89-33

89-33

In Proceedings IEEE 1989 International Conference on Systems, Man and Cybernetics

Nov 1989, Boston, MA

A Continuous Model of Robot Hand Preshaping

Thang N. Nguyen and Harry E. Stephanou
stephano@gmuvax2.gmu.edu

Center for Artificial Intelligence
School of Information Technology and Engineering
George Mason University
Fairfax, Virginia 22030

Abstract

Most of the existing dextrous robot hand preshape schemes offer grasp choice from a finite and discrete set of grasp types. Such a discrete set obviously hinders the applicability of preshaping schemes to complex and versatile task requirements. In this paper, we formulate the *continuous spaces* of grasps and derive an algorithmic procedure for robot hand preshaping. The proposed procedure consists of a mapping from a task space to a topological space of hand configurations using a barycentric coordinate system and a barycentric subdivision scheme. A model implementation architecture is discussed. An example showing the preshaping of a four-finger hand in preparation for pinch grasping is also described.

1. Introduction

The preshaping of a robot hand is a preparatory step prior to grasping, aimed at satisfying the anticipated physical constraints. In the case of a gripper (capable of opening and closing motions), the preshape phase is trivial. For a multifingered hand, however, the preshape is much more complex. To reduce complexity, some current schemes select hand preshapes from a finite, discrete number of grasp types (e.g. Cutkosky, 1989). Such a discrete set obviously hinders the applicability of preshaping schemes to complex and versatile task requirements. To overcome this limitation, it is natural to consider the possibility of representing the set of all possible grasps as a continuous set. In this paper, we formulate *continuous spaces* for robot grasps and derive a *procedure* for robot hand preshaping. The proposed algorithmic procedure consists of (i) a set of heuristic rules, and (ii) a convergent mapping from a given, high-level task space to a topological space of hand configurations using a barycentric subdivision scheme.

In section 2, we review some previous work on high-level preshaping and grasp planning schemes. In section 3, we briefly describe our topological model of multifingered prehension. An algorithmic procedure for the selection of hand preshapes from a continuum set of hand configurations is then proposed in section 4, and a model implementation is presented in section 5.

2. Related work

Most high-level knowledge-based schemes for grasp selection or for preshaping operate on a finite and discrete set of grasps. Lyons (1985) proposed a set of three simple grasps: encompass grasp, precision grasp and lateral grasp. Iberall (1987) grouped different grasp types into three categories in terms of force-opposability: pad, side, and palm oppositions. Cutkosky (1977) used a tree-like hierarchy of grasp types which are described in terms of relations between task requirements and object geometry. The common point of departure of these classifications is based on two patterns: *power grips* and *precision grips*, observed in human hand activities and formulated by Napier, a surgeon (1956). The common characteristic of these classifications is discreteness. These finite and discrete sets of grasps have been used in various grasp planning schemes. Examples include Tomovic et al. (1987) in their work on hand preshaping and orientation during approach phase, and Erkmen and Stephanou(1989) in mapping of hand preshapes into finger trajectories.

3. A topological model of prehension

In an earlier paper, we have formulated a topological model of prehension (Nguyen and Stephanou, 1989), based primarily on our observations of the functionalities and sensorimotor activities of the human hand. More recently, we have extended the topological model as a *hierarchy of continuous spaces* representing the set of robot grasps, in constrast to many existing taxonomies of finite and discrete grasps describing grasp types (e.g. Cutkosky, 1989).

In addition to the notion of *power* and *precision* (Napier 1956), our model captures the notion of *support* grasps to account for quasi-prehensile patterns of the hand which are used in hook grasps (four fingers form a hook to hold the handle of a suitcase for example), for pushing a large object, or for holding a large plate with a flat hand. Those three notions constitute the independent variables of the task space.

