VIRTUAL HUMAN HAND: AUTONOMOUS GRASPING STRATEGY

Esteban Peña Pitarch[1,3], Jingzhou Yang[3], Montserrat Abenoza Guardiola[2], Neus Tico[2] , and Karim Abdel-Malek[3]
[1]Departament Enginyeria Mecànica
Universitat Politècnica de Catalunya (UPC)
[2]Servicio de Rehabilitación y Medicina Física, Althaia
Av. Bases de Manresa, 61-73
08240 Manresa, Spain
[3]Virtual Soldier Research (VSR) Program
Center for Computer-Aided Design
The University of Iowa
111 Engineering Research Facility
Iowa City, IA 52242-1000, USA
esteban.pena@upc.edu

ABSTRACT
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 paper, 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.

KEY WORDS
Virtual human, virtual environment, autonomous grasping, virtual hand model, semi-intelligence grasp.

1. Introduction

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.

1.1 Previous work

Early [1] proposed a Neural Network architecture for robot hand control and used neural network control of force distribution for power grasp [2].
Nuseirat and Abu-Zitar [3] used a special neural network to find minimal finger forces in the first stage. The second stage was to use the results obtained in the first stage as a static mapping in training another neural network.

A system is presented by Bernardin et al. [4] that uses hand shape and contact-point information obtained from a data glove and tactile sensors to recognize continuous human-grasp sequences. The sensor fusion, grasp classification, and task segmentation are made by a hidden Markov model recognizer.

Bowers and Lumia [5] investigate "intelligent" grasping schemes using a fuzzy logic rule-based expert system and use a vision system, robot arm and mechanical hand to locate and manipulate unmodeled, randomly placed objects of various sizes and shapes.

Molina-Vilaplana and Lopez-Coronado [6] proposed involves the design and development of a Library of Hand Gestures consisting of motor primitives for finger pre-shaping of an anthropomorphic dextrous hand.

Moussa [7] in experiments in a simulated environment using a 28-object database, showed how the algorithm dynamically combined and expanded a mixture of neural networks to achieve the learning task.

Virtual hand models are also presented by Miyata et al. [8] and Peña-Pitarch et al. [9].

1.2 Outline

The purpose of this paper is to implement a new algorithm for autonomous grasp in a VE. To do this, we first define the grasping parameters that our virtual human uses for grasping in a VE and establish the relationship between these parameters. In the next section, we show a grasping strategy based on grasping parameters defined earlier and based on support vector machine to help make a decision about how to grasp the object. One example was implemented without lost generality. Conclusions are presented in the last section.

2. Grasping Parameters

Before grasping the object, the VH needs do some actions. We classify these actions as pre-grasp, grasp, and after-grasp, see [10] and [11]. 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.

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 the 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.
• *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.

2.1 Relationship between Grasping Parameters

Figure 1 shows the relationship independent of the dominant hand; for our VH, the dominant hand is usually the right hand.

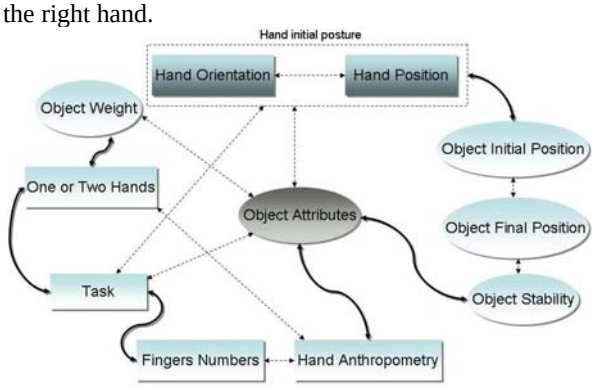

Figure 1. Relation between grasping parameters

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 the 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 on 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_o 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.

2.1.1 Object Attributes

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

- Temperature
- Fragility
- Surface shape

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 is filled with hot coffee and the action is to move, we do not 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 references [12] and [13], 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 the object and represents the object i selected by the user. $\mathbf{X}^i(u, v)$ is called the *parameterization* of the surface S for the object i, and u, v are called the parameters of this representation.

The unit normal vector to the surface can compute for the object, using the vector product, as

$$\mathbf{N}_o = \frac{\boldsymbol{X}_u^i \times \boldsymbol{X}_v^i}{|\boldsymbol{X}_u^i \times \boldsymbol{X}_v^i|}$$

2.1.2 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 simplicity, 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 2. If we change to other coordinates (u_1, v_1), shown for the hand in Figure 2, the new parametric representation is

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

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

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

where

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

The orientation in the new parameterization is the same if the **Jacobian** of the transformation is positive, and is opposed if it is negative.

