Computer Animation of Knowledge-Based Human Grasping

Hans Rijpkema and Michael Girard

SCAN (National Institute for Computer Animation)
Groningen, the Netherlands

Abstract

The synthesis of human hand motion and grasping of arbitrary shaped objects is a very complex problem. Therefore high-level control is needed to perform these actions. In order to satisfy the kinematic and physical constraints associated with the human hand and to reduce the enormous search space associated with the problem of grasping objects, a knowledge based approach is used. A three-phased scheme is presented which incorporates the role of the hand, the object, the environment and the animator. The implementation of a hand simulation system HANDS is discussed.

CR Categories: I.3.5: computational geometry and object modeling; I.3.7: Three-Dimensional graphics and realism;
Keywords: Grasp Planning, Animation, Simulation, Robotics

1.Introduction

Although there has been some progress on simulating the geometric deformation of the hand during a grasping contact [Gourret 89], animating the grasping motion behavior of the hand remains a difficult task for the computer animator. Even the use of advanced inverse-kinematic and physically-based limb control techniques demand that the animator tediously position the palm, the thumb, and each finger of the hand until the grasped object appears to be trapped by the hand in a natural, physically credible way.

Special input devices that attempt to digitize hand motion, such as the *data-glove*, do not yet record precise individual finger and thumb joint motion or provide the feedback required for intuitive interactive grasping [Fisher 86] [Iwata 90]. Augmenting digitized motion with some grasping intelligence may help to reduce the need for such extensive feedback. However, the focus of this paper is on the problem of *synthesizing grasping motion*, rather than simply recording it.

Since the hand is a multi-limbed system, recent computer animation and robotics research directed at problems associated with modelling limb kinematics and dynamics [Armstrong 86] [Badler 87] [Girard 87] [Isaacs 87] [Korein 82] [Walker 82] [Barzel 88] [Wilhelms 87] [Schoner 90], collision detection [Gilbert 89] [Moore 88] [Baraff 89], motion planning [Lozano-Perez 82] [Brooks 83], and optimizing motion in the presence of kinematic and physically-based constraints [Girard 90] [Kirckanski 82] [Lin 83] [Sahar 85] [Tan 88] [Witkin 87] have helped to lay the basis for controlling individual fingers. However, the selection of grasping positions, the coordination of fingers, and the determination of the palm's motion trajectory during a grasping action requires a higher-level analysis and a control system that operates as a function of the hand's geometric, kinematic, and physical characterics taken as a whole.

Although we are able to easily pick up most objects with little effort, the human capability for manually grasping objects is a non-trivial task. Grasping strategies must take into account the geometry and dynamic characteristics of the object to be grasped, the selection of contact between the object and the fingers, thumb and palm of the hand, and the problems associated with finding collision-free paths in the context of the general environment.

Our approach begins with the realization that the ease with which a person is able to decide how and where to grasp an object depends on the person's familiarity with that object. We view this human capability as a multi-stage process, in an approach that is similar to that suggested by Tomovic [Tomovic 87] in the robotics literature.

First, the object is identified according to its similarity to a given class of shapes, such as a block, sphere, torus, cone, or cylinder. Then, in the second stage, a grasping strategy associated with the object's classification is chosen from a knowlege-base of class specific, parameterized techniques. In the third stage, the grasp is marginally adjusted to manage the object's deviation in shape from its classified shape. In this way, the astronomical search space of grasping techniques and grasping locations which are possible between the hand and an arbitrary three-dimensional object may be restricted to the much smaller set of frequently used human grasping methods.

In the next section, we begin with a kinematic description of the hand. In section 3 the high-level control of the hand is discussed. In section 4, we give an overview of the grasp planning problem and the knowledge-based approach toward its solution. Finally, in section 5, we give our conclusions with suggestions for future research.

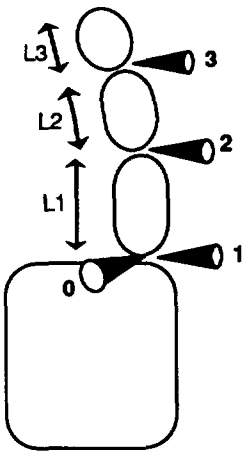
2. Kinematics of the hand

2.1 Model of a human hand

The fingers have 4 DOF, two at the connection with the palm, one at the end of the first finger part and one at the end of the second finger part [See figure 1]. From this we can establish the link coordinate frames of the fingers and obtain the four Denavit-Hartenberg parameters [Denavit 55] for each link.

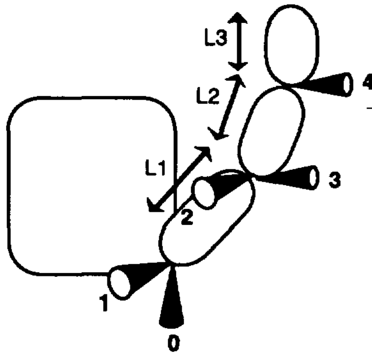
Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

©1991 ACM-0-89791-436-8/91/007/0339 $00.75



	θ	a	d	α
1	90	0	0	90
2	0	l_1	0	0
3	0	l_2	0	0
4	0	l_3	0	0

fig. 1: model of a finger



	θ	a	d	α
1	0	0	0	90
2	90	l_1	0	0
3	0	0	0	90
4	0	l_2	0	0
5	0	l_3	0	0

fig. 2: model of a thumb

The thumb is very dextrous and therefore a more complicated manipulator. Because a large part of the thumb seems to be part of the palm of the hand and the joints are moving along non-trivial axes, the motion of a thumb is not easily understood. A workable model of the thumb that approximates the motions of a real human thumb is a manipulator with 5 DOF [See figure 2]. From this we can establish the link coordinate frames of the thumb and obtain the four Denavit and Hartenberg parameters for each link.

2.2 Basic motion control

From the Denavit-Hartenberg parameters, it is possible to find the transformation matrices for adjacent coordinate systems. The forward kinematics problem is easily solved by using the product of these transformation matrices [Fu 87]. Forward kinematics is useful for bending fingers at the joints. However we are also interested in simulating the human ability to place the tip of the finger at a certain location. For this inverse kinematics is required.