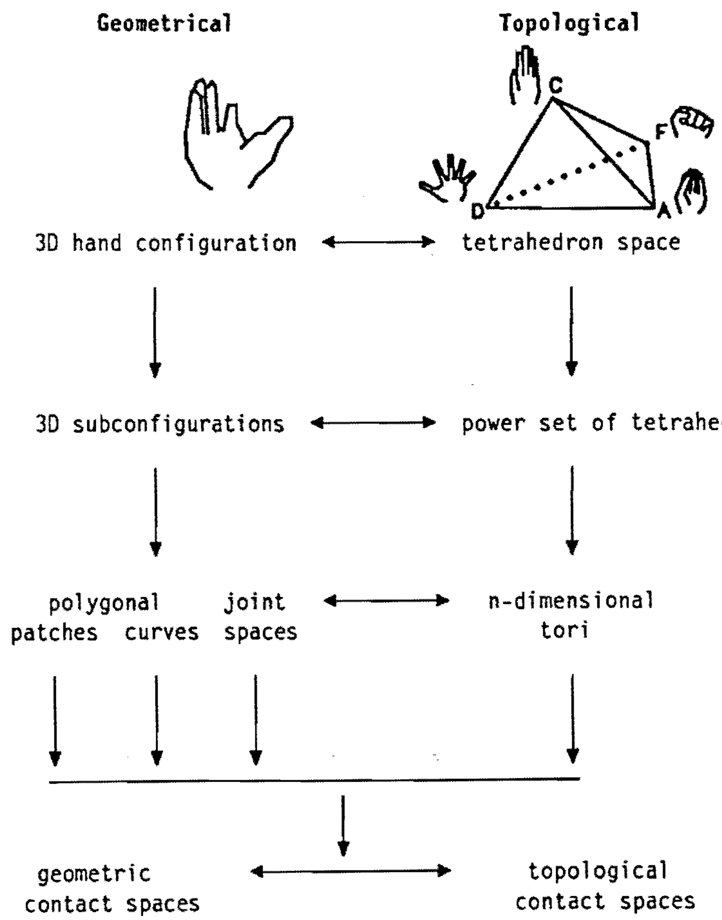


Figure 1: Spaces of hand configurations

The formulation of the continuous space hierarchy of robot grasps is summarized in figure 1 and described from two viewpoints: (i) topology, and (ii) geometry. Geometric representations of the hand shapes are on the left while their equivalent topological spaces are indicated on the right. In this paper, we investigate the preshaping of robot hands using the topological approach. A brief summary of the geometric approach is also given in this paper, for completeness. A much more complete account will be described in a forthcoming paper.

3.1 Representations of hand configurations

3.1.1 Topological point-set representation of hand configurations

We hypothesize that the set of all human hand configurations is bounded by four terminal configurations (figure 2) (i) the fist configuration F, (ii) the planar divergent configuration D (flatly opened hand with all digits divergent), (iii) the planar-convergent configuration C (flatly opened hand with all digits side by side), and (iv) the all-finger-in-opposition, configuration A.

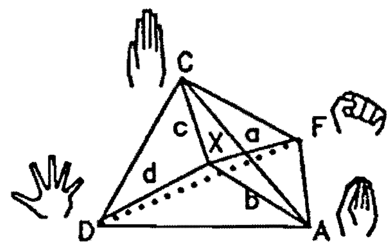


Figure 2: Representation of hand configurations

This set of hand configurations, called T_1, is modeled as a point-set tetrahedron with four vertices representing the four terminal configurations. As such, an arbitrary configuration may be represented as a "point" which lies either at the vertex, or on the boundary or inside the tetrahedron. As a point set, this topological tetrahedron is the highest level of representation of the human hand configurations.

When the four terminal positions are completely known (Nguyen and Stephanou 1989), any other position X inside or on the boundary of the tetrahedron is uniquely determined by a set of barycentric coordinates having the four terminal positions as vertices (figure 2), i.e.:

$$X = aF + bA + cC + dD$$

$$\text{where } a + b + c + d = 1 \text{ with } a,b,c,d \geq 0$$

We postulate the following:

Postulate 3.1: *For a k-finger hand, $k \geq 2$, the set of all hand configurations is bounded by 4 terminal postures, and therefore forms a convex tetrahedron.*

3.1.2 Special case of a hand with one-finger

If the hand consists of a single finger, the tetrahedron is reduced by a projection of itself onto a plane perpendicular to the edge < D-C >. The resulting triangle represents a point set of all possible finger configurations. Indeed, at the finger level, the finger also possesses three terminal positions: (i) fully flexed V, (ii) fully extended T, and (iii) in reciprocal position U (Long, 1970), as shown in figure 3. Since the finger can be considered as a polygonal line in R^3, the finger shape does not necessarily go through the reciprocal position when it changes from fully extended to fully flexed. A general position will be of the *claw*-type.

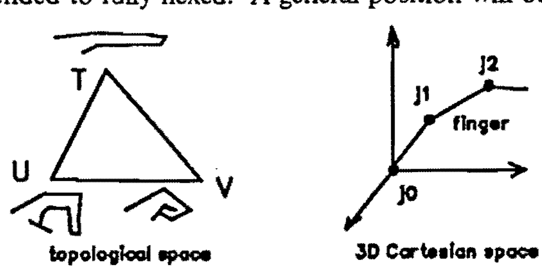


Figure 3: Representation at finger level

Thus, one may use the three terminal positions to form a two-dimensional topological simplex (triangle) with each terminal position being represented as a vertex.

