for providing me with expert help on hand modeling and the DH method. Special thanks go to Steven C. Beck, Amos Patrick and Anith Mathai for helping me implement my work. Many thanks go to all members of the VSR team at the University of Iowa for their continuous support.

Lastly, I would like to thank my thesis committee: Professor Karim Abdel-Malek, Professor Jon Kuhl, Professor Jasbir Arora, Professor Soura Dasgupta and Dr. Timothy Marler.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the recent advancements in computer processing power, 3D graphics, and animations, virtual humans are becoming more and more popular. New generations of virtual humans are highly realistic with respect to appearance, movement, and feedback. In fact, research concerning digital humans has grown to become a multidisciplinary area that involves computer graphics, biomechanics, anatomy, physiology, artificial intelligence, animation, optimization, and speech recognition, to name a few.

Digital humans have had a great impact on the field of ergonomics. An increasing number of organizations are now using digital humans in the design and testing of complex systems. Instead of classical prototyping approaches used for testing and evaluation of new designs, these organizations are relying more on simulations in virtual environments to get feedback about their designs. For example, in the automotive, aerospace, and heavy equipment industries, building a prototype of a product, testing it on human subjects, and then reformulating and adjusting the original design is costly and time consuming. It can have a devastating impact on quality, safety, and time to market. A better approach is to incorporate the human factor into the design cycle from the beginning of the design process. This can help the designer answer questions about the product that involve positioning and comfort, reaching, grasping, visibility, fatigue, and strength assessment. This can lead to a faster time to market, reduced development costs, and most importantly, improved designs that are exactly tailored to the user’s need.

Since the main goal of using digital humans is to study the interaction between the human and the product in a virtual world, digital humans must have sufficient capabilities to allow them to manipulate and grasp virtual objects. Grasping has proved to be a very challenging problem in the field of robotics and human simulation. This stems from the anatomical complexity of the human hands, as well as the cognitive abilities required to grasp objects depending on the task and the purpose of using the object. Grasping involves correctly positioning the arm and wrist with respect to the object while avoiding obstacles present in the environment. It also entails choosing a realistic and human-like hand shape that allows the virtual human to firmly hold the grasped object in place.

For this purpose, we present in this work a new interactive grasping system that helps realize such capabilities for a digital human called Santos™ that is being developed at the Virtual Soldier Research Laboratory at The University of Iowa. Our system is modular in the sense that it is composed of different subsystems that can each be used independently to solve a part of the grasping problem, but when combined, provide an invaluable tool for simulating and evaluating human grasping in the virtual world.

Problem Definition

In the most general sense, the grasping problem in the virtual world can be defined as follows:

Given an object and a model of a human, find a suitable grasp for that object.

It is clear from the above definition that there are many terms that should be defined more precisely. What information is known about the object? How accurate is the model of the human? Is it just a kinematic model, or is it physics based? How is a grasp defined? What is a "suitable" grasp? There are many different ways we can answer these questions, and each answer creates a different instance of the grasping problem. For this reason, we have made several assumptions that would help us define our instance of the grasping problem that we are trying to solve. These are stated and justified below:

1. **Only information about the shape of the object is provided.** This means that no physical or dynamic properties are included, such as temperature, weight, or texture. This is a reasonable assumption for grasping simulations using computer graphics since the object models come with complete information about the shape in the form of a polygonal mesh. If we wish to complicate the problem and

manually define physical properties for each object we import into the virtual environment, we might as well define how each object is to be grasped and save this information with the properties of the object and be done with the grasping problem.

2. **The hand model is a 25-degrees-of-freedom kinematic model with an outer skin layer.** This model was previously developed at the Virtual Soldier Research program and is one of the most realistic and anatomically correct hand models in the field. The hand model is discussed in more detail in Chapter 2 of this thesis.
3. **The grasp should be realistic and natural**, meaning that it resembles grasps used by humans. This assumption stems from the fact that we are doing human simulations and it is only reasonable to reject postures or motions that do not resemble those that a normal human would perform.
4. **No information is provided about the task.** Our system is capable of producing more than one way of grasping the object, and it is up to the user to select the desired grasp. Again, this avoids burdening the user with defining task requirements for every object model to be used and allows the system to work with completely unknown objects for which only shape information is provided.
5. **A suitable grasp is one that is mechanically stable and realizable given the kinematic constraints of the hand and the limbs.** We propose a single quality measure for rating grasps and demonstrate that other quality metrics can be easily developed to evaluate a grasp.

Considering the above assumptions, we define the grasping problem as follows:

Given:

- A 3-D CAD model of the object (polygonal mesh) to be grasped.
- The kinematic model of the hand and body of the virtual human with information about joint limits and link lengths

Find:

- Finger joint angles
- Hand position and orientation
- Upper-body posture
- The quality of the grasp

Literature Review

Although the concept of a digital human is relatively new, the problem of grasp synthesis is not. This is because researchers in the field of robotics are interested in creating robots that are able to manipulate objects in the physical world. Many of the techniques used to solve the robotic grasping problem can be readily transferred to the virtual world with only slight modifications. Other algorithms are specifically tailored to virtual humans. In the following subsections, we classify previous work in the literature according to the method used to achieve a grasp.

Rule-Based Methods

Rule-based techniques try to classify the part of the object to be grasped as one of several previously stored shape primitives, such as a sphere, pyramid, cube, or cylinder. The system contains rules for grasping each of these primitives. For example, Mas and Thalmann (1994) develop a method whereby the user selects the type of primitive that best describes the object to be grasped, and then depending on the geometrical attributes of the primitive, the system chooses one of the general hand shapes suggested by Cutkosky and Howe (1990). Similarly, Tomovic et al. (1987) postulate that a vision system is needed to describe the object as a composition of geometrical primitives according to the intended task to be performed using the object. Miller et al. (2003) conducted experiments on automatic grasping using their grasp simulator, but they assumed that an approximation of the object as a set of geometric primitives already exists, and they defined a corresponding approach vector and hand shape for each primitive. Similar work was presented by Rijpkema and Girard (1991), while Carenzi et al. (2005) incorporated learning into their rule-based system through the use of neutral networks.

The major shortcoming of all these approaches is that they are not suitable for automatic grasping of novel objects because the decision about which primitive to use is either left to the user or embedded in the object model during the design stage. One can argue that any of the systems described by Carenzi et al. (2005), Lien and Amato (2006), Zhang et al. (2003), and Pirrone and Chella (2003) can perform the task of segmenting the object automatically. However, the operation of automatically segmenting the object is by itself a complex problem, and when coupled with the grasping problem, one cannot expect real-time performance of the system.

Even if we assume that the segmentation problem has been solved, there are still many objects that cannot be intuitively classified as one of the few geometrical primitives. This will produce inaccurate or unnatural-looking grasps.

Grasp Learning Methods

Researchers have attempted to produce intelligent grasping systems that can learn from previous successful grasps and adapt them to novel objects. For example, Moussa (2004) uses a mixture of expert systems to learn primitive grasping behaviors for a library of 28 arbitrary objects. However, for input, the system requires the size, position, orientation, and class for each object. Similarly, the neural-network-based module of Gorce and Rezzoug (2005) requires information about the fingertip location of each finger and the bounds of a 6D palm search space to learn the fingers and wrist configurations. With this amount of input information, the problem has been solved in real time using optimization (Borst et al., 2002). Other examples of the use of neural networks for grasping include, but are not limited to, those presented by Taha and Wright (1997) and Jagannathan and Galan (2004). Many of these approaches attempt to learn and control low-level grasping behaviors, such as finger joint angles or wrist orientation. These actions can be more efficiently calculated using inverse kinematics. Another drawback is that most of this work has been theoretical and applied only to specific types of objects with well-known geometry. We do not know of any neural-network-based system that has been successfully employed to learn higher-level, task-related grasping behaviors and decisions.

Interesting work on supervised grasp learning for grasping is described by Pelossof et al. (2004). The authors use support vector machines (SVM) to associate a successful grasp with a given object. The objects are modeled as superquadrics (a family of three-dimensional shapes), which are not general enough to describe any arbitrary object. In addition, this work focuses on robotic grippers and is not easily adapted to humanoid grasps.

The greatest difficulty with applying learning methods to grasping is describing the shape of the object to be grasped so we can learn the mapping between different shape parameters and grasp parameters. The parameters describing the shape must be general enough to describe a broad range of objects, but at the same time, we wish to minimize the number of parameters used. There is no shape description technique that possesses the aforementioned qualities.

Optimizing a Quality Function

A large part of the grasping literature is concerned with optimization-based methods for finding grasps. Most of these techniques use the quality metric proposed by Farrari and Canny (1992), or variations of it, as an objective function in an optimization problem. Some of these optimization algorithms assume the availability of a closed-form description of the object surface and then attempt to analytically calculate points on that surface that maximize the grasp quality (Katada et al., 2001; Kim et al., 2004; Liu et al., 2004). Other grasping systems work by first calculating a large number of grasps from random starting positions and configurations or based on simple heuristics, then rank these grasps by their quality and choose the best among them. This was done by Wang et al. (2005), Borst et al. (1999), and Toth (1999) on a robotic arm and by Miller et al. (2003) using a simulation engine. Hester et al. (1999) approximate the surface of the object as a grid of points, conduct an exhaustive search for grid points that are suitable for grasping, and then analyze these point combinations based on a quality metric.

Some researchers have focused their efforts on using genetic algorithms for grasp synthesis. Globisch (2005) has experimented with different quality functions to achieve automatic grasping for multi-fingered hands in a simulation environment. Fernandez and Walker (1998) apply the same concept to robotic grippers. However, these algorithms assumed that the object was already placed at the center of the hand workspace, and they ran too slow to be suitable for real-time applications. Also, an objective function that was efficient for a certain class of objects (e.g., cylindrical objects) was not necessarily effective for other classes.

In summary, the methods that rely on optimizing a grasp quality function are not suitable for virtual-human grasping for the following reasons:

1. These approaches approximate the grasp as a collection of contact points on the object. This is only applicable to precision grasps, where only the fingertips touch the object, and cannot be applied to power grasps, which involve the palm and inner surfaces of the fingers. In such cases, there is a large number of contact points, and the number of these contacts is not known before the grasp is executed.
2. Most of these approaches separate the problem of finding optimal contact points on the object surface from the problem of finding a hand posture that can touch these points. This results in unreachable grip points due to the constraints imposed by the gripper kinematics.

3. Most of the quality measures in the literature are computationally expensive. A system that attempts to evaluate a quality measure for a large number of grasps will not be useful for real-time applications and thus will not be useful with digital humans.
4. Objects in a virtual environment are modeled as a “polygon soup,” for which no closed-form expression is available. This prevents the use of any gradient-based optimization techniques to maximize the quality of the grasp.

Quality metrics should be used to provide feedback to a user who is experimenting with a few different ways of grasping the same object, or for pruning low-quality grasps after a small number of candidate grasps have been generated.

Data-Driven Grasping Methods

Data-driven grasping techniques exploit the idea that data obtained offline about grasps can be used to synthesize similar grasps online. For example, ElKoura and Singh (2003) use a database of human grasps to enhance results obtained from an inverse kinematics algorithm. Ehrenmann et al. (2001) utilize a dataglove to record grasping actions that are later used to teach a robot manipulation tasks in similar environments. Li and Pollard (2005) use a database of grasps obtained through motion capture data and try to adapt these grasps to novel objects through the use of shape-matching algorithms. Similarly, Miyata et al. (2006) rely on motion capture data to select starting hand poses for grasp posture generation. Aleotti and Caselli (2006) use virtual reality to program their system to grasp by demonstration. These systems generally do not yield unnatural grasps because the database itself will contain precise grasps that were carefully generated from actual human postures. However, the challenge is to adapt the grasps in the database to new objects and situations in a virtual environment.

Calculating Hand Postures Given Contact Points

Methods that involve using contact points to calculate hand postures are most applicable to what typically constitutes the final stage in the grasping process. It is assumed that contact points on the surface of an object have either been generated by an automatic grasp planner using one of the methods mentioned above, or selected interactively by the user. The usual approach in solving this problem is inverse kinematics (Sanso and Thalmann, 1994; Kallman et al., 2003). Most of these algorithms focus on how to solve the redundancy in the inverse kinematics problem. One interesting approach was to optimize an objective function. Borst et al. (2002) provide a precise formulation for solving this problem by casting it as an optimization problem, with the objective function being a sum of penalty terms that guarantee that the optimal solution will obey the constraints set forth by the joint limits and target positions. Similar work was done on a virtual human (Yang et al., 2006), but the authors used deviation from the joint neutral position as an objective function to be minimized. These methods are generally fast and suitable for online grasping, especially if the problem is formulated as a continuous optimization problem with known analytical gradients. However, there is no way to check that the target points are reachable or physically realizable by the hand model before attempting to solve the optimization problem.

Motivation and Proposed Solution

Although the literature presents a wide spectrum of clever grasping techniques, there are several issues that prevent us from adapting any one of these techniques for our application:

1. None of the methods mentioned above consider the effect of the upper body on the feasibility of the grasp. They consider the grasp as finished when the positions and orientations of the fingers and wrist have been calculated.

2. Most of the research in the literature is concerned with robotic grasping, which does not necessarily apply to humans because the human hand is unique in its anatomy and complexity.
3. Very few previous methods were able to address the problem of grasping novel objects that are arbitrarily shaped.

We propose a method that can handle new objects in a simulation environment, requires minimal user input, and takes into consideration the effect of the upper body on the grasping posture. This constitutes a system for planning and evaluating grasps for a digital human, which can be invaluable in the assessment of products and engineering designs during the prototyping stage.

Keeping in mind the variety of applications in which a virtual human with grasping capabilities can be used, we provide the user with tools that accommodate different levels of interactivity, ranging from semi-automatic tools that rely on minimal user input to more user-dependent tools that achieve more customization and precision. Specifically, we aim to develop and test the following array of grasping modules:

1. A posture-prediction module for calculating angles for the joints of the fingers, which dictate the optimum hand posture that allows the digital human to touch a set of point targets assigned to the fingertips of each finger. The posture-prediction procedure involves minimizing an objective function and satisfying a set of constraints, and it results in human-like grasps.
2. A shape matching and alignment module that uses a database of prerecorded grasps to produce grasps for new objects. This module relies on minimal input from the user and is specialized for grasps in which the palm and the inner surfaces of the fingers come into contact with the object.
3. An evaluation module that can provide feedback for the user about the quality of the produced grasp. This allows the user to compare different grasps and to reject unstable grasps in which a slight external disturbance might cause the

object to slip away from the hand. This module analyzes the space of forces and torques that the given grasp can exert on the object and produces a numeric quality metric.

Overview of Thesis

Chapter 2 provides an overview of the digital human SantosTM. It focuses on the hand and describes its kinematic model and its relationship to the actual human hand. It will also introduce the basic grasping capabilities that are already implemented in SantosTM. Chapter 3 provides an overview of our grasping system and briefly describes its constituent parts. It also outlines how the different segments tie in together to form the complete system. It presents several scenarios of how the system might be used for planning and evaluating grasps. Chapter 4 describes the first part of the system, which uses a data-driven technique. It describes the process of building a database of grasps, developing a function to discriminate between shapes, and finding the best orientation of the hand for a given object and grasp type. It also describes a new method for evaluating grasps using the upper-body posture required to achieve the grasp. Chapter 5 presents a new formulation for optimization-based grasping. It discusses the design variables, the constraints, and the objective function, and presents the results obtained by solving this problem and simulating the solution in the virtual world. Chapter 6 discusses the importance of a quality metric for grasping and presents the details of the formulation and implementation of one quality metric. Finally, Chapter 7 summarizes our contributions and presents potential future work.

CHAPTER 2
THE SANTOS^{TM} HAND

In this chapter, we present the details of the hand model for Santos^{TM}, the virtual human developed at the Virtual Soldier Research program at The University of Iowa. We begin by providing an overview of the model of Santos^{TM} as a whole, explaining which measures were taken to ensure that it is anatomically correct and highly realistic in both functionality and appearance. We then focus on the current hand model, explaining its kinematics and the number of degrees of freedom (DOF) it uses. Next, we give an overview of the simulation environment in which Santos^{TM} resides and show a typical scenario of a user interacting with Santos^{TM} in that environment. Finally, we present the current hand capabilities that are implemented in Santos^{TM}, which allow for basic manipulation and grasping actions.

Overview of Santos^{TM} the Digital Human

The 3D model of Santos^{TM} can be thought of as a “skin” laid over a skeleton. The skin is obtained from a scan of an actual human being, and it defines the overall avatar’s shape (Figure 2.1). The skin is made deformable using a well-known animation technique called *skin weighting*. This prevents breaking or tearing of the skin when the body parts are moved.

Underneath the skin, there is a skeleton that allows the avatar to be controlled and moved around. This skeleton is modeled as a kinematic system, a series of links with single-DOF joints connecting each pair of links. Each joint in the skeletal model is represented by one, two, or three revolute joints. Using the Denavit-Hartenberg method (Denavit and Hartenberg, 1955), the position of different points on the body of the avatar can be related to the joint angles using a series of transformation matrices. Details of this approach are discussed by Marler (2004). When the outer skin of the avatar is attached to the skeleton, the result is a complete virtual human (Figure 2.2).

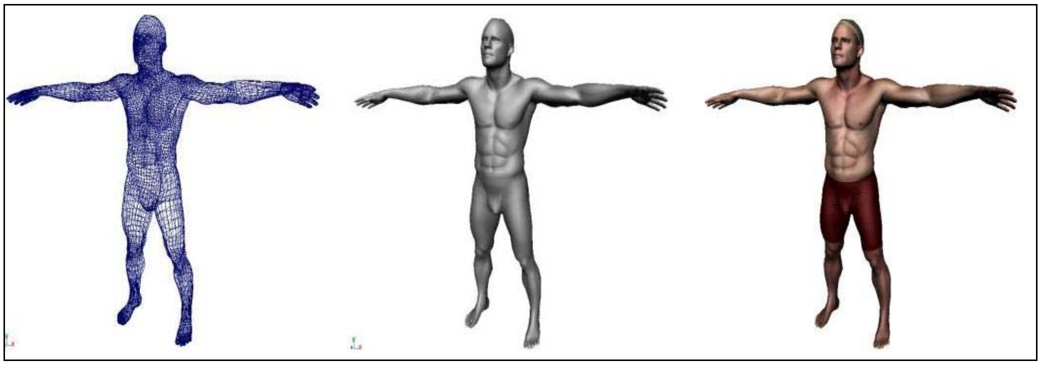

Figure 2.1: Scanning and skinning of an avatar.