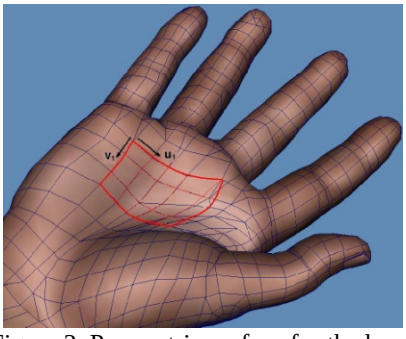

Figure 2. Parametric surface for the hand

2.1.3 Hand Position

To consider the hand position with respect to the object, we refer to this position with respect to the wrist. For us the global position of 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 the COM. That is information inherent with the object; the position of the wrist is also known in each moment.

The coordinates of the 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:

$$\mathbf{x}_w^j = A\mathbf{x}_c^i$$

where A is a transformation matrix.

2.1.4 Task

The task is the most important parameter, and many times it decides for itself how to grasp the object; i.e., when grasping a mug containing a drink, the usual response is to grasp the handle; when moving the mug, we can grasp the handle or the top. The Oxford English Dictionary defines *task* as "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., opening the door is a task, and the elemental action is pulling or pushing. This elemental action can also be used for other tasks, like moving a joystick in a machine.

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 $i = 1 \ldots 4$ 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 similar to pulling, but we add the use of the palm. We define this as PS_i, where $i = 1 \ldots 5$ 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 use is defined as PI_i, where $i = 1 \ldots 4$ 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 with the thumb and fingers, not in contact with the palm. We define this action as PH_i, where $i = 1 \ldots 4$ 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 transformed with the index. We define TO for as touch.

2.1.5 Object Initial Position

Object initial position is defined by:

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

2.1.6 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 to add some actions like walking or advancing the body.

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 $\mathbf{x}_f^i = [x_f^i \; y_f^i \; z_f^i]^T$, both with respect to the global coordinates. The relationship between the initial and final position is:

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

where B is a transformation matrix.

2.1.7 One or Two Hands

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

2.1.8 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 parameter is a function of the shape and weight of the object.

We define elemental actions, and each elemental action shows the number of fingers to be used. 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 as follows:

$$If\ 1 \le \rho < 20\ mm\ then\ PL_1$$
$$If\ 20 \le \rho < 40\ mm\ then\ PL_2$$
$$If\ 40 \le \rho < 60\ mm\ then\ PL_3$$
$$If\ 60 \le \rho < 90\ mm\ then\ PL_4$$

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

2.1.9 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 bottle can lose stability and fall. Falling is not the final reaction when we touch the object.

3. Grasping Strategy

We have defined all the parameters that work in grasping. In this section, we present the process of grasping. Figure 3 presents a flowchart of grasping. The objects are in the virtual environment, and the user chooses one object from among several. Each object has information attached about several attributes like task, shape, and position of center of mass; 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 approximately how to put the object in the workspace of the hand.

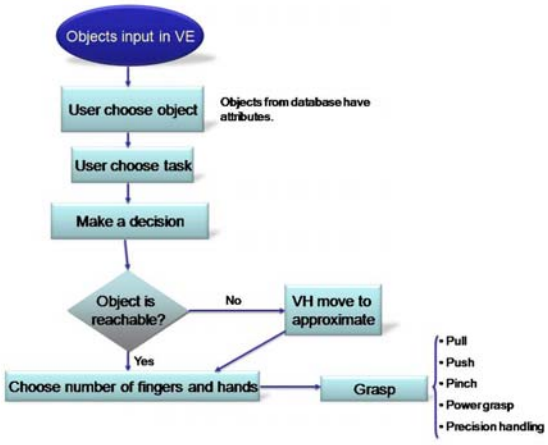

Figure 3. Grasping flowchart

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.

3.1 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.

3.2 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 having 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, including COM. Of course, a change in the position of COM is a function of the quantity of coffee. For this object, there are two reasonable tasks: drinking or moving.

3.3 User Chooses Task

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

3.4 Making a Decision

Inputs in a simple associator are task, weight, shape, temperature, etc., and the output pattern is type of grasp, i.e., pinch, pull, etc. When the VH makes a decision, it is based on a linearly separable system.