Again, an arbitrary finger configuration Y, is determined by a point in the simplex:

$$Y = tT + uU + vV$$

where $t + u + v = 1$ and $t,u,v \geq 0$,
and T, U, V correspond to the three terminal positions

3.1.3 Power set of subconfigurations

A hand configuration may be considered as an aggregation of group of fingers called subconfigurations (Nguyen and Stephanou, 1989). The simplest subconfiguration is a finger, and the most complete subconfiguration is the entire hand. The set of all such subconfigurations is the power set of hand subconfigurations.

Topologically, for a k-finger robot hand, the power set T_2 consists of 2^k elements each of which, by Postulate 3.1 above, is either a 3-dimension tetrahedron (for $2 \leq m < k$, where m is the number of fingers of a subconfiguration), or a 2-dimension triangle (for $m = 1$). $m = k$ corresponds to the case of the entire hand. Furthermore, an arbitrary whole hand configuration may be established by a *topological product* of two or more component subconfigurations, as illustrated in the case of the fine pinch (figure 4): $t_f \in T_f = t_2 \in T_2 \times t_3 \in T_3$, i.e. the configuration t_f for a fine thumb-index pinch is the topological product of two subconfigurations: (i) a two-finger subconfiguration t_2 of the two-finger tetrahedron T_2, and (ii) a three-finger subconfiguration t_3 of the three-finger tetrahedron T_3.

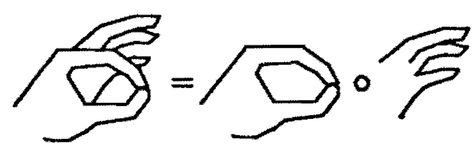


Figure 4: Fine thumb-index pinch

3.1.4 Geometric representations of hand subconfigurations

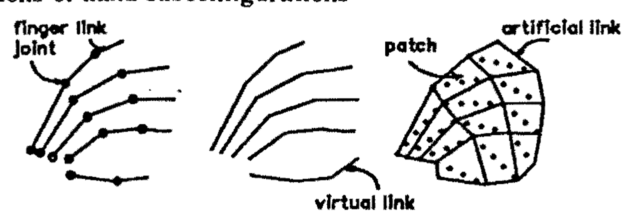


Figure 5: Joints, links, and patches

In Cartesian space, the geometric shape of a hand may be represented by (i) a set of joints, (ii) a parallel structure of k kinematic chains, (iii) a set of adjoint patches, or (iv) a collection of connected geometric tetrahedra (figures 5 and 6). The first two representations (i) and (ii) have been widely discussed in the literature. In this paper we briefly describe the third and fourth representations.

To represent a hand as a collection of patches (third representation), we use the notions of *virtual link* and *artificial link*. Virtual links are those that extend the chain so that all chains (fingers) have the same number of links. Artificial links are those that connect corresponding joints of two adjacent fingers. The triangles formed by adding those links are called *patches* (figure 5).

In the fourth representation (geometric tetrahedron) and in the general case where the hand figure is *cupped*, as shown for the two-finger, three-joint hand of figure 6, the hand posture is a union of three adjacent geometric tetrahedra, each pair of adjacent tetrahedra having a common edge. Each tetrahedron is a connected set, and two adjacent tetrahedra have a common edge. Therefore their union $s_1 \bigcup s_2 \bigcup s_3$ is also a connected set. This connected set is a 3-dimensional topological space.

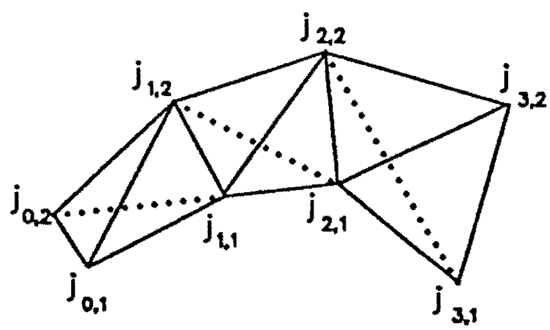


Figure 6: Hand subconfiguration as connected geometrical tetrahedra

In other words, a hand subconfiguration may be decomposed into simpler forms, called *simplexes* (Pontryagin, 1952), of smaller dimensions, which adjoin one another in some describable fashion. This geometric configuration of the hand is a polyhedron, and the decomposition scheme is called a geometric complex K. K is in fact a collection of simplexes. The 0-simplexes of a complex K are the joints, the 1-simplexes are the links, the 2-simplexes are the patches, and the 3-simplexes are the geometric tetrahedra. Based on the previous analysis of a hand configuration, we introduce the following definitions:

Definition 3.1 *A hand* **subconfiguration** *is the union of a collection of connected sets (geometric tetrahedra, patches, chains). A connected set is a set of points which is compact, continuous, and bounded by piecewise differentiable surface. A geometric subconfiguration S is said to be convex if* $x \in S$ *and* $y \in S$ *implies* $((x,y) \subset S)$.