2.2.1. Inverse kinematics of the fingers

A human finger has the property that it is (almost) impossible to move the joint of the last link (joint *4*) without moving the next to last joint (joint *3*) and vice-versa, without forcing one of the two not to move in some unnatural way. Therefore, there is a dependancy between these two joints that is caused by the tendon that runs through the finger. Careful observation reveals that there is an almost linear relationship between the joint angles q3 and q4. [See figure 3].

After measuring several human subjects, we found this could be reasonably approximated by:

$$q4 = 2/3 * q3$$

By making q4 fully dependent on q3 , the number of degrees of freedom is reduced. The solution of the inverse kinematics will now be of the form:

$$q = (q1, q2, q3, 2/3*q3)$$

Landsmeer's [Landsmeer 55, 58, 63] empirical studies of the physiology of the human hand addressed the relationship between the joint angles of the fingers and the activation of the tendons. Other studies support the finding that the relationship between the joint angles is not completely linear [Armstrong 78]. We are planning to incorporate this more accurate model in the near future.

A second way to simplify the problem is to note that the finger is a planar manipulator with the execption of the first joint. From this it follows that q1 can be calculated directly from the displacement of the fingertip in the x0 and y0 direction, and that it is completely independant of the other joint angles. [See figure 4a].

From the fact that q3 and q4 are fully dependent, it can be seen that in order to reach an arbitrary point at distance d from the origin of the 0th coordinate frame, there is a unique solution for q3, and therefore

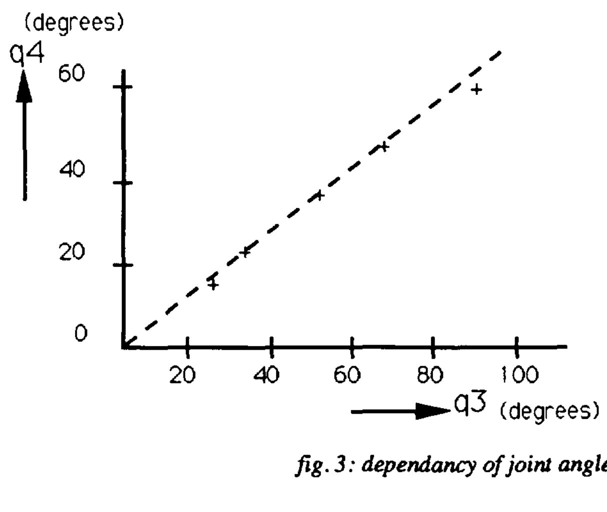


fig. 3: dependancy of joint angles

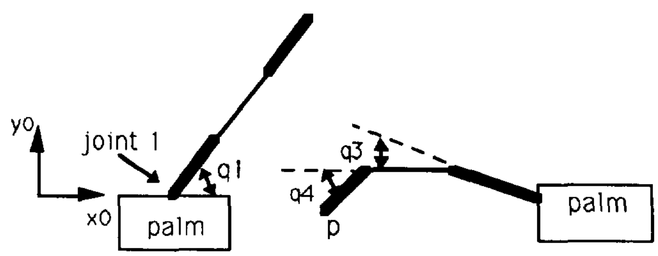


fig. 4: inverse kinematics of the finger

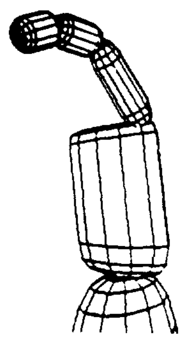


fig. 5: single finger control

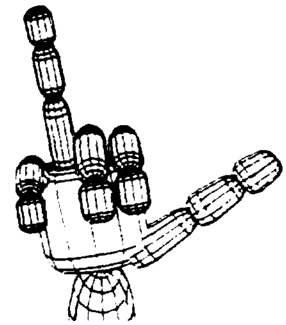


fig. 6: group control

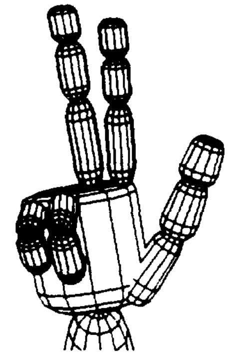


fig. 7: hand control

also for q4. [See figure 4b]. These angles can be calculated using a binary search on q3 that converges quickly. The remaining joint angle q2 can now be calculated such that the tip of the finger will be at the correct location.

2.2.2. Inverse kinematics of the thumb

Due to the greater kinematic complexity of the thumb, a closed-form solution was not found. Instead we employed the resolved motion rate control method, in which the desired joint-space solution of the thumb is satisfied as a secondary goal [Liegeois 77] [Klein 83]. An excellent review of this method, along with a means of solving difficulties with singularities of the pseudo-inverse jacobian, may be found in [Maciej 90]. The thumb's joint-space secondary goal, in context of our kinematic model, is recalculated at each position to minimize deviations from joint angles matching the following experimental observations:

$q3 = 2*(q2 - 1/6*\pi)$ and

$q5 = 7/5 * q4$.

3. High level control of the hand

Attaining a desired posture by moving all the different joints of the fingers separately is a very tedious and time consuming process. Higher level control has been incorporated in our system, called HANDS, to ease the burden of manipulating many degrees of freedom and to prevent unnatural hand postures from occuring.

The interactive positioning of a hand into a desired gesture in HANDS may be accomplished by using a set of functions that give the animator different levels of control over the hand.

3.1 Single-finger control

The lowest level of control involves direct independant control over each finger. [see figure 5]. This can be done using both forward and inverse kinematics of fingers, which satisfy the constraints of natural movement discussed in the previous section.

3.2 Group control

The second level of control is that of group control. [see figure 6]. The user can select which fingers belong to a group and then use a number of functions to change the hand posture:

- *Closing and opening of a group.*
  This function closes or opens all fingers that are part of the group at the same time, in the same way as this can be done for single fingers.
- *Spreading of a group*
  The fingers of the group are spread outward or inward by changing the joint angle of the first joint of all the fingers in the group, depending on their location on the hand and the joint angles of the two most outward fingers.

