We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS

BOOK CITATION INDEX

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

4

Virtual Human Hand:
Autonomous Grasping Strategy

Esteban Peña Pitarch
Uviversitat Politecnica de Catalynia
Spain

1. Introduction

Several researchers have spent time and effort studying grasping under different points of view, grasping objects in virtual environments (VE) is one of them. Now that the grasping strategy in VE has been solved, the next step is grasping any object in a VE, with minimum interaction with the user. In this chapter, we develop a new strategy for autonomous grasping in a virtual environment, based on the knowledge of a few object attributes like size, task, and shape. When the object is input a VE, the user chooses the object from among others, chooses a task inherent to the object selected, and then we implement a semi-intelligence algorithm, which makes a decision about how to grasp the selected object. When the system makes a decision, it determines whether the object is in the workspace of the hand. If it is then grasps, if is not, the virtual human (VH) moves to closer to the object so that it is now graspable. Simulation in VE is becoming more relevant every day and becoming a tool for designing new products. In the grasping area, grasping objects in a VE is commonly used by several researchers, but they don't pay attention to autonomous grasp. Autonomous grasp is an interesting problem, and its application in VE can help teach grasping to people with some diseases like ictus or those with amputations. It can also be used in robotics to teach the robotics hand to grasp.

2. Basic Concepts

For basic concepts, we introduce definitions from the literature Kapandji (1996), Tubiana (1981), Tubiana et al. (1996).

Free Motion In free motion, the hand moves freely in space. The basic free motions possible for the hand are:

- *Opening*: Fully extending the fingers and the thumb until the hand is fully open, as in the anatomical position.

- *Closing*: Fully flexing until the hand is closed in a fist with the thumb overlapping the index and middle fingers.

- *Clawing*: The motion that reaches the terminal position of metacarpophalangeal (MP) extension, interphalangeal (IP) flexion.

- *Reciprocal*: The motion that reaches a terminal position of MP flexion, IP extension.

Twelve variations of these motions are observed if the terminal position of each motion is the starting point of each other motion.

Resisted Motion Resisted motion is that performed by the hand against an external resistance, for the purpose of exerting force on an external object and sometimes changing its position.

- *Power grip*: Gripping an object against the palm (primarily isometric motion).

- *Precision handling*: Manipulation of an object by the thumb and fingers, not in contact with the palm (primarily isometric motion).

- *Pinch*: Isometric compression between the thumb and fingers.

3. Grasping Parameters

Before grasping the object, the VH needs do some actions first. We classify these actions as pre-grasp, grasp, and after-grasp. To concentrate only on grasp, we consider the object and the VH in position, meaning that the object is in the workspace of the hand. Pre-grasp actions are:

- *Hand Position:* The position of the wrist respect to the object.

- *Hand Orientation:* The hand has adequate orientation for the grasp.

These two pre-grasp actions need to work together. If we consider each separately, a good position with a bad orientation, results in an inability to grasp the object, and good orientation with a bad position results in the object not being reachable. In our algorithm both actions are doing in same time.

In this dissertation, after-grasp actions are considered in terms of the final position of the object.

Grasp involves operations related to the hand and the object, and we consider grasp touch, pull, push, etc. If we take the definition from a dictionary, one of the definitions is to take hold of or seize firmly with the hand. We extend this definition to include touch, pull, etc., ultimately amplifying the concept of grasp.

Parameters for the object and the hand are considered in the new step called grasp.

- *Object Attributes:* In the virtual environment, the object was built with techniques of computer-aided geometric design, and the basic attributes there are known. Other attributes, such as temperature, are described below.

- *Hand Orientation:* Hand orientation is related to hand shape. An adequate lecture of object attributes can give all possible options for grasp, with axes, planes, etc. Hand surface and object surface can give the hand orientation relative to the object.

- *Hand Position:* Hand position is similar to hand orientation and follows the same procedures for positioning the wrist as in the function of the object attributes.

- *Task:* This parameter can help the virtual human decide how to grasp the object.

- *Object Initial Position :* In some operations we need to know the initial position.

- *Object Final Position :* In some operations we need to know the final position.

- *One or Two Hands :* This parameter is related to several of the attributes discussed above.

- *Finger Number :* The number of fingers to use, depending on the type of grasp, object shape, etc.

- *Object Weight :* Object weight can be derived from the object shape if we know the density.
- *Object Stability:* For any small movement close to the position of equilibrium, the object stays in the equilibrium position.
- *Hand Anthropometry:* Virtual humans like real humans, have different-sized hands.

4. Relationship between Grasping Parameters

In this section, we establish the relationship between parameters and demonstrate that they need work together sometimes. To show this relationship, we present Figure 1. This figure shows the relationship independent of the dominant hand; for our VH, the dominant hand is usually the right hand.

Fig. 1. Relation between grasping parameters

In this figure, fill arrows indicate functions between parameters that connect, arrows with dashed lines show relationships, the elliptic shapes are for the object, and rectangular shapes are for the hand. Hand orientation and hand position are related and closed with a dashed rectangle; both parameters work together, and we classify both as hand initial posture.
Hand initial position is a direct function with object initial position; they each depend one the other, e.g. in order to perform some action with the object, knowledge of the hand becomes necessary and with this first approximation we can know if the object is reachable or not. These parameters are related to object attributes, i.e., the object attributes permit different actions and depend on the hand initial position. In a similar way, the task can be done in relation with the hand initial position, i.e., if the object is not in the workspace, the task cannot be done.

The object weight, also relates to the object attributes, in the virtual environment, if we know the geometry and the density of the object we can know the weight with the geometric relation $W = V_0.\gamma$ in absolute terms, where W is the weight, V_0 is the volume, and γ is the specific weight .

The number of fingers and hand anthropometry are related, and the hands are used during the action too.

Object stability is a function of the object shape; a tall glass is less stable than a short glass when it is sitting on a table.

When the action is to do some particular task, the action is related directly to whether one or two hands are used or how many fingers are used. In the section below, we describe some tasks in which we can see this relation.

5. Parameters and Relationship

5.1 Object Attributes

Figure 2, shows the object attributes from engineering design. Attributes inherent to the object are volume, mass, inertia center, and inertia matrix. Some commercial programs call these characteristics, but we call them attributes. Inertia center is the center of mass (COM). Other attributes are:

- Temperature
- Fragility
- Surface shape
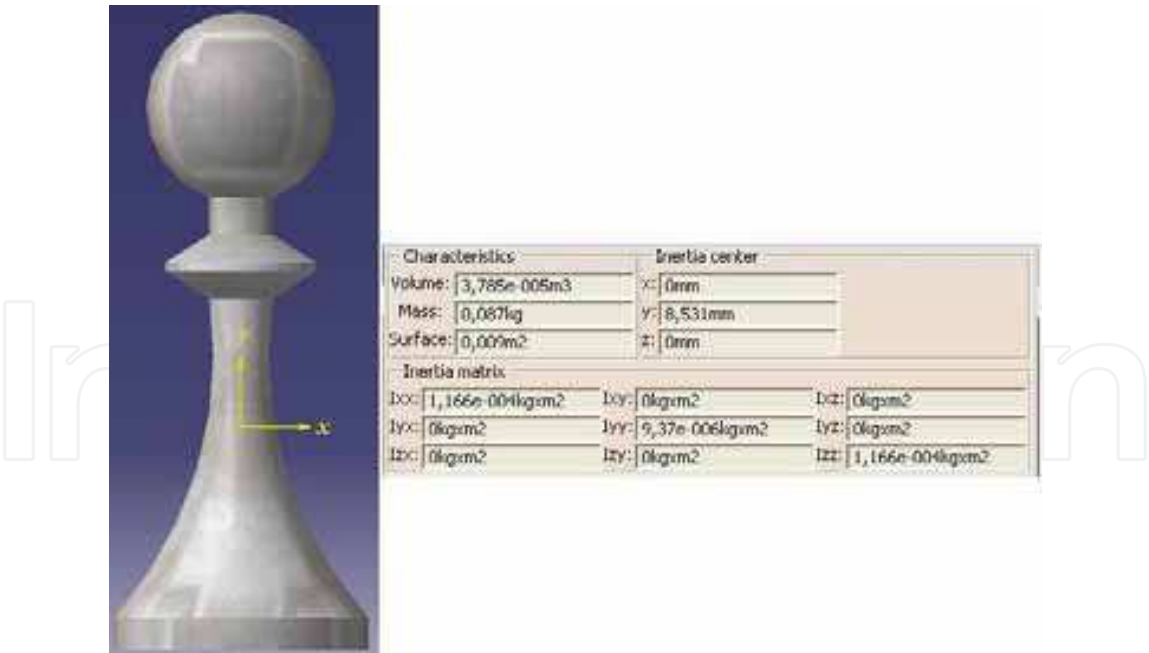
Fig. 2. Object with some attributes information

Temperature: When the objects have a temperature, this attribute can help decide what type of grasp is required.. For example, when grasping a mug, if the object are is filled with hot coffee and the action is to move we do not can grasp the side.