Definition 3.2 *Two patches A and B are said to be* **properly situated** *if they are non-intersecting or if their intersection is a common edge of A and B. Thus, a geometric hand subconfiguration is bounded by a collection of properly situated patches.*

3.2 Subconfiguration functionality

The functional activities of grasping are divided into power grips, precision grips and support grips. These patterns may be associated with hand functionality as shown in figure 7:

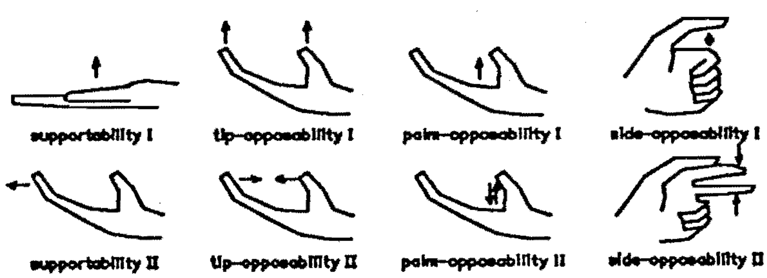


Figure 7: Eight types of hand functionality

Type I supportability occurs in flat hand support grips while type II supportability occurs in various hook grips. Type I tip-opposability is similar to type I supportability except that it only includes the fingertips. Type II tip-opposability occurs for example when holding the lid of a large jar. Type I palm-opposability uses the palm in opposition to the object while type II palm-opposability uses the palm in opposition to the other fingers, as in power grips. Type I and type II side-opposability both use the sides of two consecutive fingers. In type I, one of the fingers is the thumb. (Notion of opposability is due to Napier (1956) and opposition space of dextrous grasps is due to Iberall (1987)).

Example 1

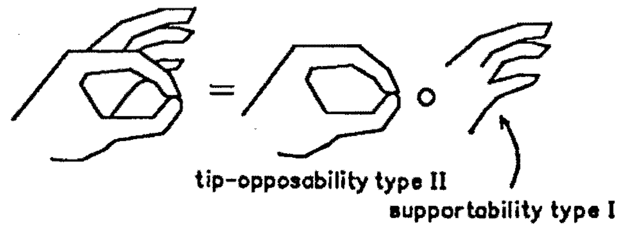


Figure 8: Fine thumb-index pinch (functionality)

In a fine thumb-index pinch (figure 8), two subconfigurations may be identified with their functionalities: (a) type I tip-opposability involving the thumb and index, which is the primary subconfiguration needed to perform the intended task (pinch), and (b) type I supportability, which is identified as a secondary subconfiguration, because of its small contribution to the grasp. In this example, the contribution of the subconfiguration (b) is to balance the overall hand. No resultant force is applied on the grasped object by a type I supportability. We call the functionality of this subconfiguration a *don't care* subconfiguration. We associate functionality with the notion of a *role*. The role associated with each subconfiguration may be measured by a numerical parameter called *Dominance factor* $df \in \{0,1\}$ When $df(I)$ of a subconfiguration I is higher than $df(J)$ of a subconfiguration J, we say that the subconfiguration I is primary, and the subconfiguration J is secondary. If the df of a subconfiguration is close to zero, then the corresponding subconfiguration becomes a *don't care* . Our fine thumb-index pinch problem is then reduced to:

- a two-finger pinch which may be analyzed as a two-dimensional preshaping problem with a *don't care* subconfiguration of three remaining fingers, and
- a hand balance problem between the two subconfigurations.

4. A simple algorithmic procedure for hand preshaping

The procedure described in this section allows the selection of a preshape from a larger set of hand configurations. The algorithmic procedure is based on (i) the *barycentric subdivision of hand subconfiguration space (topological space)* as described below (Pontryagin, 1952), and (ii) a set of heuristic rules derived from human hand preshaping activities (e.g. to determine a subconfiguration of two fingers or of three fingers).

Task and object characteristics are associated with the four terminal positions of the hand , and expressed symbolically and/or numerically in a model-based system (section 5). We derive the barycenter characteristics of the simplexes using a pattern matching scheme. These characteristics are then used for the determination of a simplex containing the hand configuration.