3.3 Hand control

The last level of control is complete hand control [see figure 7].

- *Hand posture library*
  The user can build up a hand posture library from which he can choose desired hand postures. These hand postures are made with the use of the above functions and can then be stored with an unique name in the library. Thus hand postures can be added to and deleted from the library. The advantage of this is clear: a posture can be constructed once and then easily be recalled from the library and then pasted in.

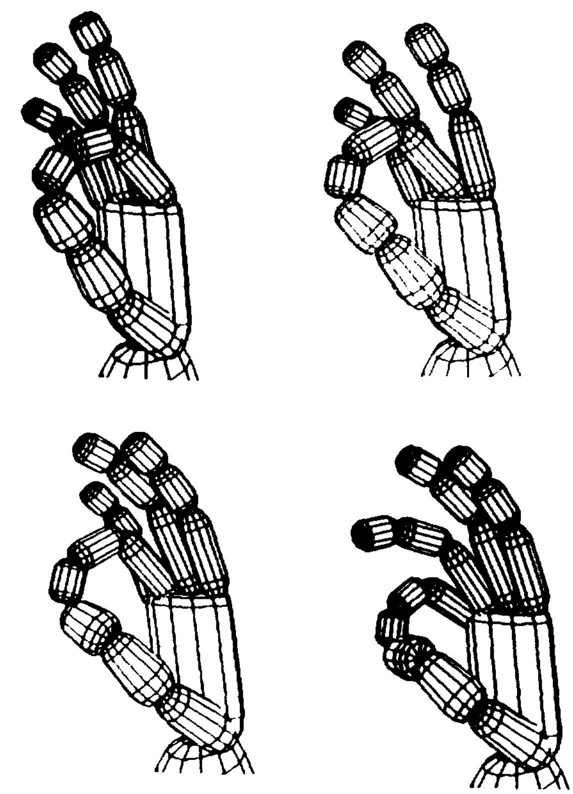


fig. 8: the pinches of the hand

Precalculcated postures

Besides the user-defined hand postures there are also system-defined hand postures. These are hand postures that might be difficult to achieve with the controls mentioned above or postures that are very often used. Examples of these hand postures are the hand at a rest position, a fist and some pinches. A pinch is the state in which the tip of the thumb is placed against the tip of a finger. These postures are calculated using collision detection [Gilbert 88] [Rypkema 90] so that the tips are exactly touching each other, and not intersecting. [see figure 8].

4. Grasp Planning

4.1 The elements involved in grasping

When grasping behavior is incorporated into an interactive computer animation system, four elements are of main importance:

- the object
- the hand
- the environment
- the user-interface

These elements have certain characteristics that influence the design of the grasping motion [see figure 9].

Characteristics of the target object:

Geometrical:	What is the size and the shape of the object?
Physical:	What are the mass, distribution of mass, and inertia of the object?
Mechanical:	What is the rigidness (i.e. is it completely rigid, elastic or flexible) and the coefficient of friction of the object? [Wang 88].

Characteristics of the hand.

Geometrical:	How large is the hand, what is the shape of the hand?
Physical:	What is the strength of the hand?
Mechanical:	Dexterity (how skilled is the hand?), grip (what is the friction coefficient of the hand?)
Naturalness:	Human sensory motor control, muscular con straints.
Topological:	What are the connections and degrees of free dom at each of the joints of the hand.

Characteristics of the environment

Information about the environment is required to determine potential obstacles and collisions.

Spatial complexity:	Where are all other objects (location and orientation)?
Dynamical complexity:	How do other objects, arms, etc. move in time?

Characteristics of the user-interface

Expression:	How does a user want to express his ideas?
Automation:	How much does the user want to be done automati cally?
Control:	Under what circumstances does a user want to be able to take control?
Output:	When there are multiple solutions, when should the system offer choices and when must it out put just one, working solution?

4.2 A Knowledge Based approach

Previous research on the analysis of human hand motion supports a knowledge-based approach to the synthesis of a grasping behavior [Tomovic 87] [Iberall 88].

Human beings perform grasping tasks by using *experience* that has been gathered over time. The approach followed here is to incorporate this experience into a knowledge base. The knowledge base can be seen as a collection of precalculated strategies for different categories of situations, thus partitioning the enormous search space of possible solutions into computationally managable subsets.

Each of the knowledge-based strategies assumes the form of a three-phased decomposition into the following subtasks [see figure 10]:

1. The task initialization phase
2. The target approach phase
3. The grasp execution phase

In the task initialization phase, the target object is classified as a primitive and the overall strategy for grasping the object is determined. During the target approach phase, all possible grasp positions are filtered to obtain the feasible ones from which the hand is preshaped to assume an optimal or user-selected grasp position. Once the hand is preshaped to the primitive, the grasp execution phase ensures that the fingers will close around the actual object.