Fragile: If the object is fragile, this attribute can help decide what type of grasp is required.
Surface Shape: In a virtual environment, when we use the B-rep form found in the references Hoschek & Lasser (1993), Farin et al. (2002), Zeid (2005), and Wang & Wang (1986) , we show the definition of the object shape. In domain M in the plane with parameters (u,v), there is continuously differentiable and locally injective mapping $M \rightarrow S$, which takes points (u,v) in M into $\Re^3$. Then every point in the image set S can be described by a vector function $\mathbf{X}^i(u,v)$, where $i = 1,\ldots,n$ and n is the number of object, and represents the object i selected by the user. $\mathbf{X}^i(u,v)$ is called the *parametrization* of the surface S for the object i, and u,v are called the parameters of this representation; see Figure 3.
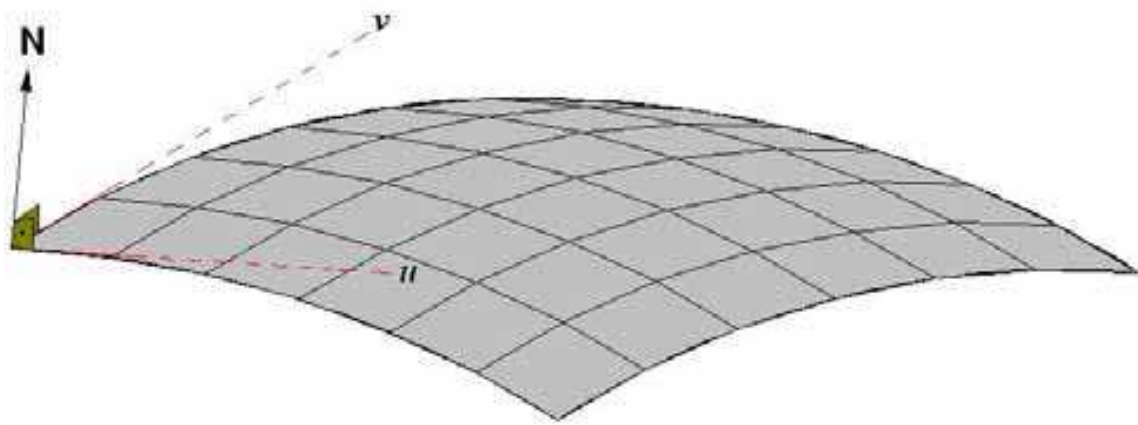
Fig. 3. Parametric surface

A parametrization is *regular* if for every point of S, the normal vector is defined:

$$\left| \frac{\partial \mathbf{X}^i}{\partial u} \times \frac{\partial \mathbf{X}^i}{\partial v} \right| \neq 0 \tag{1}$$

If $\left| \frac{\partial \mathbf{X}^i}{\partial u} \times \frac{\partial \mathbf{X}^i}{\partial v} \right|$ has a zero at the point $\mathbf{P}(u_0,v_0)$, then the surface has a *singularity* at $\mathbf{P}$.
The *tangent vector* to a surface curve $\mathbf{X}^i(u(t),v(t))$ can be computed as

$$\dot{\mathbf{X}}^i = \frac{\partial \mathbf{X}^i}{\partial u} \dot{u} + \frac{\partial \mathbf{X}^i}{\partial v} \dot{v} \tag{2}$$

In particular, the tangents to the parametric curves are given by

$$\mathbf{X}^i_u = \frac{\partial \mathbf{X}^i}{\partial u} \ (lines \ v = const.), \quad \mathbf{X}^i_v = \frac{\partial \mathbf{X}^i}{\partial v} \ (lines \ u = const.) \tag{3}$$

The two vectors $\frac{\partial \mathbf{X}^i}{\partial u}$ and $\frac{\partial \mathbf{X}^i}{\partial v}$ uniquely determine the plane, which is tangent to the surface at the point $\mathbf{P}(u,v)$.
The *unit normal vector* to the surface can compute for the object, using the vector product, as

$$\mathbf{N_o} = \frac{\mathbf{X}^i_u \times \mathbf{X}^i_v}{|\mathbf{X}^i_u \times \mathbf{X}^i_v|} \tag{4}$$

Usually we assume that the hand size is defined by two parameters, $HL = 190 \ mm$ (Hand Length) and $HB = 90 \ mm$ (Hand Breadth). With these two parameters and the object size, we

can know if the virtual human can grasp the object as a whole or if it needs to search for parts of the object for grasp, i.e., one cannot grasp a whole chair, but we can grasp the back of the chair.

5.2 Hand Orientation

For hand orientation, we use the theory for two oriented surfaces. In this case the hand surface is oriented with the object surface, and the hand can be the right, left, or both hands. For simplify, we refer in this case to the right hand; the same equations can apply for the left hand or both hands. One surface M_H, where the subscript H indicates the hand surface, is orientable if the mapping of $M_H \rightarrow S^2$ is *regular*, and the vector normal is defined in Equation 4. If we change to other coordinates (u_1, v_1), shown for the hand in Figure 4, the new parametric representation is

$$\mathbf{Y}_{u_1}^{j} \times \mathbf{Y}_{v_1}^{j} = \frac{\partial(u,v)}{\partial(u_1,v_1)} \mathbf{X}_u^i \times \mathbf{X}_v^i \tag{5}$$

where $j = 1,2$, 1 for the right hand, 2 for the left hand, and for the corresponding unit vector:

$$\mathbf{N}_H = \epsilon \mathbf{N}_o \tag{6}$$

where

$$\epsilon = sign \left| \frac{\partial(u,v)}{\partial(u_1,v_1)} \right| \tag{7}$$

The orientation in the new parametrization is the same if the **Jacobian** of the transformation is positive, and is opposed if it is negative.
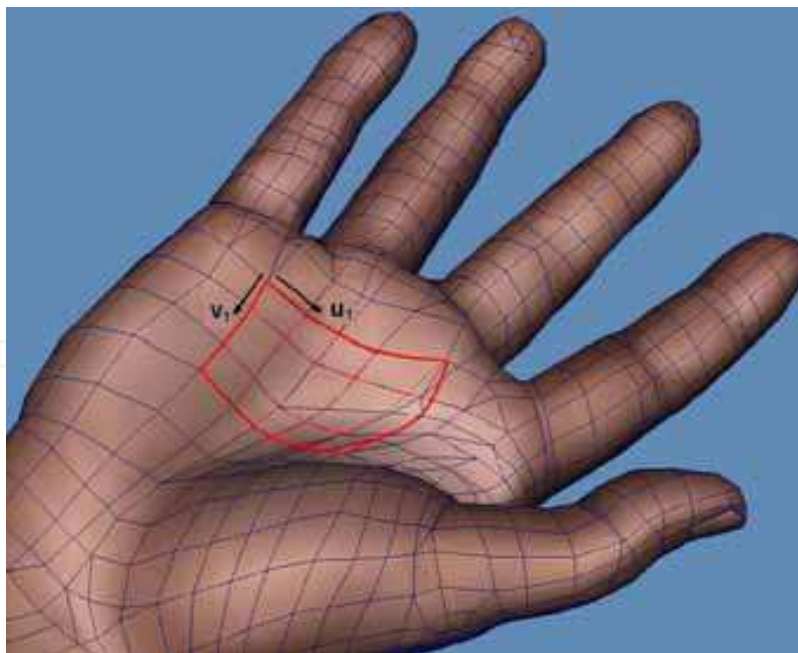
Fig. 4. Parametric surface for the hand

5.3 Hand Position

To consider the hand position with respect to the object, we refer this position with respect to the wrist. For us the global position for the hand is given by the position of the wrist. The point of reference for the object is the COM.

When the object enters the virtual environment, we know the position of COM. That is information inherent with the object; the position of the wrist is also known in each moment.