4.1 Barycentric subdivision of hand subconfiguration space

In representing the set of hand consifurations as a continuous set, the search for a particular configuration is more complex. By using a barycentric subdivision technique, however, the complexity decreases dramatically as shown below.

For a one-dimensional topological space K_1, a barycentric subdivision of K_1 is obtained by identifying the midpoint of the one-dimensional simplex K_1 (called the barycenter). The midpoint is determined by the characteristics of the end-points. For example, the mid-point of the edge CD of the tetrahedron is a point representing a hand shape which is flat and whose webspace is one-half of the the maximum webspace of the divergent position of the hand (hand configuration D).

For an n-dimensional polyhedron space, there are $(n+1)!$ simplexes which are subdivided barycentrically. This can be shown by induction.

4.2 Heuristic rules

A hand preshape need not be exactly determined. To reduce computational complexity, heuristic rules may be used to coarsely derive, for example: (i) what subconfiguration and how many fingers per subconfiguration will be involved in an anticipated grasp, (ii) the type of subconfiguration functionalities and their associated roles, (iii) how large the hand opening should be, and (iv) the orientation of the hand with respect to the object as specified by the task. Some heuristic rules for preshaping are:

1. The hand opening must be larger than the size of the object surface containing the anticipated contact points.
2. Except for the special case where type I supportability is clearly indicated (this results in a flat position of the hand), the position of the thumb generally determines the types of preshape: (i) for a precision grip preshape, the thumb and the index are in opposite direction, and (ii) for a power grip, the fingers and the palm are in opposite direction, and the thumb is not indispensable (Napier 1956, Landsmeer 1962).
3. A power grip is associated with the transverve axis of the hand. A precision grip is associated with the longitudinal axis of the fingers, and a support grip is associated with the normal to the plane of the palm.
4. In a precision grip preshape, there is one contact per finger, whereas in a power grip preshape, there are multiple contacts per fingers (fingertip, link and palm contacts),
5. During preshaping, the task determines the set of congruent axes of the object and the hand.
6. If a graspable part of an object is larger than the width of the hand (measured along the transverse axis), the preshape involves all available fingers. If it is small relative to the hand width, it involves fewer fingers. If it is small relative to the finger size, the preshape involves only two fingers, one of which is the thumb.

4.3 Determination of the hand preshape using the tetrahedron space

A hand subconfiguration is represented by a point on the boundary or inside the tetrahedron and is uniquely determined by a set of barycentric coordinates (or any equivalent three-dimensional system of coordinates). Using the barycentric subdivision scheme, if the four vertices are known, then the preshape of a hand can be determined as follows:

Case 1: Task completely specified and object known.

Using heuristic rules, the geometric characteristics of a given object are used to approximate the overall webspace of the hand (sum of openings between successive fingers). This corresponds to point X on the edge <D-C> (figure 2). Similarly, the specifications of the given task in terms of power and precision is used to determine a point Y on the edge <F-A>. The preshape of the hand is then somewhere on the line XY which is characterized by, for example, the magintude of the curvature (Nguyen and Stephanou, 1989), and the direction of its functionality axis. A binary sort procedure (dividing the line XY into halves, and evaluating which half the preshape should belong to) is used to get smaller segments containing the point representing the anticipated preshape characteristics. Some criterion is defined to stop the iteration, for example, in pseudo-code:

```
WHILE
   object axis u'u is congruent to
   longitudinal axis l'l of hand subconfiguration
DO binary_sort_procedure
UNTIL
   curvature of new_barycenter of XY ≤
   curvature of graspable surface of object
END;
```

Case 2: Task somewhat specified and object somewhat known.

This is the general case of preshaping. A barycentric subdivision procedure is used instead of binary sort. Somewhere in this tetrahedron, there exists a unique point at which no precision, no power and no support are involved. This is the hand position at rest, and is located at the centroid O of the tetrahedron. For an arbitrary hand preshape (represented by a point ξ with barycentric coordinates a, b, c, d $\neq 0$), the point ξ is either (i) completely inside one of the four smaller tetrahedra formed by the point O and the four vertices, or (ii) on a common face of two such tetrahedra, or (iii) on a common edge of two tetrahedra. It is easy to identify which of the four smaller tetrahedra ξ belongs to: this is the tetrahedron $< O >$ formed by the origin O and three vertices corresponding to the three largest barycentric coordinates. We can then obtain a smaller topological space of hand configurations containing the point ξ . One may continue this process of decomposition of the tetrahedron until some stopping criterion is met, for example, until one cannot obtain a new tetrahedron with O and at least one of the points (F, A, C or D) as vertices.

In all cases, the procedures are assured to converge due to the nature of barycentric subdivision scheme (and binary sort).