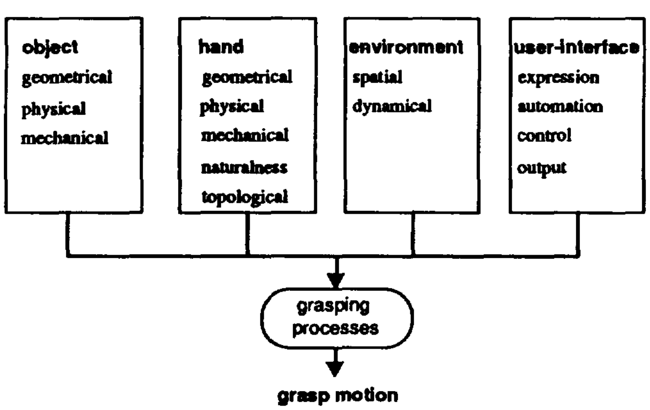


fig. 9: elements involved in grasping

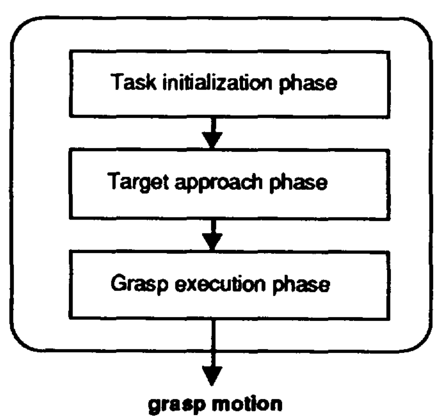


fig. 10: decomposition of grasping task

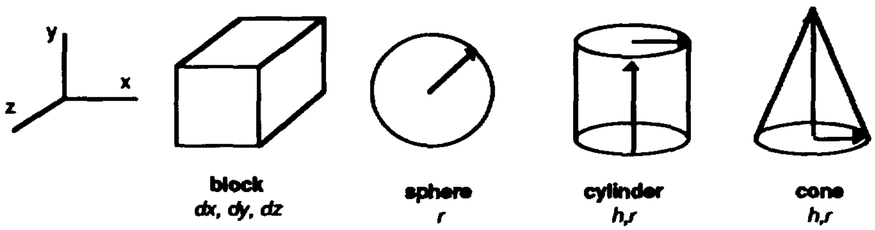


fig. 11: primitive types

4.3 The task initialization phase

When a specific grasping task is to be carried out, the motion is influenced by the *high-level goal* that leads to the grasping motion. For example a hammer should be grasped differently depending on whether one wishes to pound a nail in or pull a nail out. Therefore the *context* of the action should be made clear in the task initialization phase, in order to be able to exclude at this early stage all possible grasp configurations that do not satisfy some desired goal. The classification of grasps in terms of goals has not been implemented.

Thus far, our knowledge base consists of classifications based only on the shape of the object. During the object identification process [see figure13] the object is classified as one of the primitive object types (block, sphere, cylinder etc.) [see figure 11] and the values of the attributes are specified. Classification of complex 3D shapes as generic primitives is a difficult problem that has been addressed in the computer vision literature [Fu 87] [Marr 82]. Objects can be compared with the different primitive types by looking at volume, center of gravity, etc. The primitives must also be oriented in such a way that the best matching between the primitive and the target object is achieved by minimization of differences in their occupied volumes.

Human beings have a very good sense of classifying objects as primitives. Therefore in the current version of the grasping system, the classification of the object as a certain primitive is left to the user. This can be done in a simple interactive way by selecting a primitive from a pop-up menu and then visually positioning the primitive so that it circumscribes the object.

Once the primitive is known, the values for the attributes of the primitive can be computed automatically. These attributes are very simple. For a block they are the lengths along the three axes: dx, dy and dz. For a sphere they are the radius *r* and for a cylinder and a cone they are the radius *r* and the height *h* [see figure 11].

Finally, information about the environment should be gathered. The environment can put restrictions on the way an object is grasped. Objects other than the target object can block the path for the hand, so as to make it impossible to reach certain points or surfaces of the object.

4.4 The target approach phase

The position of the hand includes both the position of the palm and the positions of the fingers. As a convention, we will call a hand position that specifies a grasp a *grasp position*.

The search space of possible grasp positions for a given object is enormously large, so it would be very time consuming to find a correct and natural grasp by simply searching all these possibilities. This follows partially from the fact that the hand has a large number of degrees of freedom. In the model every finger has 4 DOF, the thumb has 5 DOF and the hand 3 DOF, so this gives (4x4)+5+3 = 24 DOF, which shows how dextrous a human hand is. Also, when only considering finger-object contact types the number of possible contacts is extremely large. Salisbury has shown that a hand with five three-linked fingers may touch a ball in 840 ways [Mason 85].

A grasp should be found from this large solution space that minimizes muscle tension and optimizes the stability of the grip on the object. The number of possibilities may be limited by enforcing a set of *constraints and properties* that can be derived from observing how human beings tend to grasp objects.

The first property that decreases the large number of possible grasps is the observation that humans tend to pick up objects with the fingers placed on *opposite faces*. This also makes sense physically, because in this way the forces that the fingers need to exert on the object in order to obtain a stable grasp is probably less than the forces needed when grasping the object in any other way.

A second property of human grasping is that *the thumb almost always takes part in the grasp*. Grasps without the thumb are very rare and they don't look natural. When picking up an object using opposite faces, the thumb is placed on one face of the object and the other fingers that take part in the grasp are placed on the opposite face.

These two constraints/properties mean in the case of grasping a block that the number of grasp types is limited to 24:

#opp.faces . #thumb locations . #palm locations =
3 . 2 . 4 = 24
[see figure 12]