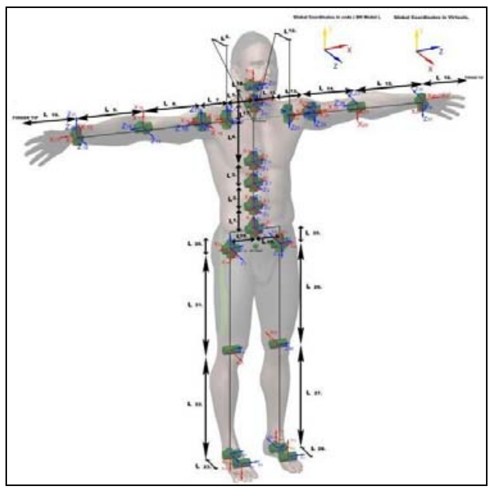

Figure 2.2: An outer mesh and a skeleton constitute a complete virtual human model.

An optimization-based approach to posture and motion prediction allows Santos^{TM} to operate autonomously without relying on stored animations and data or being restricted by inverse kinematics. The method incorporates a mixture of human performance measures that produces motions and postures that greatly mimic those of actual humans. For a more detailed discussion of this approach, refer to work by Marler (2004), Marler et al. (2005-a), Marler et al. (2005-b), and Yang et al.(2004).

Work is in progress to add a musculoskeletal model based on actual human anatomy (Figure 2.3). It simulates realistic muscle wrapping and can be used for predicting the muscle forces needed to create joint torques and muscle stress in Santos™. See the paper by Abdel-Malek et al. (2006) for details about muscle modeling, physiological modeling, reach envelopes for the limbs, and workspace zone differentiation.

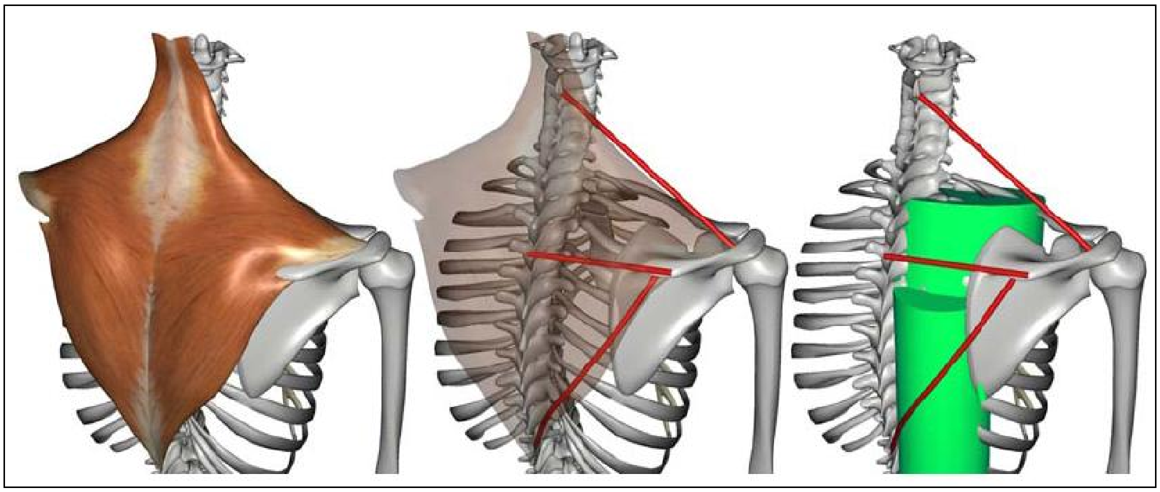

Figure 2.3: Musculoskeletal modeling for Santos™

The Hand Model

The Santos™ hand model has 25 degrees of freedom in each hand. It is based on anatomical data from the human hand. Figure 2.4 shows the skeleton of a human hand and the kinematic model for the Santos™ hand.

Due to the presence of joints in the palm (the carpometacarpal joints for the pinky and ring fingers), this hand model is considered more advanced than other models that consider the whole palm as a rigid body because it allows for arching of the palm. Due to the important role that the shape of the palm plays in grasping, this model is considered more suitable for grasping simulation.

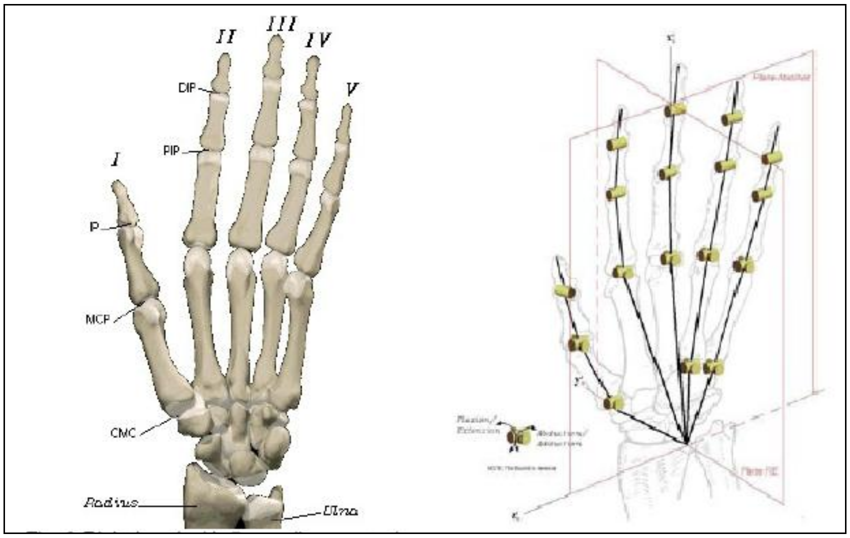

Figure 2.4: A human hand model (left) and the Santos^{TM} hand model (right).

Data obtained from Tubiana (1981) and Tubiana et al. (1996) specifies the limits for the joints of the human hand. This was implemented in our model. Table 2.1 below summarizes these limits in degrees.

The hand link lengths are a function of the size of the hand. However, there are equations that describe the relationships between the hand breadth and length and the length of the metacarpal bones. The hand model is built according to these relations (Petarch et al., 2005). In cases where the user needs to model humans of different sizes and percentiles, the anthropometric hand interface shown in Figure 2.6 insures that all the proportions between the link lengths are conserved and allows the user to input specific anthropometric data concerning joint limits and individual link lengths.

Forward and Inverse Kinematics for the Hand

In order to study the forward and inverse kinematics for the hand, the D-H method from the robotics field (Denavit and Hartenberg, 1955) was adapted to define the

Table 2. 1: Joint limits in degrees for the Santos^{TM} hand.

Finger	Joint	Minimum	Maximum
Thumb	Q1	0	60
	Q2	-25	35
	Q3	0	60
	Q4	-10	55
	Q5	-15	80
Index	Q6	-13	42
	Q7	0	80
	Q8	0	100
	Q9	-10	90
Middle	Q10	-8	35
	Q11	0	80
	Q12	0	100
	Q13	-10	90
Ring	Q14	0	10
	Q15	0	10
	Q16	-14	20
	Q17	0	80
	Q18	0	100
	Q19	-20	90
Pinky	Q20	0	20
	Q21	0	20
	Q22	-19	33
	Q23	0	80
	Q24	0	100
	Q25	-30	90

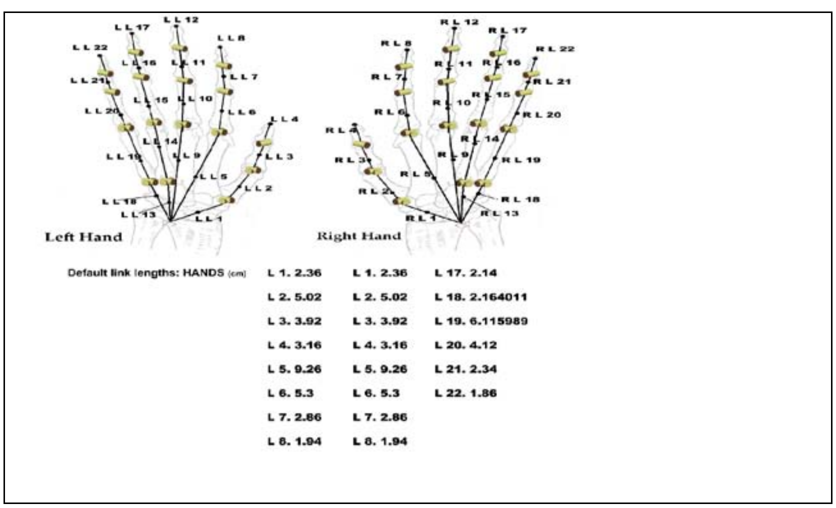

Figure 2.5: The default link lengths for the Santos^{TM} hand.

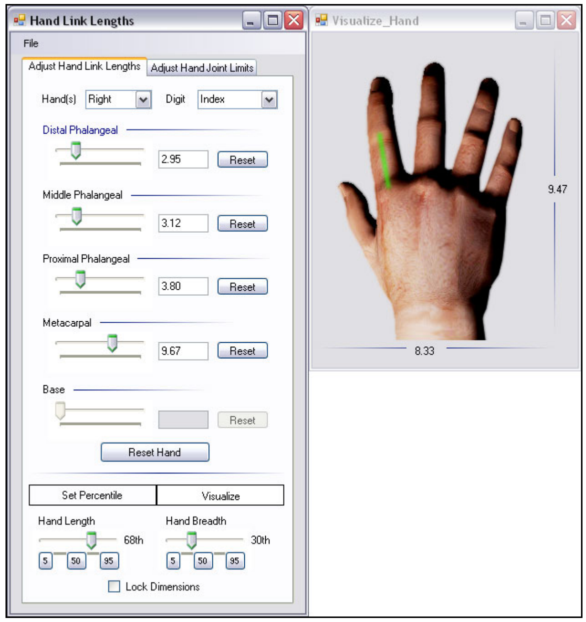

Figure 2.6: Anthropometric hand interface for Santos™.

positions of the fingertips with respect to each local coordinate system. The D-H method provides an efficient and systematic way of representing the transformations between joints in terms of only four parameters, which are:

- the angle θ_i between the $(i\text{-}1)_{th}$ and i_{th} x-axis about the $(i\text{-}1)_{th}$ z-axis
- the distance d_i from the $(i\text{-}1)_{th}$ to the i_{th} x-axis along the $(i\text{-}1)_{th}$ z-axis
- the angle α_i between the $(i\text{-}1)_{th}$ and i_{th} z-axis about the i_{th} x-axis
- the distance a_i from the $(i\text{-}1)_{th}$ to the i_{th} x-axis along the i_{th} x-axis

After we define these parameters for the $(i\text{-}1)_{th}$ and i_{th} frames (Figure 2.7), we can calculate a transformation matrix between these two frames according to the following equation:

$$
{}^{i-1}T_i = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.1}
$$

Now we can define the position vector expressed in the i_{th} coordinate frame $X_i(q)$ with respect to the j_{th} coordinate frame using:

$$\begin{bmatrix} X_j(q) \\ 1 \end{bmatrix} = \prod_{k=j}^{k=i} {}^{k-1}T_k \begin{bmatrix} X_i(q) \\ 1 \end{bmatrix} \tag{2.2}$$

Now for the forward kinematics problem, if the vector of joint angles $\mathbf{q}$ = $[q_1,q_2,\ldots,q_{25}]$ is known for the right hand, then we can calculate the position of the fingertips by directly using Equations (2.1) and (2.2). The left hand is treated similarly for the joint angles $[q_{26},q_{27},\ldots,q_{50}]$.

The inverse kinematics problem is not as straightforward. Here, the positions of end effectors in a global coordinate system are specified, and the problem is to find joint angles that, when applied to the individual joints, allow the end effectors reach their positions. The difficulty of this problem lies in the fact that there is usually more than one solution $\mathbf{q}$ for a given set of end effector positions. In older versions of Santos™, this difficulty was overcome by imposing constraints on the range of angles of the joints in order to minimize the feasible space of the solution. In Chapter 5 of this thesis, we will discuss the details of an optimization-based method for inverse kinematics and show its implementation for the Santos™ hand.

Grasp Selection and Interactive Joint Manipulation

Cutkosky (1989) classified hand shapes that workers use in manufacturing tasks into precision grasps that emphasize sensitivity and dexterity, and power grasps that emphasize stability and security. He also specified 16 hand shapes that are most commonly used by humans in practice.

This grasp taxonomy has become a standard in applications dealing with 5-fingered hands. This grasp taxonomy was adopted into Santos™ by carefully calculating the joint angles for each grasp using inverse kinematics.

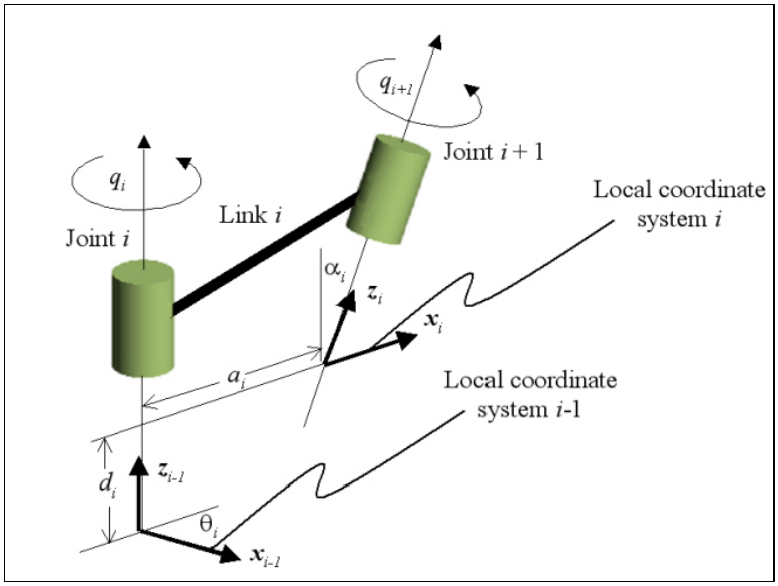

Figure 2.7: Parameters for the D-H method.

A library of grasps was created that allows the user to interactively choose a hand shape that best fits his or her needs (Figure 2.8).

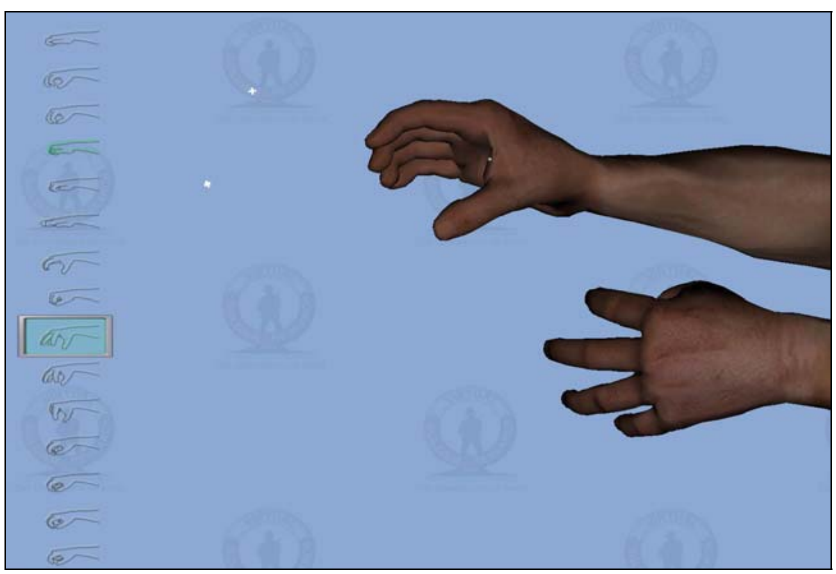

Figure 2.8: Interactive grasp selection for Santos™.

Another useful capability that was implemented in Santos^{TM} is the ability to interactively manipulate any joint in the virtual human's body. The user can click on a desired joint and then choose which specific degree of freedom he wants to manipulate through an intuitive user-friendly interface (Figure 2.9).

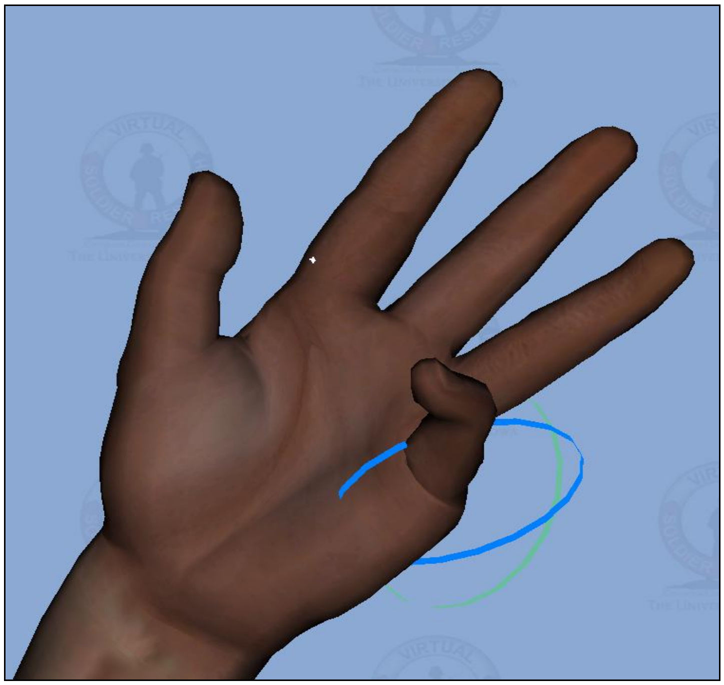

Figure 2.9: Interactive manipulation of joints in the pinky finger.