The coordinates of COM are $\mathbf{x}_c^i = [x_c^i \ y_c^i \ z_c^i]^T$, and the coordinates of the wrist are $\mathbf{x}_w^j = [x_w^j \ y_w^j \ z_w^j]^T$. We can locate the hand (wrist) with respect to the object with the linear transformation:

$$\mathbf{x}_w^j = A\mathbf{x}_c^i \qquad (8)$$

where A is a transformation matrix. Figure 5 shows the wrist position with respect to the center of mass in pre-grasp action.
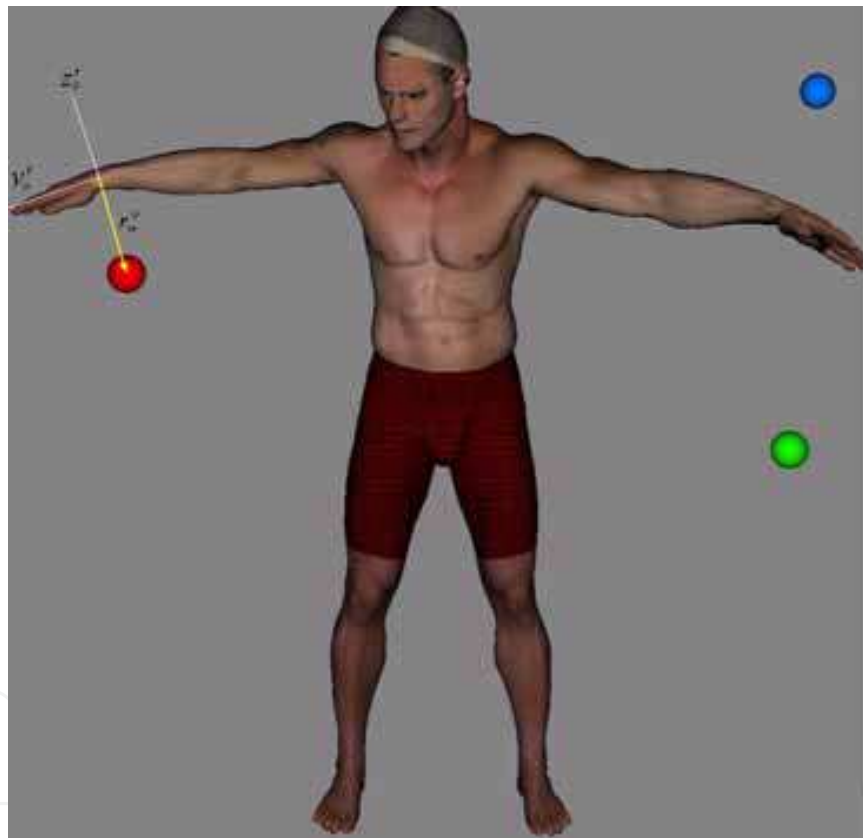
Fig. 5. Hand position respect the object

5.4 Task

The task is the most important attribute, and many times it decides for itself how to grasp the object; i.e., when grasping a mug containing drink, the usual response is to grasp the handle; when moving the mug, we can grasp the handle or the top. The *task* in the dictionary (Oxfort English Dictionary) is *a piece of work assigned or done as part of one's duties.* Follow this definition and the definition of task analysis.

Task Analysis: Task analysis is the analysis or a breakdown of exactly how a task is accomplished, such as what sub-tasks are required.

We divide tasks and sub-tasks into elemental actions, i.e., open the door is a task, and the elemental action is pulling or push. This elemental action can also be used for other tasks, like moving a joystick in a machine.

5.4.1 Elementary Actions

Pull: To pull is to apply force so as to cause or tend to cause motion toward the source of the force. This action can be done with one, two, three, or four fingers. We define this as PL_i, where $i = 1\ldots4$ and PL_1 means pulling with one finger and so on.

Push: To push is to apply pressure against for the purpose of moving. The number of fingers used is the similar as for pulling, but we add the use of the palm. We define this as PS_i, where $i = 1\ldots5$ and PS_3 is pushing with three fingers.

Pinch: A pinch is isometric compression between the thumb and the fingers. The number of fingers to used is defined as PI_i, where $i = 1\ldots4$ and PI_2 is pinching one object with the thumb and two fingers.

Power Grasp: A power grasp is the gripping of an object against the palm. We define this action as PG.

Precision Handling: Precision handling is the manipulation of an object by thumb and fingers, not in contact with the palm. We define this action as PH_i, where $i = 1\ldots4$ and PH_4 is precision handling with the thumb and four fingers.

Touch: To touch is to cause or permit a part of the body, especially the hand or fingers, to come into contact with so as to feel. We consider this action is transformed with the index. We define TO for touch.

5.5 Object Initial Position

The point of reference for object initial position is the center of mass (COM) of the object; this point is independent of the type of grasp, but the wrist position is referred to this point. In Figure 5 we can see the object position with respect to the wrist. This position in the global coordinates of the virtual environment (VE) tells the system where it is with respect to the global coordinates of the virtual human. Object initial position is defined by:

$$\mathbf{x}_c^i = [x_c^i \; y_c^i \; z_c^i]^T \tag{9}$$

5.6 Object Final Position

Knowledge of final position permits us to know a priori if the virtual human can do the task; if it cannot do the task, we need add some actions like walking, advancing the body, etc. To tell the system to add actions, a check of the workspace helps us determine the reachability of the action. In some tasks, i.e., moving a joystick in a machine to elevate the load, the object is fixed in the base and only moves with the constraints of spherical joints; the final position is always the same as the initial position.

We can use equations similar to those used for determining hand position to determine object final position. We know the object initial position relative to COM $\mathbf{x}_c^i = [x_c^i \; y_c^i \; z_c^i]^T$, and we know the object final position relative to COM $\mathbf{x}_f^i = [x_f^i \; y_f^i \; z_f^i]^T$, both with respect to the global coordinates. The relationship between the initial and final position is:

$$\mathbf{x}_f^i = B\mathbf{x}_c^i \tag{10}$$

where B is a transformation matrix.

5.7 One or Two Hands
This parameter is a function of the shape of object. Shape provides information about the size and the weight of object, which determine if one hand or both hands are used or if the action is not performed.

5.8 Number of Fingers
For touching, a virtual human only needs one finger, but for precision handling, the virtual human may need one, two, three, four, or five fingers for grasp. This parameters is a function of the shape and weight of the object.

5.9 Stability
Analysis of stability for some objects helps us to determine whether or not the object in a particular position can be touched. An example is a tall bottle. If the virtual human touch the top of the bottle and it is not stable the botlle can lose stability and fall. Falling not is the final reaction when we touch the object.

From the statics equilibrium, we can calculate if the object's position is stable, unstable, or neutral equilibrium; this is interesting to know for the actions of pulling, pinching, or touching. The general case with one degree of freedom for the object, defined by independent coordinate s, is

$$\frac{dV}{ds} = 0, \ \frac{d^2V}{ds^2} > 0 \ stable \ equilibrium$$

$$\frac{dV}{ds} = 0, \ \frac{d^2V}{ds^2} < 0 \ unstable \ equilibrium$$

$$\frac{dV}{ds} = \frac{d^2V}{ds^2} = \frac{d^3V}{ds^3} = \cdots = 0 \ neutral \ equilibrium$$

where V is the potential energy of the object.

5.10 Hand Anthropometry
We give the hand anthropometry for a 95th-percentile human hand related to the hand length and breadth. This parametric length for each bone permit the simulation of almost all the population.

6. Mathematical Model
We have defined task in elemental actions, and these actions are related to some parameters. In this section, we write each elemental action and show the equations in the function of parameters.

$$Pull(OS, HO, HP, T, OIP, OTH, FN, ST, HA)$$

$$Push(HO, HP, T, OIP, OFP, OTH, FN, ST, HA)$$

$$Pinch(OS, HO, HP, T, OIP, OFP, FN, HA)$$

$$Power \ grasp(OS, HO, HP, T, OTH, FN, HA)$$

$$Precision \ handling(OS, HO, HP, T, FN, ST, HA)$$

where:

OS= Object Shape, and the mathematical function is $\mathbf{X}^i(u,v)$

HO= Hand Orientation, $\mathbf{Y}^j(u_1,v_1)$

HP= Hand Position, $\mathbf{x}^j_w = A\mathbf{x}^i_c$

T= Task, user chooses the task

OIP= Object Initial Position, $\mathbf{x}^i_c = [x^i_c \; y^i_c \; z^i_c]^T$

OFP= Object Final Position, $\mathbf{x}^i_f = [x^i_f \; y^i_f \; z^i_f]^T$

OTH= One or Two Hands, a function of OS and represented by, $OTH = f(OS)$

FN= Finger Number, $FN = f(OS)$

ST= Object Stability, dV

HA= Hand Anthropometry, HL, HB

6.1 Pull

Pulling is a function of the object shape because in order to do this action we need to know a priori if is permitted to pull or not. For example, we can pull a door even though it is a big object, but we cannot pull a wall even though it is a similar object shape. The shape of the object can give us additional information, such as which parts of the object to pull.