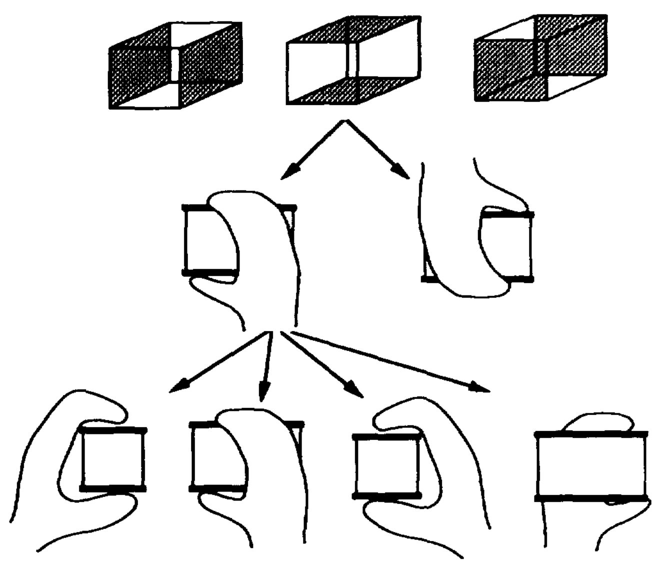


fig. 12: grasp positions for a block

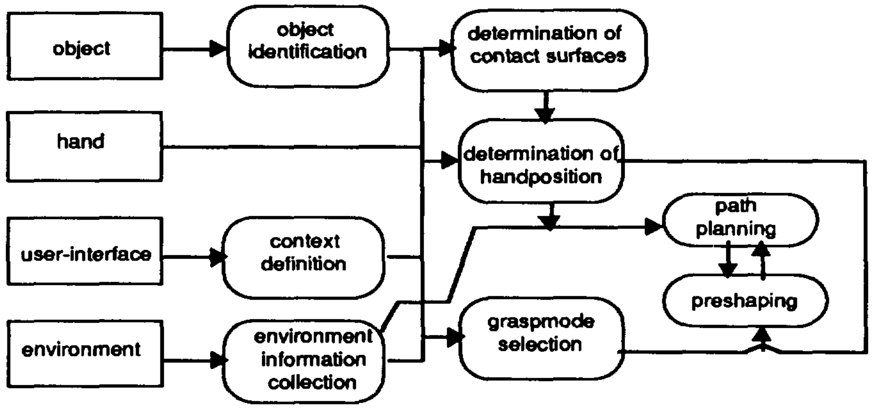


fig. 13: task initialization and target approach

In order to automate grasping we need to first determine which grasp positions are feasible. Then we wish to select the 'best' or optimal grasp out of this feasible set. Instead of computing every possible grasp position defined by the constraints and properties discussed in the previous section, a series of more computationally efficient tests may be applied to incrementally rule out infeasible grasps.

The target approach phase first applies these tests and then orders the feasible grasps in accordance with an optimization criterion. Then the hand is lead from an arbitrary position to the vicinity of the object, with the hand preshaped to grasp the target object's associated primitive. The target approach phase consists of the following subtasks [see fig. 13]:

1. determination of contact surfaces of the object's associated primitive
2. selecting the hand position with respect to the feasible contact surfaces
3. selecting the graspmode and hand structure for the chosen hand position
4. preshaping of all fingers to grasp the object's primitive
5. path generation of the palm towards the preshaped hand position

4.4.1 Contact surface determination

The first phase in which infeasible grasp positions are eliminated is the determination of contact surfaces. To determine whether a certain contact surface combination will lead to incorrect grasps, four tests can be applied:

1. are the contact surfaces reachable?
2. is it possible to spread the hand enough so that the fingers can close around the object?
3. are both the contact surfaces free, i.e. are they not blocked by other objects?
4. does it make sense to grasp the object with these contact surfaces, i.e. does the contact surface combination conflict with the high level goal?

If any of the above tests is not satisfied, that contact surface combination should be deleted from the list of possibilities. In the current version of the grasping system only the first two tests are applied. (The other two tests need information that should be collected during the task initialization phase, outlined in section 4.3).

The first test, to determine if the contact surfaces are reachable, is done by calculating the distance from the base of the arm to each contact surface. If the distance for at least one of the two contact surfaces is larger than the length of the arm, it means that the contact surface combination is not reachable, and therefore it has failed the test.

The second test deals with the *spread of the hand* and the size of the primitive. The spread of the hand is a measure for the distance between the tip of the thumb and the the tip of another finger. When the hand is flat and the thumb is pointing outwards the maximum spread can be determined for each thumb-finger combination by calculating the distance between the two tips. The maximum spread of the hand is then the maximum of all these maximum spreads of the fingers. Objects can only be grasped with contact surfaces that are no further apart than the maximum spread of the hand.

4.4.2 Determination of the grasp position

The second level of deleting infeasible grasps is the selection of the grasp position. To do this the following tests can be applied:

- is the hand position within the reach of the arm?
- will the grasp follow from an feasible (and optimal) arm motion?

The first test to determine whether the hand position is within reach of the arm is done by calculating the location of the wrist at the desired hand position. Then the distance from the base of the arm to the desired wrist location is calculated. If this distance is larger than the length of the arm minus the length of the hand then the desired hand position is not reachable and is therefore excluded from further consideration.

In the case of the second test a difficulty is that the selection of the best grasp must take into account the motion of the entire arm. For example, the best grasp may be the one which is reached by the minimum energy path. The constrained optimization of collision free limb trajectories requires numerical methods such as steepest descent gradient techniques [Witkin 87], or dynamic programming [Girard 90]. These techniques are extremely costly, requiring optimization of path and speed distribution in terms of cost criteria involving both kinematic and dynamics based quantities. A further complexity arises due to the need to calculate the actual tension in the tendons and muscles rather than the idealized rotational torques of inverse-dynamics formulations.

Therefore we use a heuristic approach that orders the feasible grasps, but leaves the final decision to the user. A heuristic that has proven effective is to minimize the weighted sum of the translational and rotational distance between the initial hand position and the final grasp position. The translational displacement is given by the distance from the initial wrist location to the final wrist location. The rotational displacement of the hand can be calculated by using the quaternion formulation [Shoemake 85] [Pletinckx 89].

4.4.3 Grasp mode and hand structure selection

With the contact surfaces for the thumb and the fingers known, we must still determine the grasp mode and hand structure for the grasp. The selection of the grasp mode depends mainly on the purpose of the action. The grasp mode may be a lateral or palmar grasp [see figure 14]. A glass is picked up most of the time with a lateral grasp when the goal is to put the glass on the shelf, but when the same glass is used for drinking it will probably be picked up with a palmar grasp (unless the contents of the glass are very hot). Sometimes the selection of the grasp mode can also depend on the characteristics of the object. If an object is very heavy, a power grasp is needed to be able to lift the object. So when restricting the grasp modes to lateral and palmar grasps the determination of which of the two should be applied depends on the high-level goal. In in our current implementation the selection of graspmode is left to the user.

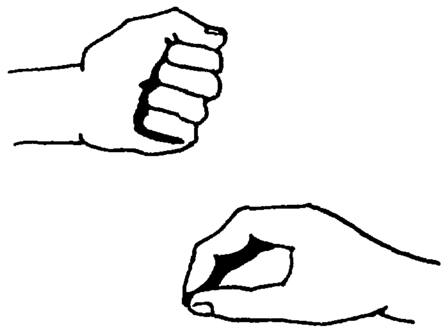


fig. 14: palmar and lateral grasp

Although there are a large number of hand structures that are possible to use when grasping objects, in practice, only a small number of them are used.