CHAPTER 3

GENERAL GRASPING STRATEGY FOR SANTOS^{TM}

Introduction

The main goal of this research is to develop a framework that can be used for simulating grasping and manipulation actions for a digital human in the virtual world. Since this could be an extremely complex problem depending on the level of autonomy of the virtual human (the most autonomous grasping functions are the most complex to model), we have broken down the problem of reaching and grasping into several simpler sub-problems. For each of these smaller problems, we have developed a module that is focused on solving that problem.

Although each of these modules is useful on its own, the true power of the system is when these modules are combined. Our grasping system can cater to varying levels of user input. It produces grasping postures that include the joints in the hand and wrist and also extend to the whole upper body. It is capable of producing both power grasps suitable for heavy objects and precision grasps required to finely manipulate objects with the fingertips. Once a grasp is produced, the system is also capable of providing feedback on the mechanical stability of the grasp.

In this chapter, we present a typical scenario in which a user would employ all of our developed modules to interactively grasp an object, refine the grasp, and finally obtain feedback on the quality of the grasp. We will begin by presenting a typical example of planning grasping and reaching actions for a virtual object using our system. We will then proceed by summarizing the functionality provided by each component of the system. Full details about each component can be found in the remaining chapters of this thesis. Finally, we discuss the effect of the level of user intervention on grasp planning and present alternative scenarios that can be used when more user input is provided.

Proposed Grasping Framework

Figure 3.1 below shows our proposed framework for virtual human grasping. The whole process is summarized visually in the flowchart in Figure 3.3.

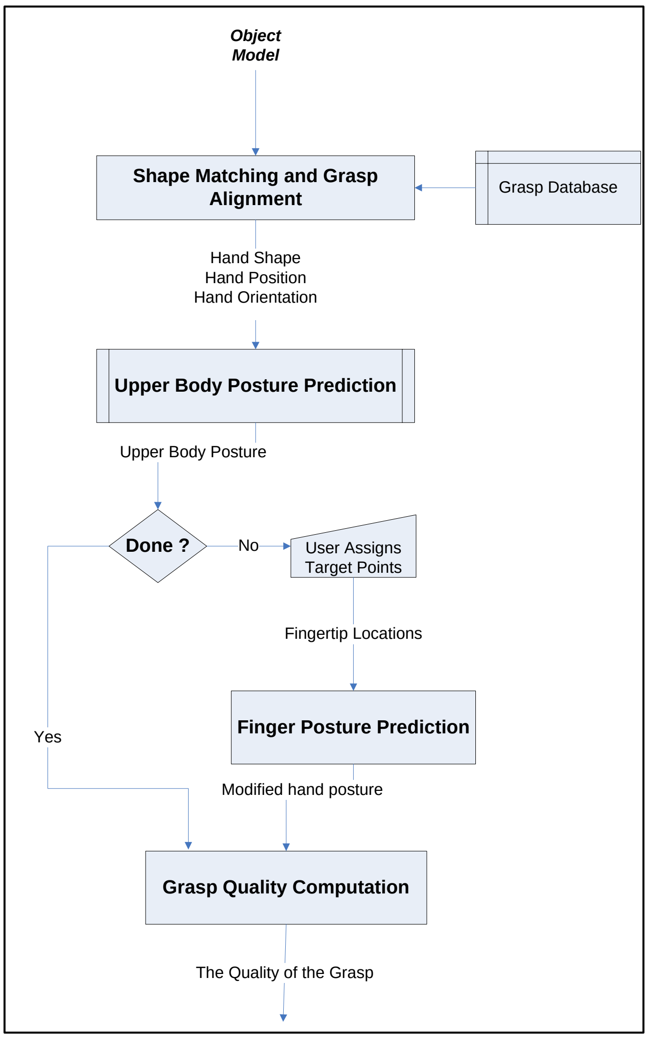

Figure 3.1: Our proposed grasping method.

First, information about the object shape is read from the object model in the virtual world. This information describes the polygonal composition of the object mesh. The first step is to analyze the shape of the object and pick a corresponding hand shape from the database. The database contains hand shapes that are frequently used in grasping. Given the hand model and the hand shape, the grasp alignment module calculates several possible hand alignments. Each of these alignments are input to an upper-body posture prediction module that calculates the proper joint angles for the upper body that result in the hand being in the given position and orientation subject to the joint limit constraints. The user picks one of the alignments presented to him. At this point, the user either proceeds to evaluate the stability of the grasp or requests further refinement. Assuming that the user wishes to refine the grasp, he is allowed to pick target points that correspond to the desired positions of the fingertips. This invokes a finger posture prediction module that calculates a hand posture with the fingertips touching the target points. Finally, the grasp quality module computes the quality of the resulting grasp.

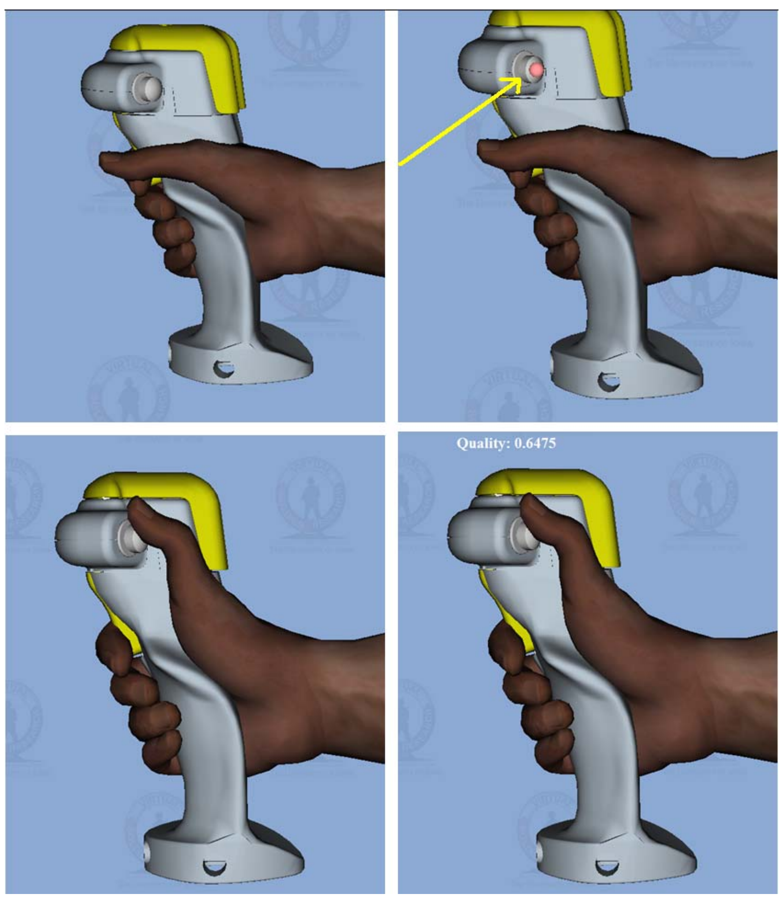

Figure 3.2: Grasp alignment (upper left), selection of target points (upper right), finger posture prediction (lower left), and grasp quality (lower right).

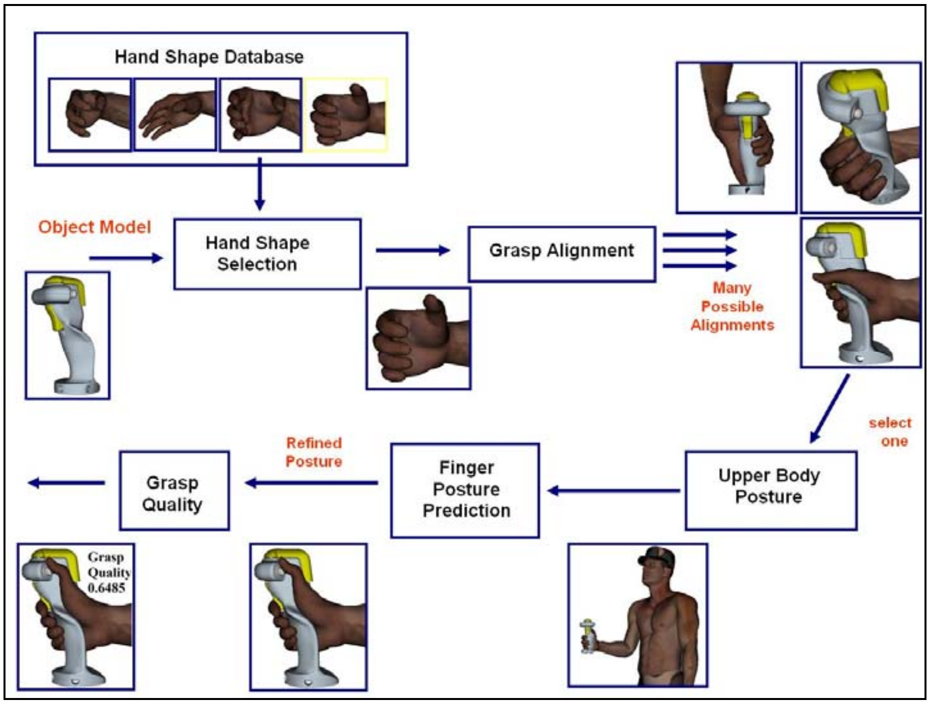

Figure 3.3: Overall system operation.

Shape Matching and Grasp Alignment

In this stage, the object is first sampled, and a mathematical description of its shape is extracted from the samples. This description is called a "shape function." We have a database of hand shapes that is populated offline with the shape functions and joint angles corresponding to each hand shape. The shape function of the object is compared to each and every hand shape function in the database, and the one that matches best is selected. Next, all the possible hand orientations are calculated and tested to see whether they result in valid grasps. All the valid grasps are clustered into n groups (n is a number set by the user), and the mean of each group is presented to the user. This module is explained in more detail in Chapter 4 of this thesis.

Upper-Body Posture Prediction

For each of the n grasps resulting from the shape matching and grasp alignment stage, an upper-body posture is calculated. This is done using an existing upper-body posture prediction module; see for example the one presented by Farrel (2005). This module takes as input the position of the end effector, which in this case is the wrist position, and the end effector orientation, which is the wrist orientation given as two orientation vectors defining the up and right directions of the palm. Using an optimization-based inverse kinematics algorithm, this module calculates the required joint angles of the upper body (excluding the fingers) that result in the given wrist position and orientation, subject to joint limit constraints.

Finger Posture Prediction

Assuming that the user is not completely satisfied with the resulting grasp and wishes to have more control over the position of the fingers, he is allowed to do so using a finger-posture prediction module. For example, while grasping a joystick, the user might wish to press a button. For this purpose, the user is asked to click on points on the object surface where he wants to position each finger tip. This invokes the finger posture prediction module, which uses the same ideas as the upper-body posture prediction but is specialized for the hands. More details are provided in Chapter 5 of this work.

Grasp Quality

After the final grasping posture is realized, the user is provided with feedback on the mechanical quality of the grasp via the grasp quality module. This module analyzes the number of contact points between the hand and the object, the normals at each of these contacts, and their distances from the center of mass of the object. Based on this information, the wrench space of the grasp is computed and used to extract the quality measure. This measure describes how well the grasp restrains the object and counteracts external disturbances. This topic is treated in depth in Chapter 6 of this thesis.

Alternative Scenarios

The scenario described above is just one of many possible scenarios in which the individual modules can be used to simulate grasping. Depending on how much user intervention is desired, there could be many possibilities. The scenario presented in the previous section is the most automatic that we can currently come up with: all the user has to do is to click on the object to be grasped and later assign the target points for the fingertips. Table 3.1 below shows other possible scenarios sorted according to increasing levels of user intervention.

Table 3.1: Grasping scenarios for different levels of user input.

	Level 1 (Automatic)	Level 2 (Semi-Automatic)	Level 3 (Manual)
Finger Joint Angles	The best matching hand shape is automatically chosen from the database	The user chooses a desired hand shape from the database	The user manipulates each joint to achieve the desired hand shape
Wrist Position and Orientation	The system chooses one of the possible alignments suggested by the grasp alignment module	The system presents n possible alignments, and the user chooses one of them	The user manually manipulates the joints of the wrist, elbow, and shoulder
Fingertip Location Refinement	The system chooses targets for the fingertips and invokes finger posture prediction	The user picks the target points and invokes finger posture prediction.	The user refines the grasp through manual manipulation of the joints of each finger
Grasp Quality	Quality is calculated at the end of the refinement	N/A	The user signals the start of quality computation

Notice that there are distinct advantages and disadvantages for each level of user intervention. While Level 3 in Table 3.1 is the most tedious for the user because he is expected to guide the grasping actions at every step, it provides the highest level of control and accuracy. Grasps produced according to the Level 3 scenario are the most natural and are hence used to populate the database in Figure 3.1. On the other hand, Level 1 represents autonomous virtual human behavior. This uses minimal user input but is not guaranteed to result in the same grasp that the user intended. For example, a grasp for a mug might result in the mug being held upside down, causing the contents to be spilled. This is due to the fact that grasps are task dependent, and we have not developed a cognitive model to make decisions about which grasps are suitable for which tasks.

CHAPTER 4

SHAPE MATCHING AND HAND ALIGNMENT

Introduction

Among the basic requirements in a virtual-human grasp planner is the ability to produce natural-looking grasps with minimal user intervention. If these requirements are satisfied, such a grasp planner would be a useful tool for interaction and product evaluation in the virtual world. This chapter presents a method that satisfies these requirements by planning a grasp based on the shape of the object to be grasped. This method is based on the observation that the shape of the hand during power grasps (grasps that involve the palm and inner surfaces of the fingers) follows the shape of the object quite closely. Our method builds on the work by Li and Pollard (2005), who reduced the grasping problem into a shape-matching problem. We couple our grasping-prediction process with whole-body posture prediction and incorporate it into the Santos™ simulation environment.

The main idea behind this approach is to use a database of hand shapes that are frequently used in grasping. Then, when a new object is to be grasped, its shape is analyzed and compared to the grasps in the database. The hand shape that matches best is chosen as the grasp. Then, an alignment phase determines the hand position and orientation to guarantee maximum contact with the object (this method is most suitable for power grasps). Finally, the output of the alignment stage, which is the hand position and orientation, is passed to an upper-body posture-prediction module that determines a suitable posture for the upper body, including the torso and the arms.

This chapter begins with a review of the related work in the literature. We then outline the overall module for grasping using shape matching before explaining the details of each component in the system. Then, we present results and finally conclude with a discussion section and future work.

Related Work

Shape matching is a well-studied problem that has numerous applications in computer graphics and image processing. There are many techniques for describing the similarity between shapes in three dimensions, and a review of all of them is beyond the scope of this work. However, of particular relevance to our application is the work of Osada and his colleagues on shape distributions (Osada et al., 2001; Osada et al., 2002). The authors also introduced the notion of shape functions, which we make use of in our work. Ohbuchi et al. (2003) introduced two enhanced shape functions for matching three-dimensional polygonal models.

Li and Pollard (2005) were the first to reduce the grasping problem to a shape-matching problem. They have built a database of grasps generated using motion capture data, and their shape-matching algorithm chooses the most suitable grasp from this database. They have also used an alignment algorithm to calculate a suitable hand position and orientation for a given object and hand pose. This is based on the Random Sample Consensus algorithm (Fischler and Bolles, 1981; Chen et al., 1997). We use a similar algorithm for our alignment stage.

The Shape-Matching and Hand-Alignment Module

Overall system

Figure 4.1 shows the overall system that we have developed to solve the grasping problem for Santos™. The major component is a grasp database that contains preprocessed hand shapes that are frequently used when Santos™ grasps virtual objects. For each entry, the database contains the joint angles of the fingers, the hand samples, and a shape function that describes the shape of the hand. The database is constructed offline, while the rest of the operations in the figure are done online. The idea is to match the shape of the object to be grasped to the most similar hand shape in the database. The first step is to sample the surface of the object. These samples are used to analyze the shape of the object and to extract a “shape function” that mathematically describes this object’s shape. The heart of the system is a shape-matching module that compares the shape of the object against each and every entry in the database. After the best matching hand shape is automatically picked from the database, the grasp alignment stage calculates all the possible hand positions and orientations to insure maximum contact between the hand and the object. Since there are many possible hand orientations, and many of these could be similar, the clustering stage groups similar grasps together and presents the user with a fixed number of grasps (four in our experiments).