It is also a function of hand orientation and hand position that if the object is outside the workspace of the hand, the virtual human cannot do this action. However, if the virtual human moves, he can reach the object.

As discussed earlier, task is the most important, because it can determine whether or not the action is pulling.

Object initial position is related to the workspace and hand position, i.e., object and hand position are two parameters that work together to define the object reachability.

When pulling, humans use one or two hands; for realistic actions, virtual humans use similar behavior and sometimes pull a door using two hands and pull a joystick using only one hand. In similar way, we can use a determinate number of fingers; for pulling a door with two hands, the virtual human uses all the fingers, and for pulling a joystick and two fingers are sufficient.

Object stability helps determine if a small variation in the equilibrium position causes the object to lose the equilibrium; or not, e.g., pulling a 1.5-liter bottle of water by the top can cause the bottle to fall, and this is not the final position that the virtual environment is looking for.

Hand anthropometry helps to pull an object and can tell the system which part to pull, e.g., for a door, the handle.

6.2 Push

Knowledge of hand orientation, hand position, task, and object initial position are similar for the action of pull.

If we know object final position in pushing, we can see if this action can completed without additional actions, like walking or moving the virtual human.

The other parameters considerations for pulling apply here.

Fig. 6. Lateral pinch

6.3 Pinch

Let me explain why pinching is function of the parameters shown above. Picture 6 shows a lateral pinch for a key; the virtual human is using the thumb and the index finger laterally.

To pinch a key, is the object shape was analyzed first, then the hand orientation and hand position with respect to the object and then the object initial position and final position were considered. If the action were putting the key in a lock and rotating it ninety degrees to open or close the door, the final position for the object is the lock. Finger number in this case, following our definition above, is PL_1 because only one finger and the thumb is used and pinching always uses the thumb. Finally, the hand anthropometry was considered because in some cases the object with respect to the hand is too big for pinching, in this case, the object and the hand are in good proportion.

6.4 Power Grasp

In the object shape function, if the object is so big compared to the hand anthropometry that the virtual human cannot grasp the whole object, it can look for a part to grasp, e.g., a frying pan was designed for power grasping the handle. Shape can tell us the possibles parts for grasping.

All the other parameters have a similar role, which we describe above for the other basic functions.

6.5 Precision Handling

Precision handling is similar to power grasping with respect to the function parameters. The only remarkable difference is that in this case normally we only use one hand, and do not use the palm. In the function of the action relative to the object, the virtual human can use one, two, or more fingers. At this time the number of fingers to use is related to the object shape and the task to perform.

7. Grasping Strategy

We have defined all the parameters that work in grasping, and in this section we present the process of grasping. Figure 7, presents a flowchart of grasping. The objects are in the virtual environment, and the user chooses one object from among several. Each object has attached

information about several attributes, like task, shape, position of center of mass, etc.; these attributes help the system make a decision. Some objects can do different tasks, e.g., if the object is a mug, the task can be moving or drinking. Therefore, in this first approximation the user chooses the task inherent to the object.

Once the user chooses a task, the system helps make a decision about how to grasp because the system knows the attributes of the object and the task. But maybe the object is not in the workspace of the virtual human; if this is true, we need to tell the virtual human approximate- lyhow to put the object in the workspace of the hand.
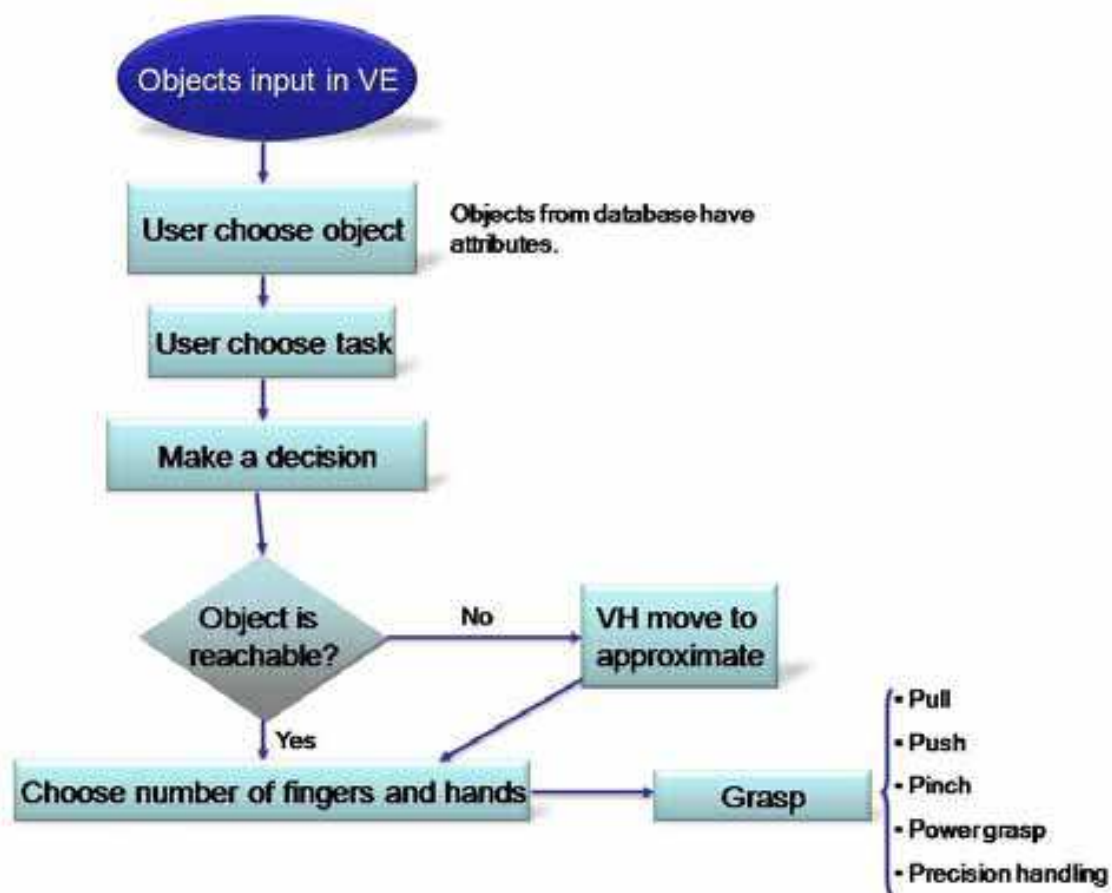
Fig. 7. Grasping flowchart

With the types of grasp and the shape of the object, the next step is to calculate the number of fingers and hands to use for grasping the object, and then grasp.

This is a heuristic definition for grasping. In the next section, we explain the semi algorithm for grasping the object.

7.1 Objects Input in the Virtual Environment

When the objects are in the virtual environment and the user choose one object to grasp, the system reads all of the attributes and saves them in a file with the extension "*.txt", which allows the file to be used in any code of language. Figure 2 shows an object with attributes like center of mass (COM) or, in this case, Inertia center, volume, mass, characteristics, and Inertia matrix.

7.2 User Chooses Object

This action occurs when there are many objects in the virtual environment; if virtual environment has only one object, the virtual human will know the object. One example is that the virtual human is taking a breakfast, and the objects on the table (virtual environment) are a coffee mug, cereal bowl, and spoon. The user chooses a coffee mug, and this inherently has all the properties with the center of mass, of course can change position of center of mass is function of quantity of coffee. For this the object, there are two reasonable tasks, drinking or moving. If the task chosen by the user is drinking, once the virtual human has drunk, it will put the coffee mug back on the table in the same position. During this process, the position of the wrist of the virtual human is known, the position of the center of mass is known, and we can calculate the hand orientation for performing this action.

7.3 User Chooses Task

This action is mentioned above, and the preceding section has an example of the task and how the user chooses.