5. Model implementation

In an unstructured robot environment hand, task representation is often knowledge-intensive while the object representation is data-intensive. The efficient execution of a task depends on skill and experience, while the object to be manipulated may be described in great detail. For a task to be performed on a graspable object, there must be a set of commonalities (properties that satisfy both task and object) established between task and object information. This set of commonalities constitutes the link between a given high-level task description and a given low-level object description (generally available in some database structure). Thus high-level task information must be translated into low-level grasp kinematics. We postulate that the set of commonalities may be determined by two types of abstraction hierarchies (Smith, 1986): (i) generalization (i.e. formulation of more abstract concept from existing concepts), and (ii) aggregation (i.e. component_of, part_of).

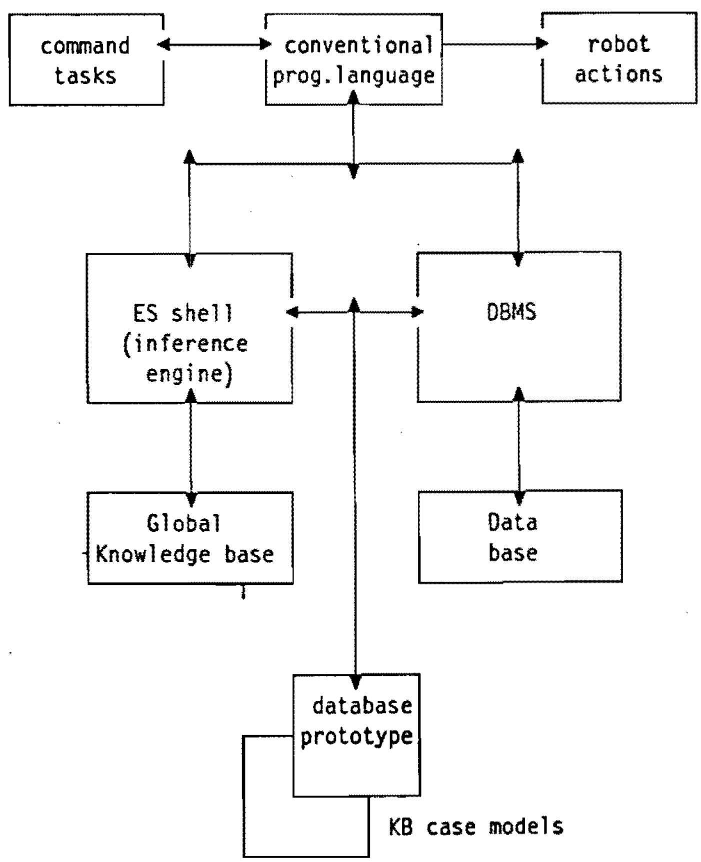


Figure 10: Implementation architecture

For implementation purposes, we introduce two notions: knowledge base *case model* and *database prototype*. A case model embodies the knowledge, skill and handling experience for a task to be performed by a robot hand, and has little to do with any particular object. A database prototype embodies higher-level characteristics of objects abstracted from a data-intensive description.

5.1 Case model

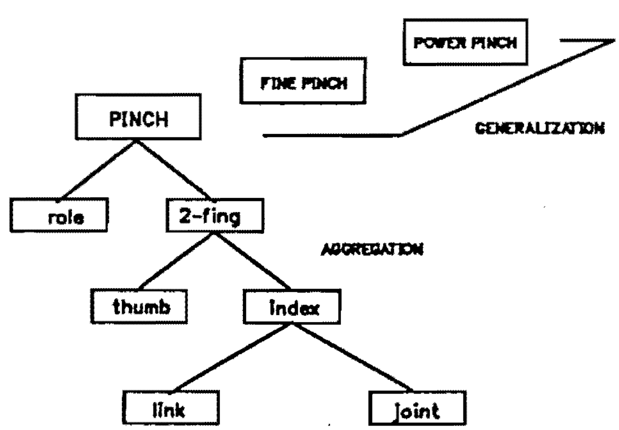
Case models are functionally dependent on its inputs, just as the contents attributes of a file are functionally dependent on the key attributes. We anticipate the existence of commonalities between case models's input and output attributes which means that the outputs of some case models could be inputs to others Such a linking of case models corresponds to a relational join in DBMS, in which case models are viewed as virtual relations (Blanning, 1987).

Frame	Slots	Facets
PINCH_PRESHAPE	TYPE	fine_pinch
	ROLE	primary
	NUMB_FINGERS	2
	PRIMARY_AXIS	x'x
	WEBSPACE	.8
	MULTI_FING_CONTACT	no
	LINK_CONTACT	no
	FINGERTIP_CONTACT	yes
	FORCE_OPPOSITION	yes
	FUNCTIONALITY	type II tip_oppo.