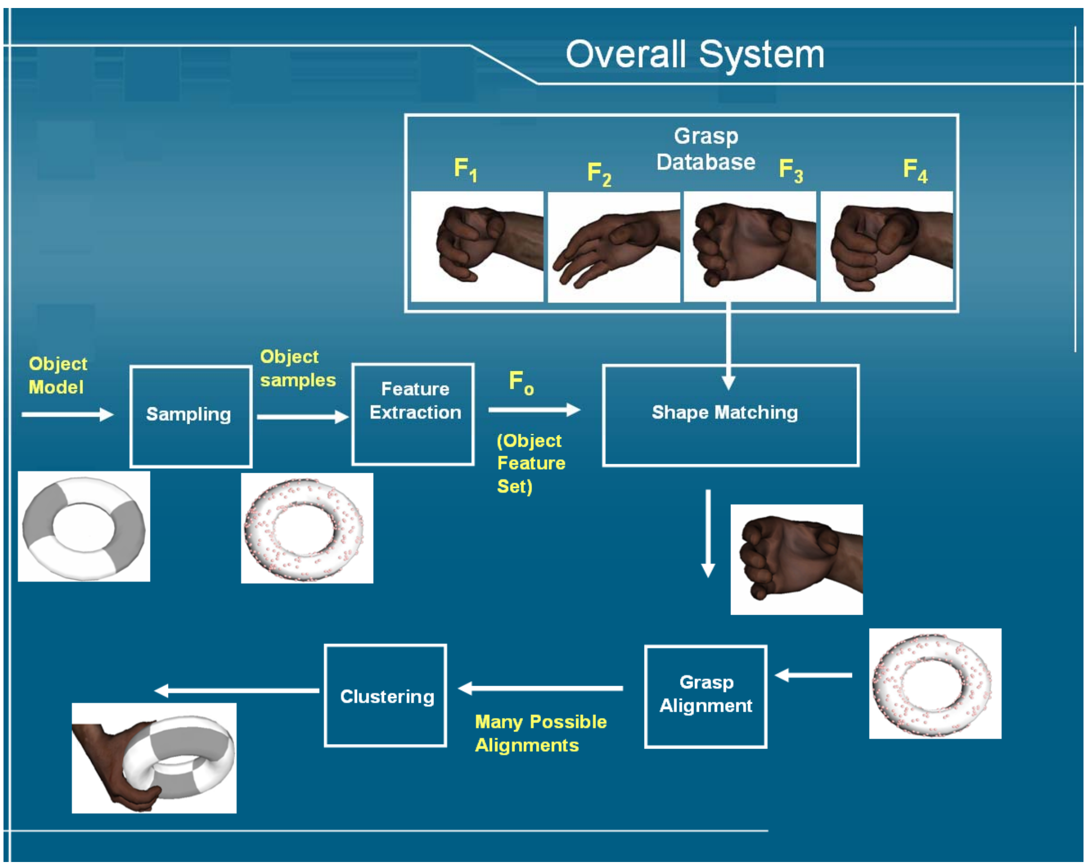

Figure 4.1: Overall system for grasping by shape matching.

After the grasps are grouped, the mean of every group is passed on to an optimization-based posture-prediction module that calculates the whole upper-body posture required to achieve the calculated grasps. We will now explain each of the components in Figure 4.1 in more detail.

Sampling

We define a sample as a Cartesian point in 3D (x,y,z) on the surface of the object. We have used the sampling algorithm presented by Osada et al. (2002), which allows the user to specify the number of samples to be taken. The algorithm chooses a triangle to be sampled at random with probability proportional to its area, so there is a higher chance of sampling a large triangle than a small one. After a triangle is chosen for sampling, the location of the sample on the triangle is randomly chosen. This algorithm is suitable for our application for the following reasons:

a) The samples are independent of the polygonal composition of the model of the object to be grasped. For example, in Figure 4.2, the rectangle on the left is composed of two triangles, yet we were able to get 1024 randomly distributed samples on its surface. The same thing was possible for the model of a rocket with 100 faces.

b) The samples are distributed evenly on the surface of the object, even on flat areas where there are not many faces, or areas that are represented by many faces, such as pointed or sharp edges.

Figure 4.2: Sampling of different objects with 1024 samples: A rocket model composed of 2 faces (left), and a model composed of 100 faces (right).

Feature Set Extraction

Once the samples are obtained, we use them to construct a mathematical description of the shape of the object, or a "shape function," For this purpose, we use the same shape function as that proposed by Li and Pollard (2005). For each pair of samples on the object, we construct one three-dimensional feature value. This is done for all possible pairs of samples on the object. Figure 4.3 shows the feature value that we have used in our experiments.

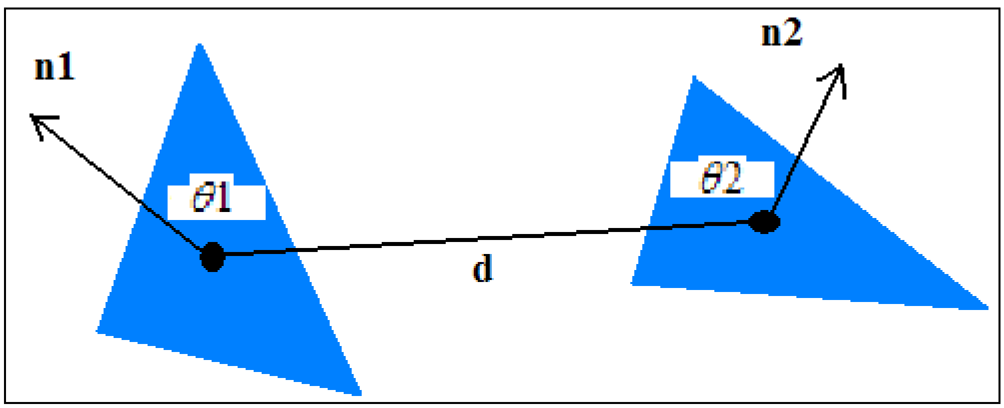

Figure 4.3: The three-dimensional feature set.

For each pair of samples, we calculate the distance between the points, the angles between the normals, and the vector joining the two samples. Hence, a feature is represented as follows:

$$f = [d \ \ \theta 1 \ \ \theta 2] \qquad (4.1)$$

This three-dimensional value is calculated for each pair of samples, and it is stored in a Feature Matrix of dimensions 3 by (N)(N-1)/2, where N is the total number of samples. N is set by the user in an options file once and does not change during execution. We refer to the set of feature values for a given hand shape as the "feature set" of that hand shape.

Database Population

The sampling and feature extraction processes described above are used to create a database of grasps to be used in our data-driven grasping system. The database is populated offline. We manually manipulate the joints of the virtual human in order to create a realistic hand posture that can be used in grasping. Once an appropriate hand shape is reached, the joint angles are saved into the database, and then the hand is sampled (Figure 4.4) and its feature set is extracted. The hand samples and the feature matrix are saved.

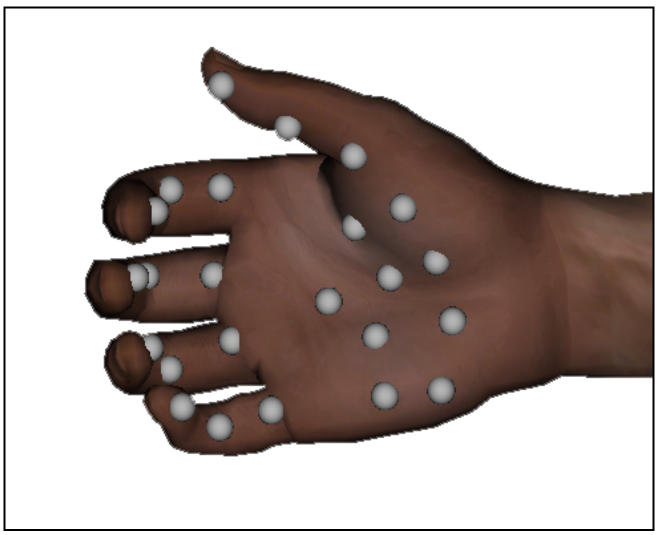

Figure 4.4: The hand samples.

The hand samples are simply Cartesian points on the surface of the hand that were chosen manually. When assigning the samples, we empirically chose the locations in a way that captures the shape of the palm and fingers as precisely as possible. We paid special consideration to the deformable parts of the palm, which play a prominent role in power grasping. We used 24 samples for the hand in our experiments.

Shape Matching

As can be seen in Figure 4.1, the input to the shape-matching stage is the object feature set and the feature sets for all the grasps in the database. The role of this shape-matching stage is to compare the object feature set to all the feature sets in the grasp database, and to rank the grasps in order according to their similarity to the object, starting with the most similar. Assuming that F_{oj} denotes feature j on the object and that F_{ij} denotes the jth feature in the feature set of the ith grasp in the database, then the difference between F_{oj} and F_{ij} can be calculated as:

$$E(F_{oj}, F_{ij}) = \sqrt{(d_{oj} - d_{ij})^2 + (\theta 1_{oj} - \theta 1_{ij})^2 + (\theta 2_{oj} - \theta 2 i_j)^2} \tag{4.2}$$

It is important to note that in order for an object and a hand shape to match perfectly, all features on the hand must be present on the object, but the inverse is not true. For this reason, we compare each feature on the hand with each and every feature on the object and pick the minimum difference value E from Equation (4.2). This means that we compare each feature on the hand with its most similar counterpart on the object surface. This is referred to as a "nearest neighbor" approach in the pattern classification literature.

Since we wish to sort the grasps according to the similarity between the hand shape and the object shape, we need a metric to describe the similarity between a given hand shape and object. From Equation (4.2) and the above discussion, we can sum all the E values between the features on the hand shape and the corresponding nearest neighbors on the object and use this to quantify the dissimilarity between the two:

$$\text{Dissimilarity}(\text{grasp}_i,\text{Object}) = \sum_{j=1}^{n} E(F_{oj}, NN(F_{ij}, Object)) \tag{4.3}$$

Dissimilarity(grasp_i,object) is the amount of dissimilarity between the hand shape in the ith grasp and the object to be grasped.

$NN(F_{ij}, Object)$ is the nearest neighbor of feature j in the ith grasp from among the features of the object.

The calculation in Equation (4.3) is carried out for all grasps in the database. The best matching hand shape is the one with least dissimilarity value. This is used to rank the grasps from the best matching to the least matching.

Grasp Alignment

Once the system has presented the best matching hand shapes, the user either chooses one or allows the system to pick the hand shape with the least dissimilarity measure. This is because the system might not always pick the grasp that the user has in mind because it only compares the shape of the object and the hand and does not take the task into consideration. It is then crucial to calculate the hand position and orientation to achieve the intended grasp. We do this with a Random Sample Consensus algorithm (Fischler and Bolles, 1981; Chen et al., 1997; Li and Pollard, 2005). A triplet of points on the hand are picked and designated as the control frame (Figure 4.5).

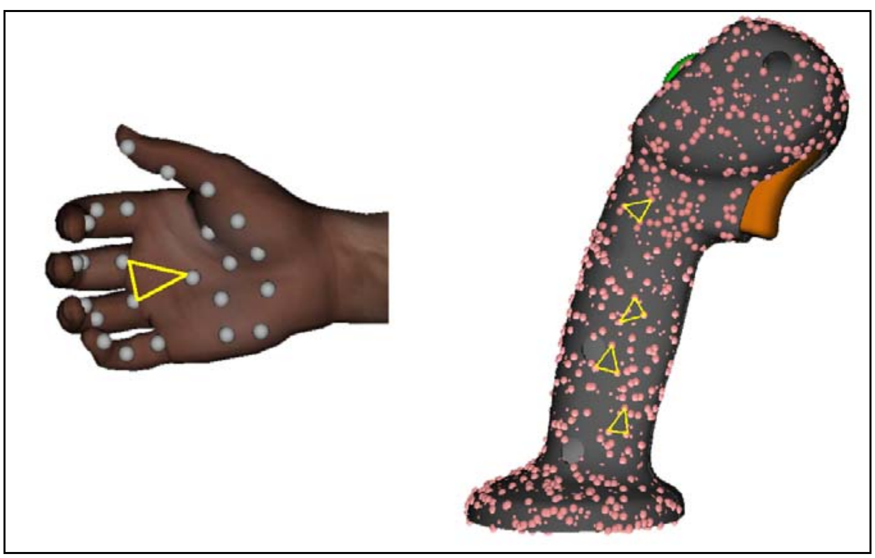

Figure 4.5: The control frame on the hand and partial frames on a joystick, shown in yellow.

We then search the samples on the surface of the objects for triangles that are similar to the control frame. For a triangle to be considered similar, we require that the lengths of the corresponding sides are within a user-specified threshold ε_d. For each triangle that matches the control frame, we calculate a 4x4 transformation matrix that transforms the control frame to the partial frame. The algorithm we use to calculate this transformation matrix is discussed in detail in the appendix. Next, the samples on the hand are transformed by this matrix and tested to see if they are within a user-specified threshold from the object surface. We measure the distance from each transformed sample to its nearest neighbor on the object surface and reject transformations that result in samples violating the distance threshold.

All the transformation matrices that obey the distance threshold are passed to the next stage. Every matrix represents a plausible hand position and orientation.

Clustering

Since the algorithm used in the previous section performs an exhaustive search on the samples of the object to find plausible grasps, we might end up with a large number of grasps alignments, most of which are similar. For this reason, we wish to sort the grasps into groups, where similar grasps are placed in the same group. This is done by employing a clustering algorithm on the transformation matrices resulting from the grasp alignment stage. We do the clustering in the 16-dimensional space of the 4x4 transformation matrices, where we have rearranged the elements of the matrix into a 16-dimensional vector. We have used a C++ implementation of the K-means clustering algorithm (Mount, 2005). In our experiments, we have used a fixed number of clusters (typically four) to be presented to the user.

Connecting to the Upper Body

The final stage of our process integrates the whole body with the grasping task. Specifying a hand position and orientation is not enough when we want the virtual human to perform the grasping task in a natural-looking manner. The avatar must use his or her elbow, shoulder, trunk, and other body parts to reach for the object and grasp it. For this reason, we use the upper-body posture-prediction module that was developed for Santos™ at the Virtual Soldier Research (VSR) group at The University of Iowa (Abdel-Malek et al., 2006). We supply this module with the position of the wrist and the orientation of the hand given as two vectors. These two vectors define two of the local coordinate axes of the wrist. In our case, one vector is perpendicular to the plane of the palm, and the other originates at the wrist and points in the direction of the fingers. More details are provided by Marler (2004). These vectors are extracted directly from the transformation matrices output by the clustering stage. In response, the posture-prediction module calculates the joint angles for the rest of the body by solving an underdetermined inverse kinematics problem using optimization. The result is an upper-body posture that insures that the virtual human's hand is in the desired position and orientation to perform the grasp. Figure 4.6 below shows the effect of using the posture-prediction module.

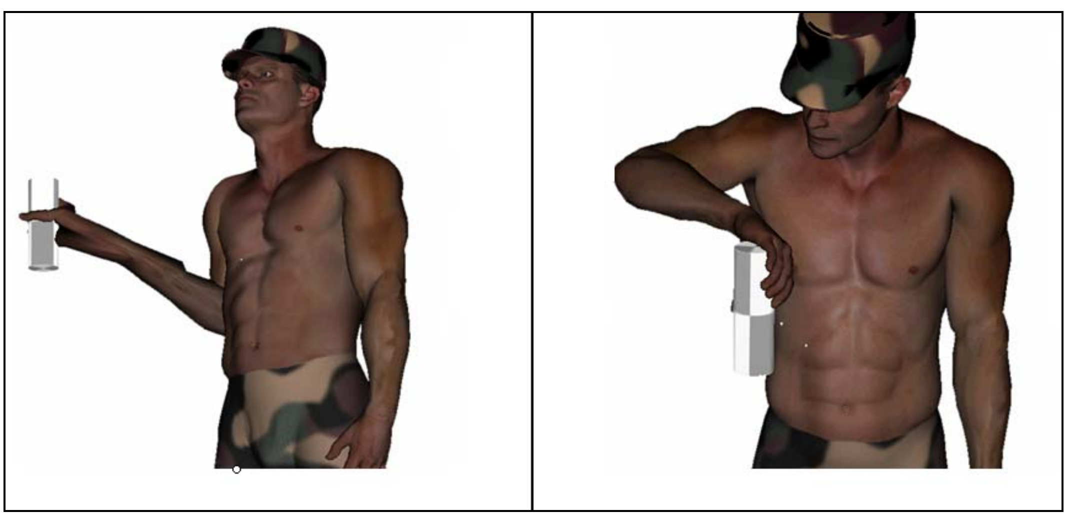

Figure 4. 6: Grasping involving the upper body: grasping without the upper body connected (left) and grasping the same object with the upper-body posture-prediction module active (right).

In the left part of the figure, we have placed the hand in the desired position and orientation without any regard to the rest of the body. The hand is detached from the rest of the body, and the posture looks odd. On the contrary, the right side of the figure shows the avatar grasping the cylinder with the posture-prediction module active. Notice how he is looking at the object and how his elbow, shoulder, and clavicle are all involved in producing a body posture that allows him to properly reach for the object.

Implementation Details

We have used Virtools as our simulation and visualization environment. We read the object mesh information and send it into our C++ code. Our code in turn uses the following external libraries:

ANN: Approximate Nearest Neighbor (Mount and Arya, 2006) to perform nearest-neighbor queries.

KMlocal (Mount, 2005): To perform k-mean clustering.

MTrand: For random number generation.

The code allows the user to control the behavior of the algorithm through an Options file. The file contains options for the number of samples per unit area, the threshold ε_d on the side lengths of similar triangles, and the maximum allowable distance between a point on the hand and its nearest neighbor on the object surface. The output of the code is the mean of each cluster obtained from the clustering stage. We are currently using four clusters in our experiments. We used a database of eight grasps as a proof of concept, but our implementation can easily be extended to include more grasps.

Results

Figures 4.7 and 4.8 show some of the simulation results we obtained from an initial database of grasps. We cluster the grasps into four groups, and we have noticed that we usually get two or three meaningful grasps and that one of the groups contains bad grasps or grasps in which the hand is not touching the object.

Figure 4.7: Two ways of grasping a sphere using the same hand shape.

Figure 4.8: Three different ways of grasping a cylinder using the same hand shape.

Figure 4.9 below is a plot of the running time of the shape-matching stage versus the number of samples per unit area on the surface of the object. These results were obtained for a spherical object being matched to a grasp in a database consisting of eight grasps, each with 24 samples. We noticed that the relative ordering of the grasps did not change when the number of samples per unit area was increased. Hence, in the remainder of our experiments we used around 10 samples per unit area.