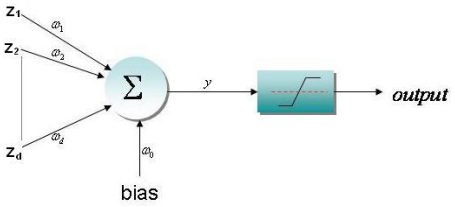

Figure 4. Single perceptron

Figure 4 shows a single perceptron from the literature [14], [15], [16] 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 $w = [\omega_1, \omega_2, \ldots, \omega_d]^T$, and the weight for the bias is ω_0.
In our case, we choose to apply these networks for a classification problem in which the inputs are binary images of attributes like task, shape, temperature, and density. The output of the perceptron is given by

$$\mathbf{y} = g\left(\sum_{j=1}^{d} \omega_j \varphi_j(z) + \omega_0\right) = g(\mathbf{z}^{\mathrm{T}}\varphi)$$

where φ denotes the vector formed from the activations $\varphi_0, \ldots, \varphi_d$, and the function of activation in this case is a symmetric saturating linear (satlins), because it is linearly separable. The input/output relation is:

$$\begin{array}{ll} g(a) = -1 & a < -1 \\ g(a) = a & -1 \le a \le 1 \\ g(a) = 1 & a > 1 \end{array}$$

4. A Task-oriented Object Grasping Approach: Implementation

In this section, we provide an example applying the Support Vector Machine (SVM) presented in the last section. 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 a mug, it has only two associated tasks: one is drinking and the other is moving. The user 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 above, such a shape, hand orientation, and number of fingers. The problem becomes a classification problem between two classes.

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

If we choose $x = 0$, as the axis for classification and $b = 0$ as 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.

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 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 Section 2.1.8, 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. These grasps are shown in Figure 5.

Figure 5. Grasping a mug: power and precision grasps

5. Conclusion

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 introduced is that the virtual human can grasp autonomously without the user once the task is chosen. The 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.

Based on the fact that every object has some inherent parameters, attributes, and tasks for which they are designed, the objects can be grasped in different situations and positions connected with the upper body. Semi-intelligent task-oriented object grasping was implemented in Vitools™ and demonstrates that the proposed algorithm works with several objects.

Acknowledgements

This work was partially supported by the projects DPI2007-63665 and the Caterpillar Inc. project: Digital Human Modeling and Simulation for Safety and Serviceability.

References

[1] H. Liu, J. Butterfass, S. Knoch, P. Meusel, & G. Hirzinger, A new control strategy for DLR's multisensory articulated hand. In *Proceedings. 1999 IEEE Control Systems*, 1999, 47–54.

[2] M. Hanes, S. Ahalt, K. Mirza, & D. Orin, Neural network control of force distribution for power grasp. In *Proceedings.IEEE International Conference on Robotics and Automation*, 1991, 746–751 vol.1.

[3] A. Al-Fahed Nuseirat & R. Abu-Zitar, Neural network approach to firm grip in the presence of small slips. *Journal of Robotic Systems*, *18*(6), 2001, 205–215.

[4] K. Bernardin, K. Ogawara, K. Ikeuchi, & R. Dillmann, A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models. *IEEE Transactions on Robotics*, *21*(1), 2005, 47–57.

[5] D. Bowers & R. Lumia, Manipulation of unmodeled objects using intelligent grasping schemes. *IEEE Transactions on Fuzzy Systems*, *11*(3), 2003, 320–330.

[6] J. Molina-Vilaplana & J. Lopez-Coronado, A neural network model for coordination of hand gesture during reach to grasp. *Neural Network*, *19*, 2006, 12–30.

[7] M. A. Moussa, Combining expert neural networks using reinforcement feedback for learning primitive grasping behavior. *IEEE Transactions on Neural Networks*, *15*(3), 2004, 629–638.

[8] N. Miyata, M. Kouchi, & M. Mochimaru, Posture estimation for design alternative screening by dhaibahand - cell phone operation. In *SAE 2006 Digital Human Modeling for Design and Engineering Conference*, 2006, 2006–01–2327.

[9] E.Peña-Pitarch, J. Yang, & K. Abdel-Malek, SANTOS™ Hand: A 25 Degree-Of-Freedom Model. *SAE 2005 Digital Human Modeling for Design and Engineering Conference*, 2005, 2005-01-2727.

[10] I.A.Kapandji, *Fisiologia articular. Miembro superior*, (Medica Panamericana, Madrid, 5 Edición, 1996).

[11] R. Tubiana, *The hand. Volume I*. (W.B. Saunders company. 2 edition, 1981).

[12] J. Hoschek & D. Lasser, *Fundamentals of computer aided geometric design* (AK Peters. 1 edition, 1993)

[13] G. Farin, J. Hoschek, and M.-S. Kim, *Handbook of computer aided geometric design* (Elsiever. 1 edition, 2002).

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

[15] S. Haykin. *Neuronal Network. A Comprensive Foundation* (Prentice Hall international, Inc. Second edition, 1999).

[16] J. A. Anderson, *An Introduction to Neural Networks* (Bradford Book. Third edition, 1997).