7.4 Making a Decision

Inputs in a simple associator are task, weight, shape, temperature, etc., and output pattern is type of grasp, i.e. pinch, pull, etc. When VH take a decision there is based in a system linearly separable.
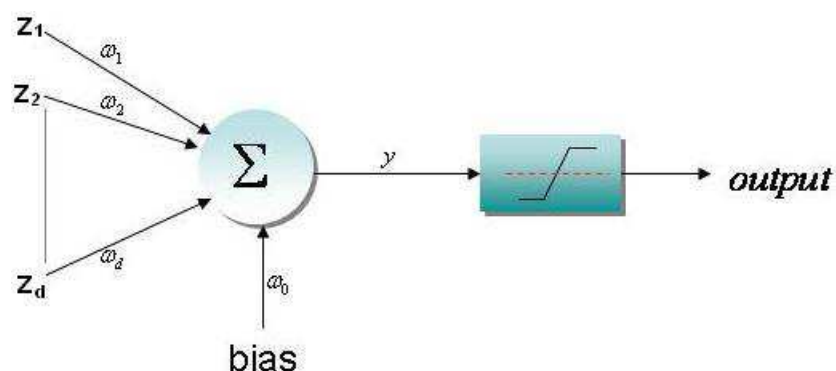
Fig. 8. Single perceptron

Figure 8 show a single perceptron from the works Hagan et al. (1996), Haykin (1999),Anderson (1997) where the input vector is $\mathbf{z} = [z_1, z_2, \ldots, z_d]^T$, bias is b, the weight vector considering each input is $\mathbf{w} = [\omega_1, \omega_2, \ldots, \omega_d]^T$, and the weight for the bias is ω_0.
In our case we choose apply these networks for a classification problems, in which the inputs are binary images of attributes like task, shape, temperature, and density. The output of the perceptron is given by

$$\mathbf{y} = g(\sum_{j=1}^{d} \omega_j \phi_j(z) + \omega_0) = g(\mathbf{z}^T \mathbf{\Phi}) \tag{11}$$

where $\mathbf{\Phi}$ denotes the vector formed from the activations $\phi_0, \ldots, \phi_d$, and the function of activation in this case is a symmetric saturating linear (satlins), because this are a linearly separable. The input/output relation is:

$$g(a) = -1 \qquad a < -1$$
$$g(a) = a \qquad -1 \le a \le 1$$
$$g(a) = 1 \qquad n > 1$$

7.5 Determining Whether Object is Reachable and Approximating

The answer to this question can be found in Chapter four, where the analysis of workspace for the hand was studied. When checking the workspace for each finger with respect to the wrist position, the virtual human can know if the object with respect to the wrist position is reachable or not. If the object not is reachable, the virtual human can perform some actions to approximate, like walk and reduce the space to a convenient distance between the wrist and the center of mass of the object.

7.6 Choosing Number of Fingers and Hands

Based on object shape and hand anthropometry, we can determine if the use of one hand is adequate or if the use of two hands is required. Based on the geometric primitives for a regular basketball and one person with a 185 mm hand length, the ball cannot be grasped with one hand; two hands are required. Some objects are designed to grasp with both hands. Normally, if the object to be grasped is based on the primitive geometries for a sphere, cylinder, or box, if the dimensions for grasp are greater than the hand length, it automatically requires two hands.

$$If\ D > HL,\ then\ use\ two\ hands$$

where D is the side dimensions to grasp, and HL is the hand length.

To determine the number of fingers to use for a grasp, parameters like object shape, hand anthropometry, type of grasp, and task are used . We define elemental actions, and each elemental action shows the number of fingers to use. Power grasp and touch do not need the number of fingers defined; normally a power grasp uses five fingers and touch uses the index finger. Other grasp types are a function of parameters mentioned earlier. For precision handling, a primitive sphere is a function of the shape, or better if the radius of equator of sphere is ρ we can define the number of fingers how:

$$If\ 1 <= \rho <= 20\ mm\ then\ PL_1$$

$$If\ 20 <= \rho <= 40\ mm\ then\ PL_2$$

$$If\ 40 <= \rho <= 60\ mm\ then\ PL_3$$

$$If\ 60 <= \rho <= 90\ mm\ then\ PL_4$$

where the subscript 2 means the thumb and two fingers, and subscript 3 means thumb and three fingers, and so on.

We can use the same relationship for pinching. Pulling and pushing are also a function of size of the object; normally, to push a door we use all the fingers, but sometimes if we know the door, we use two or three fingers. For this analysis, we always use five fingers, similar to pulling.

7.7 Grasp

The last action is grasping. Grasping is function of the parameters founded above and forward and inverse kinematics. For several reasons, we prefer to explain the action of grasping in the next chapter and work in several examples demonstrating how this theory works in a virtual human. The output of grasp in this flowchart is the type of grasp and how the grasp uses all the parameters shown in this chapter. Figure 9 shows a type of grasp based on the task and other parameters. The positions of fingers are determined as a function of all the parameters defined for grasping, i.e., when grasping a mug to drink from the side, the object in this position has a cylindric shape and the virtual human knows the COM. Based on this, we can calculate the hand position and orientation and the position for all the fingers. In the case shown in Figure 9, the task is drinking, the beverage contained in the mug is hot, and the weight is light. In this case, the decision is to grasp for a handle, and the virtual environment knows the position of the handle. The handle has a known shape, and we can calculate the number and position of fingers.

Fig. 9. Grasping a mug

8. Grasp Examples

Our approximation for to grasping is based on the movement of fingers. There are two types of movements. For grasping with power, the movement described, each finger except the thumb, is circular. For the second, when grasping with precision, the movement of fingertips, including the thumb, approximates a circle. Based on these approximations, we can simulate all the human grasping proposed by Buchcholz et al. Buchcholz et al. (1992). Pinching is a particular case of grasping with precision. Pulling, pushing, and touching we are considered positioning finger problems. For the first approximation of power grasping we can apply forward kinematics and calculate all the angles for every finger. For a cylinder with radius ρ , Figure 10 depicts cross-section of the cylinder and the schematic phalanx bones. The angles θ and q_i are obtained from the geometry relationship. This example is considered a power.

Where angle q_j for each finger, and subscript is $j = II \ldots V$ where subindex II is for the index, III middle, IV ring, and V small. These angles are for the proximal phalanx respect to the metacarpal bones for each finger. Similar for q_k, where $k = II \ldots V$, the subindex mean the same fingers related before and the angles are between proximal phalanx to medial phalanx. For q_l, where subscript is $l = II \ldots V$ are referred to angles between middle phalanx and distal
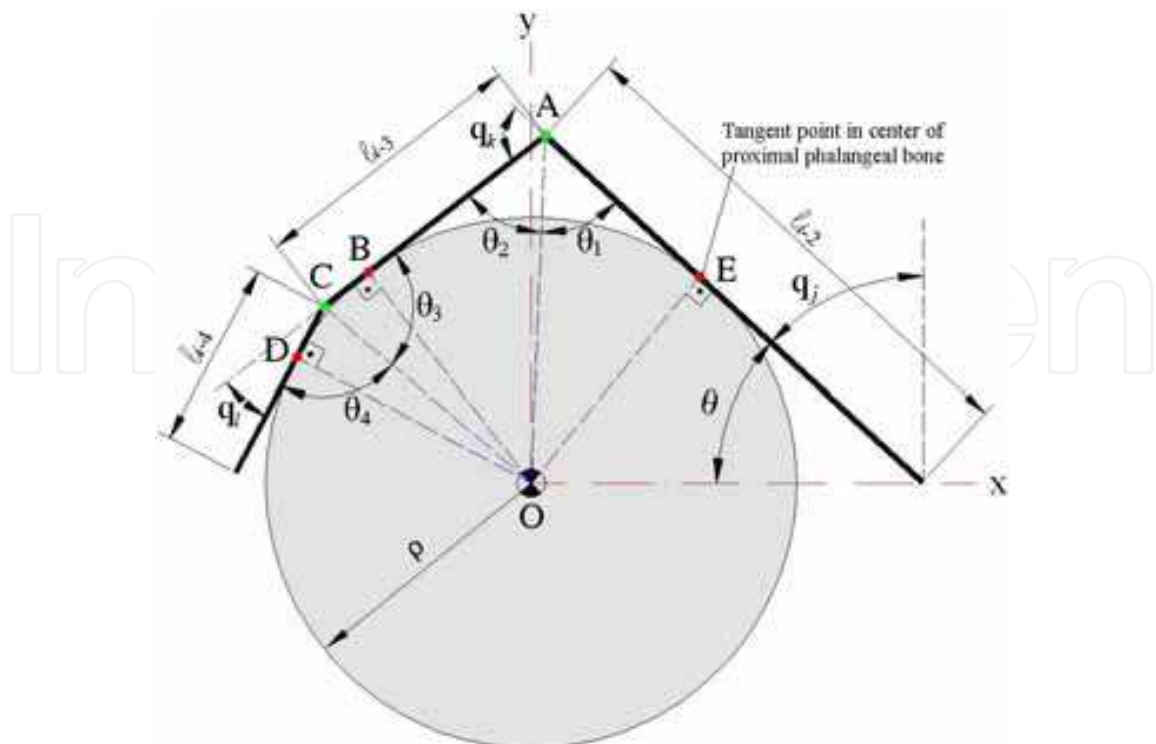
Fig. 10. The geometry relationship of finger segments

phalanx. All of these angles are calculated for geometry and changed from local to global with the transformation matrix developed in Appendix A.