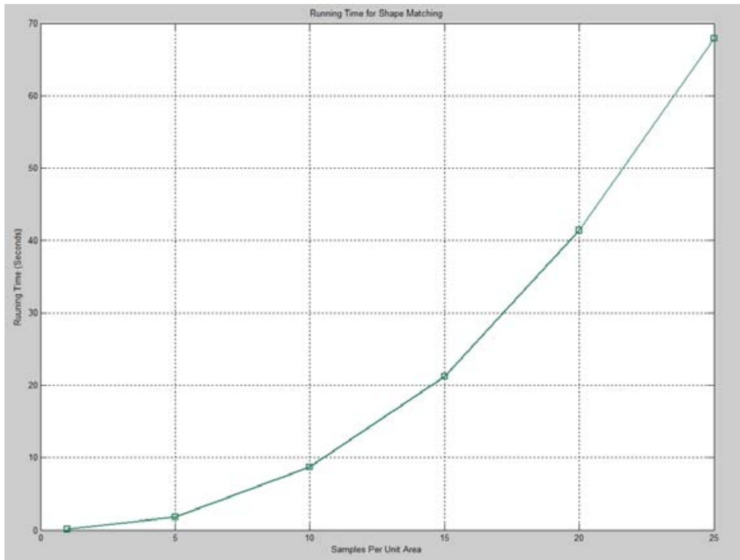

Figure 4.9: The running time of the shape-matching stage versus the number of samples per unit area on the surface of a sphere.

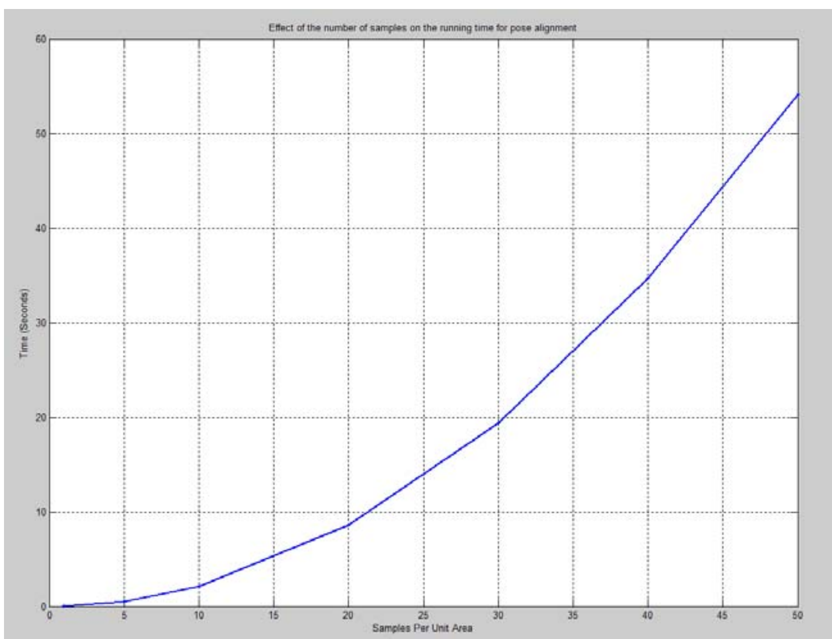

Figure 4.10: The running time for the grasp-alignment stage versus the number of samples per unit area on the surface of a sphere.

Figure 4.10 above shows the relationship between the running time of the grasp-alignment stage and the number of samples per unit area. The data points in the plot were generated by aligning a grasp with a spherical object of radius 8.2 composed of 24 faces. All the user-specified thresholds in the options file were set to a value of 1.5. Note that all the experiments were performed on an Intel Pentium 4 dual core 2.13 GHz machine with 2 GB of RAM.

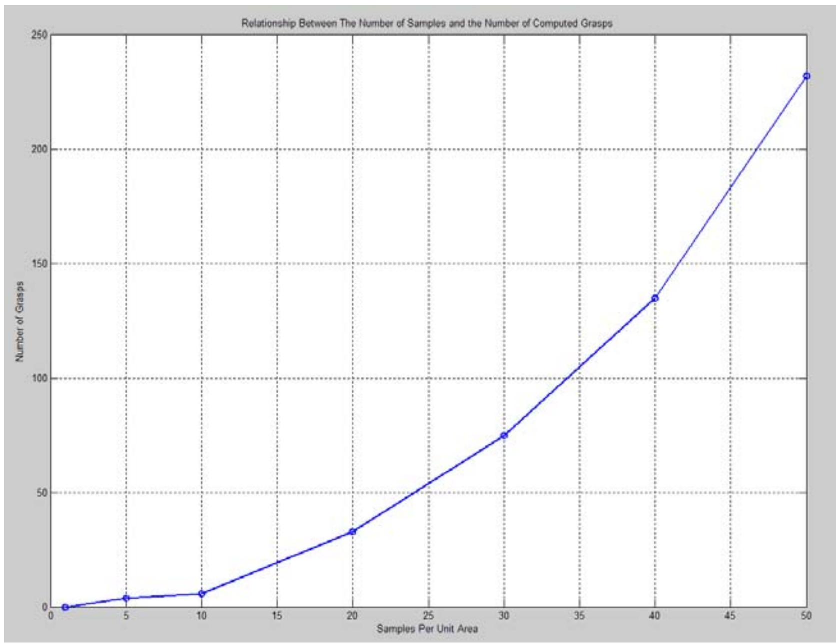

Figure 4.11: The number of possible grasp alignments versus the samples per unit area for a sphere.

We noticed that as the samples per unit area were increased, the system found more possible alignments between the hand and the object. For example, in Figure 4.11, we have plotted the number of alignments (before clustering) against the number of samples per unit area for a spherical object. Another factor that had a significant impact on the number of produced grasps was the threshold on the distance between the hand and the object in order to consider a grasp as valid. This is shown in Figure 4.12.

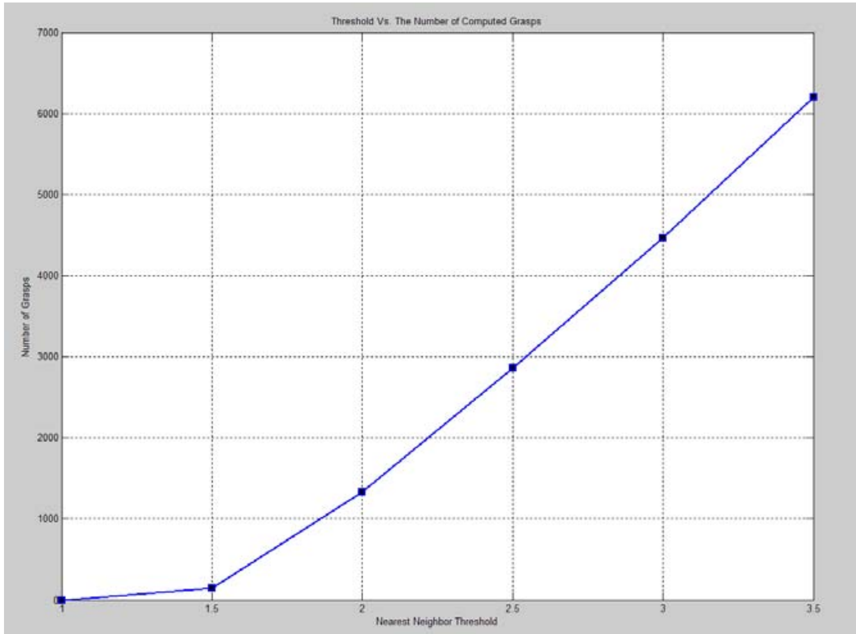

Figure 4.12: The relationship between the number of alignments found by the system and the nearest-neighbor threshold for a spherical object.

Discussion and Conclusions

The results shown in Figures 4.7 and 4.8 are not refined in any way. They look natural, and at least one of the produced grasps (the output of the clustering algorithm) corresponds to a grasping posture that an actual human would perform. However, we noticed that the results and the running time were sensitive to the number of samples taken per unit area. The more samples we took, the more time it took to complete the calculations. However, more samples also produced more possible grasps, increasing the quality of the end results. There should be a tradeoff between the number of desired grasps and the running time of the algorithm. In many cases, we needed to experiment a few times with the user-specified options before we obtained satisfying results within a reasonable amount of time. This process can be automated by allowing the algorithm to run several times, each time adjusting the options until the desired number of grasps is obtained.

Currently, we are using a database of eight grasps. Our code can perform for an arbitrary database size. However, that would increase the running time for the shape-matching stage because each grasp would be compared to the object. Our database currently contains generic hand shapes that are applicable to arbitrary objects; however, we plan to build a specialized database to handle a specific family of objects.

An interesting observation is that not all grasps that made sense from a grasping point of view resulted in natural upper-body posture. For example, the grasp shown in the right part of Figure 4.7 looks efficient when considering only the hand in relationship to the object, but it can be easily noted that the elbow and the shoulder are in a very awkward and uncomfortable position. This has inspired us to consider developing an upper-body grasp-quality metric that can filter out uncomfortable grasps.

One area that deserves special attention is the relationship between the clustering algorithm used and the results. Using a dynamic number of clusters instead of a fixed number might yield better results. Also, our algorithm has the same weaknesses that most clustering algorithms suffer from: one poisoned point in the dataset will surely affect the cluster means and hence the final results. Another point to consider is the space in which clustering is performed. Clustering in spaces other than the 16-dimensional transformation matrix space might be an idea worth pursuing in the future.

Although our work in this chapter is similar to that of Li and Pollard (2005), we believe that we have made significant contributions to this area of research. We have not relied on motion capture data from human hands in order to construct the database. When mapping the grasping data from a human hand to the Santos™ hand, errors and inaccuracies would be introduced no matter how accurate the mapping process is. Also, when modeling real objects in the virtual world, it is very hard to create perfect replicas. For this reason, we used the Santos™ framework itself to create the database. The most important contribution of this work is the coupling of the whole body with the hand to produce a complete reaching and grasping posture. The resulting upper-body posture was used to judge the feasibility of the produced grasps. We decided that grasps that resulted in awkward or uncomfortable body postures should not be considered valid.

For our future work, we wish to pursue the following topics:

- Develop an upper-body grasp-comfort measure than can prune grasps that result in uncomfortable upper-body postures.
- Add collision detection to our system to prune grasps that result in the hand penetrating the object.
- Use the existing finger-posture-prediction module to automatically refine the resulting grasps or to allow the user to interactively enhance the grasps.

CHAPTER 5

FINGER POSTURE PREDICTION

Introduction

In the previous chapters, we introduced the hand model and proposed methods for grasping objects using power grasps. In this chapter, we address the problem of predicting hand postures that correspond to specified fingertip locations. We present an optimization-based inverse kinematics method that will calculate the joint angles corresponding to a set of fingertip locations.

Many times when objects are being grasped and manipulated, whether in a real or a virtual environment, the need arises for placing a given finger at a given location. For example, the user might want to press the button of a joystick or a computer mouse. Also, when humans grasp small and light objects, they often use only their fingertips to touch and manipulate the object, resulting in what is termed "precision grasps." This can be clearly observed when a person grasps a pencil in order to write, uses a pair of tweezers, or grasps a hot cup of coffee.

Our work in this chapter leverages the work presented by Farrell (2005), Marler et al. (2005-a), and Marler et al. (2005-b) for the upper body and extends it to the hand. It is similar to the work of Borst and his colleagues on robotic hands (Borst et al., 2002), which we mentioned in Chapter 1. We begin by formally defining the finger posture-prediction problem and formulating it as a standard optimization problem. Then, we explain how this procedure is implemented in Santos™ and simulated in a virtual environment. We then present some of the results we obtain from this implementation. Finally, a discussion of the pros and cons of this method is presented, along with an outline of future research.

Formulation

We can define the finger posture prediction problem as follows: given the current hand posture and orientation, and a set of points that contains a target point for each fingertip, what is the best way to configure the joints of the fingers in order for the fingertips to touch the target points?

If we can mathematically describe the hand position and orientation, and the posture of the fingers, then the only ambiguity that remains in the aforementioned problem definition is defining the “best” way of achieving the posture. We postulate that there could be more than one way of rotating the joints of the fingers in order to touch a given set of targets. Furthermore, we assume that there is one optimal way of touching the target points that is most comfortable for humans and tends to result in natural-looking hand postures. Thus, we require that the optimal hand posture minimizes the value of a cost function that is dependent on the values of the individual joint angles. In our work, we propose a simple cost function that describes the deviation of the joints from their neutral positions. This is shown in Equation (5.1) below.

Based on the above argument, the problem can be cast as an optimization problem as follows:

Given:

Joint neutral angles, $\mathbf{q}_N$

Desired target point for each fingertip, $\mathbf{X}^{\mathbf{T}}$

Find:

The joint angles, $\mathbf{q}$

To Minimize:

$$\mathrm{f}(\mathbf{q}) = \sum_{i=1}^{n} w_i (q_i - q_{Ni})^2 \tag{5.1}$$

Subject to:

Joint limits, $q_i^L \leq q_i \leq q_i^U$ (5.2)

$||X^{fingertip}-X^T|| \leq \varepsilon$ (5.3)

where $\mathbf{X}^{\mathbf{fingertip}}$ is the vector of fingertip positions, and ε is a number that approximates zero to avoid floating point numerical errors. $\mathbf{X}^{fingertip}$ is calculated using Equations (2.1) and (2.2) in Chapter 2. The norm used in Equation (5.3) above is an L^2 norm. The neutral angles and the joint limits were adopted from the work by Tubiana (1981). As mentioned in Chapter 2, we have 25 degrees of freedom in each hand, so the dimension of our design variable vector for both hands is 50. The optimization problem is solved using the SNOPT software library (Gill et al., 2002), which uses a sequential quadratic programming algorithm

Implementation in the Virtual Environment

Figure 5.1 below shows the interconnection between the different components of the posture-prediction user interface. All the calculations are performed using the optimization solver SNOPT, which receives as input from the simulation environment (Virtools) the initial hand posture and the set of target points that are selected by the user. The final posture consists of the 50-dimensional vector of joint angles (design variables). The joint limits are the constraints of Equation (5.2) above. The link lengths and joint limits used in our experiments are shown in Chapter 2, in Table 2.1 and Figure 2.5.

A variable anthropometry interface allows the user to choose different hand sizes according to pre-calculated percentiles, or to individually customize the length of each bone in the hand model and the range of motion of the associated joints. The link length and joint limits are sent as input to the optimization solver. If there is a feasible solution (a solution to the problem that satisfies Equations (5.2) and (5.3) above) to the problem, the joint angles corresponding to this solution are sent to the simulation environment and the fingers are rotated accordingly.

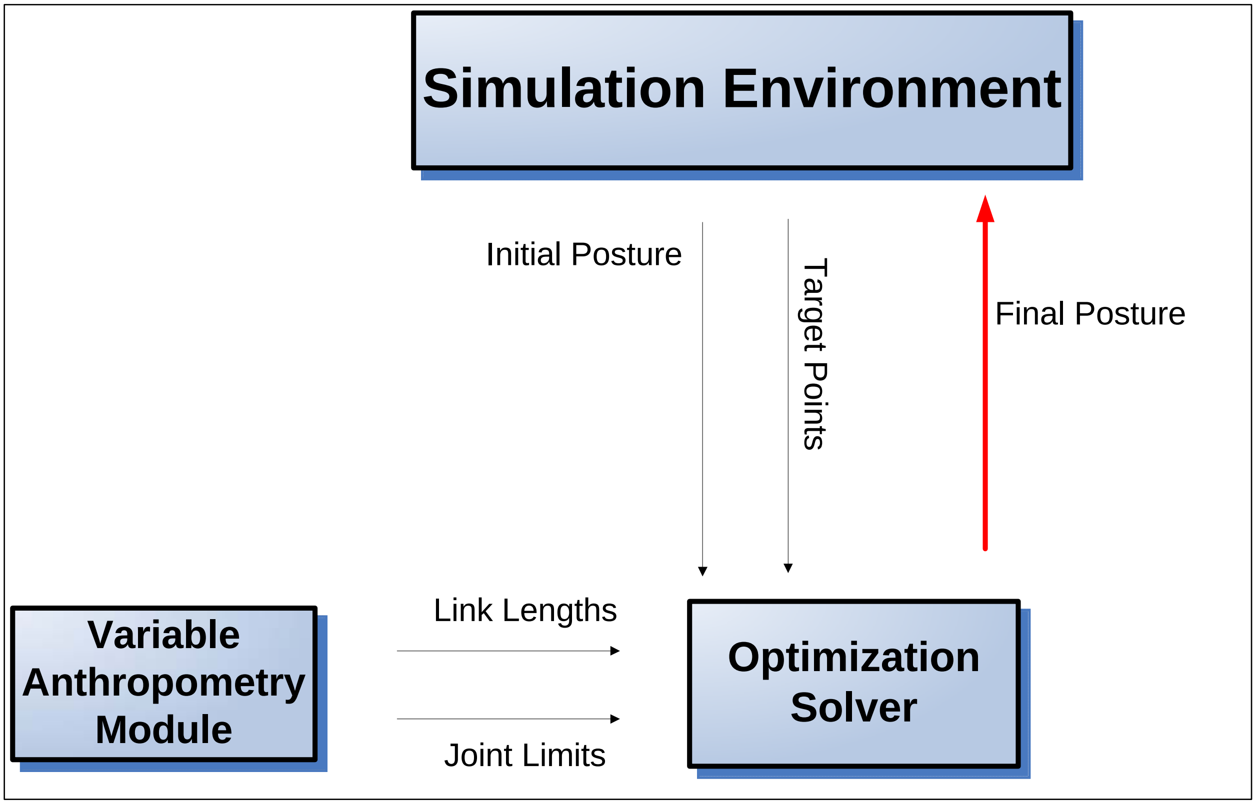

Figure 5.1: The hand posture prediction interface.

Sometimes no feasible solution to the problem exists, due to the selection of target points that are unreachable by the fingers. In this case, the solver will output a set of angles that minimizes the violation of the constraints in Equations (5.2) and (5.3) above.