With the following notation: *T = thumb, I = index finger, M = middle finger, R = ring finger and L = little finger*, the most natural grasps can be defined as [Tomovic 87]:

2-fingered structures (pinches)
pinch-TI
pinch-TM
pinch-TR
pinch-TL

3-fingered structures
three-TIM
three-TMR
three-TRL

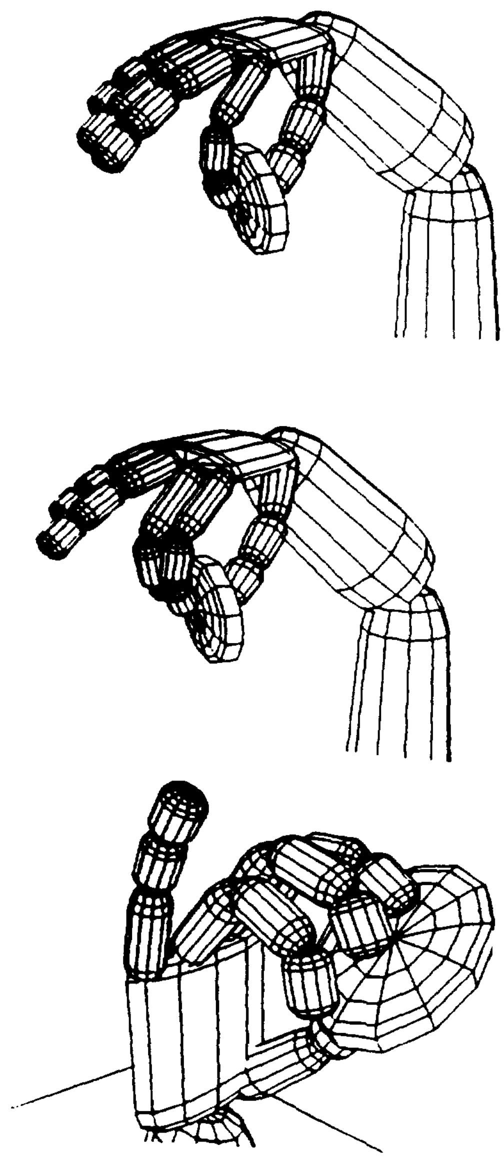


fig. 15: hand structures

4-fingered structures
four-TIMR
four-TMRL

5-fingered structures
five-TIMRL

The above add to a total of ten hand structures [see figure 15]. The selection of the hand structure can be subdivided into two different problems:
- How many fingers can be used?
- Which fingers can be used?

The maximum number of fingers that can be used in the grasp depends on the size of the object and the size of the hand. The available space on the object must be compared with the space occupied by a single finger to give an indication about the maximum number of fingers.

The determination of which fingers are valid for the desired grasp also depends on the relative size of the hand and object. The distance between the two contact surfaces determines how much the fingers must be spread to grasp the object. The selection of the contact surfaces computed at an earlier stage guarantees that there is at least one finger for which the maximum spread is larger than this distance. All fingers that have a maximum spread larger than this distance are valid grasp fingers.

After calculating the maximum number of grasp fingers and determining which fingers are valid we must still choose the best combination of fingers. Our observation of human grasping have lead us to formulate the following general rules to select a hand structure: **1)** maximize the number of fingers (since more contacts improve stability) and **2)** favour the use of fingers closer to the thumb (since they are stronger). Our implementation HANDS, picks a hand structure using the above rules [see figure 18] but allows the user to intervene and select another hand structure [see figure 19].

4.4.4 Preshaping of the fingers

Having established the contact surfaces, the approximate grasp position, hand structure and grasp mode, a more precise hand position must now be calculated. The palm position must allow the fingers to be placed on the object in such a way that the forces they exert on the object produce a stable grasp.

Using the notion of the *pinch-line*, the correct hand position can be calculated in a geometrical way. The pinch-line is the imaginary line between the thumb and a finger, called the *pinch finger* [see figure 16]. The forces that both fingers of the pinch exert on the object are directed along this pinch-line. In order to establish a stable grasp it makes sense that this pinch-line should go through the center of gravity of the object. Another assumption that can be made is that the thumb and pinch finger are placed on opposite contact surfaces in such a way that the forces exerted by the fingers are directed perpendicular to these contact surfaces.

fig. 16: the pinch-line

The choice of which of the grasp fingers is the pinch finger can be based on the same observations made in selecting the hand structure: the grasp finger closest to the thumb is most likely to be the pinch finger.

The orientation of the pinch-line varies as a complex function of the natural kinematics of the pinch fingers. The configuration in which the pinch fingers are a certain distance apart may be found by interpolating the thumb and pinch finger between their rest position and their relaxed pinch positions. The relaxed pinches of the hand, as shown in [figure 8], are automatically precalculated to satisfy the joint angle constraints of the fingers and thumb using the inverse-kinematics procedures described in section 2. The desired distance between the fingertips may be quickly achieved by using a binary search. The pinch-line is now the line between the tip of the thumb and the tip of the pinch finger expressed with respect to the hand coordinate system.

The hand can now be placed so that the pinch-line is perpendicular to the contact surfaces. To complete the preshaping of the hand all the grasp fingers are moved from their rest position to their relaxed pinch position until they collide with the primitive associated with the object.

4.4.5 Path generation

Knowing the initial configuration of the hand and arm and the preshaped configuration of the hand, the approach path of the hand towards the object needs to be determined.

Experiments have shown that this approach path has a *predictable shape* [Paillard 82] [Tomovic 87]. Seen from one side the hand travels along a *straight line* and seen from another side it travels along a *curve* [see figure 17]. This property can be incorporated into the grasping system by adding another key position, called the approach position, along the desired path. Inverse-kinematics using pseudo-inverse control [Girard 90] [Liegeois 77] [Maciej 90] is used to move the hand along the designated path.