For the second movements, we can calculate the fingertip position for each finger when grasp any object with precision and after applying inverse kinematics. Figures 11 and 12 depict the fingertip positions.

The angles β and α depend on the diameter of the ball. From the observation when people grasp a ball with radius $\rho = 27.5\ mm$, $\alpha = 0$ and $\beta = 60^{o}$. Also we impose that the middle finger stays in its neutral position. Therefore, we can determine the fingertip positions for the thumb, index, and ring fingers with respect to the wrist (global) coordinate system, and the small finger stays in the neutral position.

The inverse kinematic solutions depend on the initial values of the design variables for both iterative and optimization-based methods. Table 1 presents the solutions (q_i in degrees) for the index finger with the Newton-Raphson method, where the global coordinate is $[-11.22\ \ 152.341\ \ 77.4]$ in mm, the hand breadth is 200 mm, and the local coordinate is $[-7.3\ \ 59.9887\ \ 77.4]$ in mm.

From Table 1 it is shown that the convergence for Newton-Raphson method is very fast when the initial angles are close to the solution. For the first set of initial values, the solution for joint DIP (q_4) is negative and is in the range of motion. The negative angle for this joint represents hyper-extension. However, usually, we can observe that humans never grasp this sphere by DIP hyper-extension.

In practice, some joints in the fingers are coupled, i.e., the movement of one joint depends on the motion of another joint. For example, each finger except the thumb has two coupled joints. The distal interphalangeal joint (DIP) depends on the proximal interphalangeal joint (PIP) and
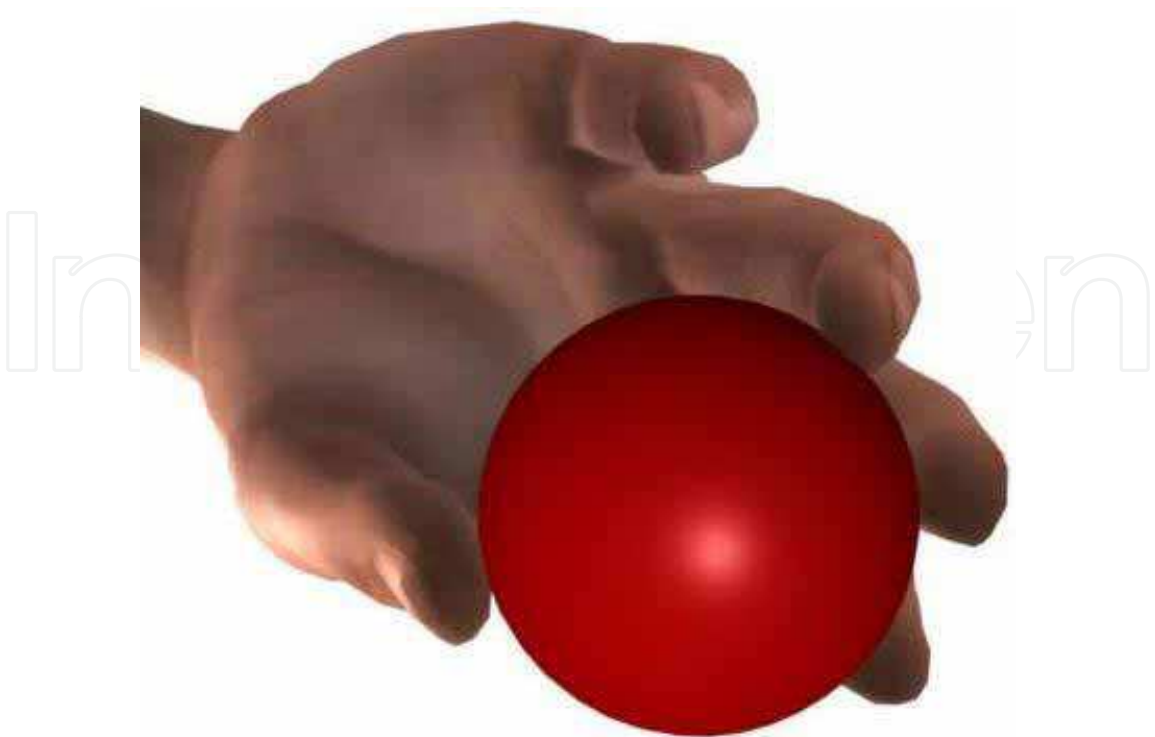
Fig. 11. Grasping a sphere

Initial	$q_1 = 0$	$q_2 = 30$	$q_3 = 30$	$q_4 = 10$
Iteration=7	$q_1 = 6.95$	$q_2 = 39.3$	$q_3 = 30$	$q_4 = -7.65$
Initial	$q_1 = 0$	$q_2 = 0$	$q_3 = 0$	$q_4 = 0$
Iteration=10	$q_1 = 6.95$	$q_2 = 42.3$	$q_3 = 10$	$q_4 = 26.6$
Initial	$q_1 = 0$	$q_2 = 10$	$q_3 = 10$	$q_4 = 0$
Iteration=7	$q_1 = 6.95$	$q_2 = 42.3$	$q_3 = 10$	$q_4 = 26.6$

Table 1. Index joint angles with Newton-Raphson method

the relationship is defined in the literature Rijpkema & Girard (1991), where the superscript i identifies the finger, beginning with the index finger and the last ending with the small finger.

$$q_{DIP}^i = \frac{2}{3} q_{PIP}^i \tag{12}$$

For the thumb, we can find a similar relationship:

$$q_3 = 2(q_2 - \frac{1}{6}\pi) \tag{13}$$

$$q_5 = \frac{7}{5} q_4 \tag{14}$$

The subindex shown above can found in Chapter 3.

When we impose the joint coupling function in Equation 12 to the index finger, it will have 3 DOF because q_4 is a function of q_3 . The result is shown in Table 2. The solution is the same for all different initial q_i values.
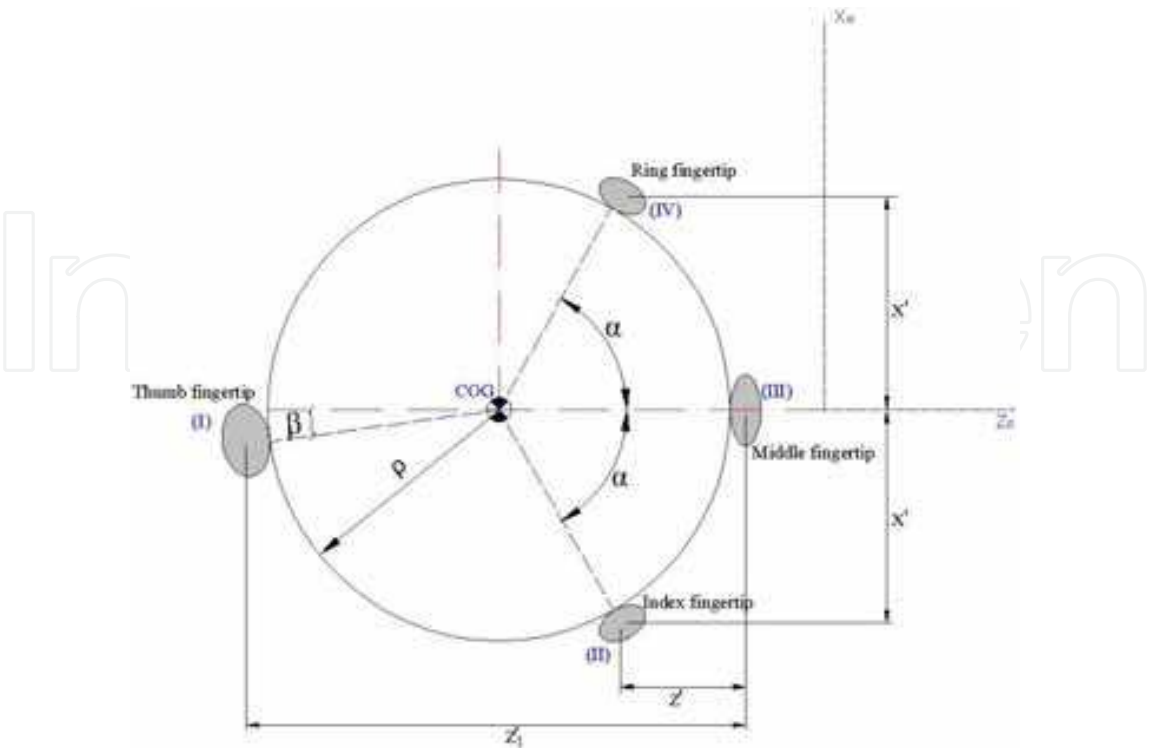
Fig. 12. Equator section with position of fingertips used