The case models are being encoded as frames (Minsky, 1975). A frame is a data structure. Frames contain named *slots* which can be filled with *facets*. Facets in turn can be simple names, identifiers or frames themselves. Slots may or may not be filled. An instance of a frame is then an individual, and each slot represents a relationship between that individual and another. A frame can be used to represent a concept, an instance of which may be generated by filling its slots. If facets of all slots are identified, then one may conclude that the concept exists. Instances of frames can be compared, that is, an instance of one concept represented by a frame may be *matched* to see if it belongs to another concept represented by another frame (Hayes, 1981).

Consider an example of a Minsky's frame hierarchy for a PINCH_PRESHAPE below. By describing the PINCH preshape as a frame, one may perform a number of inferences such as instantiation, or evidential reasoning. In instantiation, given a frame of PINCH, we may generate one instance of two-finger fine-pinch by filling its slots. In evidential reasoning, given some body of evidence which corresponds to some of its slots, we may infer that the concept of PINCH is a generic concept of FINE PINCH, and initiate a method associated with the frame to perform the required actions. As frames can be created, updated, expanded, deleted, etc., they may be used to describe tasks according to some level of specificity.

5.2 Database prototype



Database prototype is an incomplete database instance created dynamically in the database. Knowledge and data of interest are modeled as objects having properties and attached methods. Objects in the database prototype can be basic data items, entities, relationships, aggregated or generic entities. Thus .the notion of database prototypes may be abstracted from the databases, and extended to represent potential object instances that may exist in the case models or other knowledge bases.

5.3 Example 2

To illustrate our model structure, let us consider the following description which involves a *stick model* of a four-finger UTAH/MIT-like hand, a simple task and a regular object.

Given a task and an object, one key element of our scheme is that the descriptions of tasks, objects, and hand configurations are organized in four groups: (i) geometric properties, (ii) topological properties, (iii) functional properties, and (iv) behavioral properties. Those properties are called pre-runtime properties and encoded in a knowledge/data base. Other properties such as object size, weight, etc. as well as position and orientation are not considered until run-time.

5.3.1 Object properties

Objects of interest are assumed to be rigid bodies of regular shape, for example cylindrical, prismatic, spherical, etc. More complex objects can be built based on those regular shapes. Objects may be described geometrically (e.g. symmetry), topologically (e.g. number of vertices, edges, faces), functionally (e.g. used as tool), and behaviorally (e.g. rollable on its side, motion constrained in certain directions). Purely geometric object properties are not task-dependent, whereas functional and behavioral properties are very task-dependent, and topological properties are somewhat task-dependent. In terms of geometric and topological properties, we describe the characteristics of a given object through the followings steps:

1. list all the geometric characteristics of the object, (e.g. in a relational database).
2. model the object such that topological, functional and behavioral properties can be inferred using abstraction hierarchies,

Table 1 describes some properties of three types of regular objects.

Table 1. Object properties

property	cylindrical	prismatic	spherical
geometric	axes of symmetry	axes of symmetry	axes of symmetry
topological	2 edges, 3 faces	6 vertices, 9 edges, 5 faces	no vertex, 1 face
functional	weak stability on certain face	strong stability on all faces	extremely weak stability everywhere
behavioral	rollable on one of its sides	non-rollable	unconstrained in most directions

5.3.2 Task properties

As previously stated, tasks are knowledge-intensive. High-level tasks are normally described by a single or finite number of commands. Task representation consists of two components: (i) a static (environment-independent) component, and (ii) a dynamic (environment-dependent) component. The independent component is intrinsic to a given task. Tasks may be decomposed into subtasks which deal with a specific task objective. Commonly, those subtasks may be (i) partially ordered, e.g. subtask i is to be executed before subtask j for $i<j$, or (ii) performed asynchronously. The static component may be inferred by using abstraction hierarchies as described in the case of objects. The dynamic component, on the other hand, must be updated when required. An example of a task property list is given in Table 2.

Table 2. Task properties
• preshape the hand along one preferred axis direction (geometric)
• preshape the hand around the object (topological)
• preshape the hand to allow maximum contact surface (functional)
• preshape the hand to allow motion in many directions (behavioral)

5.3.3 Hand subconfiguration properties

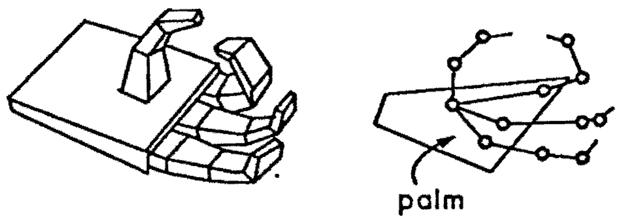


Figure 11: Four-finger stick hand model

For each element of the power set of hand subconfigurations (i.e each topological tetrahedron), we describe the characteristics of each terminal postures (namely fist, all-finger-in opposition, divergent and convergent). An example of fist grasp of the stick hand shown in figure 11 is given below.