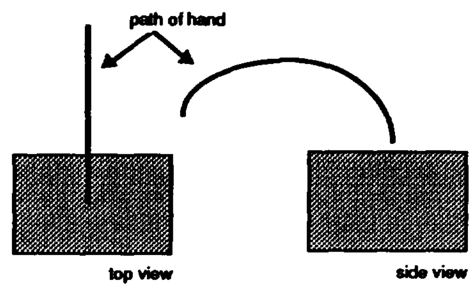


fig. 17: approach path of the hand

4.6 The grasp execution phase

To complete the grasp, the fingers need to move from their position on the primitive associated with the object so that they are touching the object itself. The thumb and pinch finger move towards each other by interpolating their current position and their completed pinch position. When one of the links of the finger collides with the object, that link cannot be moved anymore. This means that when link i collides, joints j with $j = 0,1,..i$ must be locked. The same procedure is used for the other fingers. Collisions between the fingers and the target object are calculated by using an octree spatial decomposition method for concave polygonal objects [Rijpkema 91] and a fast procedure for computing the distance between convex polygonal objects [Gilbert 88]. A discussion of our collision detection scheme is beyond the scope of this paper.

5. Conclusion

The development of knowledge-based hand behavior has made the task of computer animated grasping relatively simple, while still maintaining the creative role and guidance of the animator. We are currently extending our grasping knowledge-base to include more complex classification primitives, for example poking one's finger through the hole of a torus-like cup handle. The approach we have taken will allow us to add more complicated sequences of actions, such as shovelling a book up from a table by one's thumb before grasping it.

We think that the use of knowledge-based techniques will play an increasingly important role in the modeling of motions that involve complex physical and geometric constraints, particularly when optimal behaviors must be selected from a broad set of feasible actions.

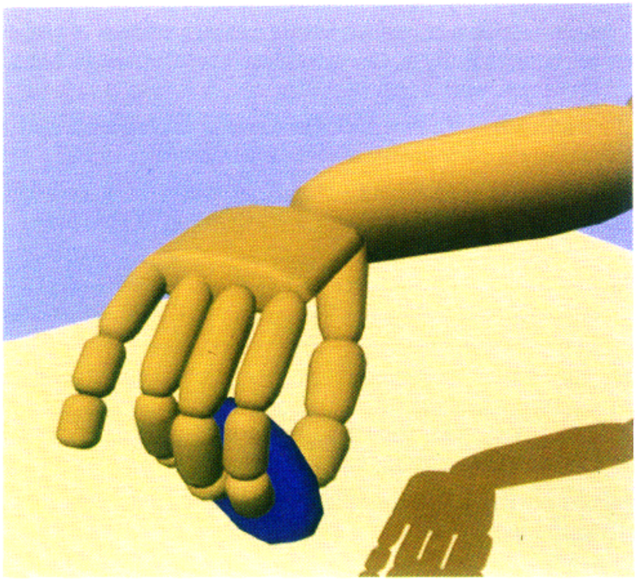


fig. 18: example of grasp selected by the system

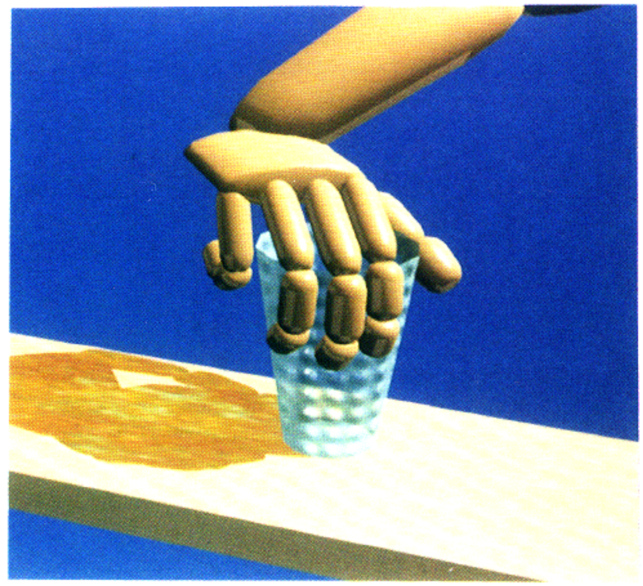


fig. 19: example of grasp selected by the user

Acknowledgements

We would like to thank Rob v.d. Weg and Wiek Vervoort of the University of Twente (the Netherlands) for their advice and constructive remarks during this research. Special thanks to Susan Amkraut for creating and adapting software for our purposes. And all at SCAN, without whom......

References

[Armstrong 86] W.W. Armstrong, M. Green and R. Lake, Near-real-time control of human figure models, Proceedings of Graphics Interface 1986

[Armstrong 78] T.J. Armstrong and D.B. Chaffin, An investigation of the relationship between displacements of the finger and wrist joints and the extrinsic finger flexor tendons, iomechanics, vol. 11, pp 119-128, Pergamon Press Ltd., 1978 (great-Britrain)

[Badler 87] N.I. Badler, K.H. Manoochehri and G. Walters, Articulated figure positioning by multiple constraints, IEEE Computer Graphics and Animation 7(6), 1987

[Baraff 89] D. Baraff, Analytical Methos for Dynamic Simulation of Non-penetrating Rigid Bodies, Computer Graphics, Vol. 23, No. 3, july 1989

[Barzell 88] R. Barzel and A.H. Barr, A Modeling system based on dynamic constraints, Proc. Siggraph, vol 22., No. 4, August 1988

[Brooks 83] R.A. Brooks, Planning Collision Free Motions for Pick-and-Place Operations, The international Journal of Robotics Research, Vol.2, No. 4, Winter 1983

[Denavit 55] J. Denavit and R. Hartenberg, A kinematic Notation for Lower Pair Mechanisms Based on Matrices, J. App. Mech., Vol. 77, pp 215-221, 1955

[Fisher 86] S.S. Fisher, M. McGreevy, J. Humphries and W. Robinett, Virtual environment display system, Proc 1986 ACM Workshop on Interactive Graphics, October 23-24, Chapel Hill, North Carolina.