Initial	$q_1 = 0$	$q_2 = 0$	$q_3 = 0$	$q_4 = - - -$
Iteration=9	$q_1 = 6.95$	$q_2 = 39.3$	$q_3 = 20.13$	$q_4 = 13.42$
Initial	$q_1 = 0$	$q_2 = 30$	$q_3 = 30$	$q_4 = - - -$
Iteration=7	$q_1 = 6.95$	$q_2 = 39.3$	$q_3 = 20.13$	$q_4 = 13.42$
Initial	$q_1 = 0$	$q_2 = 10$	$q_3 = 10$	$q_4 = - - -$
Iteration=9	$q_1 = 6.95$	$q_2 = 39.3$	$q_3 = 20.13$	$q_4 = 13.42$
Initial	$q_1 = 0$	$q_2 = 40$	$q_3 = 40$	$q_4 = - - -$
Iteration=7	$q_1 = 6.95$	$q_2 = 39.3$	$q_3 = 20.13$	$q_4 = 13.42$

Table 2. Index joint angles with the Newton-Raphson method including joint coupling

Another approach for finger inverse kinematics is the optimization-based method. The hypothesis is that the feasible space is small for fingers during grasping. The problem will be: Given a point $\mathbf{x}^p$, find the joint angles $\mathbf{q}^i$ to minimize $\left\| \mathbf{p}^i - \mathbf{x}^p \right\|$, and the joint angles should lie in the limits as constraints. Therefore, the formulation is as follows

$$Find : \mathbf{q}_i$$

$$Minimize : \left\| \mathbf{p}^i - \mathbf{x}^p \right\|$$

$$Subject\ to : q_j^{iL} \leq q_j^i \leq q_j^{iU} \tag{15}$$

A gradient-based optimizer Gill et al. (2002) is implemented to solve this problem.

Table 3 shows the predicted joint angles for the optimization problem with different initial values. The results show that the optimization problem always has a feasible solution regardless of the initial values. It also shows that and they are the same for all different cases and the final solution is more reasonable than those obtained from the Newton-Raphson method.

Initial	$q_1 = 0$	$q_2 = 30$	$q_3 = 30$	$q_4 = 10$
Solution	$q_1 = 6.95$	$q_2 = 39.22$	$q_3 = 24.54$	$q_4 = 6.24$
Initial	$q_1 = 0$	$q_2 = 0$	$q_3 = 0$	$q_4 = 0$
Solution	$q_1 = 6.95$	$q_2 = 39.36$	$q_3 = 23.37$	$q_4 = 8.47$
Initial	$q_1 = 0$	$q_2 = 10$	$q_3 = 10$	$q_4 = 0$
Solution	$q_1 = 6.95$	$q_2 = 39.22$	$q_3 = 24.56$	$q_4 = 6.22$

Table 3. Index joint angles with the optimization-based method

9. Making a Decision

In this section we provide an example applying the Support Vector Machine (SVM) presented in the last chapter. The example shown here can be extended to any case, without losing generality. This example consists of putting an object in our virtual environment, in this case a mug. If there are more objects in the virtual environment, the user chooses the mug. Based on the functionality of mug, it has only two associated tasks, one is drinking and the other is moving. The user in this case chooses between these two tasks. In this case, to simplify the example, the known input parameters for this object are as follows, the others are 0:

$$Task \begin{cases} Drink = 1 \\ Move = -1 \end{cases}$$

$$Temperature \begin{cases} Hot = 1 \\ Normal = -1 \end{cases}$$

$$Weight \begin{cases} Heavy = 1 \\ Normal = -1 \end{cases}$$

We can add all the parameters, inherent to the object and described in Section 5.6, such a shape, hand orientation, number of fingers, etc. The problem become a classification problem between two classes.

$$y = hardlim\mathbf{g}(W^T \mathbf{p} + b)$$

If we choose $x = 0$, axis for classification and $b = 0$ the bias, the weight input vector is

$$W^T = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0].$$

The output, or decision, is to grasp the side or grasp the handle. In this case, we found a problem: if the mug is hot and the task is move, the virtual human cannot grasp the side of the mug. For this reason, we need to connect these outputs to another preceptron and build another output. Figure 13 shows the proposed network connected in parallel for the problem of output that is not consistent with the inputs. In this case, the perceptrons connected in parallel are $p = t - 1$, where t is the number of types of grasp inherent to the object in the function of the task.
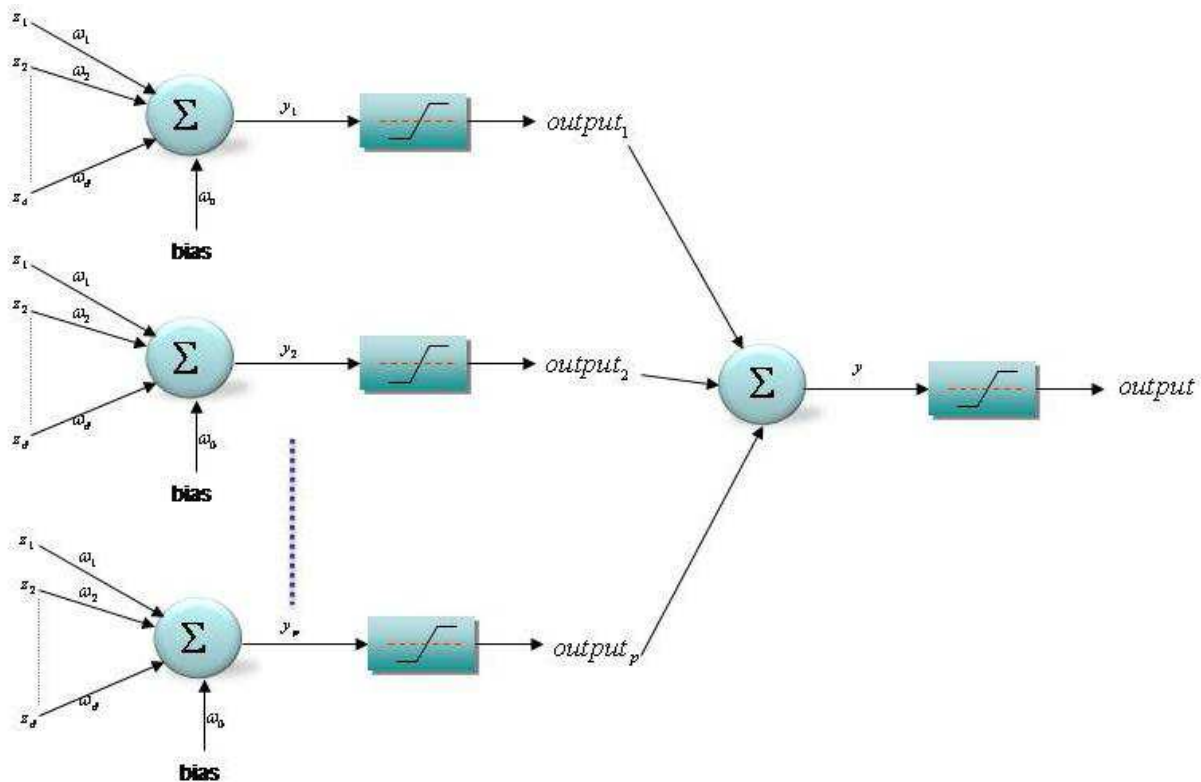
Fig. 13. Perceptrons connected in parallel

10. Choosing Number of Fingers and Hands

In this example, the parameters for grasping are any of the three ways to grasp a mug: by the side, top, or handle. The dimensions of the side to grasp are smaller than the hand length, and the virtual human uses only one hand. For the number of fingers to use, the parameter used for the first two types is $\rho = 80\ mm$. From Chapter 5, the number of fingers is PL_4, which means the thumb and four fingers. When the decision is to grasp the handle, the parameter is $\rho = 50\ mm$ and the finger number is PL_3, that is, index, middle, ring, and thumb. This grasp is shown in 8.

11. Grasping

The last action is grasping, once the user chooses the object among others in the virtual environment and the task, some objects only has designed for one task, or only there is one object in the virtual environment, if there is more than one choice. After, the algorithm take automatically makes a decision, choosing the number of fingers and hands, calculating the wrist position and orientation, and calculating the angles of every joint in our 25-DOF model. Wrists are connected with the upper body, and with this information $Santos^{TM}$ can grasp with his whole body in a realistic posture.