Fist grasp properties. The fist grasp is such that:

1. the transverse axis of the hand is parallel to or coincides with a preferred axis (axis of symmetry) of the object (axis constraint).
2. the planes containing the fingers are perpendicular to the object preferred axis (axis constraint).
3. all fingers flex such that each phalanx contacts the object (contact constraint).
4. the thumb is flexed in its flexion-extension plane such that the distal phalanx contacts the object (contact constraint)

The description of known configurations using a similar property list structure as shown above are used in the following operations: (i) derivation of a new (barycentric) configuration using known configuration descriptions (section 4), and (ii) matching of configuration properties with those in a combined task-object property list (Tables 1 and 2).

6. Conclusion

In this paper, we have described the *continuous spaces* of grasps and sketched a *algorithmic procedure* for robot hand preshaping. This partial result is based on a computational theory of prehensility that we are developing. The basis of our approach is both intuitive and mathematical. The intuitive formulation was based primarily on our observations of functionalities and motor activities of the human hand. The mathematical formulation is based on point-set topology and combinatorial topology. Point-set topology is used in the representation of the set of hand configurations as a continuous set. Combinatorial topology is used in the decomposition of a topological and geometrical hand configuration into its simplexes of smaller dimensions. We have also proposed a model implementation and introduced two concepts: database prototypes and knowledgebase case models to be used as bridges between high-level task information and low-level object geometry and hand/object kinematics.

7. Acknowledgements

This research was conducted in the Center for Artificial Intelligence at George Mason University. Research activities of the Center are supported in part by Defence Advanced Research Projects Agency under grant, administered by the Office of Naval Research, No. NOOO14-87-K-0874, in part by the Office of Naval Research under grant No. NOOO14-88-K-0226, and in part by the Office of Naval Research under grant No. NOOO14-88-K-0397.

8. References

1. Blanning R. W. (1987), "A framework for expert modelbase systems", *Proc. AFIPS*, pp.13-17.
2. Cutkosky M. R. and Wright P. K., (1986), "Manufacturing grips and correlation with the design of robotic hands", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1533-1539.
3. Cutkosky M. R. (1989), "On grasp choice, grasp models and the design of hands for manufacturing tasks", *IEEE Trans. Robotics and Automation*, Vol. 5, No. 3.
4. Erkmen A. M. and Stephanou H. E., (1989), "Preshape Jacobians for minimum momentum grasping", *Proc. IEEE Int. Conf. Systems, Man and Cybernetics.*
5. Hayes P. J. (1981), "The logic of frames", in *Readings on AI*, Webber B. L. and Nilsson N. J., Tioga Pub. Co., 1981, pp. 451-458.
6. Iberall T. (1987), "The nature of human prehension: three dextrous hands in one", *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp. 396-401.
7. Long C. II, Conrad P. W., Hall E. A. and Furler S. L. (1970), "Intrinsic-extrinsic muscle control of the hand in power grip and precision handling", *J. Bone Joint Surgery*, Vol. 52A, NO. 5, 1970, pp. 853-857.
8. Lyons D. M (1985), "A simple set of grasps for a dextrous hand", *Proc. IEEE Int. Conf. Robotics and Automation*, 1985, pp. 588-594.
9. Minsky M., (1975), "A framework for representing knowledge", in *The Psychology of Computer Vision*, P. Winston (Ed.), pp. 211-310.
10. Napier J. R. (1956), "The prehensile movement of the human hand", *J. Bone Joint Surgery*, Vol. 38B, pp. 902-913.
11. Nguyen T. N. and Stephanou H. E. (1989), "A topological model of multifingered prehension", *Proc. IEEE Int. Conf. Robotics and Automation*, 1989, pp. 446-451.
12. Pontryagin L. S., (1952), *Foundations of Combinatorial Topology*, Graylock Press, 1952.
13. Smith J. M. and Smith D. C. P. (1977), "Database Abstraction: Aggregation and Generalization", *ACM Trans. on Database Systems*, Vol. 2, No. 2, pp. 105-153.
14. Tomovic R., Bekey G. A. and Karplus W. J. (1987), "A strategy for grasp synthesis with multifingered robot hands", *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp.83-89.