Results

The initial posture-prediction results are promising. The time taken to solve the optimization problem was less than 1 second on an Intel Pentium 4 dual core 2.13 GHz machine with 2 GB of RAM. Figure (5.2) below shows the hand posture before and after grasping a small sphere. The red balls represent the target points for the thumb, index, and middle fingers, as specified manually by the user. Note that the two targets for the ring and pinky fingers are not visible in the figure because they are on the palm. The resulting posture is shown in the right-hand portion of the figure. Note that in this figure, the starting posture is the neutral posture defined for the hand, and it is the same posture that we use in Equation (5.1) throughout the experiments.

Figure (5.3) shows the hand grasping a joystick. The left part of the figure is the starting configuration, and the right side shows the hand after the thumb is positioned to press the button on the hand of the joystick.

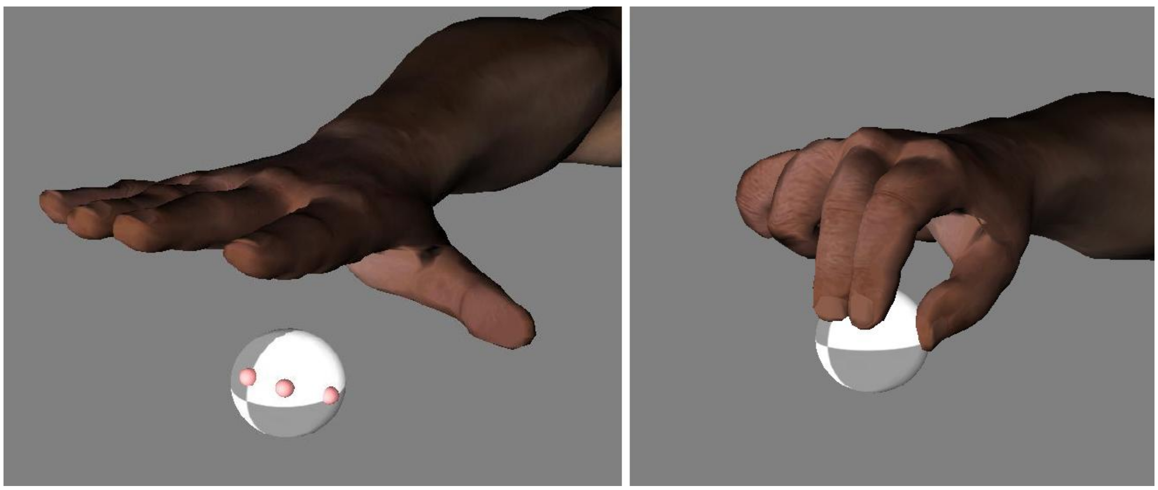

Figure 5.2: Santos performing a precision grasp on a spherical object. (Left) starting posture, (Right) final posture.

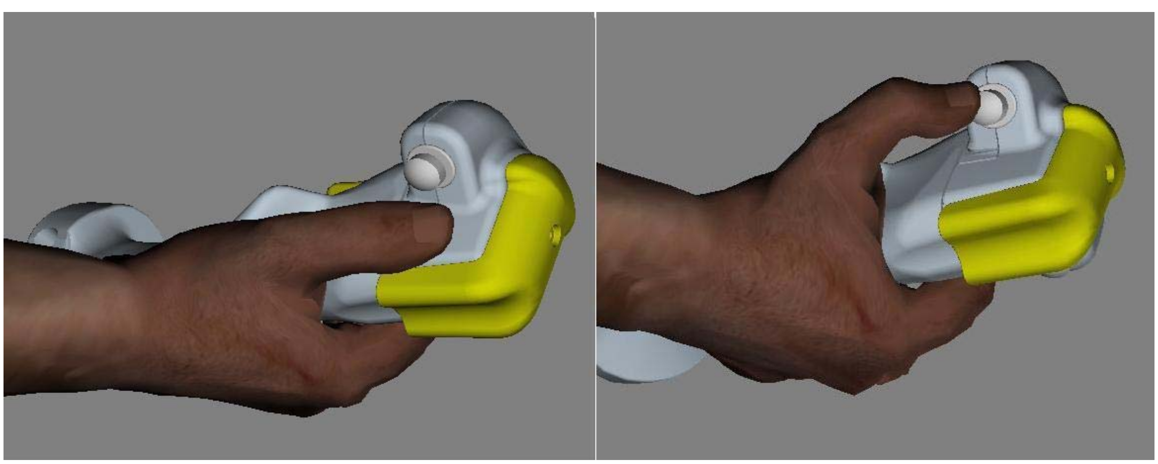

Figure 5.3: Santos pressing a button on a joystick. (Left) starting posture, (Right) final posture.

Although we have not performed rigorous validation on our results, it is evident that the resulting postures look natural and follow the posture that a real human would perform to touch the target points. Validation of our simulated results and experimentation with different cost functions remain areas of future research.

Discussion and Conclusions

In this chapter, we have formulated the precision grasp problem with known target points as a constrained optimization problem. We have demonstrated the soundness of this approach by providing simulation results on a virtual human. The virtual avatar was able to form a hand posture that corresponded to a set of user-specified contact points in real time.

The amount of redundancy in the joint angles of the fingers that satisfy a given set of target points seems to be limited. This becomes obvious when we compare posture prediction for the hand with upper body posture prediction: There are many ways to touch a target point when the arms and trunk are involved in the posture. This minimizes the role of the objective function and optimization formulation. However, this method has been shown to be faster and more flexible than other inverse kinematics methods, and it has served our purpose well in predicting posture for the fingers.

Although the proposed system is useful, there are issues that need to be addressed before the system can be a part of a reliable and automatic grasp planner for virtual environments. Some of these issues are summarized below:

1. We have assumed that the user will select the locations of the fingertips that will guarantee a stable grasp. This makes the system semi-automatic. If a more autonomous grasping behavior is desired, then it would be more appropriate to calculate the contact points on the object surface using an automatic algorithm. This has proven to be a difficult problem and is

outside the scope of this text. An in-depth treatment of this subject can be found in the robotic grasping literature.

2. Many times, the user might pick points that are unreachable by the hand. This causes the fingers to move into an unnatural posture in order to minimize the distance between the fingertips and the target points (Figure 5.4). A better approach would be to check whether the target points are within the reach envelope of the hand before attempting to solve the optimization problem. The work of Abdel-Malek and his colleagues (Abdel-Malek et al., 2006) or a similar approach can be used to achieve this goal.

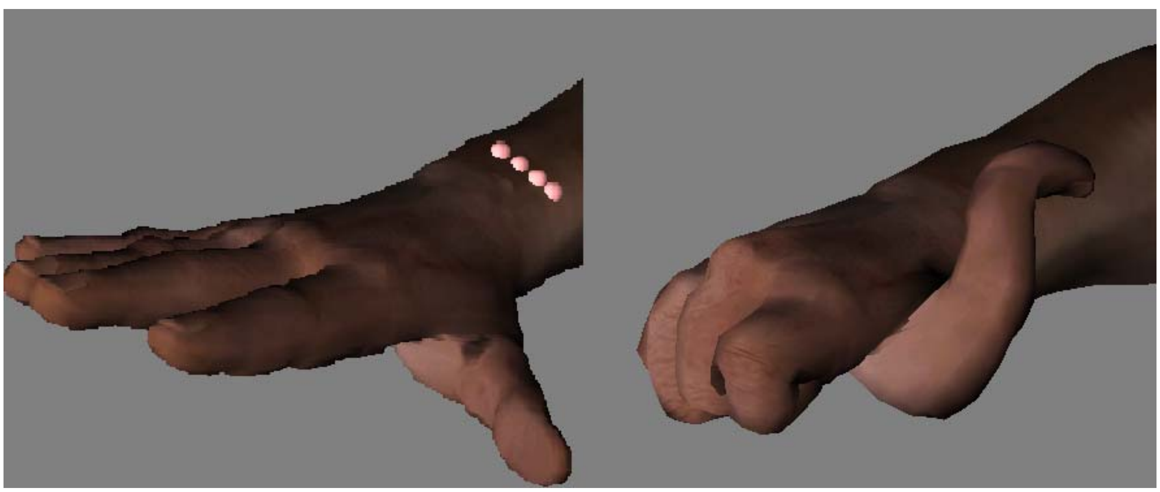

Figure 5.4: A bad choice of target points (red spheres) can result in unnatural postures while trying to minimize the distance to the target points.

3. Oftentimes, we noticed that the fingers might go inside the object in order to reach the targets via the easiest possible path. Obstacle avoidance can be readily incorporated into the formulation of the optimization problem by filling the object and the hand with spheres and using additional

constraints to prevent the spheres in the hand from touching those of the object.

4. The cost function that we have used (deviation from the neutral position) is just one of many possible cost functions. Further experimentation is needed to come up with a function that best approximates the behavior of humans when performing precision grasps. For example, we can use the sum of joint torques as a cost function that needs to be minimized.
5. It was observed that when humans move their fingers, the motion of one finger affects the motion of others to some extent. These inter-finger coupling relationships can be modeled through additional constraints in the optimization problem and will help produce postures that are more natural.

CHAPTER 6

GRASP QUALITY

Introduction

We have focused so far on the synthesis of grasps: How to find the hand position, hand orientation and joint angles for a given object geometry. Another important aspect of grasping in the virtual world is grasp analysis. Specifically, we are interested in finding whether a given grasp is good enough for our application. To achieve this, we need a quality criterion that can help us determine how well a given grasp restrains an object, and to compare between several possible grasps for the same object in order to pick the best one. For this purpose, we have implemented a grasp quality measure for Santos™. This measure is quite popular in the robotics literature, where it has been used to plan grasps for robotic grippers.

We begin with a brief review of the related literature, and then we present a precise formulation for the grasp quality problem. We provide a simple example in 2-D to clarify the process of calculating the quality of a grasp. After that, we explain the details of the implementation of the quality metric for a virtual human in a simulation environment. Finally, we include actual results for the application of this quality metric on the virtual human Santos™. We compare different grasps of the same object and show that the quality measure is intuitive and consistent with the human common sense.

Related Work

There has been a great deal of research concerning grasp quality measures. Here, we outline some of the publications that are particularly relevant to our work. Salisbury (1982) introduced the notion of *form closure grasps* in which a grasp completely restrains an object. Kirkpatrick and his colleagues (1990) proposed a general measure of grasp quality for an *n*-contact grasp which is independent of task. They defined the grasp quality as the radius of the largest sphere that can fit within the unit grasp wrench space.

Ferrari and Canny (1992) developed this measure further and visualized the wrench space using the convex hull of the contact wrenches given that the maximum contact force has an upper bound of unity. Pollard (1994) proposed to scale the torques by the maximum moment arm in order to make the quality measure independent of the object scale. All these formulations assumed that the task wrench space is unknown.

Other authors attempted to relate the task wrench space to the quality of the grasp. Li and Sastry (1988) state that a good grasp should be task oriented, and model the tasks as six dimensional ellipsoids in the wrench space of the object. Borst et al. (2004) present a method for calculating the task wrench space from the object wrench space.

Miller and Allen (1999) present a robotic grasp simulator that is capable of calculating the quality of grasp generated on different objects using different hand models. Our work is most similar to this, but is adapted for the Santos™ simulation environment. We use a task independent quality measure because of the difficulty of modeling tasks associated with every graspable object in the virtual environment.

Formulation

We assume that the grasp consists of n contact points with a coulomb model of friction. Thus, a grasp can be represented as a set of point contacts G = $\{C_1, C_2, \ldots C_n\}$. In order to ensure non-slippage of the fingers, the contact force $\boldsymbol{f}_i$ at point C_i is constrained to lie within the friction cone at that point specified by the friction coefficient μ, the contact point C_i, the cone's half angle φ, and the contact normal $\boldsymbol{n}_i$. (figure 6.1).

The friction cone constraint can be written as:

$$f_t \leq \mu f_n \tag{6.1}$$

where f_t is the tangential component of the force vector and f_n is its normal component.

Next, we approximate the friction cone by an m-sided friction pyramid. Hence, any valid force vector that satisfies equation (6.1) can be expressed as a convex

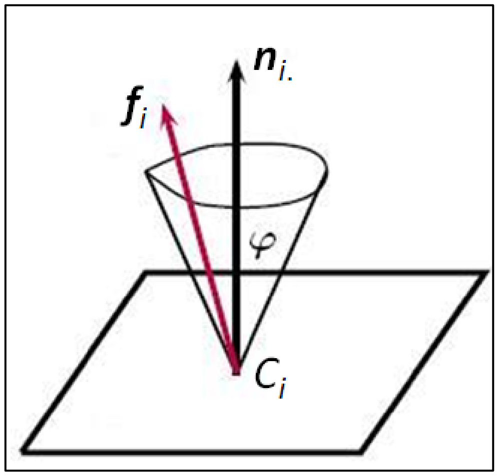

Figure 6. 1: Friction cone with half angle φ.

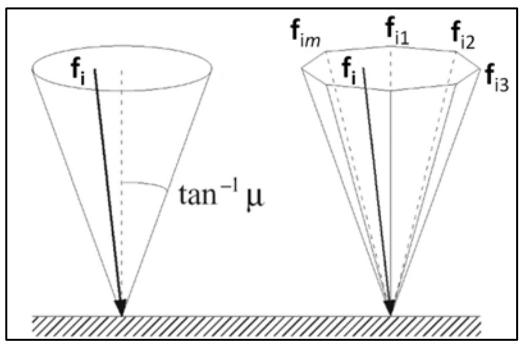

Figure 6. 2: Approximating a friction cone by an m-sided pyramid

combination of force vectors lying on the side of the friction pyramid(Figure 6.2), as follows:

$$f_i = \sum_{j=1}^{m} \alpha_{ij} f_{ij},$$
$$\alpha_{ij} \geq 0, \sum_{j=1}^{m} \alpha_{ij} = 1 \tag{6.2}$$

where f_{ij} represents the j^{th} force vector around the i^{th} friction pyramid, and α_{ij} are nonnegative convex coefficients.

We normalize the j force vectors lying on the boundary of each convex pyramid so that the following condition is satisfied:

$\boldsymbol{f}_{ij}.\boldsymbol{n}_i = 1$ (6.3)

Referring to equation (6.3), $\sum_{j=1}^{m} \alpha_{ij}$, which represents the amplitude of the normal component of the force, is restricted to be equal to 1.

Using a coordinate system at the object center of mass, the wrench $\boldsymbol{w}_i$ produced by grasp force $\boldsymbol{f}_i$ is given by:

$$w_i = \begin{pmatrix} f_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} f_i \\ r_i \times f_i \end{pmatrix} \tag{6.4}$$

where $\boldsymbol{r}_i$ is the vector pointing from the object center of mass to the contact point c_i. The wrench is 3 dimensional when considering grasps in 2D and 6 dimensional for grasps in 3D .Consideration of the force and moment components together in wrench space requires scaling of the components for compatibility. In this work, we use the torque multiplier suggested by Pollard (1994) which is the maximum moment arm:

$$\lambda = \frac{1}{|r_{\max}|} \tag{6.5}$$

We redefine equation (6.4) by replacing r_i with λr_i:

$$w_i = \begin{pmatrix} f_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} f_i \\ \lambda r_i \times f_i \end{pmatrix} \tag{6.6}$$

this will insure that the quality formulation is independent of object scale.

The wrench applied by the hand on the grasped object is given by:

$$\boldsymbol{w}_{\text{grasp}} = \sum_{i=1}^{n} w_i = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} w_{ij} \tag{6.7}$$

Note that equation (6.7) defines a convex cone in the wrench space which specifies the feasible external wrenches that the grasp can produce by applying valid contact forces $\mathbf{f}_i$, $i=1,\ldots,n$.

If we use an L_1 norm and bound the sum of magnitudes of the contact normal forces as follows (Ferrari and Canny 1992):

$$\sum_{i=1}^{n} \| f^n{}_i \| \leq 1 \tag{6.8}$$

where $f^n{}_i$ is the normal component of the i^{th} force vector, then the external wrench produced by the grasp becomes equivalent to the convex hull of the wrenches:

$$W_{L1} = ConvexHull(\bigcup_{i=1}^{n}\{w_{i1}, \ldots, w_{im}\}) \tag{6.9}$$

If the convex hull contains the wrench space origin, and there is no external force on the object, then the grasp can achieve equilibrium, and the grasp is said to be *force closure*. However, when an external force or moment is applied, the convex hull of the given grasp must enclose the wrench caused by this external disturbance to achieve equilibrium.

One quality measure that is often proposed in the literature is the radius of the largest 6D ball, centered at the origin, which can be enclosed within the hull. This represents the smallest maximum wrench over all directions that can be applied by the grasp to resist any external disturbance applied to the object. This quality measure assumes that no information about the task wrench space is known and that the disturbance wrenches could come from any direction (Ferrari and Canny, 1992 and Kirkpatrick et al. 1990).

An Example of Grasp Quality Computation

To clarify the procedure of calculating the grasp quality, we present a simple example in 2D. The resulting wrench space will be 3-dimensional and can be easily visualized (as opposed to the 6D wrench space that results from a problem in 3D). Figure 6.3 shows an object in the plane that is being grasped at 4 contact points. Each contact force is directed along the normal to the surface at that point.

We assume forces of unit magnitude and calculate the wrenches at each contact point according to the equations presented in the previous section. Finally we construct the convex hull of the contact wrenches and use that to calculate the grasp quality.

We begin by calculating the projection of the force at each point along the two vectors delimiting the friction cone. We assume a friction coefficient of 0.577, which corresponds to a friction cone with a half angle of 30 degrees.