Figure 14 shows the results of applying the task-oriented object semi-intelligent system for grasping a mug. For this example, the three parameters drink, normal, and light, are choosen randomly, without lost generality. We can leave it so the user chooses the task. Once the decision is to grasp the handle, this is the result.

Fig. 14. Grasping a mug

Other types of grasp for different types of tasks and parameters for a mug are shown in Figures 15 and 16. Figure 15 makes a decision shows a grasp with power, and Figure 16 grasp shows grasping the top with precision.
Similar process and results are shown in Figures 17 and 18 for a joystick. In this case, there are only two tasks and the parameters are inherent to the joystick.

12. Conclusions

We have presented a novel theory for grasping based on the objects and their functionality. When the object is selected for the user, it is associated with more parameters, which we describe below. After the user chooses the task, the virtual human, if the object is feasible, grasps with the type of grasp calculated as a function of the mathematical model. The new concept in this chapter is that the virtual human can grasp autonomosly without the user once the task is chosen. Support Vector Machine (SVM) theory, for a perceptron, was applied for this autonomous grasp. The position and orientation of the hand or hands was calculated in reference to the center of mass of the object. Angles for each finger were calculated with the equations of forward and inverse kinematics for the novel 25-DOF hand model.
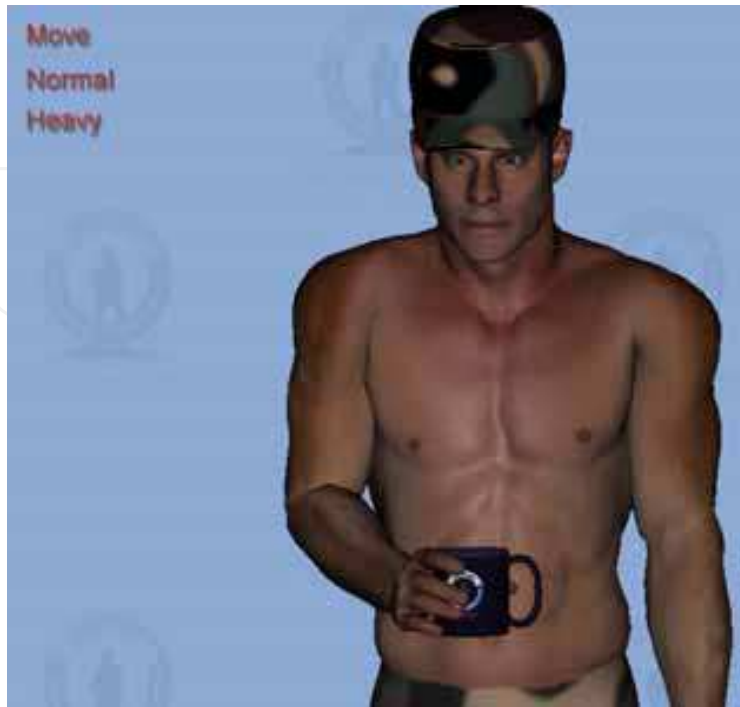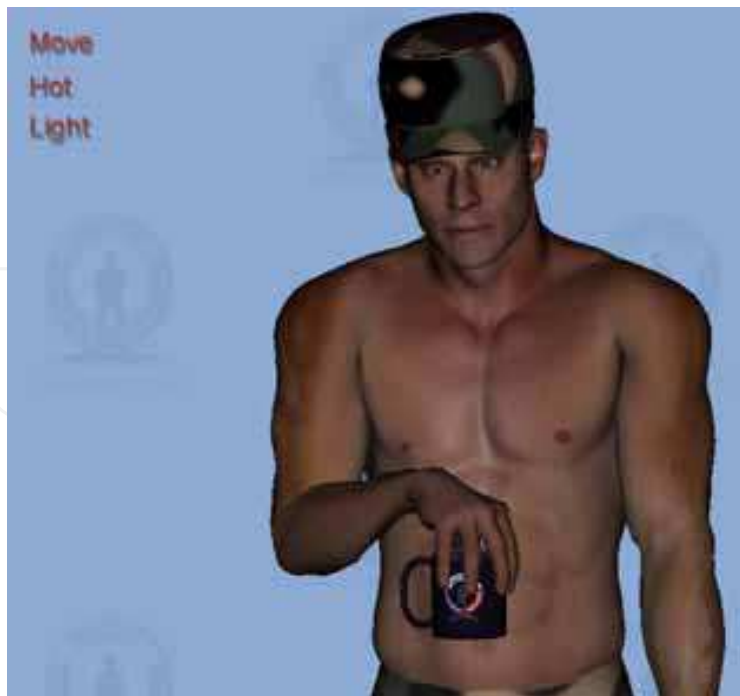
Fig. 15. Grasping a mug; power grasp

Fig. 16. Grasping a mug; precision grasp

Fig. 17. Grasping a joystick; power grasp

Fig. 18. Grasping a joystick; power grasp

13. References

Anderson, J. A. (1997). *An Introduction to Neural Networks*, Bradford Book, third edn.

Buchcholz, B., Armstrong, T. & Goldstein, S. (1992). Anthropometric data for describing the kinematics of the human hand, *Ergonomics* **35**(3): 261–273.

Farin, G., Hoschek, J. & Kim, M.-S. (2002). *Handbook of COMPUTER AIDED GEOMETRIC DESIGN*, Elsiever, 1 edn.

Gill, P., Murray, W. & Saunders, A. (2002). Snopt: An sqp algorithm for large-scale constrained optimization, *SIAM Journal of Optimization* **12**(4): 979–1006.

Hagan, M. T., Demuth, H. B. & Beale, M. H. (1996). *Neural network design*, Boston : PWS Pub.

Haykin, S. (1999). *Neuronal Network. A Comprensive Foundation*, Prentice Hall international, Inc., second edn.

Hoschek, J. & Lasser, D. (1993). *Fundamentals of COMPUTER AIDED GEOMETRIC DESIGN*, AK Peters, 1 edn.

Kapandji, I. (1996). *Fisiologia articular. Miembro superior*, Medica Panamericana, Madrid, 5 edn.

Rijpkema, H. & Girard, M. (1991). Computer animation of knowledge-based human grasping, *Computer Graphics* **25**(4): 339–348.

Tubiana, R. (1981). *The hand. Volume I*, W.B. Saunders Company, 2 edn.

Tubiana, R., Thomine, J. & Mackin, E. (1996). *Examination of the hand and wrist*, Martin Dunitz, 2 edn.

Wang, W. & Wang, K. (1986). Geometric modeling for swept volume of moving solids, *IEEE CG and A* **86**: 8–17.

Zeid, I. (2005). *Mastering CAD/CAM*, McGraw-Hill, Higher Education, 1 edn.

Modelling Simulation and Optimization

Edited by Gregorio Romero Rey and Luisa Martinez Muneta

Computer-Aided Design and system analysis aim to find mathematical models that allow emulating the behaviour of components and facilities. The high competitiveness in industry, the little time available for product development and the high cost in terms of time and money of producing the initial prototypes means that the computer-aided design and analysis of products are taking on major importance. On the other hand, in most areas of engineering the components of a system are interconnected and belong to different domains of physics (mechanics, electrics, hydraulics, thermal...). When developing a complete multidisciplinary system, it needs to integrate a design procedure to ensure that it will be successfully achieved. Engineering systems require an analysis of their dynamic behaviour (evolution over time or path of their different variables). The purpose of modelling and simulating dynamic systems is to generate a set of algebraic and differential equations or a mathematical model. In order to perform rapid product optimisation iterations, the models must be formulated and evaluated in the most efficient way. Automated environments contribute to this. One of the pioneers of simulation technology in medicine defines simulation as a technique, not a technology, that replaces real experiences with guided experiences reproducing important aspects of the real world in a fully interactive fashion [iii]. In the following chapters the reader will be introduced to the world of simulation in topics of current interest such as medicine, military purposes and their use in industry for diverse applications that range from the use of networks to combining thermal, chemical or electrical aspects, among others. We hope that after reading the different sections of this book we will have succeeded in bringing across what the scientific community is doing in the field of simulation and that it will be to your interest and liking. Lastly, we would like to thank all the authors for their excellent contributions in the different areas of simulation.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Esteban Pena Pitarch (2010). Virtual Human Hand: Autonomous Grasping Strategy, Modelling Simulation and Optimization, Gregorio Romero Rey and Luisa Martinez Muneta (Ed.), ISBN: 978-953-307-048-3, InTech, Available from: http://www.intechopen.com/books/modelling-simulation-and-optimization/virtual-human-hand-autonomous-grasping-strategy

INTECH

open science | open minds

51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821