[Fu 87] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, Robotics: Control, Sensing, Vision and Intelligence, McGraw-Hill Book Company, 1987

[Gilbert 88] E. Gilbert, D.W. Johnson and S. Sathiya Keerthi, A fast Procedure fot Computing the Distance Between Complex Objects in Three-Dimensional Space, IEEE Journal of Robotics and Automation, Vol. 4, No. 2, april 1988

[Girard 87] M. Girard, Interactive design of 3D Computer Animated Legged Animal Motion. Computer Graphics and Applications june 1987.

[Girard 90] M. Girard, Constrained optimization of articulated animal movement in computer animation, Making them move (mechanics, control, and animation of articulated figures), Eds: Badler, Barsky and Zeltzer, Morgan Kaufmann Publishers, 1990

[Gourret 89] J.P. Gourret, N.M. Thalmann, D. Thalmann, Simulation of object and human skin deformations in a grasping task., ACM Siggraph Proceedings 1989.

[Iberall 88] T. Iberall, J. Jackson, L. Labbe and R. Zampang, Knowledge-based prehension: Capturing Human Dexterity, Proceedings of the IEEE on Robotics and Automation 1988. pp 82-87.

[Isaacs 87] P. Isaacs and R. Cohen, Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics, Computer Graphics, ACM Siggraph Proceedings 1987

[Iwata 90] H. Iwata, Artificial Reality with force-feedback: development of desktop virtual space with compact master manipulator, Computer Graphics, ACM Siggraph Proceedings 1990

[Kirckanski 82] M. Kirckanski and M. Vukobratovic, A method for optimal synthesis of manipulation robot trajectories, Trans. ASME, J. Dynamic Systems, Measurements and Control 104, 1982

[Klein 83] C.A. Klein and C.H. Huang, Review of pseudo-inverse control for use with kinematically redundant manipulators, IEEE Transactions on systems, Man and Cybernetics, SMC-13(2), march/april 1983

[Korein 82] J.U. Korein and N.I. Badler, Techniques for generating the goal-directed motion of articulated structures, IEEE Computer Graphics and applications, pp 71-81, 1982

[Landsmeer 55] J.M.F. Landsmeer, Anatomical and functional investigations on the articulations of the human fingers, Acta anatomica, suppl. 25, 1-69, 1955

[Landsmeer 58] J.M.F. Landsmeer, A report on the coordination of the interphalangeal joints of the human finger and it's disturbances, Acta Morphologica Neerlando-Scandinavica 2. 59-84, 1958

[Landsmeer 63] J.M.F. Landsmeer, The coordination of finger joint motions, J. Bone Jnt. Sur. 45, 1654-1662, 1963

[Liegeois 77] A. Liegeois, Automatic Supervisory control of the configuration and behavior of multibody mechanisms, IEEE Transactions on systems, Man and Cybernetics, SMC-7 (12), december 1977

[Lin 83] C. Lin, P. Chang and J. Luh, Formulation and optimization cubic polynomial joint trajectories for industrial robots, IEEE Trans. Automatic Control AC-28(12), 1983

[Lozano-Perez] T. Lozano-Perez, Spatial Planning: a Configuration Approach, IEEE Transactions on Computers, Vol C-32, No.2, feb 1982

[Maciej 90] A.A. Maciejewski, Dealing with the ill-conditioned equations of motion for articulated figures, IEEE Computer Graphics and Applications, May 1990

[Marr 82] Vision, Freeman Press, San Fransisco, California, 1982

[Moore 88] M. Moore and J. Wilhelms, Collision detection and response for computer animation, Proc. ACM Siggraph 1988, Computer Graphics 22(4)

[Paillard 82] J. Paillard, The contribution of perifpheral and central vision to visuallly guided reaching, Analysis of visual behavior, (eds: Ingle, Goodale, Mansfield) Cambridge, Mass. MIT Press, pp 367-385, 1982

[Pletinckx 89] D. Pletinckx, Quaternion calculus as a basic tool in computer graphics, The Visual Computer 1989

[Rijpkema 91] M. Girard and H. Rijpkema, efficient collision detection for convex and concave polyhedral objects, to be submitted.

[Sahar 85] G. Sahar and J. Hollerbach, Planning of minimum time trajectories for robot arms, IEEE International Conference on Robotics and Automation, march 1985

[Schoner 90] P. Schoner and D. Zeltzer, The virtual erector set: Dynamic simulation with linear recursive constraint propagation. Proc. 1990 Symposium on Interactive 3D Graphics March 25-28, Snowbird, Utah

[Shoemake 85] K. Shoemake, Animating Rotation with Quaternion Curves, ACM Siggraph Proceedings 1985

[Tan 88] H. Tan and R. Potts, Minimum time trajectory planner for discrete dynamic robot model with dynamic constraints, IEEE J. of Robotics and Automation 4(2), 1988

[Tomovic 87] R. Tomovic, G.A. Bekey and W.J. Karplus, A strategy for grasp synthesis with multifingered robot hands, Proceedings of the IEEE on Robotics and Automation 1987. pp 83-89.

[Walker 82] M. Walker and D. Orin, Efficient dynamic simulation of robot mechanisms, Trans. ASME, J. Dynamic Systems, Measurements and Control, 1982

[Wang 88] G. Wang and H.E. Stephanou, Chopstick manipulation with an articulated hand: a qualitative analysis, Proceedings of the IEEE on Robotics and Automation 1988. pp 94-99.

[Whitney 69] D.E. Whitney, Resolved motion rate control of manipulators and human protheses, IEEE Transactions on Man-Machine systemsm MMS-10(2) pp 47-53, june 1969

[Wilhelms 87] J. Wilhelms, Using dynamic analysis for realistic animation of articulated bodies, IEEE Computer Graphics and Applications 7(6), 1987

[Witkin 87] A. Witkin and M. Kass, Spacetime constraints, ACM Computer Graphics, Siggraph Proceedings 1987