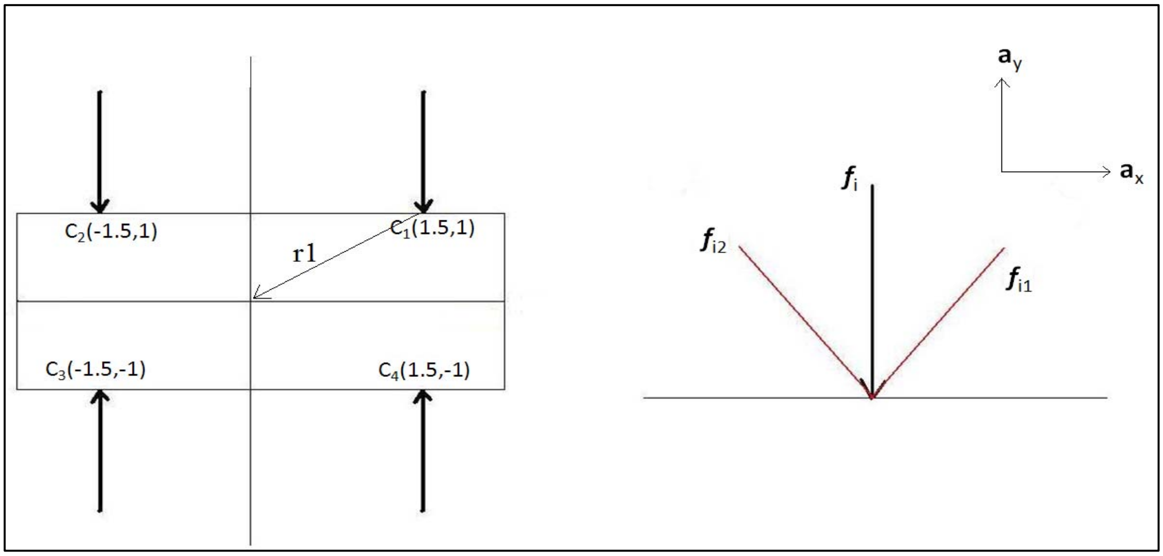

Figure 6. 3: Example grasp in 2D (left). The friction cone at each contact point (right).

In the following, $\mathbf{a}_x$,$\mathbf{a}_y$, and $\mathbf{a}_z$ denote unit vectors in the direction of the x,y and z axes respectively. For each force $\boldsymbol{f}_i$ at point C_i, we calculate two values $\boldsymbol{f}_{i1}$ and $\boldsymbol{f}_{i2}$ which are the projections of $\boldsymbol{f}_i$ along the boundary of the friction cone (Figure 6.2). Since all force vectors $\boldsymbol{f}_{ij}$ will be normalized according to Equation (6.3), we assume that they have unit magnitude.

For C1, we have:

$\boldsymbol{f}_{11} = [-\sin(30)\mathbf{a}_x,-\cos(30)\mathbf{a}_y] = [-0.5\ \mathbf{a}_x,-0.866\ \mathbf{a}_y]$

$\boldsymbol{f}_{12} = [\sin(30)\ \mathbf{a}_x,-\cos(30)\ \mathbf{a}_y] = [0.5\ \mathbf{a}_x,-0.866\ \mathbf{a}_y]$

We now normalize $\boldsymbol{f}_{i1}$ and $\boldsymbol{f}_{i2}$ according to Equation (6.3). That equation requires that $|\boldsymbol{f}_{ij}|.|\boldsymbol{n}_i|.\cos(\theta) = 1$. Since we assume a unit normal, we need to scale the forces by a factor of $1/\cos(\theta) = 1/\cos(30) = 1.155$. This yields the following normalized values of $\boldsymbol{f}_{11}$ and $\boldsymbol{f}_{12}$:

$\boldsymbol{f}_{11}{}' = [-0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y]$

$\boldsymbol{f}_{12}{}' = [0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y]$

Next, we calculate the torque using Equations (6.4) and (6.5) as follows:

$$\tau_{11} = \frac{1}{|\boldsymbol{r}_{max}|} \boldsymbol{f}_{11} \times \boldsymbol{r}_1$$

where $|r_{max}|$ in this case is equal to $\sqrt{1^2 + 1.5^2}$. This yields the following:

$$\tau_{11} = \frac{1}{\sqrt{1^2 + 1.5^2}} \left[-0.577\mathrm{a}_x, -1\mathrm{a}_y\right] \times \left[-1.5\mathrm{a}_x, -1\mathrm{a}_y\right]$$

Finally we use Equation (6.6) to find out the wrench $\boldsymbol{w}_{11}$. This whole procedure is repeated for all the contact points and we end up with the following three-dimensional wrenches:

$\boldsymbol{w}_{11} = [-0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y, -0.512\ \mathbf{a}_z]$

$\boldsymbol{w}_{12} = [0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y, -1.15\ \mathbf{a}_z]$

$\boldsymbol{w}_{21} = [-0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y, 1.15\ \mathbf{a}_z]$

$\boldsymbol{w}_{22} = [0.577\ \mathbf{a}_x, -1\ \mathbf{a}_y, 0.512\ \mathbf{a}_z]$

$\boldsymbol{w}_{31} = [0.577\ \mathbf{a}_x, 1\ \mathbf{a}_y, -0.512\ \mathbf{a}_z]$

$\boldsymbol{w}_{32} = [-0.577\ \mathbf{a}_x, 1\ \mathbf{a}_y, -1.15\ \mathbf{a}_z]$

$\boldsymbol{w}_{41} = [0.577\ \mathbf{a}_x, 1\ \mathbf{a}_y, 1.15\ \mathbf{a}_z]$

$\boldsymbol{w}_{42} = [-0.577\ \mathbf{a}_x, 1\ \mathbf{a}_y, 0.512\ \mathbf{a}_z]$

Next, we calculate the convex hull of these wrenches using the software package QHull (Barber et al. 1996). We set the output options to display the hull as a list of normals to each facet with an associated offset. The minimum offset is the grasp quality.

Figure 6.3 shows the results for different contact points. In part (a), the convex hull is plotted using MATLAB, along with the grasp quality represented by the largest

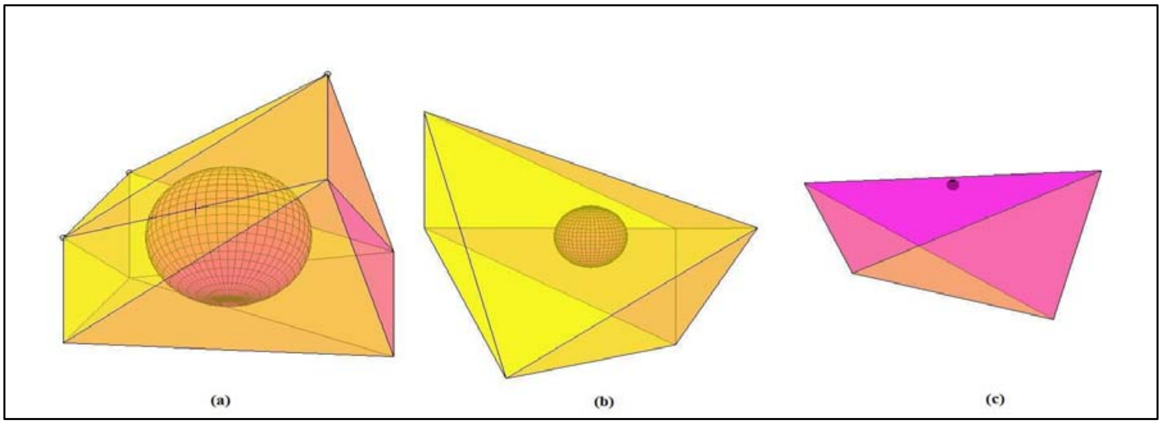

Figure 6. 4: The convex hull and grasp quality for different contact points on the same object. (a) All the contact points shown in Figure 6.2, the quality is 0.5(b) after contact points C1 is removed, the quality is 0.22 and (c) after both contacts C1 and C3 are removed. The quality in this case is 0.04.

sphere, centered at the origin, that can fit inside the hull. In part (b), we removed the contact point C1. Notice how the size of the hull and the quality ball are both reduced. However, both cases show the center of the wrench space contained within the hull. This means that both grasps can resist arbitrary external disturbances in any direction. However, in part (c), we removed both points C1 and C3 from the grasp. The quality is 0.04, which is conceptually zero if we neglect floating point errors in the computation. This result can be intuitively explained by the fact that when these points are removed, the grasp cannot resist forces that cause clockwise rotation of the object. Note also that the origin of the wrench space is on the boundary of the convex hull.

Integration into the Santos™ Environment

We have implemented grasp quality calculation for Santos™ using Virtools, C++ and QHull. Figure 6.5 below shows the interconnections between the different components of our system.

After the hand of Santos™ grasps an object in the virtual environment, collision between the hand and the object is detected and the contact points are recorded.

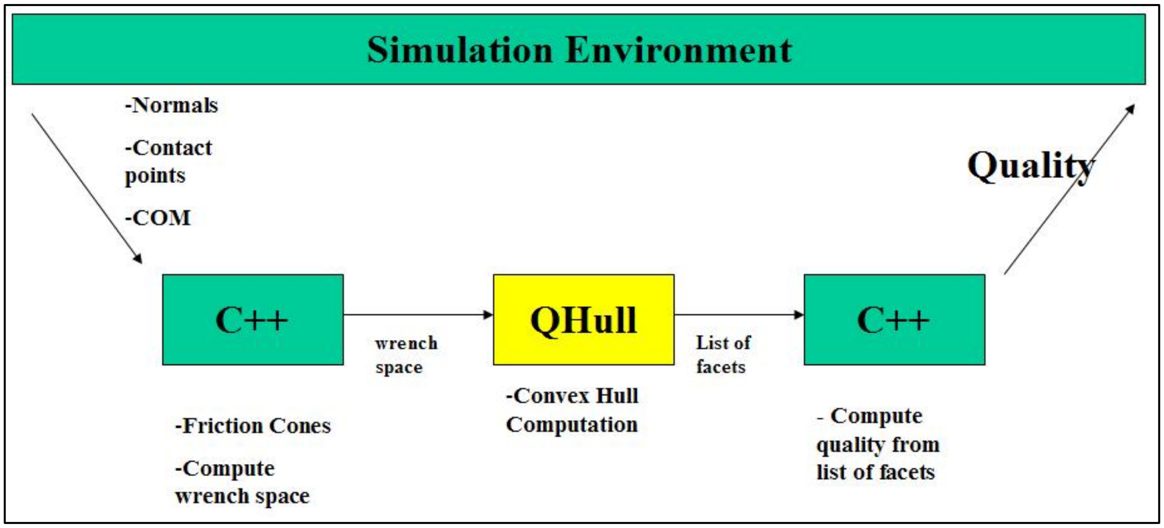

Figure 6. 5: Grasp quality implementation for Santos™.

After the hand of Santos™ grasps an object in the virtual environment, collision between the hand and the object is detected and the contact points are recorded. Since the hand is covered with a layer of deformable skin, we facilitate the collision detection process by inserting virtual spheres at each joint in the hand and at certain locations in the palm, and we perform collision detection between the spheres and the object. For each contact point, we have an associated contact normal perpendicular to the object surface at that point. The contact points and normals along with the object center of mass are passed to the next stage of the system. We assume that the object is composed of a single material of uniform density, which allows us to use the geometric center of the object as the center of mass.

Next, we calculate 8 force vectors lying on the boundary of the friction cone for each contact point. We thus approximate the friction cone by a "friction pyramid". Using these force vectors, we compute the resulting torques and form 8 six-dimensional wrenches for each contact point. This collection of wrenches is then passed to QHull, which in turn calculates the convex hull for these wrenches. The output of QHull is a list of facet normals and facet offsets from the origin. This list is passed to a C++ module which goes through the individual offsets and chooses the minimum one as the grasp quality. Finally, the result is passed back to the simulation environment and displayed for the user.

Results

Figures 6.6 and 6.7 below show some of the grasp quality results. The grasp quality module was able to calculate the quality in approximately 3 seconds on an Intel Pentium 4 dual core 2.13 GHz machine with 2 GB of RAM. Figure (6.6) shows a comparison between grasping a long cylinder and a shorter one. The module output a higher quality index for the shorter cylinder. This makes sense because in the case of grasping the shorter cylinder, the hand is closer to the center of mass of the object and the fingers wrap around the cylinder, thus providing more control and extra resistance to external disturbances.

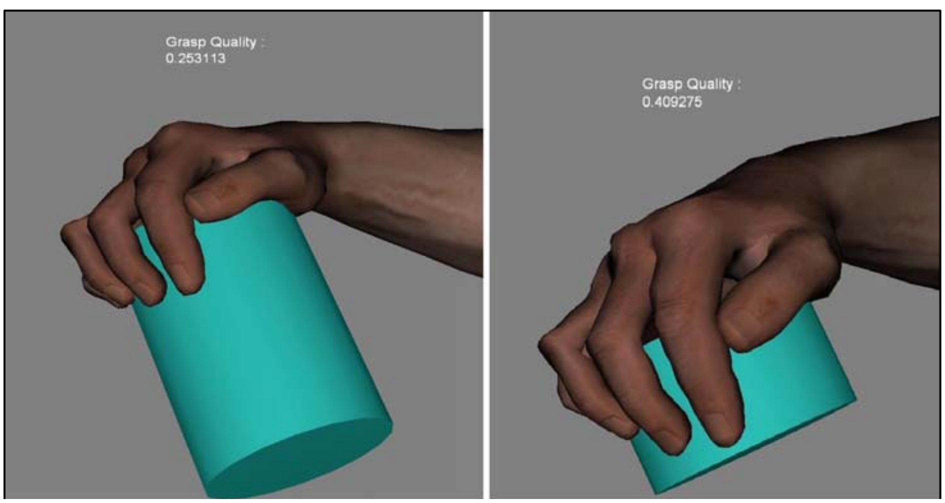

Figure 6. 6: Grasp quality results for 2 cylinders. The quality is higher for the shorter object

Figure (6.7) below shows Santos™ grasping a long cylinder. It also shows the virtual spheres that we placed inside the hand to act as collision detection sensors.

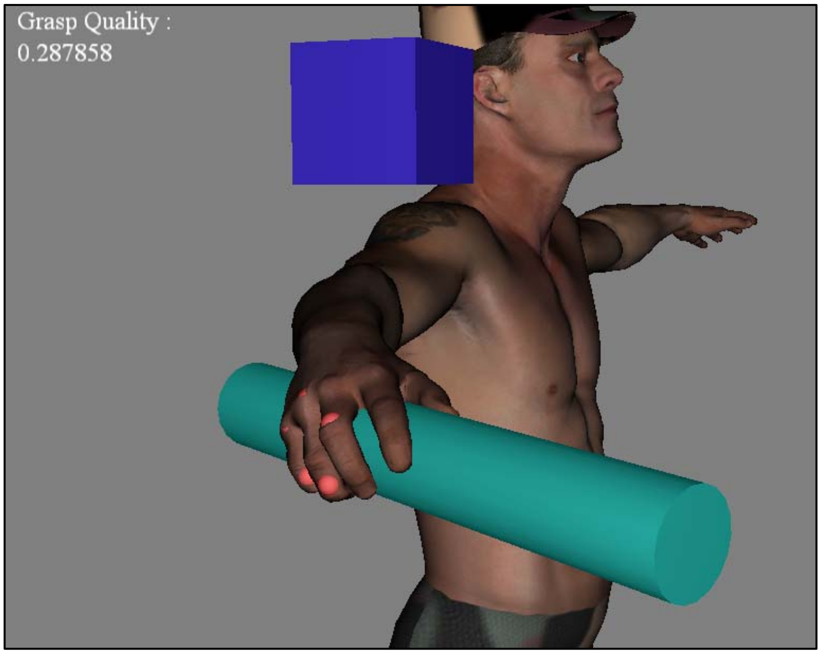

Figure 6. 7: Spherical sensors for collision detection (red spheres).

Discussion and Conclusions

In this chapter, we have presented a real-time grasp quality module that can be used to assess the quality of produced grasps. We have successfully used this module with posture prediction and shape matching modules to assess the quality of our results.

It is important to note that although this module is useful as it is, it does not take into consideration the task or the function of the object. It is useful to judge how well the hand can mechanically restrain the object. For example, this quality measure will not be able to penalize a grasp of a joystick in which the hand is faraway from the buttons, or if it is grasping a mug upside down. This can be handled by specialized quality measures that use the concept of a “Task Wrench Space”, see (Borst et al 2004) for example. The reason we decide to use a task independent quality measure is because we want a quality metric that is applicable to novel unknown objects. We do not wish to build a database of objects that stores the task associated with each one, because that would limit our grasping system to only the objects in the database. If we decide to have additional object information embedded inside the object, then we might as well store information on how to grasp the object and not bother with developing a sophisticated automatic grasp planner.

An interesting extension to the existing quality measure would be to incorporate the upper body posture and comfort into the grasp quality. This would enable us to describe as bad a grasp that involves the wrist rotating close to its joint limits, or one that has the shoulder twisted into an uncomfortable angle.

CHAPTER 7

CONCLUSION

Summary of Contributions

In this work we have presented a new and complete interactive system for simulating human grasping and interaction. We have presented several examples of the use of this system for solving the grasping problem defined in Chapter 1. Although we have used several concepts from the fields of robotics and human simulation and built upon existing work, we have enhanced some of the previous grasping algorithms and presented many novel ideas. These new ideas are summarized as follows:

1. We have extended the posture prediction capabilities of the upper body to include all 50 degrees of freedom of the hands. This was demonstrated using an intuitive user interface for selection of target points. This module was used independently to create precision grasps and also was part of a complete grasping system, where it was used to refine power grasps generated by the shape-matching and grasp-alignment module.
2. We have enhanced previous algorithms for shape matching and pose alignment. Specifically, we have created an application for synthesizing grasps in an interactive manner and saving them into a database of grasps, thus eliminating the use of motion capture and the issues associated with mapping the motion capture data to the hand model. We have also enhanced the performance of shape-matching and pose-alignment algorithms by using data structures specialized for nearest-neighbor calculations and modifying the pose-alignment algorithm to require fewer samples on the object.
3. We have laid the groundwork for studying the connection between the required hand posture for grasping and the kinematic limitations of the

upper body. By including the whole body in the grasp execution, we have identified cases in which planned grasps resulted in unnatural or awkward upper-body postures (the shoulder had to be twisted to its maximum joint limit, for example).

4. By using a grasp-quality measure from the robotics literature and applying it to a digital human environment, we have created a module for the assessment of simulated grasps in terms of mechanical stability. This was used to provide feedback on the grasps and to help the user choose between many ways of grasping a given object.
5. We have combined all the previously mentioned modules into one complete interactive grasping simulator, thus creating an environment for planning and evaluating grasps using digital humans.

Conclusions

Human grasping simulation has proved to be a complicated subject. We have proposed an interactive system to plan and assess human grasps and have obtained promising results using this system. During the process of design and implementation, we have drawn a few key insights, which we summarize as follows:

- The upper body is an integral part of the grasp. It can be used as a filter to reject hand alignments that result in awkward and unnatural arm postures.
- The proposed system can be used for product design and assessment because it gives the user the chance to independently guide the system toward the intended grasp with minimal effort. It does not force the user to accept one way of grasping the object, as a completely automatic system would, and at the same time does not require joint-by-joint manipulation to achieve the grasp.
- We have validated our work visually and subjectively by comparing the results with our observations of human grasping and manipulation actions. This of course

is not enough, and more thorough analysis is needed. We suggest using motion capture data to acquire actual human grasps and comparing them with the grasps obtained from our system.

- The grasping problem is extremely complex, mainly because it depends on the task and functionality that the object being grasped is intended for. The biggest obstacle facing people who develop automatic grasping simulators is designing a system that can judge the task requirements given the object's shape and physical properties. The first step toward automating grasp simulation is an advanced cognitive module that uses artificial intelligence algorithms to infer task requirements from shape information and previous successful grasping attempts.

Future Work

This work has achieved the objectives that were laid out in Chapter 1. Nonetheless, there is potential for improvement, and this work has shed light on exciting areas of future research. Areas for future work are summarized as follows:

- **Developing a grasp comfort measure that includes the upper body**. This can be used to filter grasps that make sense from a grasping point of view but are impossible to realize due to the joint limit and reachability constraints of the upper body. This would also help reject uncomfortable grasping postures.
- **Developing more objective functions for the finger posture prediction problem** and experimenting to determine the objective function (or combination of functions) that gives the most human-like postures. For example, we can use an objective function that measures the sum of joint torques.
- **Validation of the finger posture prediction results** by comparing them to results obtained from motion capture data.

- **Formulating grasp-quality measures specialized for human hands**. Such quality measures will take into consideration the unique structure of the human hand and how it is different from conventional robotic grippers. This measure would favor grasps that are more comfortable for humans.
- **Incorporating workspace analysis in finger posture prediction**. A major pitfall of the current finger posture prediction module is that the user can easily choose points that cannot be realized by the hand due to kinematic constraints (such as joint limits). It would be very helpful to use workspace analysis to check whether the chosen target points lie within the workspace of the hand before attempting to solve the optimization problem.
- **Adding collision detection and avoidance.** Since grasping always involves contact between the hand and the grasped objects, adding collision-detection capabilities would prove invaluable to this work. This can be used in posture prediction to avoid situations in which the fingers go through the object, or in grasp alignment to discard hand orientations that result in the fingers or the palm being inside the object. This can also be useful in the simple case where the user wishes to manually construct a grasp by direct joint manipulation.

APPENDIX

Given 2 congruent triangles in 3-dimensional space:

T_1: $A_1B_1C_1$

T_2: $A_2B_2C_2$

We wish to calculate a 4-by-4 transformation matrix **M** that, when applied to T_1, will result in T_2.

M is composed of a 3x3 rotational part, **R**, and a 3x1 translational part **t:**

$$\mathbf{M} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \tag{A.1}$$

The steps we follow are as follows:

1) Define local axes for each triangle.
   - For each triangle, the local x-axis is defined as the normalized vector **AB**.
   - The local y-axis is the normal to the plane of the triangle. This is obtained by performing the cross product **AC** X **AB** then normalizing the resulting vector.
   - The local z-axis is the cross product of the local x and the local y axes for each triangle.

   The local x,y and z axes for T1 are $\mathbf{x_1}$,$\mathbf{y_1}$, and $\mathbf{z_1}$ respectively.

   The local x,y and z axes for T2 are $\mathbf{x_2}$,$\mathbf{y_2}$, and $\mathbf{z_2}$ respectively.

2) Calculate the rotational part, **R**, of the transformation matrix using the following relationship:

$$\mathbf{R} = \begin{pmatrix} \mathbf{x}_2.\mathbf{x}_1 & \mathbf{x}_2.\mathbf{y}_1 & \mathbf{x}_2.\mathbf{z}_1 \\ \mathbf{y}_2.\mathbf{x}_1 & \mathbf{y}_2.\mathbf{y}_1 & \mathbf{y}_2.\mathbf{z}_1 \\ \mathbf{z}_2.\mathbf{x}_1 & \mathbf{z}_2.\mathbf{y}_1 & \mathbf{z}_2.\mathbf{z}_1 \end{pmatrix} \tag{A.2}$$

   where the "." denotes the dot product of two vectors.

3) Calculate $\mathbf{C_1}$, the centroid of T_1, and $\mathbf{C_2}$, the centroid of T_2. The centroid is calculated by averaging the three vertices of a triangle.

4) Calculate **t** using the following relationship:

$$\mathbf{t} = \mathbf{M_2} - \mathbf{R}\mathbf{M_1}$$

5) Construct **M** by substituting the values of **t** and **R** in equation (A.1).

REFERENCES

Abdel-Malek, K., Yang, J., Marler, T., Beck, S., Mathai, A., Zhou, X., Patrick, A and Arora, J. (2006) 'Towards a new generation of virtual humans', International Journal of Human Factors Modelling and Simulation, Vol.1, No. 1, pp. 2-39.

Aleotti, J. and Caselli, S. (2006), 'Grasp recognition in virtual reality for robot pregrasp planning by demonstration', proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, Florida.

Barber, C., Dobkin, D. and Huhdanpaa, H.(1996), 'The Quickhull algorithm for convex hulls', ACM Transactions on Mathematical Software, Vol. 22, No. 4, pp. 469-483.

Borst, C., Fischer, M. and Hirzinger, G. (1999), ' A fast and robust grasp planner for arbitrary 3D objects', proceedings of the IEEE International Conference on Robotics and Automation, Detroit, Michigan, May 1999.

Borst, C., Fischer, M. and Hirzinger, G. (2002), 'Calculating hand configurations for precision and pinch grasps', proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland.

Borst, C., Fischer, M. and Hirzinger, G. (2004), 'Grasp planning: how to choose a suitable task wrench space', proceedings of the 2005 IEEE International Conference on Robotics and Automation, New Orleans, LA.

Carenzi, F., Gorce, P., Burnod, Y. and Maier, M. (2005), 'Using generic neural networks in the control and prediction of grasp postures', The European Symposium on Artificial Neural Networks, Bruges, Belgium.

Chen, C., Hung, Y. and Cheng, J. (1997), 'RANSAC-based DARCES: A new approach to fast automatic registration of partially overlapping range images', Technical Report, Institute of Information Science, Academia Sinica, TR-IIS-97-019.

Cutkosky, M. (1989), 'On grasp choice, grasp models, and the design of hands for manufacturing tasks', IEEE Transactions on Robotics and Automation, Vol. 5, No. 3, pp269-279.

Cutkosky, M. R. and Howe, R. D. (1990), 'Human grasp choice and robotic grasp analysis', Dextrous Robot Hands, S. T. Venkataraman and T. Iberall, editors, Springer-Verlag, New York, NY, 5-31.

Denavit, J. and Hartenberg, R.S. (1955) 'A kinematic notation for lower-pair mechanisms based on matrices', Journal of Applied Mechanics, Vol. 77, pp.215–221.

Duda, R., Hart, P. and Stork D. (2001), 'Pattern Classification', John Wiley and Sons,New York, 2001.

Ehrenmann, M., Rogalla, O., Zollner, R. and Dillmann, R. (2001), 'Teaching service robots complex tasks:programming by demonstration for workshop and household environments', proceedings of the 2001 International Conference on Field and Service Robots, Vol. 1, pp. 397-402, Helsinki, Finland.

ElKoura, G. and Singh, K. (2003), 'Handrix: animating the human hand', in ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2003.

Farrell, K. (2005), 'Kinematic human modeling and simulation using optimization based posture prediction', masters Thesis, The University of Iowa, December 2005.

Fernandez, J. and Walker, I., (1998), 'Biologically inspired robot grasping using genetic programming', proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium.

Ferrari, C. and Canny, J. (1992), 'Planning optimal grasps', proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France.

Fischler, M. and Bolles, R. (1981), 'Random sample consensus: a paradigm for model fitting with applications to image analysis and cartography', Communications of the ACM, Vol. 24, No. 6, pp. 381-395.

Gill, P., Murray, W. and Saunders, A. (2002), 'SNOPT: An SQP algorithm for large-scale constrained optimization', SIAM Journal of Optimization, Vol. 12, No. 4 pp. 979-1006.

Globisch, R.(2005), 'Automated grasping for articulated structures using evolutionary learning algorithms', masters thesis, University of Johannesburg, South Africa, April 2005.

Gorce, P. and Rezzoug, N. (2005), 'Grasping posture learning with noisy sensing information for a large scale of multifingered robotic systems', The Journal of Robotic Systems, Vol. 22 , No. 12, pp. 711-724.

Hester, R., Cetin, M., Kapoor, C. and Tesar, D. (1999), 'A criteria-based approach to grasp synthesis', proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan.

Jagannathan, S. and Galan, G. (2004), 'Adaptive critic neural network based object grasping control using a three finger gripper', IEEE transactions on neural networks, Vol. 15, No. 2, pp.395-407.

Kallman, M., Aubel, A., Abaci, T. and Thalmann, D. (2003), 'Planning collision-free reaching motions for interactive object manipulation and grasping', Eurographics 2003, Vol. 22, No. 3, pp.313-322.

Katada, Y., Svinin, M., Matsumura, Y., Ohkura, K. and Ueda, K. (2001), 'Optimization of stable grasps by evolutionary programming', Proceedings of the 32nd International Symposium on Robotics, pp.1503-1508.

Kim, B., Yi, B., Oh, S. and Suh, I. (2004), 'Non dimensionalized performance indices based optimal grasping for multi-fingered robot hands', Mechatronics Vol.14,No.3, pp. 255-280.

Kirkpatrick, D., Mishra, B. and Yap, C. (1990), ' Quantitative Steinitz's theorems with applications to multi-fingered grasping', Proceedings of the 20th ACM Symposium on Theory of Computing, pp. 341-351, Baltimore, Maryland.

Li, Y. and Pollard, N. (2005), 'A shape matching algorithm for synthesizing humanlike enveloping grasps', IEEE-RAS International Conference on Humanoid Robots (Humanoids 2005), 2005.

Li, Z. and Sastry, S. (1988), 'Task oriented optimal grasping by multifingered robot hands', IEEE Journal of Robotics and Automation, Vol. 4, No. 1, pp. 32-44.

Lien, J. and Amato, N. (2006), 'Approximate convex decomposition of polygons', Computer Geometry Theory and Applications, Vol. 35 pp. 100-123.

Lien, J., Keyser, J. and Amato N. (2006), 'Simultaneous Shape Decomposition and Skeletonization', proceedings of the ACM Solid and Physical Modeling Symposium. (SPM), pp. 219-228, Cardiff, Wales, UK, June 2006.

Liu, G., Xu, J., Wang, X. and Li, Z. (2004), 'On quality functions for grasp synthesis, fixture planning and coordinated manipulation', IEEE Transactions on Automation Science and Engineering, Vol. 1, No. 2, pp.146-162.

Liu, Y., Starr, G., Wood, J. and Lumia, R. (2005), 'Spatial grasp synthesis for complex objects using model-based simulation', Industrial Robot: An International Journal, Vol. 32, No. 1, pp. 24-31.

Magnenat-Thalman, N and Thalmann, D.(2004) Handbook of Virtual Humans, John Wiley and sons.

Mantriota, G. (1999), 'Communication on optimal grip points for contact stability', The International Journal of Robotics Research, Vol. 18, No. 5, pp. 502-513.

Marler, T. (2004), 'Development of an orientation-constraint for human posture-prediction', University of Iowa, Virtual Soldier Research Program, Technical Report Number VSR-04.12.

Marler, T., Rahmatalla, S., Shanahan, M. and Abdel-Malek, K. (2005-a), 'A new discomfort function for optimization-based posture prediction', SAE Human Modeling for Design and Engineering Conference, Iowa City, IA.

Marler, T., Yang, J., Arora, J. and Abdel-Malek, K. (2005-b), ' Study of bi-criterion upper body posture prediction using pareto optimal sets, IASTED International Conference on Modeling, Simulation and Optimization, Oranjestad, Aruba.

Miller, A. and Allen, P. (1999), 'Examples of 3D grasp quality computations', IEEE International Conference on Robotics and Automation, pp. 1240-1245.

Miller, A., Knoop, S., Christensen, H. and Allen, P. (2003), 'Automatic grasp planning using shape primitives', IEEE International Conference on Robotics and Automation, ICRA '03, Vol. 2, pp. 1824-1829.

Miyata, N., Kouchi, M. and Mochimaru, M.(2006), 'Posture estimation for screening design alternatives by DhaibaHand-cell phone operation', proceedings of the SAE 2006 Digital Human Modeling for Design and Engineering Conference, pp. 2006-01-2327.

Mount, D. and Arya, S. (2006), 'ANN: A library for approximate nearest neighbor searching', available at http://www.cs.umd.edu/~mount/ANN/.

Mount, D., (2005), ‘KMlocal: a testbed for k-means clustering algorithms.’, Available at http://www.cs.umd.edu/mount/projects/kmeans/km-localdoc.pdf

Moussa, M. (2004), ‘Combining expert neural networks using reinforcement feedback for learning primitive grasping behavior’, IEEE transactions on Neural Networks, Vol. 15, No. 3, pp.629-638.

Ohbuchi, R., Minamitani, T. and Takei, T. (2003), ‘Shape –Similarity search of 3D models by using enhanced shape functions’, Proceedings of the Theory and Practice of Computer Graphics TPCG’03, pp. 97-104, Washington D.C., USA.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin, D. (2002), ‘Shape distributions’, ACM Transactions on Graphics, Vol. 21, No. 4, pp. 807-832.

Osada, R., Funkhouser, T., Chazelle, B. and Dobkin,D. (2001), ‘Matching 3D models with shape distributions’, SMI 2001International Conference on Shape Modeling and Applications, pp. 154-166,Genova, Italy.

Patrick, A. (2005), ‘Development of a 3D model of the human arm for real time interaction and muscle activation prediction’, masters thesis, The University of Iowa, December 2005.

Pelossof, R., Miller, A., Allen, P. and Jebara, T. (2004), ‘ An SVM learning approach to robotic grasping’, proceedings of the IEEE International Conference on Robotics and Automation, Vol. 4, pp. 3512-3518.

Pirrone,R., Chella, A., (2003), ‘A Neural architecture for segmentation and modelling of range data’, the 8th Congress of the Italian Association for Artificial Intelligence, September 2003, Pisa, Italy.

Pitarch, E., Yang, J. and Abdel-Malek, K. (2005), ‘SANTOS™ hand: a 25 degree-of-freedom model’, proceedings of the SAE Digital Human Modeling for Design and Engineering, Iowa, USA.

Pollard, N. (1994), ‘Parallel methods for synthesizing whole hand grasps from generalized prototypes’, PhD thesis, Department of Electrical Engineering and Computer Science, MIT, 1994.

Pollard, N. (2004), ‘Closure and Quality Equivalence for Efficient Synthesis of Grasps from Examples’, the International Journal of Robotics Research, Vol. 23, No. 6, pp. 595-613.

Rijpkema, H., Girard, M., (1991), ‘Computer Animation of Knowledge-based Grasping’, proceedings of the ACM.SIGGRAPH‘91, Vol.25, No.4, pp.339-348.

Salisbury, J. (1982), ‘Kinematic and force analysis of articulated hands’, PhD thesis, Department of Mechanical Engineering, Stanford University, 1982.

Sanso,R. and Thalmann,D., (1994) ‘A hand control and automatic grasping system for synthetic actors’, Eurographics '94, pp.167-178, Oslo, Norway.

Taha, Z., Brown, R. and Wright, D. (1997), ‘Modeling and simulation of the hand grasping using neural networks’, Medical Engineering and Physics, Vol. 19, No. 6, pp. 536-538.

Tomovic, R., Bekey, G. and Karplus, W. (1987), 'A strategy for grasp synthesis with multifingered robot hands', proceedings of the IEEE International Conference on Robotics and Automation, Vol. 4 pp. 83-89.

Toth, E. (1999), 'Stable object grasping with dexterous hand in three dimension', Periodica Polytechnica SER EL. ENG., Vol.43, No. 3, pp. 207-214.

Tubiana, R (1981), 'The hand', Vol. 1, W.B. Sanders Company, 1981.

Tubiana, R., Thomine, J. and Mackin, E. (1996), 'Examination of the hand and wrist', Second Edition, Martin Dunitz, 1996.

Wang, B., Jiang, L., Li, J., Cai, H. and Liu, H. (2005), 'Grasping unknown objects based on 3D model reconstruction', proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, California, USA.

Yang, J., Marler, R.T., Kim, H., Arora, J. and Abdel-Malek, K. (2004) 'Multi-objective optimisation for upper body posture prediction', 10th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference, Albany, NY.

Yang, J., Pena-Petarch, E., Kim, J. and Abdel-Malek, K. (2006), 'Posture prediction and force/torque analysis for human hands', The 2006 SAE DHM Conference, Lyon, France.

Zhang, Y., Koschan, A. and M. Abidi (2003) 'Superquadrics based 3D object representation of automotive parts utilizing part decomposition', proceedings of SPIE 6th International Conference on Quality Control by Artificial Vision, Vol. 5132 pp. 241-251, Tennessee, USA.